# Analyzing colony dynamics and visualizing cell diversity in spatiotemporal experiments


A DISSERTATION SUBMITTED
BY
GEORGES HATTAB
TO THE
FACULTY OF TECHNOLOGY


IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR RERUM NATURALIUM
IN THE SUBJECT OF
BIOINFORMATICS


BIELEFELD UNIVERSITY
BIELEFELD, GERMANY

SEPTEMBER 2017

Printed on non-ageing paper according to DIN-ISO 9706.

Thesis advisor: apl. Prof. Dr.-Ing. Tim W. Nattkemper
Thesis co-advisor: Prof. Dr. Tamara Munzner Georges Hattab

## Abstract

Bioimaging technologies enable the description of the life cycle of organisms at the microscopic scale, for example bacterial cells. In the particular case of time lapse imaging, the coupling of experimental setups and marker protocols results in the acquisition of biological changes in spatiotemporal experiments. Such experiments are devised to obtain a time-lapse image data, which I refer to as biomovies. Understanding how a cell behaves at every time point is crucial. In fact, this motivated all cell studies in the literature, which are single cell oriented. For the present biomovies, the task is to identify similarly fluorescing subpopulations across space and time. My interest lies in isogenic bacterial populations of *Sinorhizobium meliloti*. The biomovies' particularity is a dynamic range of high values for a set of different properties (e.g. cell density, cell count, etc), herein, leading to a bottleneck. State of the art methods cannot address such a task, which is partly due to their inability to handle highly dense populations and their adaptability to different experimental setups. In particular, they fall short either at the segmentation step (to delineate individual cells and extract their abstraction, e.g. cell centroid) or at the tracking step (to follow identified cells in each frame). To gain insight into bacterial growth at the population level, I claim that one does not really need to know the fate of each single cell. In the context of this thesis, I present a series of pipelines and algorithms. First, preprocessing pipelines to reduce noise and enhance the object-to-background contrast. Second, an adaptive algorithm to correct spatial shift in the images (i.e. registration) and of each biomovie. Third and last, a modular algorithm that constructs coherent patch lineages by employing two adapted data abstractions, the particle and the patch, that are essential to solving the aforementioned bottleneck and are defined as follows: A particle is an intuitive geometric abstraction that results from considering whether the neighborhood around a pixel falls within a cell by checking for signal characteristics such as signal intensity, edge orientation, fluorescence signals, or texture. A patch is the aggregation of spatially contiguous particle trajectories that feature similar fluorescence patterns. The methodology that creates coherent patch lineages is automatic and modular. By integrating aspects of object recognition and spatiotemporal changes, it lays down the foundation for investigating colony growth. All of the aforementioned pipelines represent a new methodological contribution to the field of lineage analysis and colony growth. I evaluate the proposed pipelines and algorithms on simulated and biological data, respectively. In turn this enabled me to validate the algorithms, interpret changes in the colony growth and differences among conditions of an experiment. In particular, I found that in a same condition, two isogenic bacterial colonies grew differently when faced with the same stress. The methods pioneered herein provide a key step to investigating colony growth.

# Contents

The following authors contributed to Chapter 1: Georges Hattab drafted and revised the content. Tamara Munzner contributed to the clarification of the biomovies properties. Tim W. Nattkemper contributed to content revisions.

The following authors contributed to Chapter 2: Georges Hattab composed the initial content and revised it. Jan-Philip Schlüter recorded and kindly provided the first experiment of the biological data. Matthew McIntosh conducted and provided the second experiment of the biological data. Anke Becker introduced the biological motivation behind such experiments. Veit Wiesmann created and kindly provided the simulated data. Tim W. Nattkemper revised the content.

The following authors contributed to Chapter 3: Georges Hattab wrote the initial draft. Tim W. Nattkemper revised the content.

The following authors contributed to Chapter 4: Georges Hattab conceived the approach, led its development, the data analysis and composed the content. Anke Becker and Tim W. Nattkemper revised the content. Torben Müller reviewed the formal description of the approach.

The following authors contributed to Chapter 5: Georges Hattab conceived the framework, led its development and the data analysis. Veit Wiesmann contributed to content revisions. Tamara Munzner contributed to framework development, content writing and revisions. Anke Becker and Tim W. Nattkemper revised the content.

The following authors contributed to Appendix A: Georges Hattab revised the initial draft. Veit Wiesmann formally described the cell simulation approach and wrote the initial draft.

The following authors contributed to Appendix B: Georges Hattab conceived the amino acids task-oriented visualization, led its development, designed the amino acids cards and the 2-dimensional structural formulae. Benedikt Brink suggested an appropriate data abstraction, implemented the gamification approach, and contributed to the execution of the data abstraction design. Tamara Munzner refined the description of the visual encoding of the cards. Tim W. Nattkemper aided the explication of the motivation behind the visualization.

The following authors contributed to Appendix C: Georges Hattab wrote and revised the content. Daniel Langenkämper contributed to the initial study and its development.

The following authors contributed to Appendix D: Georges Hattab wrote the initial draft. Tim W. Nattkemper revised the content.

For my parents, whose openness and dedication nurtured me.

# Acknowledgments

Achieving this thesis is best understood by analogy to cell division. In the context of cell population growth, one 'mother' cell grows and divides to produce two 'daughter cells' and so on. Its peculiarity is the exponential growth called doubling. In comparison, tackling head on this project resulted in doubling ideas and doubling the possible directions of inquiry. Like the journey of a cell, from one point to another, reaching the end matters. Yet I learn that the whole point is to grow. Like this cell, there are particular conditions, and brilliant people that helped me grow.

Big hugs for both my mum and dad. I would like to thank them for being who they are, and for freely providing for me. They have always encouraged me to go further, even where there is no path to create a new trail. My supervisor, Tim W. Nattkemper, has offered me his leading spirit, sense of direction, and encouragement. Without his support, I would neither have enjoyed this project as much as I have, nor would I have been able to pursue it with patience and commitment. I could not have asked for a more effective mentor and human being, for that I am very grateful. Another faculty member, Roland Wittler, whose sense of planning and drive have enriched me. I am thankful for his continuous care, and for providing the best possible work environment. I am exceptionally lucky to have experienced the duality of this graduate program, to have travelled to Vancouver and met my second supervisor, Tamara Munzner. I am very grateful for all the time she invested, the stupendous number of research ideas, and the contagious enthusiasm throughout our continuous meetings. By knowing her, I got the privilege of meeting the visualization community, opening my own world to another. Next, I would like to acknowledge funding from the International DFG Research Training Group 'Computational Methods for the Analysis of the Diversity and Dynamics of Genomes' GRK 1906/1 for the last three years.

I owe a great deal of insight to all the groups I have been part of: the 'Biodata Mining Group', the 'Diversity and Dynamics' research group, the 'InfoVis' group. Thanks to their members, I will always remember my time in Germany, and Canada as some of the most transformative, fun, and challenging years of my life. Last but not least, I have also had the fortune to connect and create friendships with several graduate students. With whom I have had great fun, from traveling together to discussing novel ideas. Thanks to Benedikt Brink, Jia Yu, Karol Szczypkowski, Liren Huang, Marten Heidemeyer, Jan Kölling, Markus Lux, Nicole Althermeler, and several others with whom I have went on adventures, and enjoyed simple things. They are some of the most wonderful people I know.

# 0 Introduction

## 0.1 Background

Since 1887, the Petri dish has been used for the culture of microorganisms[1]. Particular microorganisms, such as bacteria, moss, and protozoa have been widely studied[2,3,4]. While Petri dishes are widespread in microbiological research, they have a limited amount of space, and limited nutrients for the bacteria to grow in. Faced with such limiting factors, smaller dishes are used for large-scale studies yet can be relatively expensive and labor-intensive. In turn, this motivated the fabrication of smaller structures with the help of micro-engineering adapted to biological experimentation.

The notion of the micro-scale was taken from the domain of fluid mechanics in physics and was introduced in the early 1880s into the study of microorganisms[5,6,7]. The word **micro** typically involves the following features: small size or small volumes (i.e. from the microliter to the femtoliter), or low energy consumption, or the effects of the microscale. In physics, the microscale is often defined as the relative strength of forces, or changes due to confinement, or due to scale. One prominent example is the relative dominance of surface tension in three-dimensional fluidic spaces, i.e. microfluidic devices. The different reasons that motivated the domain of microfluidics are: The ability to create and control flow configurations at very small

scales. The detection of small quantities in an affordable and portable way. And extending microfluidics methods to other domains, such as biology (c.f. chapter 1).

From over a century of neuron culture to bacterial cell culture, the study of cellular and sub-cellular elements attracted biologists. Recent work use microfluidics to conduct fundamental experiments[8,9]. Thanks to a multiplicity of knowledge domains, microfluidic devices permit the study of individuality over time, at the single cell level and in an automated manner. Many successful and pivotal studies are reported in developmental biology (i.e. embryogenesis)[10,11,12], synthetic biology, and systems biology (i.e. variations in gene expression, and genotype-phenotype linkage)[13,14].

In biology, in particular synthetic biology, genetic engineering is extended to focus on whole organisms and their gene products[15]. Analyzing biological organisms in their entirety is shared by the discipline of systems biology. The standardization and automation of processes in the lab led to a shift of efforts towards engineering cells, with novel functions and in a novel hierarchy of biological devices, modules, cells, and multicellular systems. Microfluidics is a prominent example, it provides the means to obtain high cell proliferation and high density which are neither limited by the depletion of nutrients nor the accumulation of metabolites in the medium[16]. The coupling of microfluidics and time-lapse imaging provides functional insight into cell development. For example the how bacterial cells develop resistance to antibiotics, in small populations, and in a short period of time.

Time-lapse imaging is a technique whereby serial images are taken at regular time points to capture the dynamics of what is being observed. Recorded images can be played back at different speeds to aid analysis. Hence, by recording such experiments into time-lapse image data, it is possible to gain knowledge into the life of microorganisms, i.e. the becoming of one cell into a cell colony. Time-lapse imaging captures motility, cell morphology, as well as changes in multiple fluorescence channels. The resulting data volume requires a combination of automated cell detection (cell segmentation), automated cell tracking methods, and cell lineage construction. These steps comprise the general paradigm employed by all state of the art tools. While many tools have implemented software solutions to deal with such data, a bottleneck remains in the context of adaptability and scaling.

This entails generalizing the methods to other data sets and/or experiments and adapting them to the data volume, respectively. Various tools attempt fully automated cell tracking, which can contain errors and thus need manual data curation[17,18]. Other approaches ana-

lyze the data while allowing some user control, yet are data specific, and lack functionality to process long term experiments[19]. For example, they provide limited or no support for cell tracking correction (e.g. over multiple fields of view). However, there exist software solutions that address multi-dimensional image data, yet lack interfaces for manual data curation[20,21]. More complete approaches exist, addressing single cell quantification in an iterative image analysis workflow, where image preprocessing is followed by an inspection of the data where only relevant parts are loaded. It is often followed by automatic tracking, inspection, correction, then visualization[22,23]. Confronted with strong image variability and high values for data properties such as cell density, most promising state of the art solutions fall short either at the segmentation step (to delineate individual cells and extract their abstraction, e.g. cell centroid) or at the tracking step (to follow identified cells in each frame). By addressing a level of biological organization, for instance the cell, its corresponding level of abstraction is needed: either at the image space, lexically, or even computationally.

### 0.1.1 Abstractions: From ideas to events

The quality of dealing with ideas rather than events is an abstraction. Topics vary in degrees of abstraction. In the particular domain knowledge of biology, abstractions range from the human, to the organs, to the tissue, to the cell, to the organelles, to the molecules, to the infinitesimal (i.e. atoms and their constituents). The study of a scientific question entails a process of considering associations and context. This denotes establishing relationships and drawing away concepts or abstract ideas. As a result, the exploration of representational forms or abstractions that exist in multiple domains (e.g. biochemical formula) provides a freedom of thought. In this work, I present abstractions that are capable of tackling not only ideas, but also events. The idea I tackle concerns the representation or visualization of amino acids (c.f. chapter 3), part of a cell, cells, and subpopulations (c.f. chapter 5).

### 0.1.2 Life begins with cells

As a fundamental unit of life, an individual cell can grow, process information, respond to stimuli, and carry out an array of biochemical reactions. In the context of cell growth, these points vary from an organism to another. Cells are either eukaryotic or prokaryotic. Unlike eukaryotic cells, prokaryotic cells consist of a single closed compartment that is surrounded

by the plasma membrane, lack a defined nucleus, and have a relatively simple internal organization. All prokaryotes have cells of this type. Bacteria, the most numerous prokaryotes, are single-celled organisms. For cyanobacteria, the organism is unicellular or is observed with filamentous chains of cells.

In this thesis, I focus on isogenic bacterial populations of *Sinorhizobium meliloti*, where similar cellular behaviors result from similar gene expression profiles. *S. meliloti* is a soil bacterium and model organism that has been the central interest of gene regulation studies (e.g. quorum sensing) and investigations of symbiotic and pathogenic plant-microbe interactions[24,25,26]. Provided different conditions, the records of each population results in diverse and dynamic data. I define diversity and dynamics by employing a multiplicity of data properties and the ranges in which they vary, respectively (c.f. chapter 1). This enables me to describe the particular data at hand, in turn, permitting us to create and tailor the different data abstractions. In the particular case of cell growth, the different levels of biological organization are represented by different data abstractions. For instance, the cell can be represented by a centroid or even a connected component[17,27]. The recorded images at every time point of the bacterial growth result in such time-lapse data. They are employed to track each cell and ultimately to construct cell lineages.

### 0.1.3 Bioimaging: Insight into the microscale

Bioimaging relates to methods that non-invasively image all levels of biological self organization, from molecules to human organs. To portray biological processes, from sub-cellular structures, to entire cells, to tissues, to entire multicellular organisms, biological specimens are imaged using a variety of imaging sources, among others: electron, positrons, light, fluorescence, ultrasound, X-ray, magnetic resonance[28,29,30,31]. These imaging sources are employed with imaging modalities, such as positron emission tomography (PET), single photon emission computed tomography (SPECT), optical imaging, and magnetic resonance imaging (MRI). Such modalities differ in spatial resolution, depth penetration, and detection sensitivity. For example, in clinical studies, imaging intracellular compartments, cells, and tissues enables more accurate diagnosis and treatment of disease, respectively.

Bioimaging integrates a wide range of applications, coupling technologies, such as flow cytometry, functional magnetic resonance imaging (fMRI), or functional photoacoustic microscopy, and tomography[32,33,34,35]. In cell biology, flow cytometry is employed for cell

counting, cell sorting, biomarker detection, and protein engineering. It allows simultaneous multi-parametric analysis of the physical and chemical characteristics of up to thousands of particles per second.

In the case of optical imaging, the aim is to produce a picture of the activities of biological molecules, cells and tissues. It is achieved by tagging the specimen with different emitting fluorescent probes and observing their unique colors to identify biological activities. For example, to quantify ion or metabolite levels and to measure or localize molecular interactions.

By integrating the aforementioned technologies and tackling a range of applications, bioimaging serves as means to investigating the living. The imaging step is of paramount importance to study prospect changes in the imaged specimen (e.g. in the environment of cellular growth), in turn leading to an eventual understanding of the biological processes. I briefly present two example applications in bioimaging for dedicated tasks: gene therapy[36] and nanoparticle-based imaging for cancer research[37,38].

First, gene- and stem cell-based therapies have been known to have the potential to treat a variety of diseases. In this example, researchers have identified the function of individual genes in human cells thanks to time-lapse imaging. They have accomplished this great task by perturbing each of the 21 000 human protein-coding genes using short interfering RNA molecules (siRNA), and by then observing their effects on the fluorescence profile of the labelled chromosomes over a span of two days[36]. This method enables the detection of basic cell functions such as cell division, proliferation, and migration.

Second, in cancer imaging, the ultimate goal is the development of an imaging probe that is sensitive enough to find tumors in the early stages of disease. Recent developments in bioimaging include three-photon imaging[28], three-dimensional super-resolution microscopy[39], and nanoparticle-based imaging[37,38]. The latter strategies result from a strong interaction between molecular biology and bioimaging. Once the nanoparticles design is perfected, their injection in the tissue of interest ultimately leads to its incorporation in cells. This is referred to as cell targeting, where the cells become self-reporting, e.g. for the metabolite in question, hence are clearly seen when imaged. Such strategies have been used to study cancer, in particular by imaging angiogenesis, lymph nodes, and tumor microenvironment. Moreover, they are crucial in helping image guided surgery, minimally invasive therapy, and image-guided drug delivery and release.

### 0.1.4 Problem statement

In the context of this project, recording cell growth results in time-lapse image data. In the case of long term experiments, phototoxicity occurs and the signal-to-noise (SNR) decreases, preventing a robust cell segmentation. A sufficient fluorescence intensity and high acquisition frequency (temporal resolution) are both necessary for reliable cell segmentation[40]. However, most software tools for single cell tracking and quantification are either specifically designed for a single cell type and/or image acquisition modality, or are not robust enough to deal with the strong image variability[22,41,42]. Errors in automatic approaches at the segmentation step result in the distortion of whole cellular pedigrees. Moreover, the diverse set of experimental conditions and constraints lead to poor performances, hence requiring manual tracking and data curation[43].

Pushed by the desire for automation and high accuracy, software tools that employ the single cell paradigm, are adapted to specific data sets and/or particular cell types. Nevertheless, single cell based approaches are not adapted to all tasks. For example, in live cell imaging of somatic cells, the study of cellular reprogramming raises the questions of: what happens during reprogramming and when does it occur? Experts denote that in certain experiments, they 'cannot distinguish between an early stochastic event versus the existence of a predetermined subset of cells that are in some way primed for reprogramming'. Additionally, provided state of the art methods, they could not trace the origin of a subset of different colonies[44]. As indicated by a review on synthetic biology, novel strategies that focus on cellular context are a must, so as to accomplish tasks using cell populations rather than individual cells[15].

The problem inscribes itself in a context that spans from the particularity of the *S. meliloti* bacterium and the study of its growth from the mother cell, to subpopulations, to an isogenic bacterial colony. The resulting time-lapse image data provides insights into how the isogenic bacterial population adapts to environmental changes, which raises an array of questions: Why are there differences in behavior for bacterial cells of *Sinorhizobium meliloti* that share the same genetic material? Under what conditions, and when does it occur? These questions raise the problem of cellular context, which I formulate as follows: **How do we reliably take into account the cellular context to follow cell-to-subpopulation, subpopulation-to-subpopulation events within a colony?**

### 0.1.5 Thesis statement

The investigation of a higher level of biological self organization is motivated by biological questions, e.g. cellular stress response. If software tools for single cell tracking and quantification fail, manual data curation requires substantial computational support, is time consuming, and error-prone. Moreover, in high-throughput and/or long term experiments and/or highly variable experiments, the single cell paradigm (segmentation, tracking, lineage construction) is neither adapted to answer biological questions in a timely manner nor to address higher levels of biological self organization (i.e. subpopulations and colonies). Such a case scenario occurs when one of the three steps of the paradigm fails. In this thesis, a daunting combination of high values for different data properties (e.g. noise, cell density) hinders the usage of this paradigm. Due to the particularity of the image data presented herein, it fails at the segmentation step. To reliably take into account the cellular context and ultimately follow subpopulations, I claim that we do not really need to know the fate of every single bacterial cell. I develop a tailored solution, where adapted data abstractions are derived from the raw data using a novel framework. Such abstractions are biologically driven, and the framework relies on an algorithm capable of handling this task. The latter successfully identifies and tracks changes for similarly fluorescing subpopulations across space and time.

## 0.2 Thesis overview

The methods and analyses presented in this thesis aim to analyze colony growth, ultimately at the subpopulation level. First, I present and explain the aspects of diversity and dynamics in cell colony development, and situate this thesis at the intersection of microfluidics, biology, bioengineering, bioimaging, and bioimage informatics. Afterwards, I elaborate on the how such aspects influence the data properties, hence the predictability of the resulting image data. This motivates adaptive approaches and raises an array of questions for the data at hand.

In chapter 2, I tackle the different scales of analysis and the diversity and dynamics of the time-lapse image data. This data depicts the behavior of *S. meliloti* bacterial cells over time and under controlled conditions. It is referred to as biomovies. I lay the context in which the data is produced and the reasons that make it challenging by explaining both the image acquisition step, and presenting both biological (real) and simulated (synthetic) data,

respectively. The latter comprises categorizing the biomovies using the aforementioned data properties, particularly the five data properties: cell shape diversity, cell density, cell count, spatial resolution, and noise. The simulated data or synthetic biomovies helps us establish a ground truth. Whereas real biomovies are challenging since they exhibit high values for all of the aforementioned five properties. A bottleneck results, where both manual and automatic annotations prove difficult, which in turn motivates this thesis. In chapter 3, I present the nested model of visualization, which brings the possibility of using visualization methodology to tackling such a bottleneck. I give an account of related spatiotemporal visualizations, and examine the different visualization classes for cell live imaging. Later on, I address a unique representation of amino acids in the known domain knowledge of biochemistry. As an example, it illustrates the power of visualization methods helping users perceive and retain relevant information by employing sensible visual encoding and appropriate abstractions.

To investigate colony growth in biomovies, a series of preprocessing steps are necessary. In chapter 4, I explain, illustrate, and evaluate the first preprocessing pipeline coupled with an adaptive algorithm to correct spatial shift in biomovies. Chapter 5 follows, where I introduce the particle abstraction enabling us to follow the colony structure without delineating every individual cell. Then, I extract all particles from the raw data, track each individual particle over space and time, which results into a multitude of particle trajectories. Next, I extend a visualization method, the space time cube, by devising three color codings for particle or cell trajectories so as to better perceive spatiotemporal cell pedigrees in biomovies.

Then, I introduce the second abstraction, the patch, to weigh in contextual information (i.e. spatial and fluorescence information) enabling us to delineate subsets of the colony that showcase varying fluorescence information or behaviors by grouping particle trajectories. Next, I pioneer a modular algorithm that handles splits and merges for subpopulations, where a patch trajectory represents a subpopulation throughout space and time. Preceding its application, I validate the algorithm on the synthetic biomovies for ground truth. Its application on the bioimage data is then followed by a minimal working visualization to represent and interpret the resulting patch lineages on a frame-by-frame basis. The proposed modular algorithm provides insight into the biology of subpopulations and across experiments.

The body of work presented in this thesis was developed with varied degrees of conceptual and technical range and depth. More importantly, it is the beginning of a long-term research trajectory into cell-to-subpopulation, subpopulation-to-subpopulation interactions with potential applications in stem cell research and cancer imaging.

*Clouds are not spheres, mountains are not cones,*
*coastlines are not circles, and bark is not smooth,*
*nor does lightning travel in a straight line.*

Benoit Mandelbrot

# 1

# Cell colony development: aspects of diversity and dynamics

Monitoring the growth of a cell colony in a time-lapse imaging experiment permits the assessment of challenging biological and medical applications at the single cell level. By coupling of key advances in different domains, i.e. microfluidics, *in vivo* fluorescence light microscopy and computational image processing, the assessment of cell colony development is rendered possible.

At the intersection of these domains, I describe in this chapter the general context of this thesis. It entails the study of cell colony development of a model organism, the *Sinorhizobium meliloti* bacterium, by coupling imaging protocols and computational methods. The investigation of its growth occurs across different biological conditions, herein requiring a record of each condition. Different biological conditions encompass two aspects: diversity and dynamics of the image content in each condition. To tackle such data, I define the different data properties so as to consider both aspects.

## 1.1 Microfluidics: A multidisciplinary field

Microfluidics is at the intersection of engineering, physics, chemistry, biochemistry, nanotechnology, and biotechnology, with practical applications in the design of systems in which low volumes of fluids are processed to achieve multiplexing (i.e. processing of simultaneous signals), automation, and high-throughput screening. Four main reasons motivated the domain of microfluidics. First, the existence of methods to create flow configurations at a very small scale, that is the order of hundreds of microns, and smaller[45,46,7]. Second, the rapid developments to detect small quantities and manipulate very small volumes[47,48,49,50]. Third, the quest for affordable and portable devices that are able to perform simple analytical tasks in precise and controlled conditions. Fourth and last, the potential to conduct fundamental experiments in multiple domains, i.e. physics, chemistry, biology. These reasons make microfluidics the ideal tool to study the microscale[51].

### 1.1.1 The Biology: Bacterial growth

The early principles of fluid mechanics in colloid science – the study of a colloid, i.e. a homogeneous non-crystalline substance consisting of large molecules or ultramicroscopic particles of one substance dispersed through a second substance – were first adapted to plant biology[5,52,53]. Advances in microfluidics technology revolutionized molecular biology tasks, DNA analysis, and proteomics. The fundamental idea of microfluidics-based devices is to integrate assay operations such as detection, sample treatment, and preparation at the microscale. Applied to biology, microfluidic systems grant a diverse set of example applications: microenvironmental control, precise concentration gradients, fast temperature control, tissue culture, plant on a chip, single cell studies, etc[54,55]. Microfluidic cell culture devices have been used for applications such as tissue engineering, drug screening, cancer studies, stem cell proliferation and differentiation, and many other studies.

In the particular case of bacterial cell studies, conventional culturing techniques, bacterial proliferation, and high density are limited by the depletion of nutrients and the accumulation of metabolites in the medium[8]. By employing a microfluidics device, the bacterial cells are cultured in chemostatic and thermostatic conditions in an array of microscopic chambers, permitting cell populations to reach extremely high densities. Thanks to microfluidic systems, it is possible to deliver continuous nutrient supplies for long term cell culture. This offers many

opportunities to mimic cell-to-cell and cell-to-extracellular matrix interactions, provides the means to monitor cell colony growth, and analyze cell responses to gradient concentrations of biochemical signals (e.g. growth factors, antibiotics, hormones) in a detailed manner[56,13,57,54].

### 1.1.2 The model organism: *Sinorhizobium meliloti*

In the scope of this thesis, I focus on experiments that employ microfluidic devices as heterogeneous environments for a particular bacterial microorganism: *Sinorhizobium meliloti*. This model organism is a soil bacterium that forms nitrogen-fixing nodules on the roots of certain genera of leguminous plants. *S. meliloti* is a gram negative bacterium with a thin layer of peptidoglycan between two membranes, also referred to as diderms (e.g. *E. coli*). The nodules it forms grants it to be a symbiont, where both the bacteria and the plant are in a mutually beneficial relationship (i.e symbiosis). This symbiosis led to investigations of the molecular aspects of pathogenic and symbiotic plant-microbe interactions. Moreover, it has been studied for its gene regulation and phenotypic heterogeneity[24,26]. These studies are not possible without employing imaging protocols. Hence, bioimaging provides us with records of each experimental condition, ultimately resulting in different experiments.

### 1.1.3 From Bioimaging to Bioimage Informatics

Bioimaging is the domain knowledge of employing imaging technologies (e.g. microscopy, ultrasound) dedicated to the understanding of life at the different scales of biological levels of organization. The human body for example encompasses four levels of organization: a cell, a tissue, an organ, an organism. In the context of this work, I address bacterial growth and its corresponding levels of self-organization. I refer to these levels as the scales of colony development (c.f. chapter 2). With the advent of technologies that permit advancements in standardization as well as automation in the laboratory, the aim of bioimage informatics is to help research in the following steps: acquisition, analysis, mining, and visualization of images produced by imaging technologies. To do so, bioimage informatics employs novel computational methods and techniques that tackle challenging and significant biological problems[58]. In turn, bioimage informatics provides an understanding of the diversity and the dynamics of the recorded biological processes on the aforementioned scales of organization. Such methods

implicitly rely on the diversity and the dynamics of data properties. I define both in the following sections.

## 1.2 Diversity: A multiplicity of properties

In the particular case of time lapse imaging data for cell studies, I report five data properties, and briefly describe each property:

- cell count: the quantity, or the total number of observable cells
- cell shape diversity: the morphology, the external form, or outline of cells
- cell density: the quantity of cells per spatial unit of the image
- image noise: signal fluctuations that obscure, or do not contain meaningful data
- resolution: the degree of detail visible in an image.

The data properties vary from experiment to experiment, from cell to cell. I limit the scope to bacterial cells, herein lies my interest. Since many of the aforementioned properties are interdependent, e.g. cell count and cell density, I focus on the three most important properties: shape diversity, image noise, and image resolution.

First, I report cell shape diversity for bacteria. As a data property, it varies from a circle-, to rod-, to filament-shaped like cells; as seen in Figure 1.1. Such variation not only impacts how well the human eye can distinguish a cell from another, but also the generalizability of available software tools to delineate or segment each individual cell. Second, I tackle image noise, as a result of multiple factors. In the particular case of light microscopy, and in a microfluidics setup, I present a couple of possible factors that contribute to noisy images: (a) A pixel size that is smaller than the optical resolution. This is possible when employing super-resolution microscopy technologies with a resolution limit that is higher than the optical limit. In such a scenario, image resolution is inextricably linked to image noise. And (b) possible focus shift due to vibrations and/or heat over time. The electronic instrumentation used for the acquisition may create heat and/or vibrations. This results in background instability, also known as spatial shift in sample images.

Third and last, I cover image resolution. Image resolution depends on the employed microscopy technologies to image the specimen. The latter range from light microscopy, to super-resolution microscopy, to differential interference contrast microscopy, to fluorescence confocal microscopy[10,59].

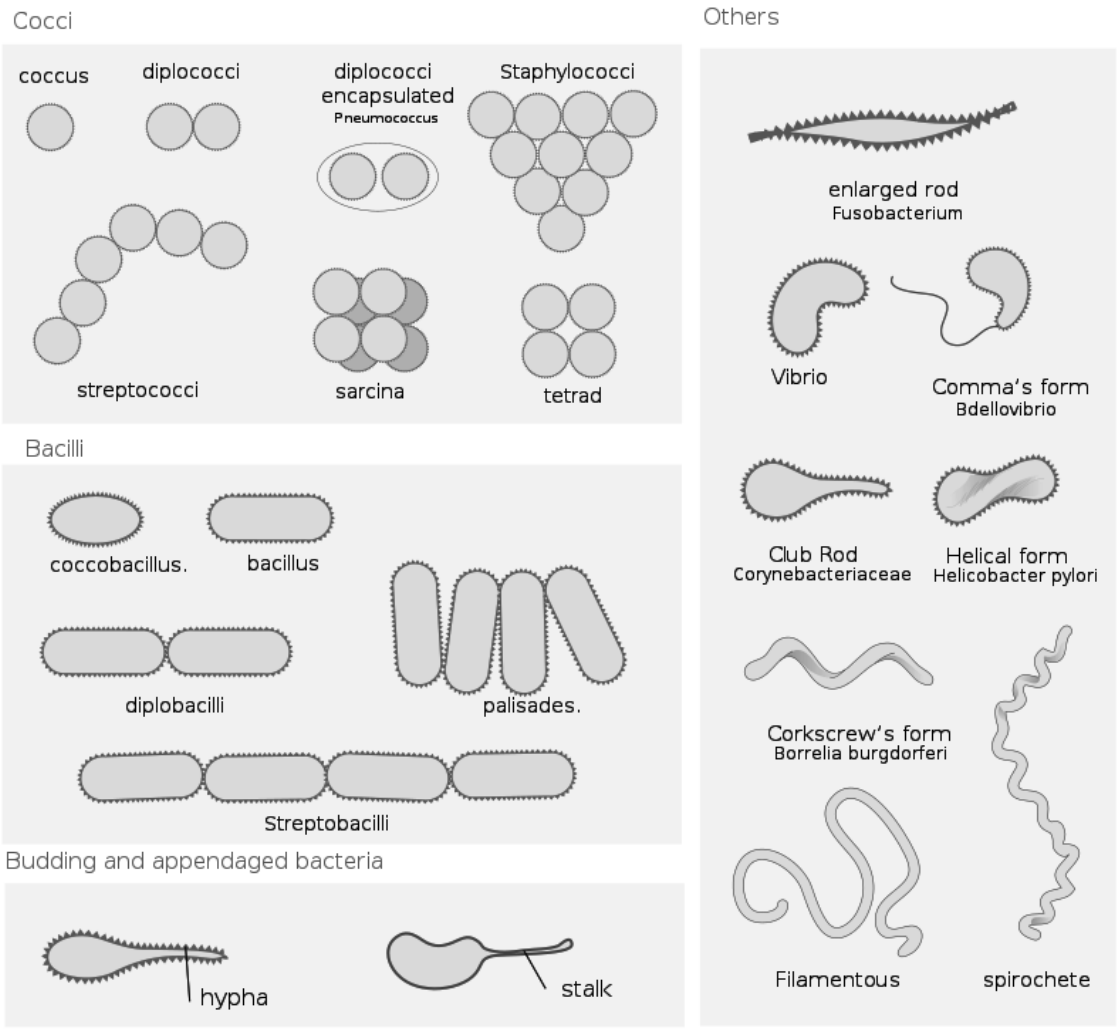
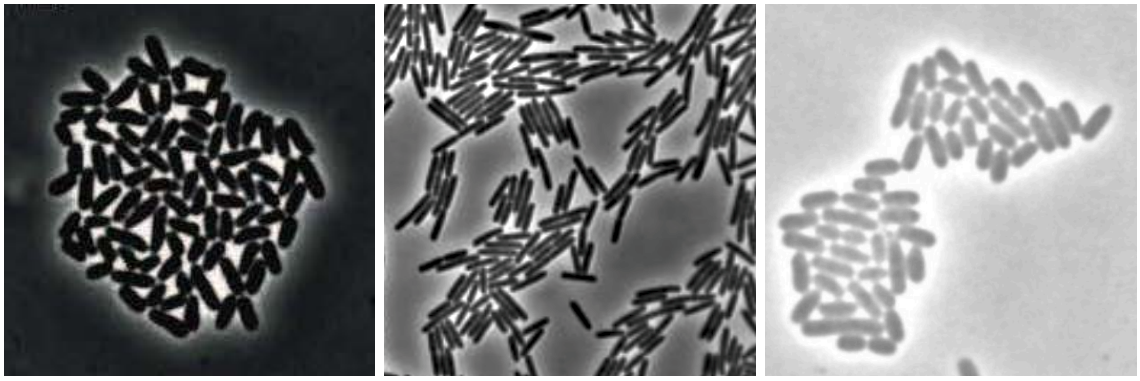



**Figure 1.1:** Bacterial morphologies. Bacterial shapes range from cocci, to bacilli, to budding and appendaged, to other bacteria.

## 1.3 Dynamics: The forces which vary data properties

The forces that stimulate cell growth directly affect data properties and result in highly variable data. Such variability comprises varying combinations of data properties with varying values (low, moderate, high). Figure 1.2 illustrates few examples of such dynamics for different microorganisms.



**(a)** *Corynebacterium glutamicum* (120 nm/px)

**(b)** *Pseudomonas aeruginosa* (64.5 nm/px)

**(c)** *Escherichia Coli* (not stated)

**Figure 1.2:** Phase contrast sample images of different bacterial colonies. The caption presents the studied species and the spatial resolution in parentheses. (a) Adapted from Grünberger et al.[56], (b) Adapted from Vallotton et al.[60], and (c) Adapted from Wang et al.[61].

In this chapter, I briefly presented, and defined the different domains in which I situate this thesis. From microfluidics, to its application in the biology of cell growth, to the computational domain of bioimage informatics that addresses such image data, to the diversity and dynamics that arise from such experiments. At the microscale, cell colony development is controlled with minute precision using microfluidics technologies. The dynamics of the diverse set of data properties hinder the predictability of the produced data, and motivates data-driven and versatile approaches.

*We may have knowledge of the past but cannot control it, we may control the future but have no knowledge of it.*

Claude E. Shannon

# 2

# biomovies: a particular time lapse image data

Time-lapse imaging of cell colonies in microfluidics chambers results in a novel and challenging category of bioimage data, namely biomovies, showing the behavior of cells over time under controlled conditions.

This chapter presents the different scales of analysis, and the diversity and dynamics of the particular time-lapse image data at hand. The time-lapse image data is referred to as biomovies and are at the core of this thesis. Biomovies showcase the growth of a single *Sinorhizobium meliloti* bacterium into a colony. I first describe the particular imaging technologies used to record these biomovies. Then, I present the diversity and dynamics of these biomovies using the predefined data properties (see Chapter 1). I address their peculiarity and the consequent bottleneck, where all data properties have high values and renders the analysis of these biomovies a complex task.

## 2.1 Bioimaging of microfluidics experiments

The imaging experiments are carried out with a high level of automation and standardization by employing a microfluidics device to host the growing bacteria and coupling phase contrast microscopy with total internal reflection fluorescence, or TIRF, for the imaging, respectively. To record the biomovies of the *S. meliloti* bacterium, a bioimaging system is employed and is described as follows. I refer to these biomovies as original biomovies, to denote the cell colony development.

In each biomovie, a micro-colony grows on a flat plane between two membranes that fit onto the microplate of the microfluidics device. Such membranes are designed to prevent bacterial cells from overlapping each other (i.e in the z-axis). The microplate is linked to a microfluidics pump that continually moves a specific medium and permits to control the environment in which the bacterial cells grow. Provided the microfluidics system, a single bacterium of *S. meliloti* is monitored until it becomes a colony.

Phase contrast microscopy is employed by using a 100x objective and is coupled with TIRF to record the biomovies. Phase contrast microscopy is a technique in microscopy that introduces a phase difference between parts of the light supplied so as to enhance the outlines of the imaged specimen, or the boundaries between parts differing in optical density, i.e. the bacterial colony. One image is generated every 30 minutes (i.e. temporal resolution) and is taken using a laser as incident light to reduce the noise, which increases the SNR. Each biomovie comprises one colony of finally 200 to 300 individuals. Coupling TIRF allows imaging of fluorescent molecules located close to the microplate/medium or microplate/specimen interface. This is achieved by employing an electromagnetic field for excitation of the fluorophores instead of direct illumination via light delivered by TIRF lasers. This technology relies on creating an electromagnetic field, known as the evanescent field. In this biological application, the incident light is a laser light, the interface is the microplate's plastic, and the bacterial cells are in the flowing media between the two membranes.

On occurrence of total internal reflection, a portion of the energy of the incident light is converted into an electromagnetic field, which then passes through the specimen(s) at the interface. This electromagnetic field excites the fluorophores and permits imaging sensors to capture the fluorescence. This occurs in particular conditions and depends on many factors: the laser's angle of incidence, its wavelength and the refractive indices of both the specimen,

and the total reflection of the incident laser. To record the biomovies in this work, my collaborators employ high power laser light to create sufficient energy for the excitation of fluorophores. Consequently, this reduces the frequency at which the specimen is excited for imaging and affects the specimen's viability (i.e. bacterial cells). Enhancing the image rate or temporal resolution to every 15 min without damaging the cells is possible, yet would require expensive upgrades (e.g. microfluidics pump). In principle, the resolution limit of light microscopy is about 200 nm. Only with super-resolution microscopy technologies a higher resolution can be achieved. The presented biomovies are based on high resolution microscopy with a 2000 nm limit, where the pixel size is smaller than the optical resolution[62]. In this particular endeavor, I investigate biomovies of living *S. meliloti* bacterial cells at different biological scales of development.

## 2.2 Scales of cell colony development: Cell, colony, colonies

For a comprehensive investigation, I define three scales of cell colony development. They are: the small scale (i.e. individual cell), the larger scale (i.e. an entire cell colony), and the full scale (i.e. different cell colonies). Individual cells are the building blocks of tissues, organs, and organisms (see Chapter 1). In the case of bacteria, bacterial cells grow from a single mother cell and accumulate to form a visible mass. Two reasons motivate the acquisition of such biomovies: reproducibility, where one condition is repeated, and experimentation, or screening, where multiple conditions are considered. As a result, multiple biomovies are investigated for changes in fluorescence and other properties.

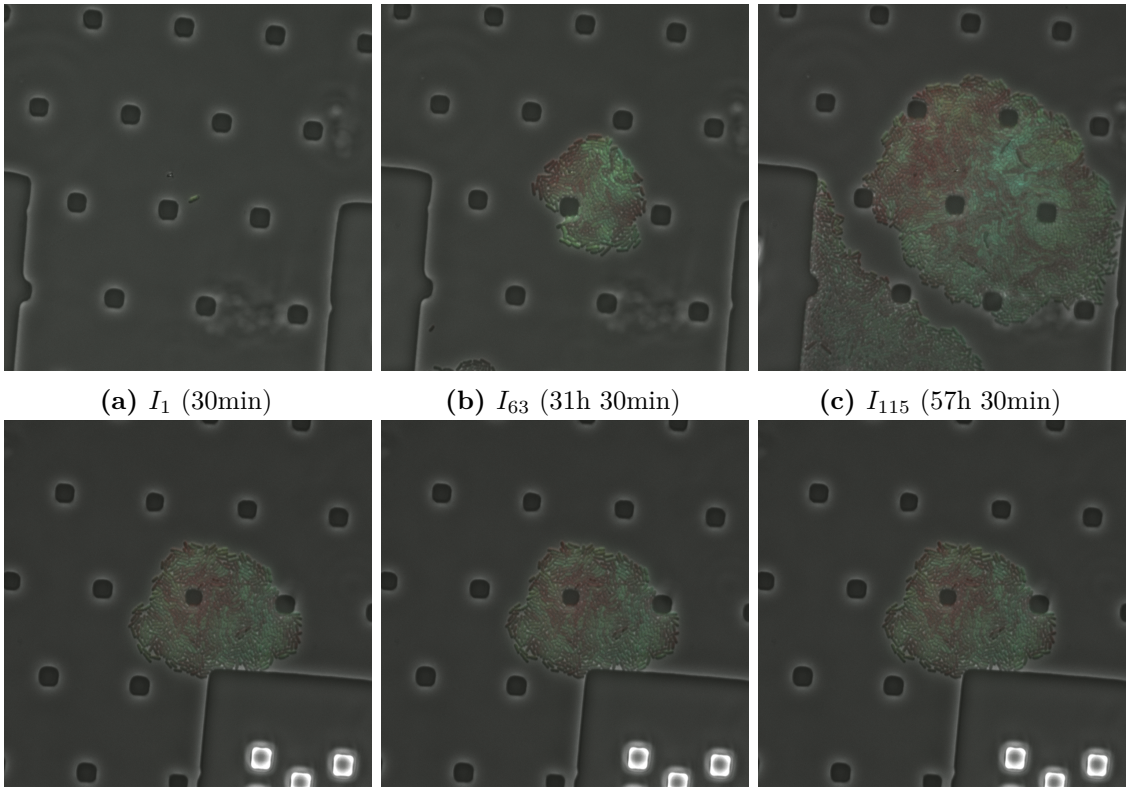
## 2.3 Diversity and dynamics of biomovies

In this work, I employ a total of nine biomovies. They comprise four biological data or original biomovies and five simulations or simulated biomovies. The latter rely on simulation software, as described in the upcoming section and detailed in Appendix A. The wet-lab biomovies were acquired using the aforementioned bioimaging system and are described below.

### 2.3.1 Biological data: The biomovies

The original biomovies arise from two different experiments: phenotypic heterogeneity and bacterial communication. Bacterial cells grow from one single mother with one particular genotype. The motivation of both experiments is to monitor the phenotype of bacterial cells in the isogenic cell populations. In the first experiment, the heterogeneity of a particular promoter is monitored. This promoter is responsible for the expression state of the galactoglucan biosynthesis gene cluster. To express this exopolysaccharide, two copies are employed: one fused to cerulean and one to mVenus coding regions, in turn representing the expression state of this gene cluster. For this experiment two biomovies result: D1, D2[26,63]. In the second experiment, the aim is to understand colony behavior and other phenomena such as quorum sensing. The activity of a promoter representing the quorum sensing state of the cell is monitored. This promoter is fused to the mVenus coding region, in addition to monitoring the activity state of one of the aforementioned promoters that is fused to the cerulean coding region. mVenus (yellow) is driven by cell division, meaning that any cell fluorescing in yellow has either recently undergone cell division, or is about to, or both. Cerulean (blue) represents exopolysaccharide production, which can only occur in the presence of sufficient quorum sensing signal, the AHLs (N-Acyl homoserine lactones). In the first experiment the constitutive T5 promoter fused to the mCherry coding region was used as marker to label viable and metabolic active cells. In this experiment the red channel represents the quorum sensing signal production and is the most heterogenous. This results into two other biomovies: D3, D4[64]. For easy recollection I refer to each experiment as follows: first, the heterogeneity experiment and second, the bacterial communication experiment. Biomovies D1–D4 were acquired by using the aforementioned bioimaging system at a temporal resolution of one image every 30 min, and a spatial resolution of 60 nm for each pixel (px). Every half hour, the bioimaging system outputs four images, one for each channel (luminance, and RGB) in the uncompressed Tagged Image File Format, or TIFF, and of size: 1004 x 1002 pixels.

Changes in fluorescence reflect changes in cell state, they are mediated by promoter-reporter gene fusions and are triggered by various factors[26]. These include stochastic effects, adaptation to environmental conditions, such as diffusible signals, nutrient availability, antibiotic resistance, or other factors. For the communication experiment, the green channel encodes an homogeneous fluorescence for cells that are alive. The red and blue channels show certain behavior in response to changes of conditions. In this case, the bacterial cells are of



**(a)** $I_1$ (30min) **(b)** $I_{63}$ (31h 30min) **(c)** $I_{115}$ (57h 30min)

**(d)** $I_1$ **(e)** $I_{63}$ **(f)** $I_{115}$

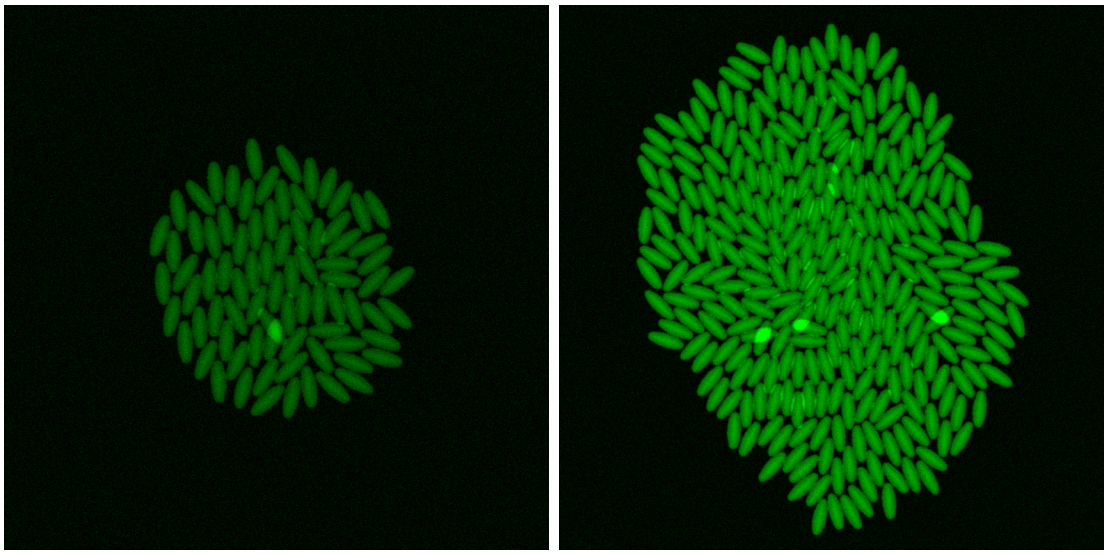
**Figure 2.1:** A set of original images in RGB color space for both openly accessible data sets: D1 (a, b, c) and D2 (d, e, f). A manual annotation is tedious and proves to be impossible before even reaching the middle time point of the time series. This is due to both a compromised sentience of individual bacteria, and sample spatial shift. D1 and D2 are accessible at `http://doi.org/10.4119/unibi/2777409` under the Open Data Commons Attribution License[63].

wild type and exposed to high concentrations of phosphate, influencing bacterial communication. As for the phenotypic heterogeneity experiment in Figure 2.1, bacteria exhibit an active fluorescence in the red channel, that is expressing a red fluorescing mCherry protein. Once a bacteria undergoes changes in expression of the monitored genes, the fluorescence profile shifts from exclusively red to a yellow-green while expressing other fluorophores. No fluorescence indicates that the cell is most likely dead or in a persisting state showing only very low metabolic activity. Disrupting the medium in which the bacteria grows permits to further analyze and understand adaptation to stress. In both experiments, spatial information is crucial to identifying and locating proximate regions with similar intensity patterns. Biologically, spatial information is crucial. It helps investigate proximate regions with similar

intensity patterns to uncover similar behaviors (e.g. antibiotic resistance in screening experiments). In these particular biomovies (D1–D4), the bacterial colonies of *S. meliloti* have high cell densities. Ordinarily, the cell shape of a *S. meliloti* bacterium is rod-shaped and anisotropic. Yet the bacteria may appear to have different shapes due to contact between cells. Such a factor contributes to limit both our human capabilities and computational methods, to successfully delineate and follow each individual cell, respectively.

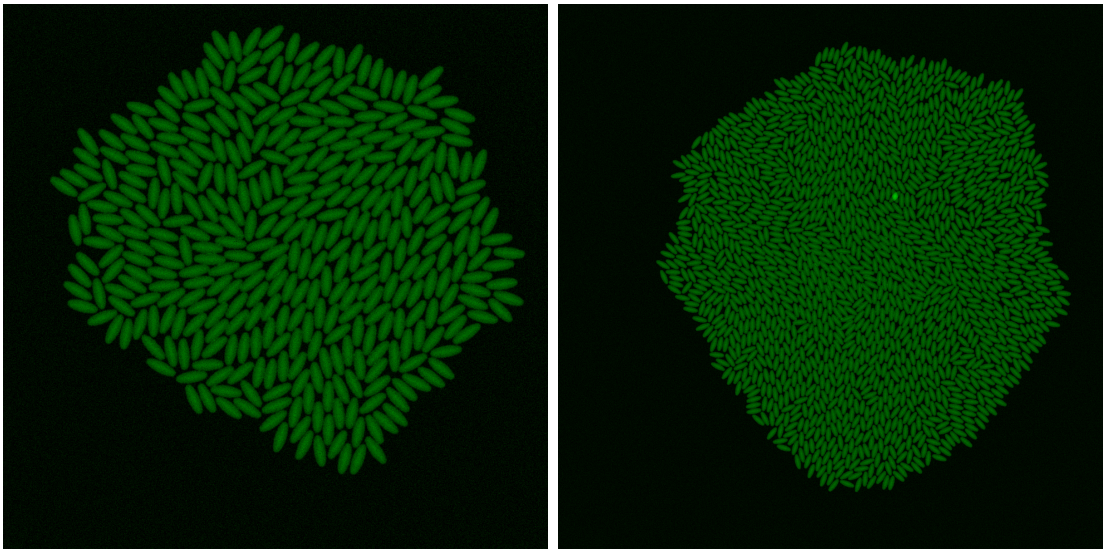
### 2.3.2 Simulated data

In order to have a test data set with a structure similar to the experimental data D1–D4, my collaborator extended a previously proposed cell simulation software for the computation of the simulated cell colony biomovies (DS1–DS5). The bacterial cell shapes are modeled as ellipses with a texture computed by a sigmoid function[65]. Moreover, cell positions are determined on a frame by frame basis by an energy minimization approach. Appendix A provides extensive details of this computation. Example biomovies are shown in Figures 2.2, and 2.3.



**(a)** DS1 - $\hat{I}_{25}$ **(b)** DS2 - $\hat{I}_{60}$

**Figure 2.2:** RGB images of simulated biomovies (final images). Enhanced images (exposure: 50%). (a) DS1. (b) DS2.



**(a)** DS3 - $\hat{I}_{63}$ **(b)** DS4 - $\hat{I}_{78}$

**Figure 2.3:** RGB images of simulated biomovies (final images). Enhanced images (exposure: 50%). (a) DS3. (b) DS4.

### 2.3.3 Data properties

Image content and background vary greatly due to both the bioimaging system's instrumentation and the rapid changes in the imaged microplate, respectively. The former is due to heat and/or vibrations in the instrumentation used to image the microplate. The latter is caused by the exponential bacterial growth. I employ and extend the aforementioned data properties in Chapter 1 to describe and categorize the nine biomovies as presented in Table 2.1. The acquired biomovies are characterized by high values for all of the previously defined data properties. Even if their acquisition can now be carried out with a high level of automation and standardization, a major bottleneck remains at the extraction of the cell lineage information.

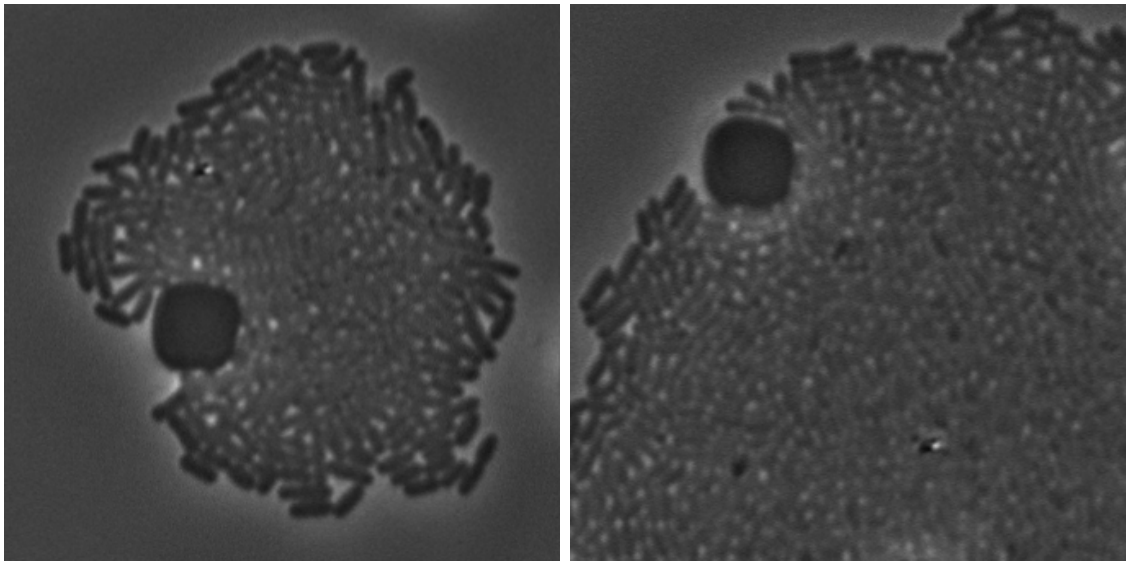
## 2.4 Bottleneck: Human and computer based limitations

Lineage extraction by human observers requires weeks of work and suffers from quality problems with low inter-/intra-observer agreement, even with substantial computational support to solve the correspondence problem. Manual annotation by human observers suffers from a

perceptual phenomenon that occurs when a change in visual stimulus is introduced, and an observer does not notice it. It is referred to as change blindness[66]. Failing to notice major differences in a biomovie while the cell growth occurs, results in fundamental limitations to analyzing such data. My collaborators produce these biomovies and require a minimum of two working days to manually analyze the data using proprietary software. Depending on cell density and movie length, manual annotation can take up to two weeks. Moreover, due to the dynamics and diversity of cell colony expansion, it is impossible to use an optimization criterion that can be solved. This is reflected by the fact that the biomovie data cannot be adequately handled by previous automatic imaging methods. Provided the data properties, it is difficult to know how diversity and dynamics 'look like', especially for a computer. To derive information from such complex data, the domain knowledge of visualization provides useful leads and hints.

| **Data properties** | **Original biomovies** | **Simulated biomovies** |
|---|---|---|
| datasets/biomovies | 4 sets (D1–D4) | 5 sets (DS1–DS5) |
| channels | RGB | 4 sets green-only, 1 set RGB (DS5) |
| image count | N = 115 (D1, D2), 44 (D3, D4) | N = 25, 60, 63, 78, 76 |
| hours of recording | 57.5h (D1, D2), 22h (D3, D4) | varying times |
| spatial resolution | 60 nm/px (high) | varying (low - moderate) |
| experiments | 2 experiments w/ 2 conditions each | 5 simulations |
| cell organism | *S. meliloti (in situ)* | cell model *(in silico)* |
| cell count | ∼300 cells (D1, D2), 80 cells (D3, D4) | vary from ∼70 to ∼400 cells |
| cell shape diversity | high variation (from rod-shape to contiguous cells) | low variation (elliptical or oval) |
| cell shape size | high variation | no variation |
| cell density | high density | high density |
| cells in contact | touching cells (no overlay) | touching with few overlays, no touching |

**Table 2.1:** Data properties for the four biological, and the five simulated biomovies.



**(a)** $I_{59}$ **(b)** $I_{115}$

**Figure 2.4:** Phase contrast images of biomovie D1 depicting the challenge at hand. (a) D1 - $I_{59}$. (b) D1 - $I_{115}$ Enhanced cropped views at different time points (contrast: 10%, exposure: 30%). The grown colony in (a) and (b) show that each individual bacterium is indiscernible from the neighboring one, especially in its center. This is due to high cell density and cell count, hence leading cells to touch, which results in high cell shape diversity and strong noise.

In this chapter, I presented the data at hand and its particularities for both biological and simulated biomovies. The biomovies are described by a diverse set of data properties (e.g. cell density) and scales of cell colony development which are intrinsically depicted throughout space and time. Simulated biomovies help us establish a ground truth in future steps of the analysis. Tackling the original biomovies requires us to consider the diversity and the dynamics of five data properties: cell shape diversity, cell density, cell count, spatial resolution, and noise. High values for all of these properties result in a bottleneck, where both manual and automatic annotations prove difficult. The coupling of such a bottleneck and the inability to know how diversity and dynamics 'look like' across different experiments motivate this thesis.

*The greatest value of a picture is when it forces us to notice what we never expected to see.*

John W. Tukey

# 3

# Visualization: a means to understanding and discovery

A visualization is a visual representation of an object, situation, or set of information. It augments human capabilities and help them carry out tasks more effectively. In the context of this thesis, the domain of visualization provides a methodology that could be adapted to manage and know how diversity and dynamics look like.

This chapter covers the nested model of visualization, related visualizations from Minard's spatiotemporal graphic, visualization methods for live cell imaging, and other domains where the data and design space are well established. It ends with the peculiar example of the biomovies, where we present the challenge we are faced with.

## 3.1 The nested model: Relationship and meaning

Based on the nested model of visualization design and validation[67,68], an analysis framework of four levels can be defined: domain, abstraction, idiom, and algorithm. Firstly, a do-

main denotes a situation in which domain-specific users are interacting with the visualization (i.e. target users). Secondly, an abstraction translates from the specifics of a domain to the vocabulary of visualization. It encompasses the data abstraction and the task abstraction. Thirdly, the idiom encompasses a visual encoding idiom and an interaction idiom. To link the visualization to the domain and empower the user, three questions are formulated at the levels of abstraction and idiom. On one hand, the data and task abstractions refer to the what is shown and why is the user looking at it, respectively. Often, a data abstraction is transformed data. In general, it refers to deriving new data elements that are essential to carry out the task without presenting domain-specific details. On another, the idiom refers to the how the visualization is shown. Lastly, the algorithm stands for an efficient computation that enables the extraction of a data abstraction, to ultimately visualize it.

## 3.2 Related spatiotemporal visualizations

In the context of this work, I only focus on spatiotemporal related visualizations. Spatiotemporal visualizations present changes of information in space and time. Such visualizations have a natural advantage of revealing overall tendencies and movement patterns. In the case of biomovies, information unravels itself frame by frame at a rapid pace. It is therefore hard to follow the narrative thread to uncover the biological growth patterns. 'Designing for narrative is very different from designing for information seeking'[66]. It is challenging to visually narrate such temporal data while preserving the original storyline, yet visualizing temporal data in a static display is possible. In the following sections, I present related spatiotemporal visualizations, starting with Minard's graphic in which a series of data (location, temperature, etc) is mapped onto a geographical map.

### 3.2.1 Aggregate plots

Aggregate plots range from aggregating classes, or features, to representing the spatiotemporal information where data is selected by relevant data attributes and filtered by specifying a feature value or interval of values, respectively. From Minard's graphic of Napoleon in 1812 to Sankey's diagram of the first energy flow diagram in 1896[69], the numerical data is represented either on a map or on the steam engine's blueprint. Sankey diagrams are aggregate plots, where information is rendered accessible at the large scale or population. They both

represent a flow chart, or a flow diagram in which the sequence of movements, or actions of people, or things are depicted in the complex system, or activity in which they are involved. Napoleon's march in Fig. 3.1, is one of the most prominent examples of spatiotemporal visualizations. It reveals information without superfluous details, E. Tufte refers to it as the 'best statistical graphic ever created'[70]. After Czar Alexander of Russia refused Napoleon's embargo, Napoleon gathered a grand army to attack Russia in June 1812, also referred to as Napoleon's march.

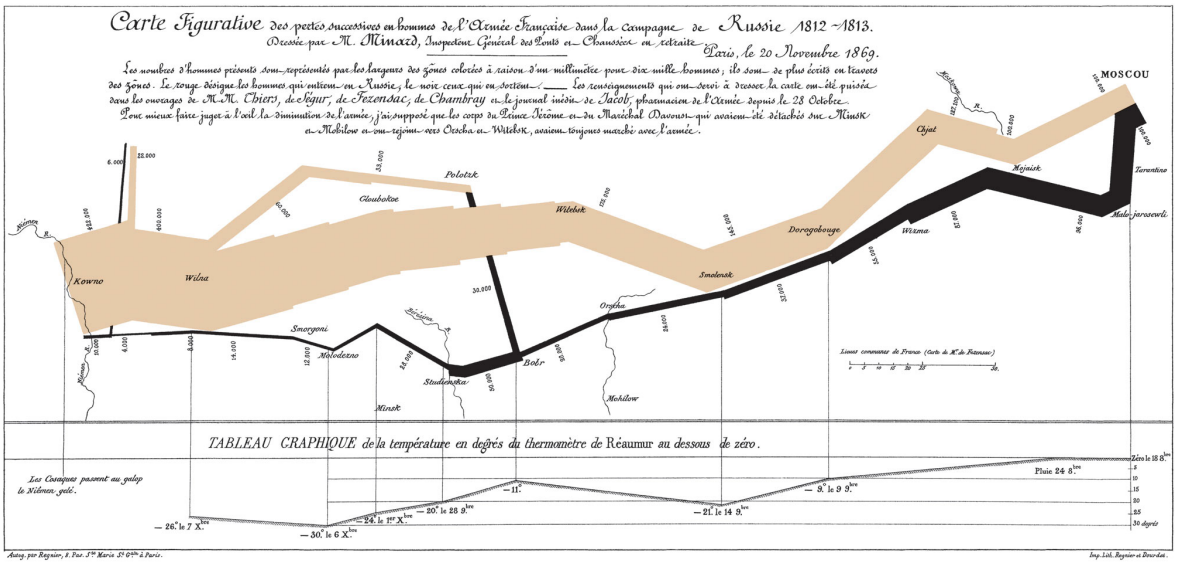
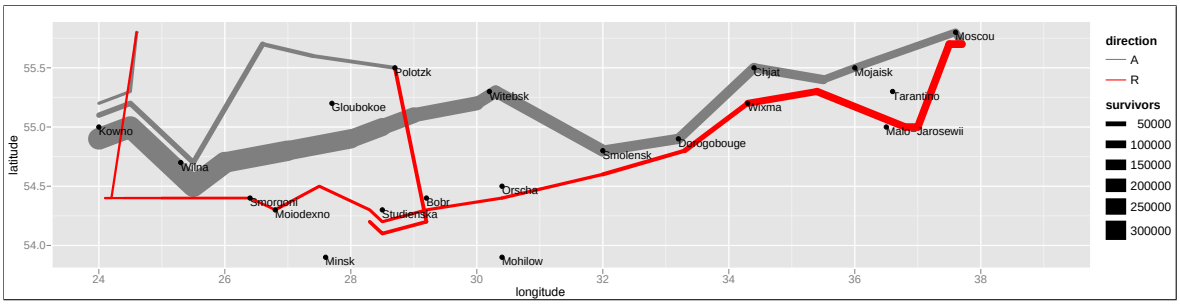
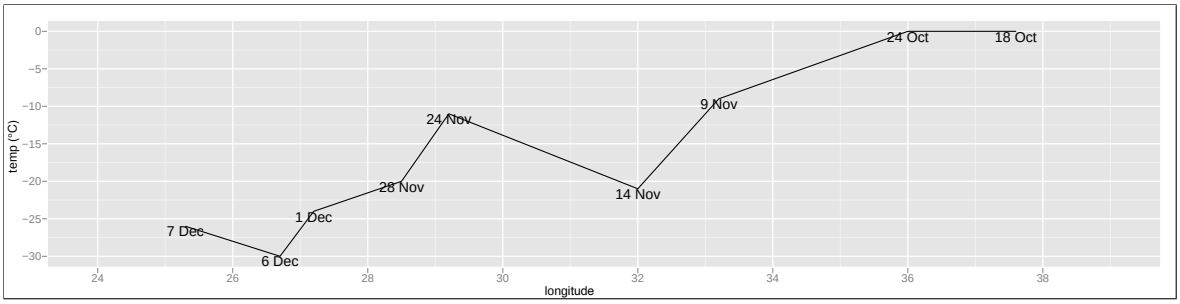



**Figure 3.1:** Charles Minard's graphic of Napoleon's march. Source: `https://www.edwardtufte.com/tufte/minard`

Figure 3.1 describes the outward progress, and returning paths of the army in a spatiotemporal manner, and employs six types of information: location, time, temperature, the course, the direction of the army's movement, and the number of troops remaining. The widths of the gold (advancing) and black (retreating) paths represent the size of the force, one millimeter to 10 000 men. Geographical features (space) and major battles are marked and plummeting temperatures on the return journey are indicated along the bottom, respectively. The visualization clearly tells the story of losing such a grand army, which set out from Poland with approximately 430 000 soldiers, where only 100 000 reached Moscow and only 10 000 returned. As men tried and mostly failed to cross the Bérézina river under heavy attack, the width of the black line halves: another 20 000 or so gone. I decomposed this visualization into two graphs showing the march in space and the temporal evolution of the temperature for the




**(a)** The geographical evolution of Napoleon's march. The gray and red correspond to advancing and retreating, respectively. The line thickness encodes the number of survivors. Major cities along the paths are indicated.




**(b)** Variation of the temperature in degrees Celsius mapped onto the longitude, for the march's duration.

**Figure 3.2:** Decomposed graphics of Minard's visualization.

march's duration (see Fig. 3.2). In Figure 3.3, an example xkcd webcomic, employs a very similar flow visualization or Sankey diagrams, and shows the different interactions between all main characters of a movie.

### 3.2.2 Cell imaging visualization

In the particular domain of live cell imaging, the data is more challenging, and requires ample visualization methods to perceive the whole context. This is mainly due to the dynamics and diversity of cell growth. This heightens the variability of the outcome and lessens its predictability. Pretorius et al.[23] separated the related visualization methods in six classes (see Fig. 3.4). These classes enable users to access the data in different ways and are represented using: (a) Spatial embedding, where cells are visualized in the field of observations (2D or 3D). (b) Space-time cubes, where cell positions are mapped to the x- and y-axis, and time is mapped to the z-axis. (c) Temporal plots, where derived cell features are encoded as time

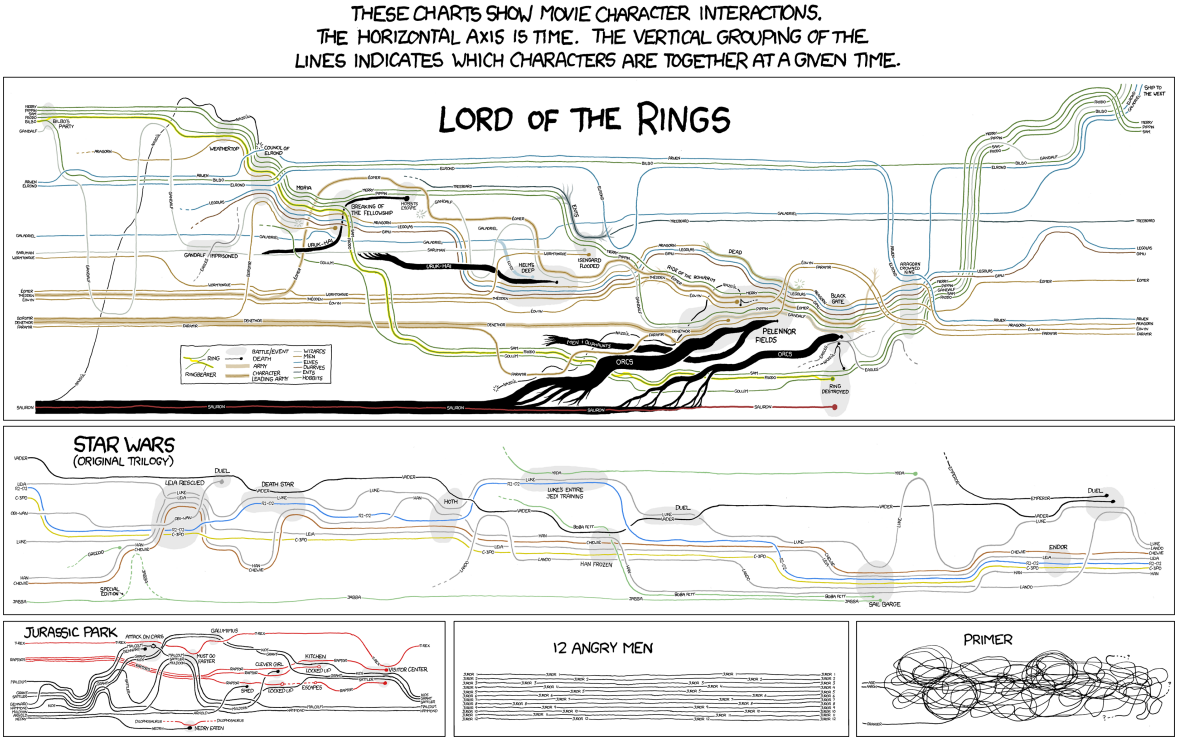



**Figure 3.3:** Movie narrative charts. The term narrative chart is used, where time is represented along the horizontal x-axis, and a sort of nominal ordering is employed for the y-axis. Source: `https://xkcd.com/657/`

series with a line function of time (i), bar plots (ii), or event sequences (iii). (d) Aggregate plots, see the aforementioned Sankey diagrams section. (e) Dimension reduction, where data clusters or classes are located by classification algorithms to lay out relationships between them. (f) Lineage diagrams, where cell lineages are shown as branching tree structures, with a typical temporal orientation, function of either the elapsed time or the successive cell generations. This comprehensive list is reported in Pretorius et al.[23] and is extensively detailed in the context of the four levels of design (see the nested model in section 3.1). For brevity, I report and discuss the most prominent related work.

In the literature, most approaches addressed a particular data modality at a particular level of detail (e.g. by using dimension reduction, or temporal plots)[27,71,72]. Moreover, most well known tools dealt with relatively scattered cells, an acceptable signal-to-noise ratio (SNR) in sample images[20,27,72,73]. Only a limited number of tools successfully handled biomovies with high cell density[14,60].

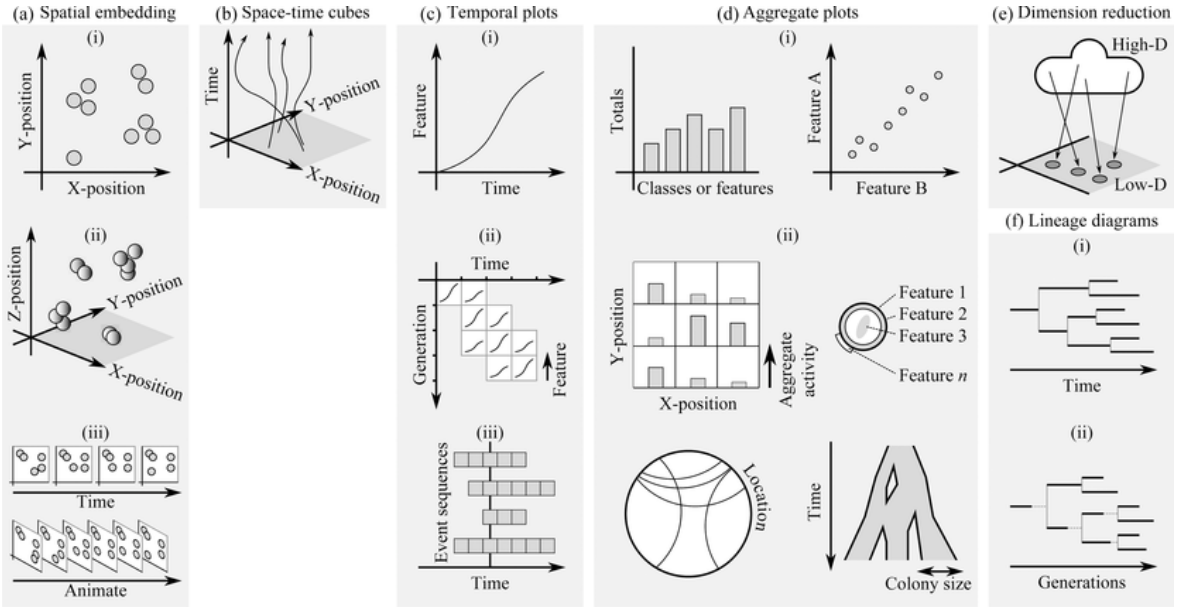



**Figure 3.4:** The six different classes of visualization methods for live cell imaging data. Figure from Pretorius et al.[23]

From related works, only a handful of tools provide support for explorative analysis across multiple coordinated views and are able to assess cell colony growth : Cell-o-pane[74] and CellProfiler[20]. As a cell lineage visualization tool, Cell-o-pane employs a range of techniques. These range from clustering cell attributes, to analyzing aggregate behaviors, to filtering, to comparing spatial and structural detail of selected lineages[74]. CellProfiler is the only notable approach that tackled comprehensively different data modalities and different levels of abstraction[20]. It provides interactive support, different types of aggregate plots, standard interaction methods (i.e. brush, select, filter, drill-down), and combines visual analysis of structural abstractions with spatial representations. The visualization methods implemented in CellProfiler range from space-time cubes (or XYT plots), to aggregate plots, to temporal plots (e.g. synchrograms, or an image sequence of an individual cell), to lineage diagrams (or lineage trees)[20]. The aforementioned six classes provide a broad and flexible range of visualization methods to represent live imaging data, its modalities and its abstractions. As stated in the survey, in the broad analytical context of live cell imaging, these visualizations are mixed with other methods of analysis[23]. The insufficiency of insights and hypotheses from visual analysis leads to combining visual data mining and non visual methods (e.g. statistics). Moreover, I argue that it is necessary to adapt some visualization methods (e.g. space-time cube) to cell-, subpopulation-, or colony- events; herein lies my interest.

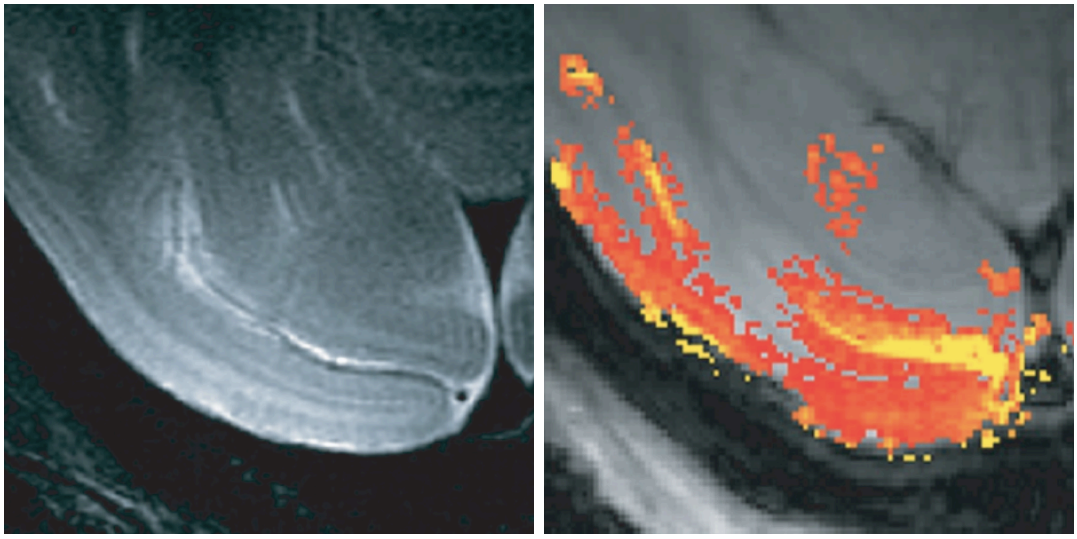
Visualization techniques exist in other domains where the data is well-known. In that case, the focus shifts to the design space, where established representations are examined or even strengthened. This is the example of brain imaging.

### 3.2.3 Functional magnetic resonance imaging (fMRI)

fMRI is a functional neuro-imaging procedure that measures brain activity by detecting associated changes in blood flow (i.e. hemodynamics). It relies on MRI, which is the most important imaging advancement since the introduction of X-rays by C. Röntgen in 1895. Since its introduction, MRI is coupled with exogenous contrast agents and is used as a diagnostic tool. Its primary usage is either to produce structural images of organs (e.g. the brain) or to provide information on the physicochemical state of tissues. The emergence of fMRI in the early 1990s, led to an upturn in related works. It started with the use of fMRI without contrast agents, to measure hemodynamics after enhanced neural activity. The first papers employed fMRI to explore functional localization and/or cognitive anatomy associated with some cognitive tasks, to examine the physiological properties of different brain structures, to study brain plasticity and a multitude of other experimental methods. When an area of the brain is in use, blood flow to that region also increases. The primary form of fMRI uses the blood-oxygen-level dependent, also referred to as BOLD contrast, which was discovered by S. Ogawa[75]. In the example of brain fMRI, regions of increased blood flow overlay the anatomical scans, as seen in Figure 3.5

Moreover, it is common to see a full brain map either in 2-, or 3-D to help visualize the region of interest in its original context. In general, the domain knowledge of brain imaging is well established (e.g. Allen brain atlas), which enables accurate analyses[76]. In spatiotemporal experiments, the data varies across the anatomical brain map and over time, resulting in image sequences that depict the same field of view over time (c.f. Figure 3.6).

There is a range of challenges for visualizing such data; particularly when the signal's location is buried deep within the brain, as opposed to a superficial location (i.e. surface of the brain)[78]. For instance, to deal with multiple superficial signals, 3D-visualizations are coupled with mirror effects, which helps users perceive all the different signals simultaneously[79]. In another example, brain symmetry is employed for topographic analysis of specific lateral events. A simplified schematic representation ensues enabling a quick insight into different conditions and events[80]. These examples clearly suggest a task-oriented design.



**(a)** **(b)**

**Figure 3.5:** Anatomical scan alongside a high-resolution fMRIs demonstrating the high functional signal-to-noise ratio (SNR) of the images. (a) Anatomical scan of the cortex using spin-echo echoplanar imaging (SE-EPI). (b) Functional SNR: red indicates low, and yellow indicates high. The yellow regions showcase the strong contribution of blood vessels. Adapted from Logothetis et al. 2002[75].

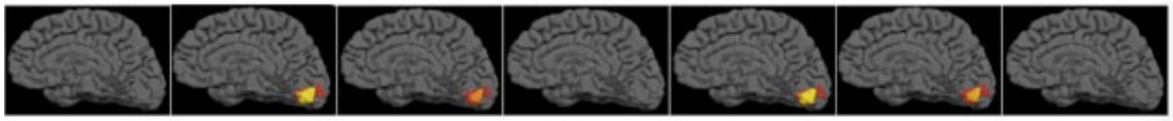


**Figure 3.6:** Spatiotemporal brain imaging of the cortical activity of a subject during a cognitive task. Combined analysis of electroencephalography (EEG) and functional magnetic resonance imaging (fMRI). Adapted from Bonmassar et al. 200[77]

In the broad domain knowledge of biology, representations are diverse and in few cases they are provided as educational mediums or even as an initiator of dialogue and engagement with the public[81,82].

In appendix B, I focus on molecular representations, in the particular domain knowledge of organic chemistry. It is described as a task-oriented visualization. I present the design space of molecular structures in the domain knowledge of organic chemistry and design my own visualization of a special class of biomolecules: the amino acids. This particular work was presented at the Information plus conference in Vancouver, Canada[83].

## 3.3 Summary

The representation of an object, a situation, or a set of information is imperative to solve a lot of real life problems. This chapter covers an important subset of spatiotemporal related visualizations in the domain knowledge of visualization. Moreover, it portrays why visualization is crucial to understand relationships between objects. In this particular endeavor (c.f. appendix B), I present a task-oriented study for the molecular representation of amino acids. This was possible by designing a data abstraction, employing a visual encoding, and ultimately representing the data. The domain knowledge of visualization provides both flexibility and accuracy by the presence of alternative representations and task-oriented visualizations.

In this chapter, I presented the specific nested model of visualization, as a tool that provides the means to address a task to ultimately visualize the data. Next, I discussed related spatiotemporal visualizations in different domains, where the data varies greatly (flow charts, cell live imaging), and where it is well established (fMRI). The applications of the nested model spans from the domain knowledge of visualization to data mining, across multiple domains (e.g. brain imaging). The latter permit users to tackle the raw data at a higher level, herein lies my interest. For the distinct case of biomovies, I employ the essential concept of visualization to tackle the data, as reported in chapter 5.

*Adaptability is not imitation. It means power of resistance and assimilation.*

Mahatma Ghandi

# 4

# Spatial shift: a hindrance to knowledge discovery

To track and analyze the development of cells, the correction of spatial shift in the image sample (i.e. registration) is a prerequisite for posterior analyses: manual annotation, automatic segmentation of dividing cells, visualization of cell growth.

> This chapter presents the first robust and data-driven method to registering time lapse images in phase contrast microscopy by finding major cues in the image space. In turn, it enables us to conduct the next step in data analysis, i.e. to extract biological information at the microscopic scale.

As seen in chapter 2, biomovies exhibit high values for all of the aforementioned properties. This is due to a non-negligible variability in both image content and background content. The background changes due to the multi-wells technology for microfluidics chambers (i.e. many conditions or experiments on one microfluidics device) resulting in multiple visual fields. The image content varies greatly due to the rapid changes in the sample (i.e. doubling phenomenon: exponential bacterial growth). Such data influences the performance of state of the art methods for image registration.

## 4.1 Related work

Promising methods relevant to different spatial resolutions have been found, yet requiring either an *a posteriori* insight of the data or an evaluation of the algorithms' adaptability for higher-resolution images. Moreover, other automatic methods, such as TurboReg[84] are designed to minimize the mean-square difference (between the target and the source image), are esteemed fast and robust. Yet such automatic solutions are unable to handle the highly dynamic image content of bacterial growth (see Fig. 2.1) without preprocessing steps and by solely relying on one metric between the consecutive images. To remain in the scope of this work, related methods are briefly reviewed (i.e. similar spatial resolution: 1 px = 60 nm). Found methods pertain to either live fluorescence microscopy of a single cell[85,86] or histochemical staining based on cellular structures[87], yet not about cell lineages on the population scale.

A range of approaches exists for cell lineage analysis, yet they do not address the registration problem explicitly[14,17,88,89] and in cases deal with sparser data[90]. In a survey of microscopy cell-lineage related work, one candidate method was found. It is an automatic approach to track and align *Arabidopsis Thaliana*'s growing sepals[91]. However, the employed data used to demonstrate its effectiveness contains comparably sparse cells and a low to moderate cell count.

## 4.2 Methods

This approach finds particular polygons, or (vi)sual (c)ues, and applies an (a)daptive (r)egistration, also referred to as ViCAR. It employs the following three steps to correct the spatial shift: preprocessing, polygon finding, and registration.

### 4.2.1 Preprocessing

As a first step in the ViCAR registration process, a customized pipeline of standard filter operations is applied to each image $I_t$, of a recorded image series, so as to reduce noise and increase the contrast between the background and the structural elements of the image. The preprocessing steps involve many constants, which are in this example set to moderate

values. These constants are chosen after conducting a sensitivity analysis, that is to vary the constants and verify their incidence on the resulting images. The whole process is illustrated in Fig. 5.9, so as to probe for particular polygons, and expand their respective shapes in the input image.

(a) RGB to greyscale transformation (Fig. 4.2a)

(b) Denoise Bilateral Filtering[92] (Fig. 4.2b)
- spatial closeness $\sigma_{\text{spatial}} = 75$
- radiometric similarity $\sigma_{\text{range}} = 75$
- diameter $\delta = 10$ px of each pixel neighborhood that is used during filtering.

(c) Contrast Limited Adaptive Histogram Equalization (CLAHE)[93] (Fig. 4.2c)
- tile size $\tau = 10^2$ pixels
- contrast limit of 2, to clip, and uniformly distribute any histogram bin above that limit.

Next, for each image $I_t$ a binary image $\hat{I}_t$ is computed to serve as a basis for finding polygons.

(d) Adaptive mean thresholding (Fig. 4.2d)
- block size $\tau = 11^2$ pixels
- a constant $c = 2$ is subtracted from the weighted mean in order to prevent noise to pop up at background regions.

(e) Dilation[94] (Fig. 4.2e): morphological operation in each image $I_t$ with a 3×3 window.

(f) Border clearing (Fig. 4.2f): it replaces all elements alongside or stemming from the borders of the binary image with background pixels.

(g) Masking (Fig. 4.2f): a binary mask of image dimensions $(r \times c)$ is initialized. It contains a circle of origin $o = (\frac{r}{2}, \frac{c}{2})$ and diameter $d = \frac{3}{5} \cdot r$ to removing any connected components external to its perimeter using a bitwise comparison.

### 4.2.2 Polygon finding

The output binary images $\hat{I}_1, \ldots, \hat{I}_t, \ldots, \hat{I}_T$ are employed to find the polygons $P_{tj}$. Each polygon has an index $j$ and a time index $t$. In each image, the border following algorithm[95] is used to obtain closed boundaries, that is, the polygons which are depicted in Figure 5.4a as connected components. For the sake of clarity, the polygon index $t$ is omitted for polygons in the next sections. Once all polygons are found throughout the time-series, they are filtered based on their individual perimeter-to-area ratio. The perimeter, area, and the ratio are defined in the following sections.

#### 4.2.2.1 Polygon perimeter

The perimeter of a polygon $S$ is:

$$S = \sum_{n=1}^{N} |C_n| \tag{4.1}$$

With the number of sides $N$ or smooth curves, equal to the number of vertices $n$, and the length of a smooth curve $|C_n|$.

#### 4.2.2.2 Polygon area

For any simple polygon, the area $A$ can be calculated:

$$A = \sum_{k=0}^{N} \frac{(x_{k+1} + x_k)(y_{k+1} - y_k)}{2} \tag{4.2}$$

With the number of vertices $n$ and the $k$-th vertex $(x_k, y_k)$. Since the first vertex of the boundary $C$ happens to also be the last vertex, this results in a summation of $n + 1$ terms where: $(x_{n+1}, y_{n+1}) = (x_0, y_0)$. Given Green's Theorem, for a piecewise smooth curve $C$ forming the boundary of a region $D$ the area $A$ is computed by:

$$A = \oint_C x \, dy \tag{4.3}$$

#### 4.2.2.3 Perimeter-to-area ratio

The perimeter-to-area ratio is used to find a particular kind of outlined polygons, referred to as visual cues. For each polygon $P_j \in P_1, \ldots, P_J$ with the number of polygons $J$, the perimeter-to-area ratio is:

$$r_j = \frac{S_j}{A_j} \tag{4.4}$$

With $S_j$ and $A_j$, the perimeter and area of a polygon $P_j$, respectively. The perimeter-to-area ratio $r_j$ is a descriptor of shape irregularity, and is polygon size dependent. If holding shape constant, an increase in size results in a decrease in ratio. Polygons are retained if and only if they satisfy the following empirically derived threshold:

$$r_j < 5 \times 10^{-2} \tag{4.5}$$

This threshold permits to consistently find particular polygons with a lowest complexity. As a consequence, the polygons found in the microfluidics data considered here are the spacers, i.e. squares and square-like structures (c.f. Fig. 5.9). In contrast, if complex polygons are found (e.g. self-intersecting polygons) they are retained only if no other polygons satisfy the aforementioned threshold. All retained polygons are referred to as visual cues.

### 4.2.3 Registration

Registration happens in a pairwise manner $I_t, I_{t+1}$, and adaptively based on the number of visual cues $J$ across all image points $T$. All indexed intervals are registered to the reference image, i.e. $I_1$.

#### 4.2.3.1 Interval adaptability

To correct for spatial shift there are two possibilities: (a) All images contain the same number of $J$ visual cues, then the computation iterates using a reference polygon as explicated in the next section. (b) Intervals of consecutive images contain different numbers $J$ and $J'$ of visual cues: In each interval, the aforementioned method in (a) is handled independently, and iteratively while using the reference polygon for registering all intervals to the first image.

One requirement to this adaptability is the minimum of two consecutive images with $J$ visual cues.

#### 4.2.3.2 Reference polygon

Image registration requires reference coordinates for correspondence among the consecutive image points of the time-series. By coupling both border clearing and circle masking, polygons that are mostly in the image centre are obtained (see preprocessing section 3.1). The first coordinate is found by ordering all coordinate pairs (along both $x$ and $y$ axes). The first visual cue has the first coordinate $x_{j=0}$ at $t = 1$ set as reference for the registration.

#### 4.2.3.3 Affine transform

From each image $I_t$, anchors points $x_t, y_t, z_t$ are extracted to apply the affine transform to $I_{t+1}$, mapping the points $x_{t+1}, y_{t+1}, z_{t+1}$ to $x_t, y_t, z_t$.

This way, the phase contrast images are transformed, and then their corresponding RGB channels. Which is similar to strategies applied in multi-tag fluorescence microscopy [96]. Once the alignment is done, the robustness of this approach is evaluated. It is conducted on preprocessed and transformed images, where only visual cues are observable, as seen in Fig. 4.4.

These anchors points are extracted from the retained polygons/visual cues. A decision is made based on the number of retained polygons $J$, three scenarios are possible: (a) One visual cue is found, an oriented bounding box (OBB) is used to retain three coordinate pairs[97]. (b) Two visual cues are found, an OBB is used for both and the first coordinate pair from each polygon along one axis is retained. (c) In the case of three or more visual cues, their respective centers are extracted.

The affine transform integrates different components of the ordinary procrustes analysis: rotation, skew, uniform scaling, translation.

### 4.2.4 Evaluation

To assess the performance of the ViCAR approach, results are evaluated by addressing both: (a) The spatial shift, by computing the pairwise root mean square difference for all $T$ images compared to $I_0$, the reference image. (b) The average elapsed time, ViCAR took to align preprocess and align one image. Results obtained with ViCAR are compared to those obtained with a Probabilistic Hough Transform (PHT) based method in Table 4.1.

(1) **Image closeness**: $\Phi$, in %, can be formulated as follows: $\Phi = 100 - (\text{rmsd} \times {}^{100}/_{r})$. Using the average root mean square difference, noted rmsd, as a measure to assess how accurate is the spatial presence of the visual cues in $I_t$ compared to $I_1$.

(2) **Performance**: elapsed computation time ($\Delta t_c$), in seconds, is computed using real system time by subtracting initial from final. The evaluation was carried out on data sets D1–D4 (see chapter 2 for more details). It ran on a MacBook Air (Mid 2013) with a 1.7 GHz Intel core i7 and 8 GB 1600 MHz DDR3 memory.

(3) **Comparison to state of the art method**: in particular, the PHT. Since it has been extensively proven to be successful[98,99] with complexity and memory requirements lower in higher dimensions. The PHT based method comprises the following steps: (a) reduce each image to a set of edges using an edge detector (i.e. Canny), (b) apply the Hough process (particularly, the PHT), (c) retain a best fitted subset of points (i.e. four points), and (d) a geometric transformation (e.g. using the least-squares method).

(4) **Visual verification**: the dataset is visualized in a space-time cube, as proposed in[23], before and after ViCAR has been applied. Using a SIFT operator[100] and a customized preprocessing pipeline, approximations for cell positions are computed in each image $I_t$. These positions were subject to the visualization shown in Fig. 4.5. The $x-$ and $y-$axes represent the original image plane. The z-axis represents the time $t$. The lowest point in time represents the first image $I_0$. The original data suggests a shift of the entire colony inside the microfluidics chamber. After ViCAR has been applied, the correct colony location and spatial distribution can be visually appreciated.

## 4.3 Results

The examples in Figure 2.1 back the necessity of preprocessing steps. Figure 4.3 shows how all visual cues are correctly aligned, for two different time points among the four aforementioned data sets (D1-D2, D3-D4). This figure is a noteworthy evidence of the adaptability and robustness of this registration approach. As reported in Table 4.1, the state of the art based approach, namely employing the PHT, resulted in correct performances on D1 and D3. Whereas, on D2 and D4, the state of the art method has proven to fail, i.e. it crashed. This is mainly due to data set variability where either the data contains no major structuring lines or a detected line disappears after an elapsed time. In the case of D1 and D3, performance results are affected by skew due to disappearing line portions. Hence, it is inappropriate to use the PHT based approach since it requires a prerequisite of the image data. To conclude, ViCAR achieved a satisfying performance, close to 100% and proved its adaptability on different data sets from two different experiments.

| | PHT based | | | ViCAR | | |
|---|---|---|---|---|---|---|
| | $\overline{\Delta t_c}$ (s) | rmsd (px) | $\Phi$ (%) | $\overline{\Delta t_c}$ | rmsd | $\Phi$ |
| D1 | 1.3 | 13.9 | 98.6 | 0.64 | $4.10^{-2}$ | 99.9 |
| D2 | — | — | — | 0.65 | $6.10^{-2}$ | 99.9 |
| D3 | 0.7 | 19.2 | 98.1 | 0.36 | $4.10^{-2}$ | 99.9 |
| D4 | — | — | — | 0.44 | $5.10^{-2}$ | 99.9 |

**Table 4.1:** Benchmark results for bacterial time series (Datasets: D1, D2, D3, D4) using both approaches: probabilistic hough transform (PHT), and visual cues adaptive registration (ViCAR). $\overline{\Delta t_c}$ is the average elapsed time per image, in seconds. The rmsd is the root means square difference in px. Images closeness $\Phi$ relies on the rmsd, see the Evaluation subsection. The PHT based approach fails due to disappearing elements of the image space crucial to the PHT based registration.

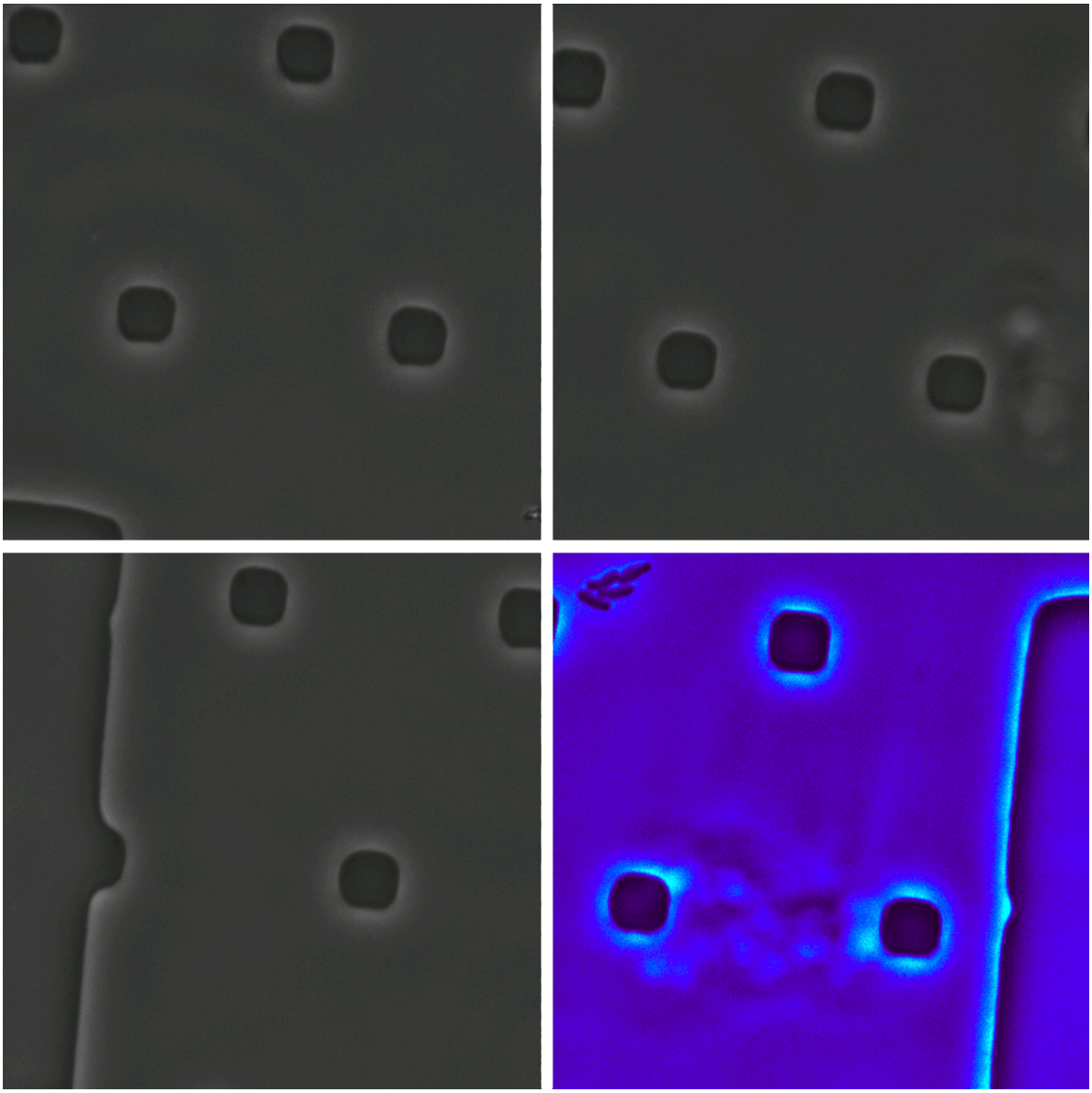
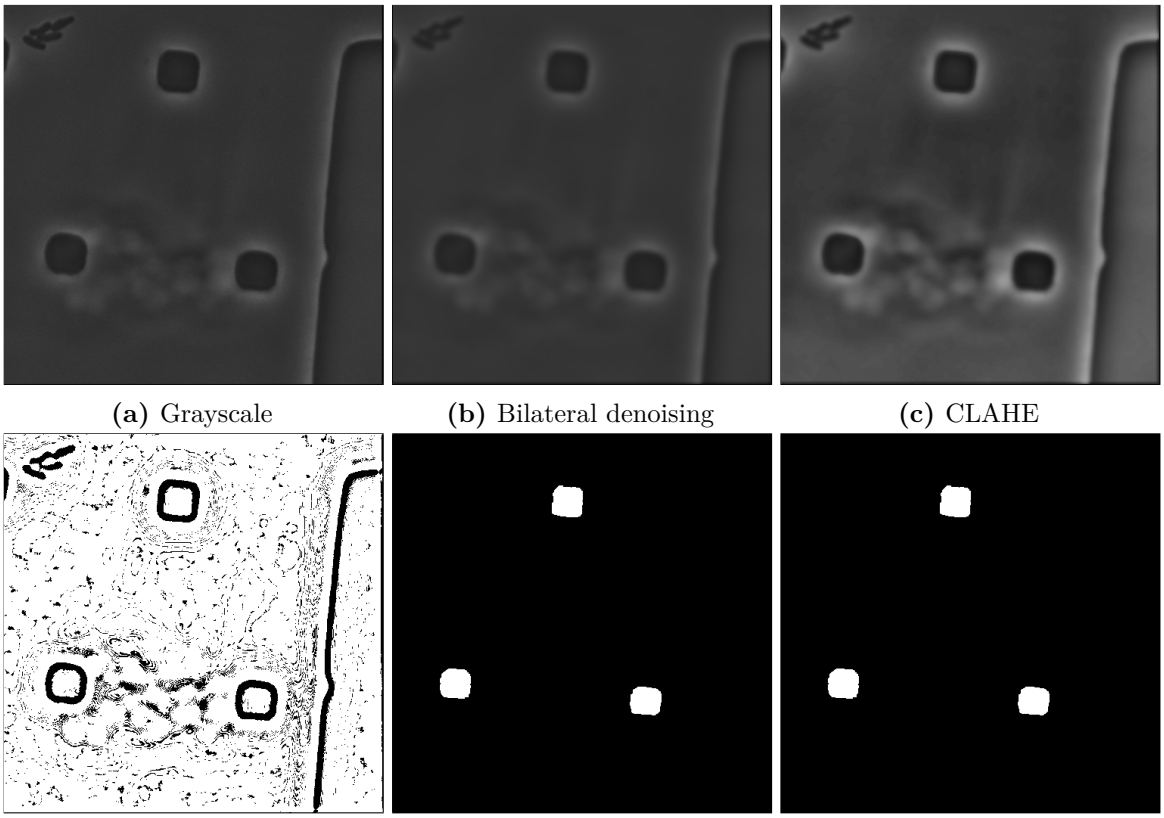


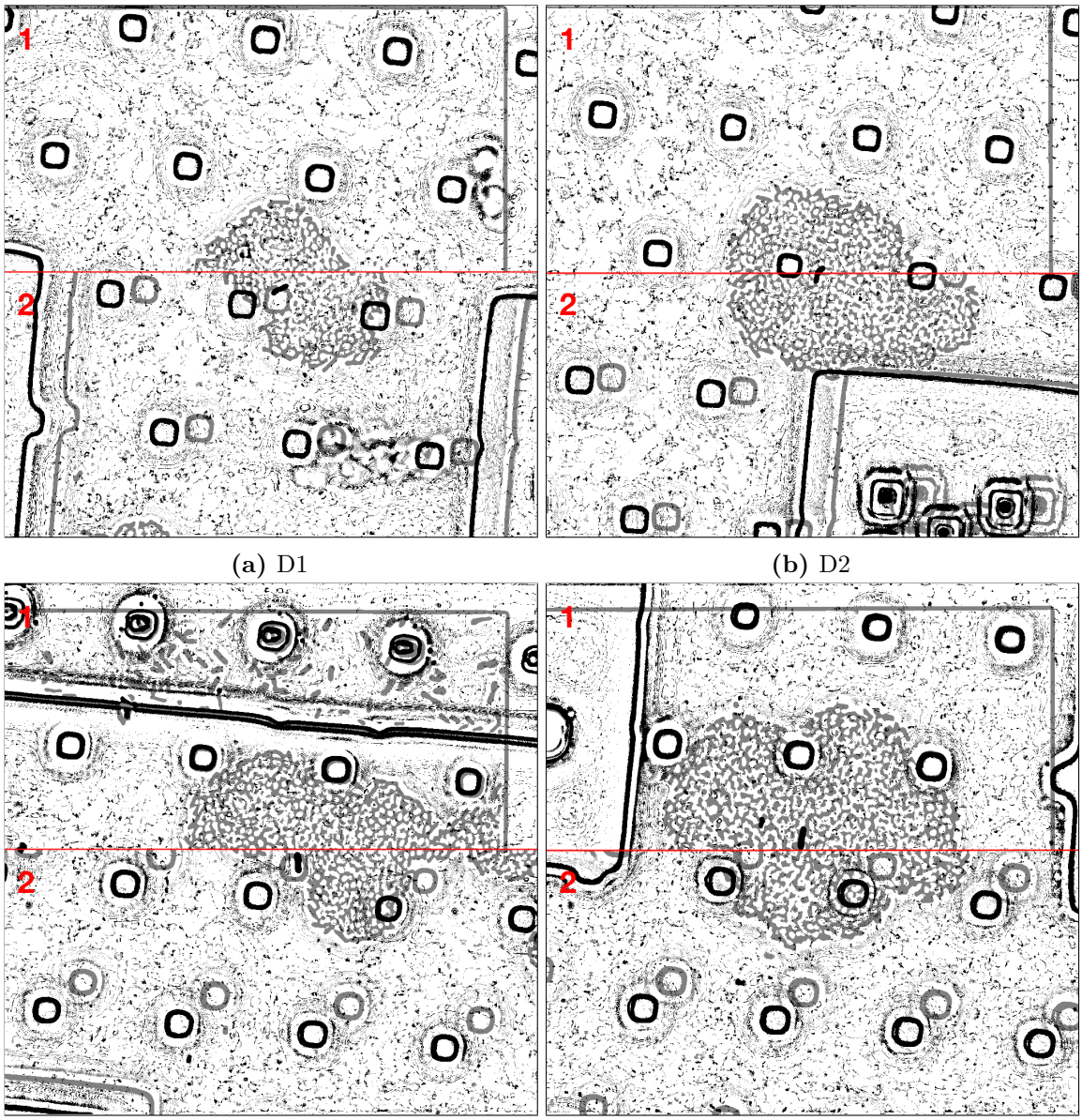
**Figure 4.1:** Quadrants of image $I_{23}$. The quadrants are delimited, by opaque white lines. The bottom right quadrant is rendered as a false-color image, so as to highlight edges in the image space. This quadrant is employed to showcase the result of the preprocessing steps in Figure 5.4a.



**(a)** Grayscale **(b)** Bilateral denoising **(c)** CLAHE

**(d)** Adaptive mean thresholding **(e)** Dilation **(f)** Border clearing then masking

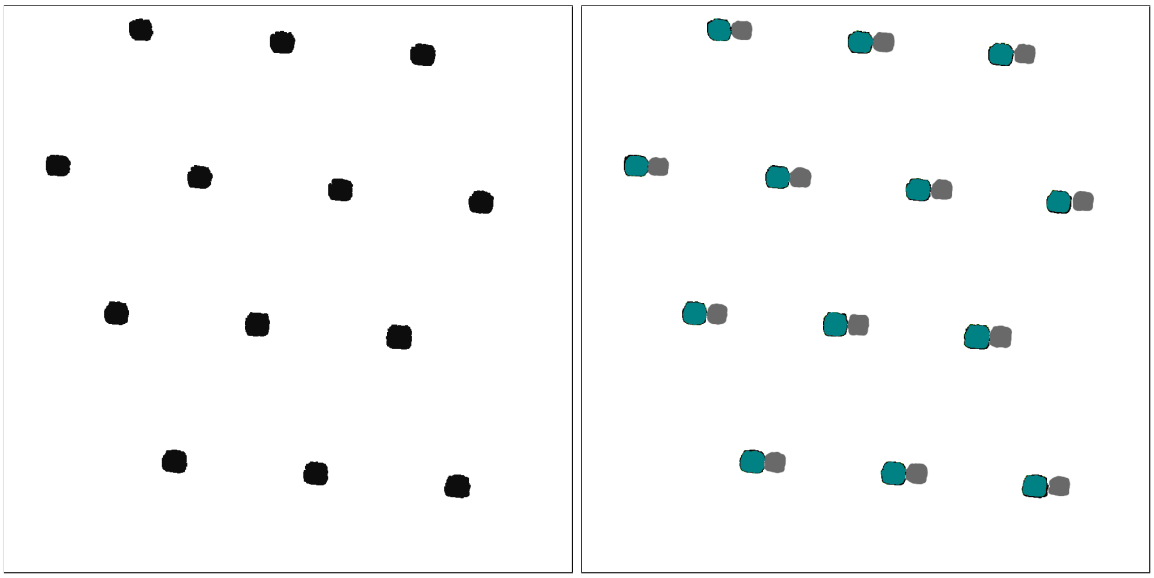
**Figure 4.2:** Example result of the preprocessing steps Dataset 1 (D1). (a) The quadrant of interest is grey-scaled. The particular polygons are observed as square-like polygons. They are an intrinsic part of the microfluidics chamber. (a–f) show the output of each preprocessing step on this particular quadrant. (b) The bilateral filter preserves edges and reduces noise by employing a smoothing filter. (c) The contrast limited adaptive histogram equalization, or CLAHE, is used to improve the contrast of the image. This favors the contrast between the background and the square-like polygons. (d) The adaptive mean threshold computes thresholds for regions of the image with varying illumination. It results in a binary image and a clear outline of the particular polygons. (e) Dilation, as a morphological operation, probes and expands the square-like shapes contained in the input image. (f) Border clearing and masking depict no effects. Such a coupling serves as a validation step so as to palliate for any great image variability (e.g. rotation of objects entering/exiting the field of view).



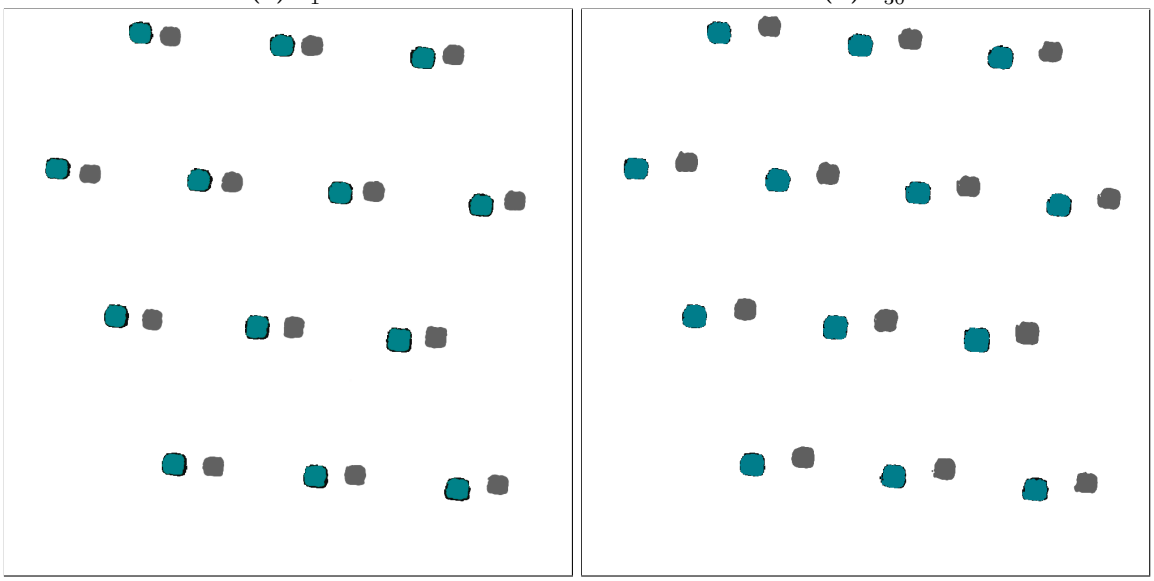

**(a)** D1 **(b)** D2

**(c)** D3 **(d)** D4

**Figure 4.3:** The effectiveness of ViCAR is demonstrated for data sets D1 (a) and D2 (b) and two other data sets D3 (c) and D4 (d) from another experiment, respectively. The upper half in (1) each image (a)–(d) shows one aligned image frame selected from four different data sets, recorded in four different experiments. For the sake of interpretability, results are shown after applying the adaptive threshold. In the lower half (2), an overlay of the non-aligned image is shown with an opacity of 50 % so the shift can be observed. The examples show the robustness of this adaptive visual cues based approach. This indeed justifies using a flexible algorithm so as to handle the varying number and positions of distractors.

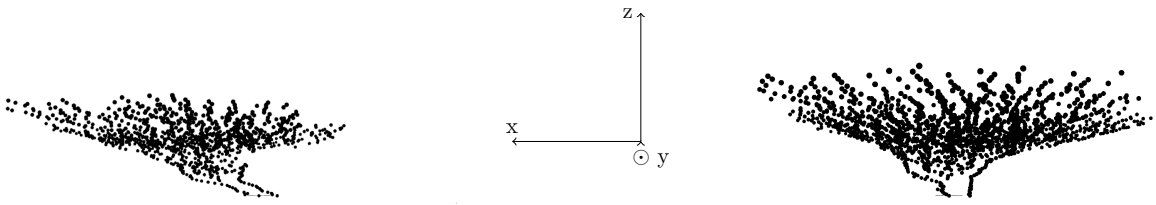


**(a)** $I_1^\star$ **(b)** $I_{30}^\star$



**(c)** $I_{60}^\star$ **(d)** $I_{115}^\star$

**Figure 4.4:** Temporal change of the found polygons before, and after ViCAR's registration for data set D1. Such square-like polygons represent the structure of the microfluidics chamber. The polygons are either shown in black, in grey, or in teal blue. Grey polygons represent the square-like polygons of the microfluidics chamber without applying image registration. Teal blue polygons depict the overlay of the polygons found in the reference image $\hat{I}_1$. Teal blue polygons are positioned in the foreground of black polygons, resulting into the impression of an outline. (a) In the first time point, only one set of anchors is observable. This is explained by the fact that the first image serves as reference for the registration. (b–d) Throughout the temporal progression of the time-series, a distancing of both grey and black outlined polygons is observable; making explicit the spatial shift. By employing the first image as reference, a correct overlay of the first image polygons is observable; as shown in teal blue.




**(a)** Pre-ViCAR **(b)** Post-ViCAR

**Figure 4.5:** Cell positions as a 3D scatter-plot for data set 1 (D1) before the ViCAR's method (a) and after (b). The $x-$ and $y-$axis represent the original image plane, and the pixel coordinates while the $z$-axis represents time. Each dot represents the position of an image feature computed with the SIFT operator[100]. Thereby the dots in one $z$-plane (i.e. at one time point $t_z$) approximate the spatial distribution, and density of the bacterial colony at this time point. On the left side (a) the bacterial colony seem to move or shift inside the chamber. A visual inspection of the original data shows that this is not the case but an artifact of the misalignment. On the right (b), the ViCAR - aligned is displayed, showing the actual spatial colony development over time.

## 4.4 Implementation

This data-driven registration approach has been published[101], and is freely available for download at `http://github.com/ghattab/vicar` under the MIT License. It is implemented in Python and supported on UNIX-based operating systems.

## 4.5 Discussion

Compared to other registration methods in biomedical imaging, this method requires neither a parametric model of the data (e.g. brain atlas, alignment of brain MRI scans)[76,102], nor explicit landmarks (e.g. anatomical landmarks in medical imaging[103], developmental biology[104]). ViCAR properly registered the image data at hand, and has demonstrated promising results for upcoming high-throughput image data analysis. Due to the highly dynamic image content in the biomovies, other methods have failed to register such time-lapse image data. An improvement of image quality might be possible using differential interference contrast microscopy, yet it is not possible to get the same quality at the same magnification. ViCAR relies on consistently finding polygons that are part of the background. Provided a re-evaluation of the preprocessing pipeline, ViCAR may adapt to different experimental

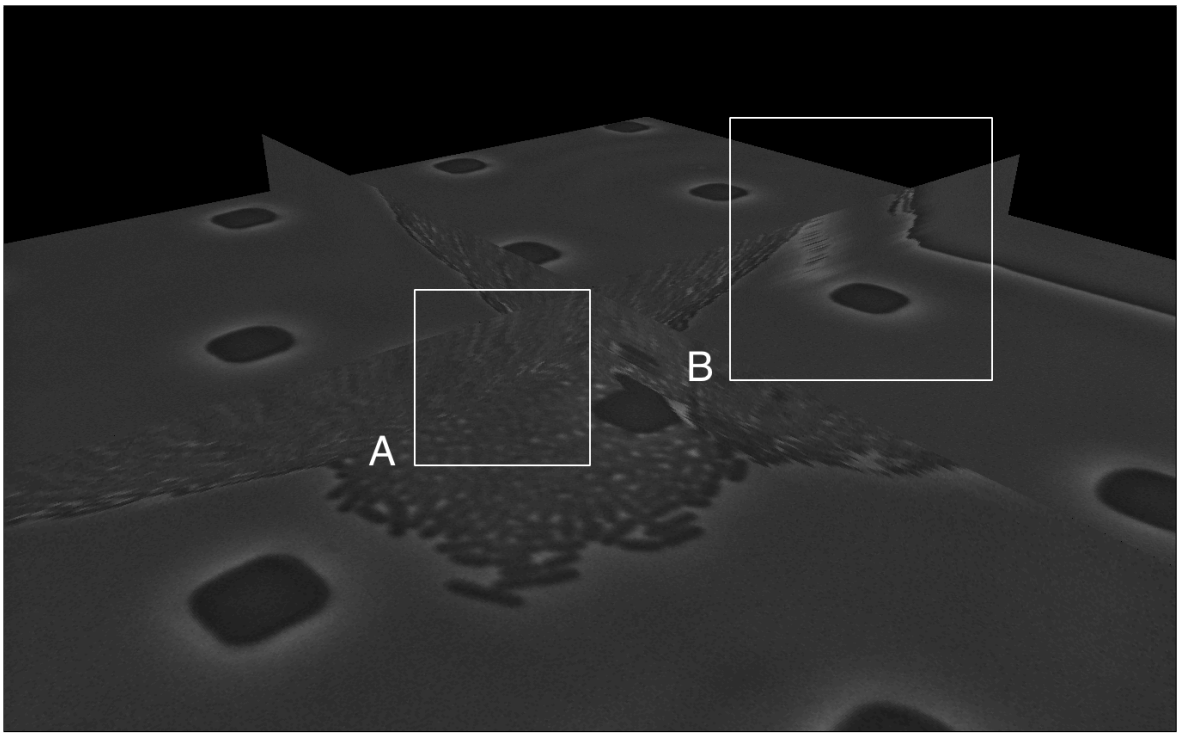



**Figure 4.6:** Example image slice from a single image stack before registering biomovie D1. The image stack solely represent phase contrast images. The three-dimensional image stack is sliced at the middle of the x and y dimensions. The example slice reveals the temporal dimension at the image center. Both the colony (A) and the microplate (B) show visible shift and temporal inconsistencies. (A) The imaged bacterial colony is shown with twisting or snaking cells in the sliced image stack. Such a spatial shift hinders tracking bacterial growth in a biomovie. (B) The large square-like structure of the microplate is observed with a lot of x- and y-axis variability.

setups. The polygon finding step is capable of handling any size, shape, and number of polygons. To find the special polygons, also referred to as visual cues, the perimeter-to-area ratio retains the polygons with least complexity. A limiting factor lies at the registration step, where two consecutive images bearing the same number of visual cues are required.

In special yet few cases, where image content and background vary greatly, it is necessary to reduce the circle mask parameter (see 4.2.1(g)) so to limit the cues to the central image area. The amount of visual cues $J$ assumes they are the same ones. If the shift is larger than half the width of the first image, there is no guarantee that the algorithm successfully registers the biomovie frames. This case scenario occurs when the first visual cues that have been found may, or may no longer be in the visual field. This aspect is to be considered for these exclusive cases, I reckon it is rather a special case than being a negative aspect

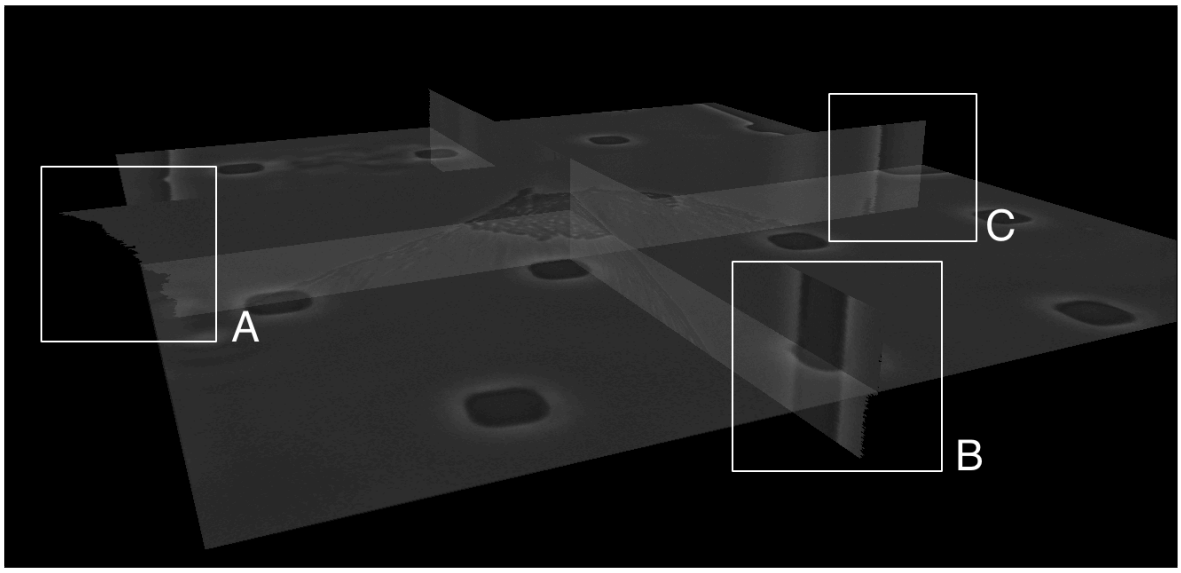
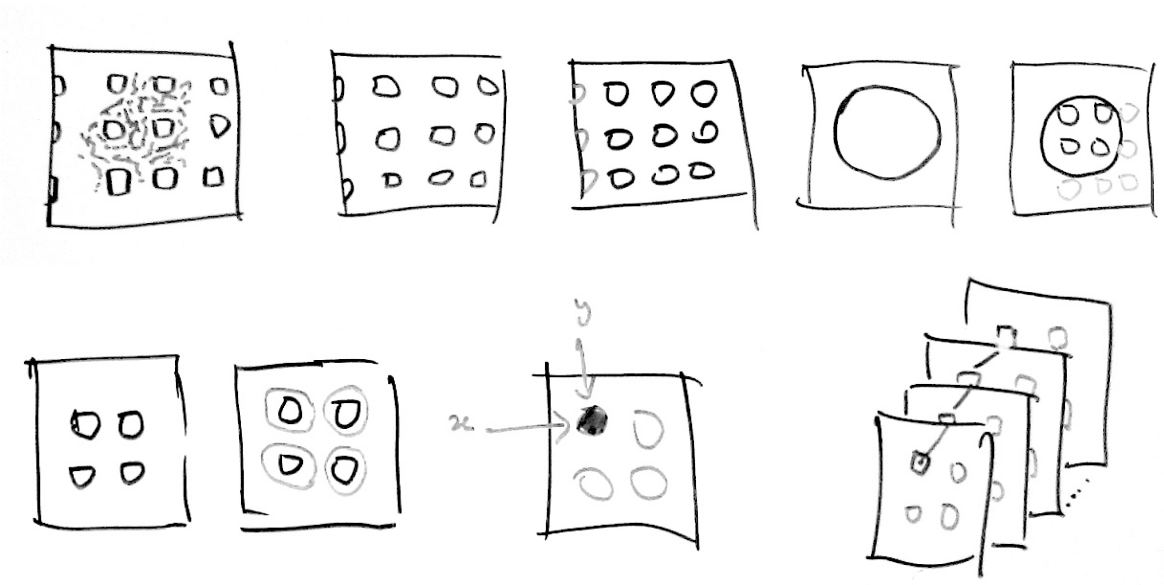



**Figure 4.7:** Example image slice from a single image stack after registering biomovie D1. As seen in Fig. 4.6, the image stack is sliced at the image center. (A–B) After registration, the spatial shift is observable in both x- and y-dimensions. (B–C) Registration correctly matches the square-like structures in the microfluidics device. Compared to Fig. 4.6, the square-like structures of the microplate are well aligned throughout time or the z-axis.

of this method. Due to these reasons, ViCAR has the strength of coupling state of the art image processing steps to a particularly flexible algorithm. Using a perimeter-to-area ratio based filtering proved robust in the filtering step. This step warrants a better adaptability of the method. If deemed decisive, the use of further shape descriptors would permit for an extended structural analysis. To conclude, the reported performance denotes a particularly fast and robust approach that is morphology-free and generalizable.

In this chapter, I described the methodology behind ViCAR, demonstrated its good adaptability and high performance, to align the multiple image frames of a biomovie. This approach helped overcome a range of issues: image rotation, scale, skew, a low SNR, a focus shift due to vibrations and/or variations in temperature, and a particularly variable image/background content. ViCAR provided an effective spatial alignment thereby paving the way to extract temporal features pertinent to each resulting bacterial colony. By using ViCAR, image registration was achieved with 99.9% of image closeness, based on the average rmsd of $4.10^{-2}$ pixels and superior results compared to a state of the art algorithm.


y
x

*All abstract sciences are nothing but the study of relations between signs.*

Denis Diderot

# 5

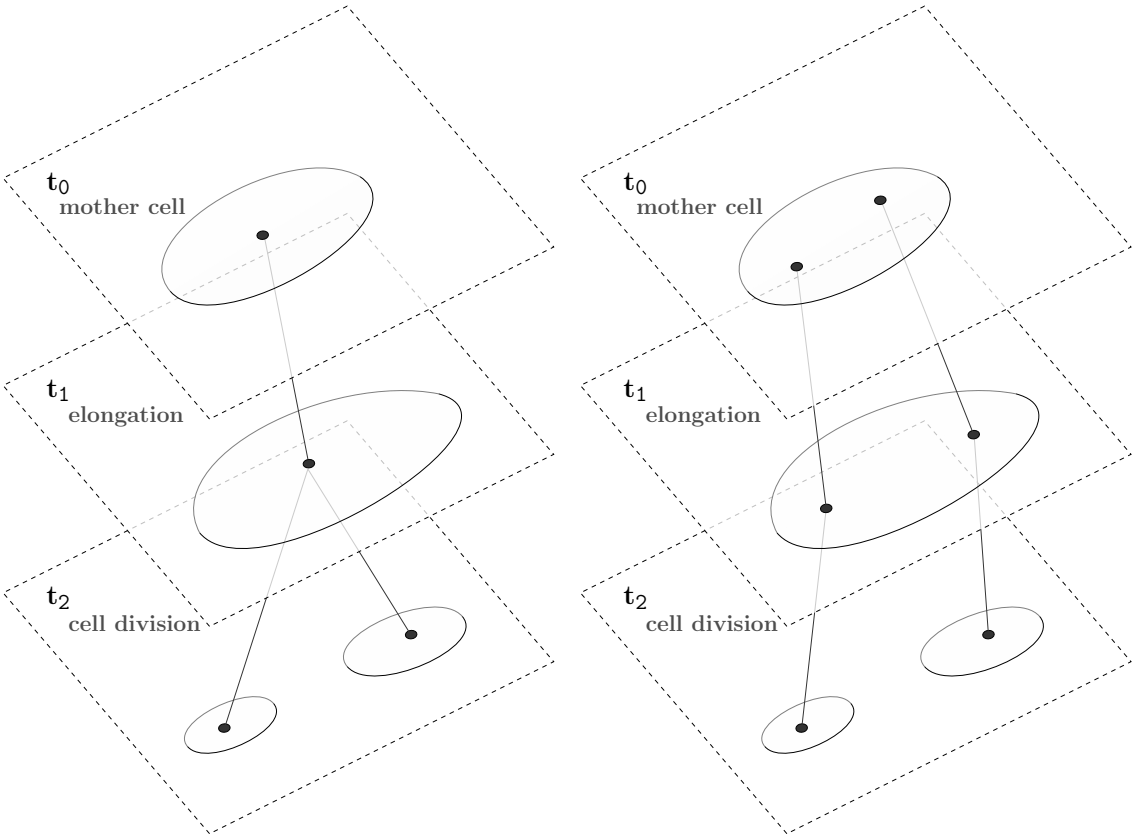
# Data abstractions to understand cell growth

Cellular behaviors may emerge in either one condition or subset of the data, or across all conditions of an experiment. To identify and follow different cellular behaviors in a cell colony, it is required to identify and track subpopulations in biomovies with particularly high values for different properties (e.g. density, shape diversity, etc); see chapter 2. To identify and follow different cellular behaviors in a cell colony, a novel approach is required.

This chapter presents a data-driven framework to identify subpopulations with similar fluorescence. It details the role of two novel data abstractions that are adapted to spatiotemporal changes: the particle and the patch. By employing them, I tackle biomovies with high values for all of the aforementioned five properties without using prior information, or single-cell segmentation (i.e. general paradigm). The presented framework integrates spatial and temporal coherence with a modular algorithm to create a patch lineage graph from particle trajectories.

## 5.1 General paradigm

The general paradigm for the analysis of such data is focused on the extraction of single cell lineage information for all visible cells. A cell lineage is a sequence of cells that developed from a common ancestor. This extraction step comprises single cell segmentation, tracking, and lineage construction. Segmentation refers to spatial coherence and entails delineating individual cells in each frame. Tracking refers to temporal coherence and involves following identified cells throughout a biomovie. Lineage construction is meant to identify cell division events so as to solve the correspondence problem of identifying cell ancestry (see Fig. 5.1).




**(a)** Single-cell segmentation (centroids) **(b)** The approach (particles)

**Figure 5.1:** Comparative illustration of single-cell segmentation approach to the particle-based solution for constructing lineages in biomovies. (A) Single-cell segmentation is used to track object centroids, detecting cell mitosis explicitly and constructing cell lineages accordingly. (B) Multiple particles are detected within regions and tracked over time, detecting mitosis implicitly.

However, extracting cell lineages from microfluidics biomovies is a challenge because of a daunting combination of high cell count (approximately 300 cells), considerable variation in cell size and shape, high cell density, strong noise, and high resolution (60 nm/px) (see Chapter 2). The inadequacy of automatic methods for this data leads to a manual annotation process that is extremely time-consuming, arduous, and also error-prone in terms of low intra- and inter-observer agreement. My collaborators require a period of approximately two full working days to fully annotate one biomovie and create a bacterial cell lineage. Better computational support for biomovie analysis and the extraction of lineage information is a clear need.

## 5.2 The idea

To access a higher level of analysis, i.e. biologically relevant, the focus shifts from single cells to entire subpopulations of cells that share similar signal characteristics. The task of identifying and tracking such subpopulations relies on finding relevant information enclosed across multiple domain fields. To locate where the latter is, I separate the domain fields in Figure 5.2. The domains are: microfluidics, biology, bioengineering, bioimaging, and bioimage informatics. The biology motivates the analysis, while bioimaging is employed to record the biological data – the biomovies – using microfluidics technology and bioengineering (see chapter 2). Thanks to bioengineering, genes are chosen as reporters. The characteristics they confer on the bacteria expressing them are easily identified and measured. This information confers a position and a fluorescence of the reporter genes. Bioimage Informatics is at the analysis side of the spectrum, where cells are delineated and tracked over time. However, cell segmentation is not easily generalizable and fails, especially when confronted with high values for all data properties. Moreover, it is not required to know the fate of every single bacterial cell to address the present task.

Faced with the bottleneck I formulated in chapter 2, the nested model of visualization[67,68] inspires the following. Detecting and following similarly fluorescing subpopulations requires data abstractions. The data abstractions shall enclose the aforementioned characteristics so as to be fitting to the task. That is to say, the abstractions must be related to both the spatial domain and the variation of these characteristics throughout time.

The idea is to exploit both temporal and spatial coherence using two new data abstrac-

tions. *The particle* relies on the edge orientation and fluorescence signal in the image space. Once particles are detected, the algorithm assembles them into particle trajectories. This confers both a spatial and a temporal coherence. To move onto the biology, a second abstraction is required: *the patch*. It relies on the spatiotemporal coherence and signal characteristics of particle trajectories. Conceptually, within one frame, a patch is a set of particles fluorescing similarly. To know whether these particles fluoresce differently in later frames, I introduce CYCASP. It is a flexible modular framework in which particles are detected and tracked, where patches are created, propagated, and evaluated. In turn, this results in patch trajectories that are spatially and temporally coherent. While circumventing the use of *proper* cell segmentation, these abstractions allow the identification of subpopulations from a microfluidics biomovie. My alternative to the single-cell oriented paradigm is designed to handle the dynamics of rapid growth and the shape diversity of bacterial cells.

## 5.3 Related work

The previous work on computational biomovie analysis is summarized in Table 5.1. No previous work handles high values for all five properties of cell count, cell shape diversity, cell density, noise, and resolution, so they cannot handle the data generated I considered here. The closest relevant effort was reported in Grünberger et al.[56]; although they do discuss data with high values for these properties, their system does not actually compute the cell lineage for the large or moderate sized experiments. Instead, they quantify the cell area of interest by computing its logarithm. All of this previous work follows the general paradigm of analysis described above, with single-cell oriented methods that rely on an initial segmentation run before continuing with tracking and lineage construction.

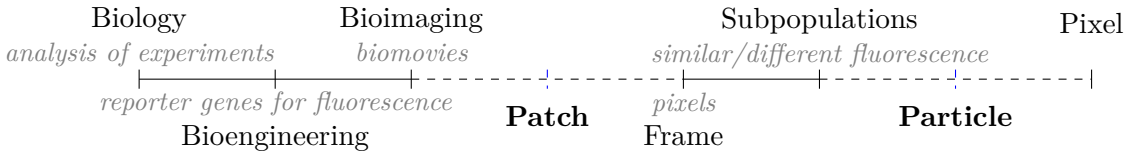



**Figure 5.2:** The diversity of the domains that constitute the scenario and the devised data abstractions.

| | Prokaryotic | | | Eukaryotic | | | Both |
|---|---|---|---|---|---|---|---|
| **Properties / Papers** | Klein et al.[17] | Mekterovic et al.[89] | Grünberger et al.[56] | Kanade et al.[61] | Bao et al.[105] | Li et al.[73] | Wang et al.[27] |
| cell count | moderate (∼100) | low (∼30) | low (∼50) | high (>200) | high (350) | very high (>500) | low |
| cell shape diversity | low | high | moderate | low | low | high | low |
| cell density | low | high | moderate | moderate | low | moderate | moderate |
| noise | low | low | low | high | moderate | high | low |
| resolution (nm/px) | NS | moderate (129) | low–moderate (<120) | moderate (130) | moderate (100) | moderate (130) | NS |
| species | *B. megaterium* | *M. smegmatis* | *C. glutamicum* | *B. Taurus* | *C. elegans* | *H. Sapiens* | *E. coli, etc* |
| tool availability | yes | yes | no | no | yes | no | yes |

**Table 5.1:** Related work is catalogued according to cell type and image parameters (NS = not stated in the publication). In the case of Grünberger et al., larger colonies are considered, yet they did not have their lineage constructed.

## 5.4 Preprocessing

This first step applies a pipeline of standard image processing steps to the RGB channels of each frame $I_t$ to reduce noise, enhance the object-to-background contrast, and spatially align the images. The output is a binary image $\hat{I}_t$ for each time point. Pipeline details are provided below. Unlike the preprocessing in chapter 4, where the square-like structures are enhanced; this following one focuses on finding bacterial signal in the foreground.

1. Enhancing the signal to noise ratio (SNR)
   (a) RGB to greyscale transformation
   (b) image inversion
   (c) contrast limited adaptive histogram equalization (CLAHE) using a tile size $\tau = 3^2$ px and contrast limit of 3, to clip and uniformly distribute any histogram bin above that limit[93]
   (d) pixel intensities transformation for a global contrast enhancement using the following formula: $I'_t = \frac{L}{\phi} \times (\hat{I}_t \times \frac{1}{L/\theta})^2$ with maximum intensity $L$=255 and $\phi = \theta = 1$ (see Fig. 5.4d).

2. Subtracting and enhancing local signals

   (a) denoise bilateral filtering with spatial closeness $s_{\text{spatial}} = 75$, radiometric similarity $s_{\text{range}} = 75$ and pixel neighborhood size $\delta = 5$ px of each pixel neighborhood that is used during filtering[92]

   (b) adaptive mean thresholding with block size $\tau = 13^2$ px and constant C = 2, that is subtracted from the weighted mean in order to prevent noise to pop up at background regions (see Fig. 5.9f).

3. Adaptive background masking

   (a) median blurring with an aperture linear size $k = 15^2$ px

   (b) binary thresholding with $h = 255$ and maximum value $V_{\max} = 255$

   (c) masking, by using a binary mask of image dimensions $(r \times c)$ is initialized, containing the background. A bitwise comparison (disjunction) returns the foreground, which contains the colony.

### 5.4.1 Benchmarks

I present the results of the preprocessing step for four original (D1-D4) and five simulated (DS1-DS5) biomovie data sets. The biological data sets feature high values for the five properties targeted by this work: cell count, cell shape diversity, cell density, image noise, and image resolution (see Table 2.1 for full details). A mid-2013 MacBook Air (1.7GHz dual-core Intel Core i7, 8Gb of 1600MHz memory) was employed for all presented benchmarks. Benchmark results for all considered biomovies are reported in Figure 5.3.

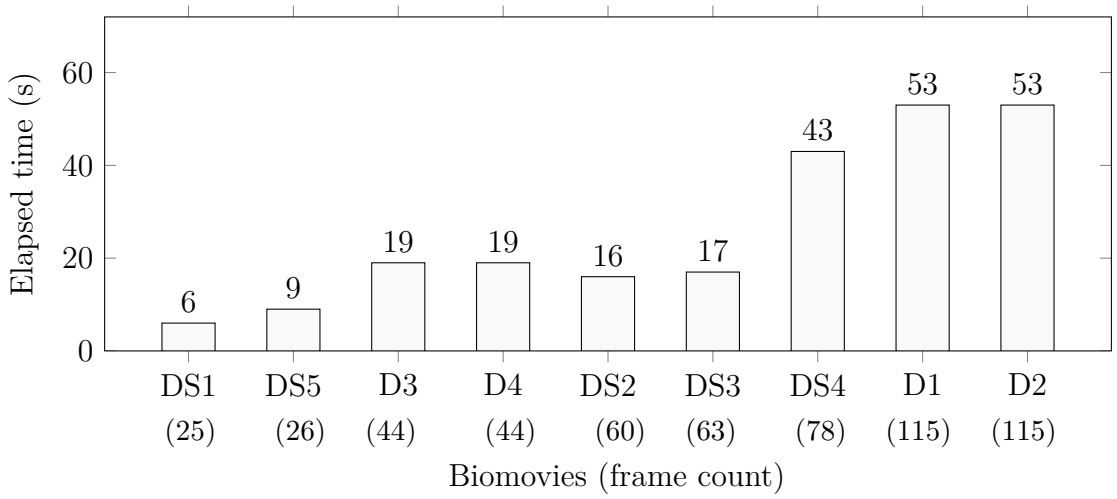
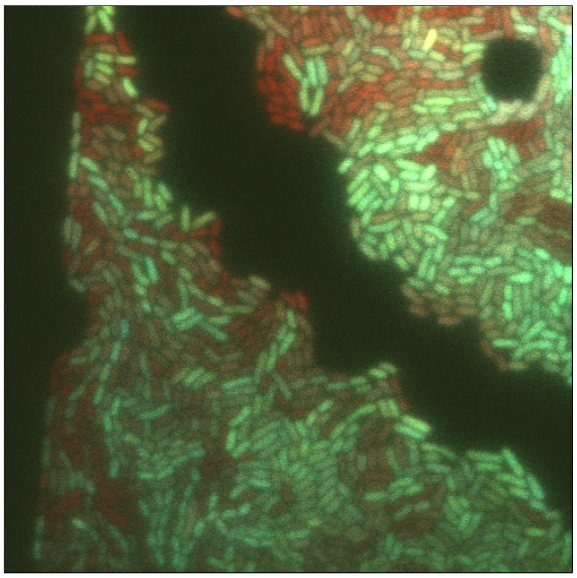
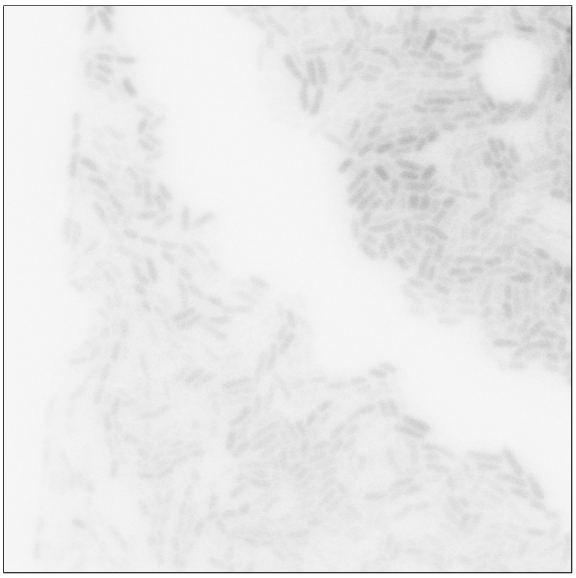



**Figure 5.3:** Average elapsed time of 100 runs of the preprocessing step for all biomovies, in seconds. Biomovies in the x-axis are sorted by frame count, from lowest to highest (as indicated in parentheses). We observe an approximate correlation between frame count and preprocessing time. The average time varies with a $\Delta\pm1$ second(s). An approximate correlation between frame count and preprocessing time is noticeable.

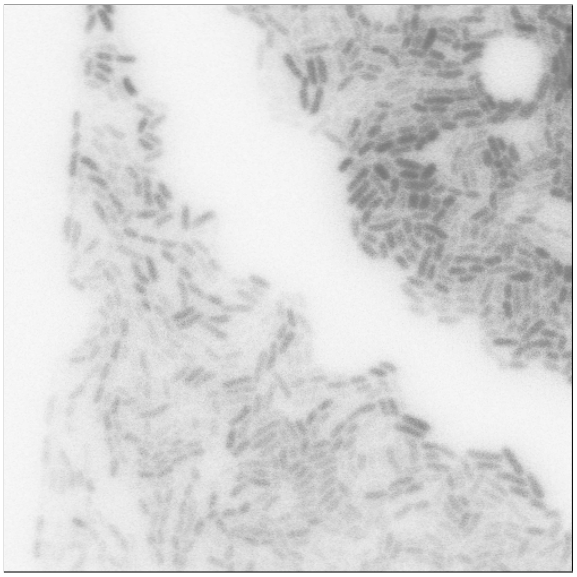
Figure 5.13(a) shows the result of the preprocessing to enhance the cell-background contrast in the RGB images shown in Fig. 2.1(c–f) (frame $I_{115}$ of D1). Whereas, each step of the preprocessing pipeline for the final frame of biomovie D1 is depicted in Figure 5.9. The final binary image of each biomovie is showcased in the following Figures 5.10 – 5.12.



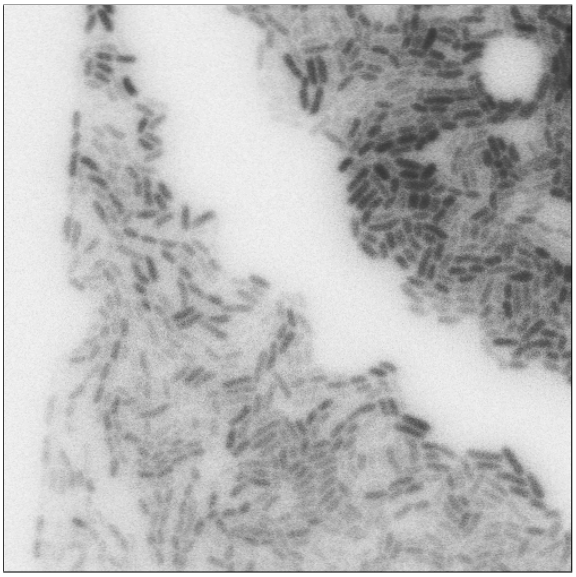
**(a)** input: RGB



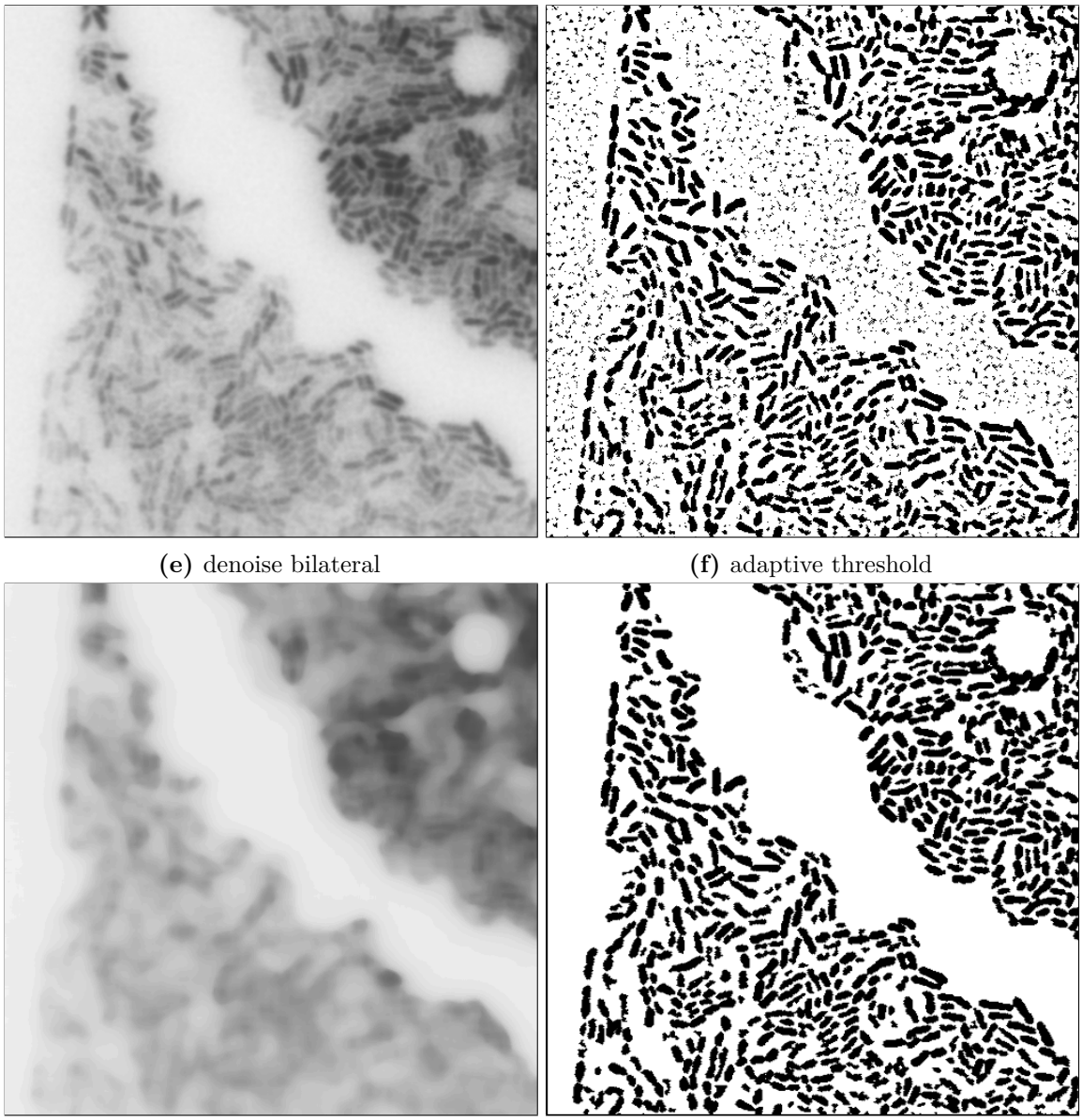
**(b)** greyscale, invert



**(c)** CLAHE



**(d)** global contrast



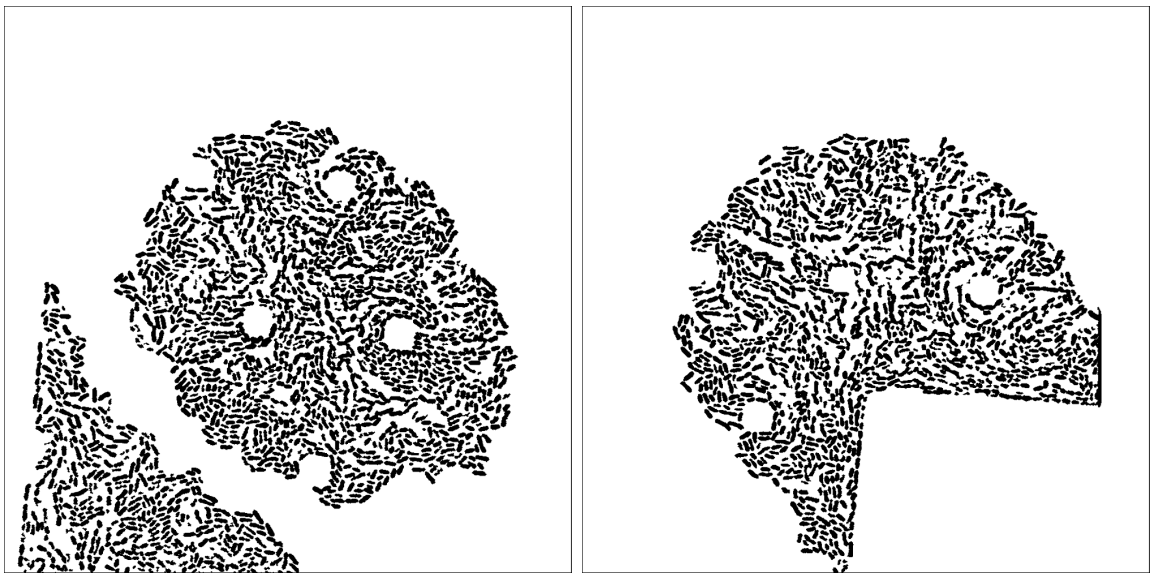
**(e)** denoise bilateral

**(f)** adaptive threshold

**(g)** median blur

**(h)** output: binary image $\hat{I}_{115}$

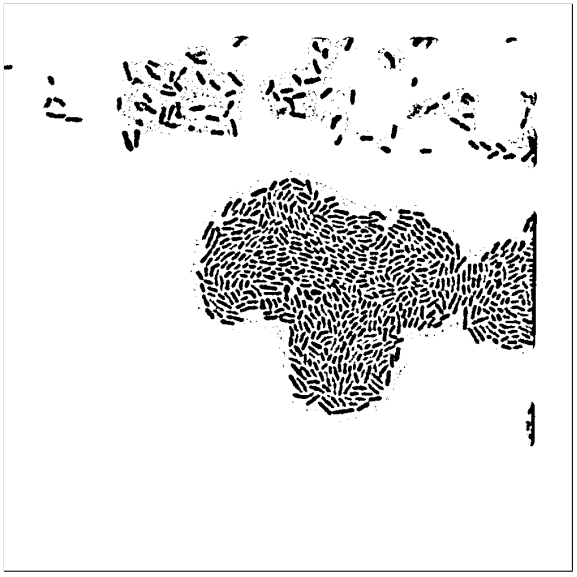
**Figure 5.9:** Example results after each preprocessing step for original biomovie D1 at $t = 115$, the final frame. The RGB image is showcased here at 100% exposure, with close-up detail of the bottom left quadrant. (a) The input RGB image. (b) After the greyscale transformation and image inversion. (c) After the contrast limited adaptive histogram equalization (CLAHE). (d) After the global contrast enhancement. (e) After the denoise bilateral filtering. (f) After the adaptive mean thresholding. (g) After the median blurring. (h) After masking, the final output is a binary image. For the detailed preprocessing see chapter 4.



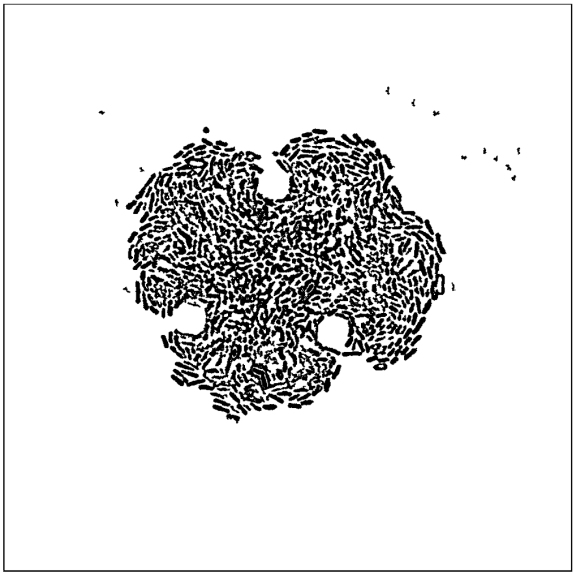
**(a)** D1 - $\hat{I}_{115}$



**(b)** D2 - $\hat{I}_{115}$

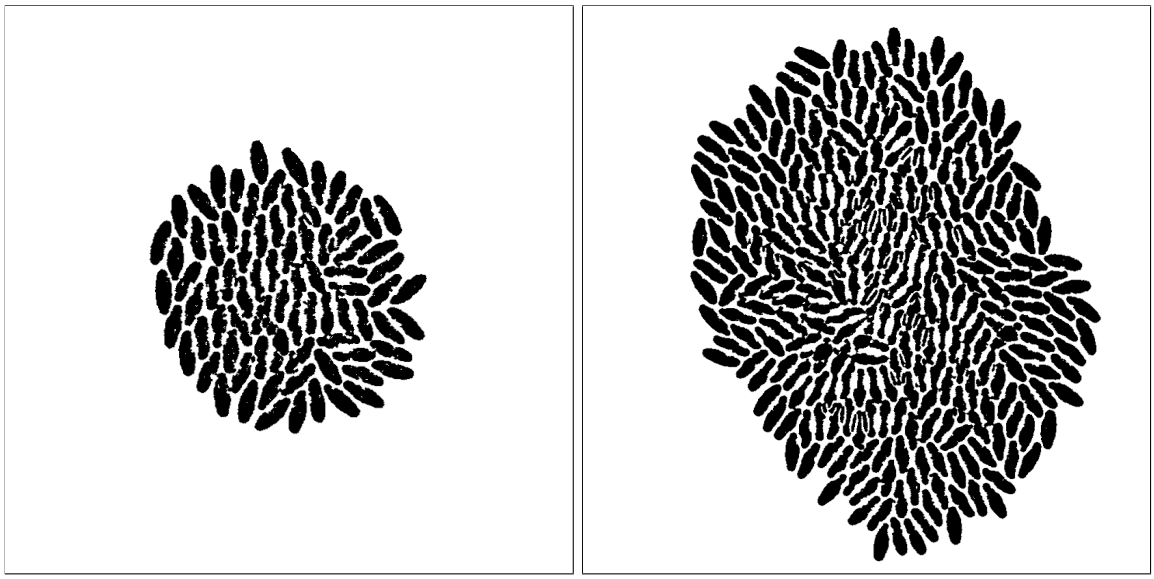


**(c)** D3 - $\hat{I}_{44}$



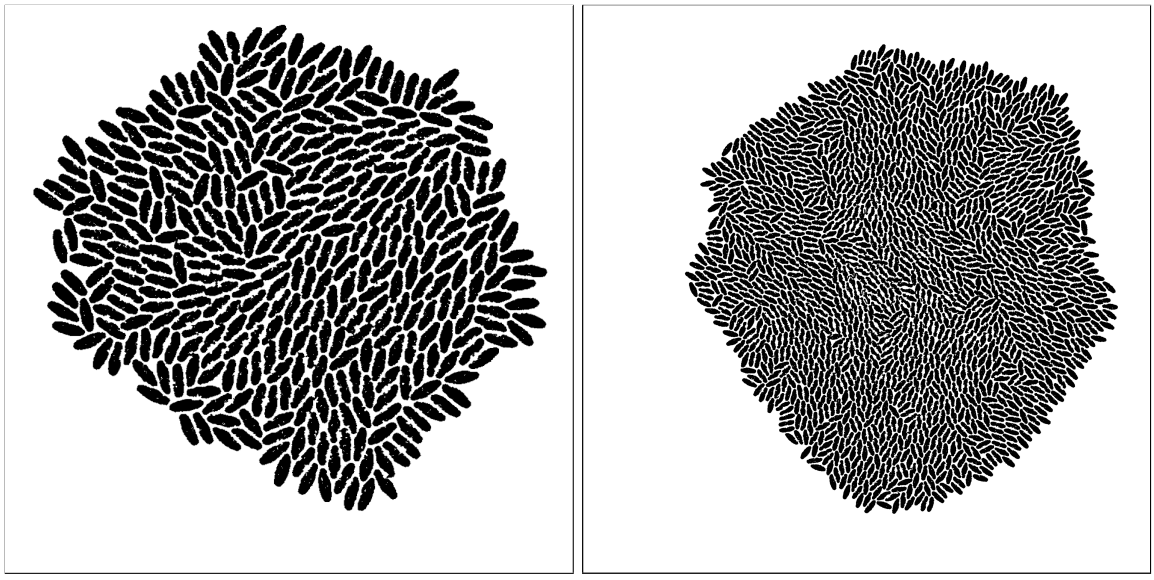
**(d)** D4 - $\hat{I}_{44}$

**Figure 5.10:** Binary images after preprocessing of the biomovie final frames. (a) Biomovie D1 shows a phenotypic heterogeneity experiment, with two separate colonies visible. (b) Biomovie D2 is an alternate condition of the same experiment. (c) Biomovie D3 shows an experiment on bacterial communication by quorum sensing. (d) Biomovie D4 is an alternate condition of the same experiment.



**(a)** DS1 - $\hat{I}_{25}$ **(b)** DS2 - $\hat{I}_{60}$

**Figure 5.11:** Binary images after preprocessing of simulated biomovies (final frames). (a) DS1. (b) DS2. See Fig. 2.2 for the original RGB images.



**(a)** DS3 - $\hat{I}_{63}$ **(b)** DS4 - $\hat{I}_{78}$

**Figure 5.12:** Binary images after preprocessing of simulated biomovies (final frames). (a) DS3. (b) DS4. See Fig. 2.3 for the original RGB images.

## 5.5 Particle analysis

In the second step described below, particles are detected and assembled into temporally coherent particle trajectories in order to incorporate the temporal domain as well as the spatial domain. The goal is to have enough particles to ensure that there are no false negatives (i.e. each cell is represented by at least one particle); false positives are detected and filtered out using the parameter windows described below.

### 5.5.1 Particle detection

A particle is an intuitive geometric abstraction that results from considering whether the neighborhood around a pixel falls within a cell by checking for signal characteristics such as signal intensity, edge orientation, fluorescence signals, or texture. As a geometrical abstraction, it is borrowed from the domain of fluid mechanics. It is employed for images depicting droplets and in general circular shapes. In this work, the interest shifts from droplets to blobs that depict bacteria. The latter have no distinct shape or definition, even though they are theoretically rod shaped. In this work, the binary images are the input and clearly depict the bacterial fluorescence signal in the foreground. Thus, the signal is a constant; e.g. Fig. 5.10.

Particle detection employs a Gaussian-like blob operator for each binary image (i.e. Crocker-Grier algorithm)[106,107]. It computes observable features in each time point for cells with a given expected diameter $d$ (set to $d = 11$ px for the biomovies shown in this paper, see illustration in Fig. 5.13). In this implementation, when a particle spans multiple pixels, the algorithm finds the position of a particle with sub-pixel accuracy. It is achieved by taking the average position of these pixels (i.e. radius of the Gaussian), weighted by the brightness. Thus, the resolution of a particle may exceed the traditional diffraction-limited resolution of the microscope. Given an appropriate particle diameter, each detected particle can be inscribed within a cell or a contiguous group of cells. Compared to greyscale or RGB images, the uncertainty in the location of a Gaussian blob is non-existent for binary images. To deal with the anisotropic bacterial shapes, I suggest computing $d$ based on the bacterial cell size in the image space where $l$ = average bacterial length, $w$ = average bacterial width (and $d$ is odd):

$$d = \begin{cases} \texttt{floor}(max, min, \frac{1}{2} * (max - min)) & \text{if} \quad l \neq w \\ \texttt{floor}(w, \frac{1}{2} * w) & \text{else} \end{cases} \tag{5.1}$$

The diameter size has an important effect over the precision. If a user under-estimates the particle diameter, precision suffers. Hence, it is best to over-estimate the diameter, although larger diameters come at some cost in performance[106]. Moreover, a too small diameter often biases the location of a particle towards the pixel edges.

Often, a particle has visually distinct qualities, features, or attributes. They span from the spatial position of a particle to the color information embedded in the RGB domain. These features are introduced once particles are tracked throughout time.

### 5.5.2 Particle trajectories

A particle trajectory is assembled by tracking a particle over time, exploiting temporal coherence. This filters out spurious signals that do not persist across multiple frames. The life cycle of a particle, that is, induced changes over time, ranges from creation, bifurcation, continuation, and dissipation to amalgamation[108]. Particle tracking is employed between consecutive frames, throughout the biomovie, on the found particles with the Crocker and Grier's algorithm[107]. The algorithm's Python implementation `trackpy` is employed[106]. All particle positions are evaluated across space and time by employing trajectory linking and filtering, respectively.

**Particle trajectory linking**: To link particle positions $(x, y)_t$ into particle trajectories $\{J_k\}$, the KDTree neighbor-finding strategy is employed (default method of `trackpy`) with the two parameter windows of distance and time. The distance radius $\sigma_{\max} = d - 2$ px determines the maximum distance each particle is allowed to move from the initial position between consecutive images. The size of the time interval $W_{\max} = \texttt{floor}(15\%$ frame count), determines the maximum number of consecutive images to be considered for (dis-)appearing particles. Particle trajectories $\{J_k\}$ are defined as:

$$\{J_k\} = \{(x, y)_{t,p}\} \tag{5.2}$$

with $1 \leq k \leq K$ where $K$ = number of particle trajectories and $p$ = particle index. Particle trajectories are disjunct: if $(x, y)_{t,p} \in J_k$ then

$$(x, y)_{t,p} \notin J_{k'} \; \forall k \neq k' \tag{5.3}$$

**Particle trajectory filtering**: Spurious trajectories are filtered out according to a time window $W_{\min} = \texttt{floor}(10\%$ frame count). If $(x, y)_{t,p} \in J_k$ with $t = t_{\max} < W_{\min}$ then $J_k$ is omitted. Otherwise, the algorithm finds no spurious trajectories and continues onto the next computation.

**Particle trajectory color information**: A particle trajectory is re-associated with its underlying color information by extracting fluorescence values from the RGB channels at the given particle positions. RGB values are referred to with $(r_{x,y}, g_{x,y}, b_{x,y})_{t,p}$ and are linearly normalized given the minimum and maximum values in each channel and across all images. The resulting RGB values are within the bounded range $[0, 255]$, normalized to diminish low fluorescence and intensify high fluorescence signals. I argue that the minimum value either corresponds to noise artifacts or to spurious trajectories (dying cells that may prove difficult to follow). I thus filter out particles that are completely black.

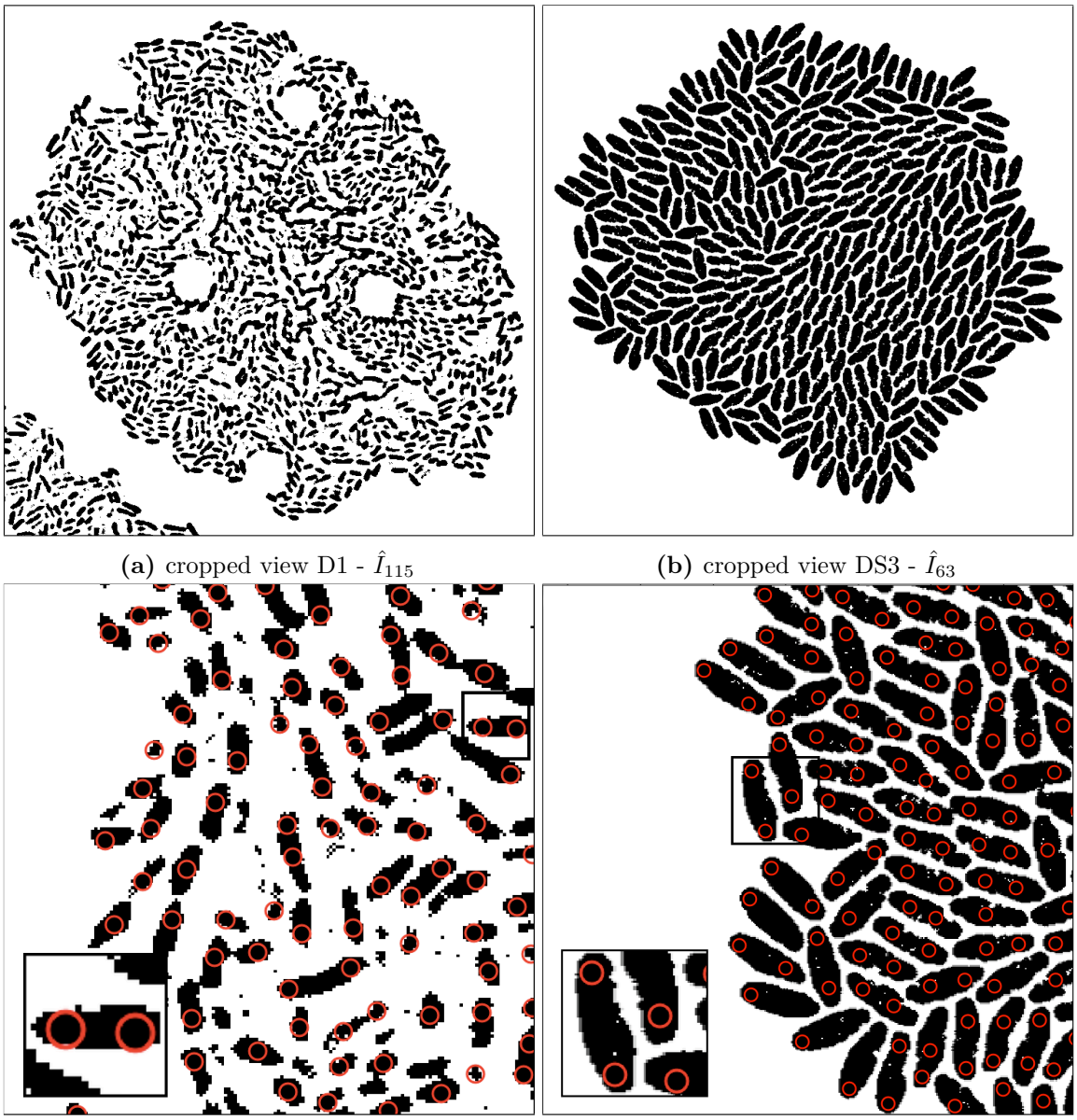
### 5.5.3 Particle evaluation

I evaluated the results of the particle analysis on a technical level in terms of success at capturing the spatial coherence, the temporal coherence present in the binary images, and the computational performance. Moreover, additional work I published answers the biological question of whether particle trajectories reflect the colony growth trend[109].

**Spatial coherence**: The results show that the particle approach used by SEEVIS and CYCASP successfully captures the spatial coherence of cell subpopulations. In this section, I also report particle visualization results using SEEVIS. Figures 5.13 and 5.14, depict computed particle locations annotated as red circles on the RGB and the binary images, respectively. These particles capture the salient structure for both original and simulated biomovies, where appropriate choices for particle diameter $d$ yield an average of two particles per cell. Figure 5.16 illustrates how particles account for cell growth, where elongation triggers an intermediate particle and then cell division results in additional particles that track the new cells, for the difficult case of strong noise and directly touching cells in biomovie D1.

**Temporal coherence**: The use of particle tracking to link particles into trajectories removes spurious phenomena while capturing the temporal coherence within the biomovie. Figure 5.16 compares the two different time intervals of 5 frames (a) and 3 frames (b) for biomovie D3, where 38 vs. 31 particles respectively are filtered. These results are characteristic of the sensitivity analysis showing that the algorithm is robust to small changes of this parameter, even as setting larger time windows results in a smaller number of particle trajectories remaining after the filtering step. Figure 5.15 shows particle linking over 25 frames of a simple simulated biomovie, where 383 particle positions were detected resulting in 63 unique trajectories after linking, reducing to 34 trajectories after time filtering.

**Computational performance**: Nine data sets are shown in Fig. 5.17. The shorter biomovies required between 27 and 39 seconds, and the longer ones between 1.2 and 6 minutes. The processing time roughly corresponds to the density of cells within the biomovie, more so than simply the number of frames. The most time consuming 6-minute computation was for the special case of a highly dense and highly populated colony (DS4 with $\sim$1700 cells). I chose a particle diameter of 7 px given a cell minimum diameter of 17px, which resulted in 7661 particles identified and tracked. I then chose a small time filtering window of 5 frames and after that step 95% of 7661 particles were eliminated. This example demonstrates the necessity of adjusting the user-settable parameters appropriately for the biomovie data set, on a case-by-case basis.



**(a)** cropped view D1 - $\hat{I}_{115}$

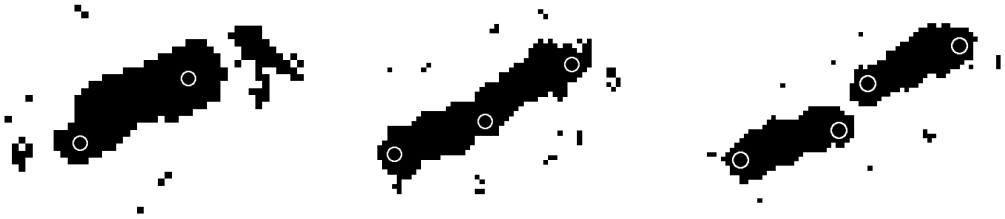
**(b)** cropped view DS3 - $\hat{I}_{63}$

**(c)** image portion $\hat{I}_{115}$: $d = 9$px

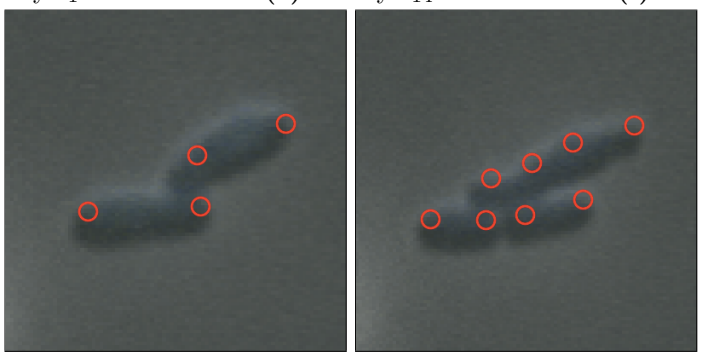
**(d)** image portion $\hat{I}_{63}$: $d = 17$px

**Figure 5.13:** Binary images annotated with computed particle positions (shown as red circles). (a) Original biomovie D1 binary image. (b) Simulated biomovie binary image. (c) Original biomovie crop of D1 showing 1-2 particles detected within each cell. A particle diameter value of $d = 9$px yields no false negatives and some false positives that will be eliminated in subsequent processing that exploits temporal coherence. (d) Simulated biomovie crop showing 2 particles detected per cell, with a particle diameter $d = 17$px.

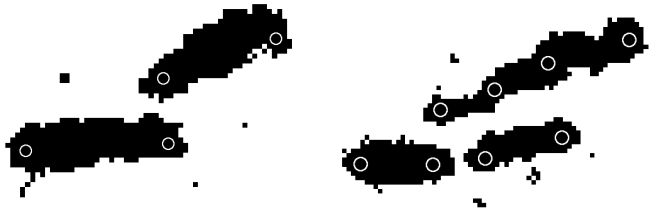
**(a)** RGB $I_1$ **(b)** RGB $I_{14}$ **(c)** RGB $I_{17}$



**(d)** Binary $\hat{I}_1$ **(e)** Binary $\hat{I}_{14}$ **(f)** Binary $\hat{I}_{17}$

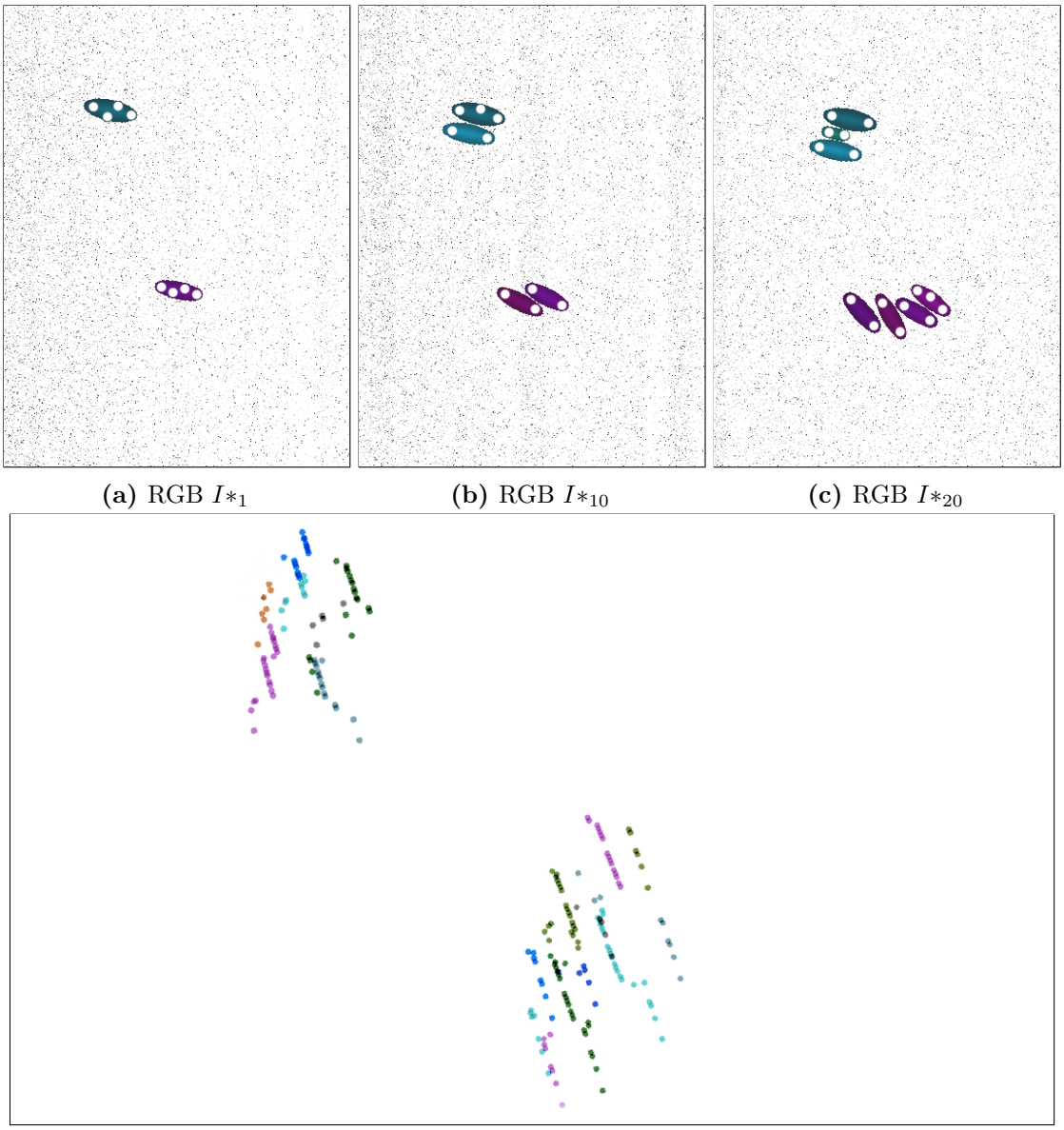


**(g)** RGB $I_{18}$ **(h)** RGB $I_{21}$



**(i)** Binary $\hat{I}_{18}$ **(j)** Binary $\hat{I}_{21}$

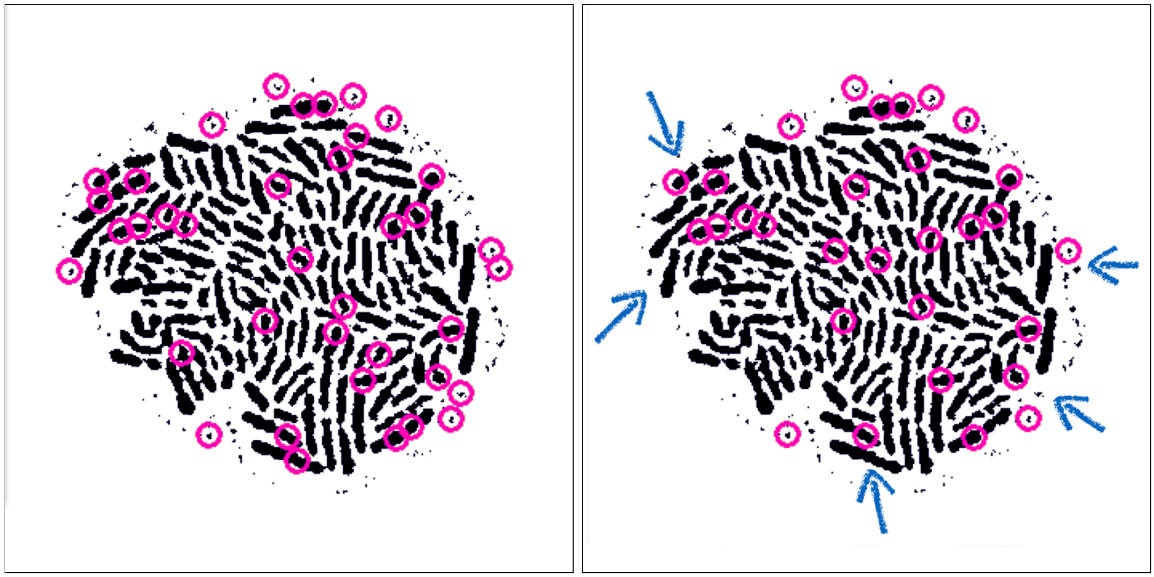
**Figure 5.14:** Particle detection for biomovie D1 across cell division events, with detected particle locations annotated as red circles on original images (a-c, g-h) and white circles on binary images (d-f, i-j), respectively. The particle paradigm handles cell division cleanly despite high levels of noise and the direct contacts between cells: when the cell elongates, a new particle is created in the centre when the width between the previous particles surpasses the distance threshold.



**(a)** RGB $I_{*1}$ **(b)** RGB $I_{*10}$ **(c)** RGB $I_{*20}$

**(d)** Particle trajectories found across time: $t_1$–$t_{23}$

**Figure 5.15:** Particle linking result for the simple simulated biomovie DS5, shown for cropped 375x500 px subsets of the original 2048x2048 px images show four to seven cells appearing in: cyan (top) and magenta (bottom). The black background was replaced by white pixels to better notice the cells. The threshold for particle finding was diameter $d = 13$ px and for particle linking the time filtering window was set to 3 frames. Computed particle locations annotated as 10 px white dots in (a–c). (a) Time point 1 shows two ancestor cells. (b) By time point 10 both ancestors have divided once. (c) By time point 20 the upper cyan colony has 3 cells and the lower purple one has 4. (d) Particle trajectories covering the first 23 time points are shown by mapping each particle differently according to the unique ID of the computed particle trajectory. This image crop contains 19 unique trajectories, all of which show an overall downward drift. For the entire DS5 biomovie, I globally found 383 particle positions resulting in 63 unique trajectories after linking, reduced to 34 trajectories after time filtering.



**(a)** (D3 $\hat{I}_{33}$) $d$= 9 px, time filter window 5 frames

**(b)** (D3 $\hat{I}_{33}$) $d$= 9 px, time filter window 3 frames

**Figure 5.16:** Effect of the time filtering window on particle trajectories, showing binary images annotated with eliminated particle positions with large 7-px magenta circles. (a) A 5-frame window filters out 38 particles. (b) A 3-frame window filters out 31 particles. Blue arrows highlight some of the particles kept for the shorter window but filtered out in the longer window.

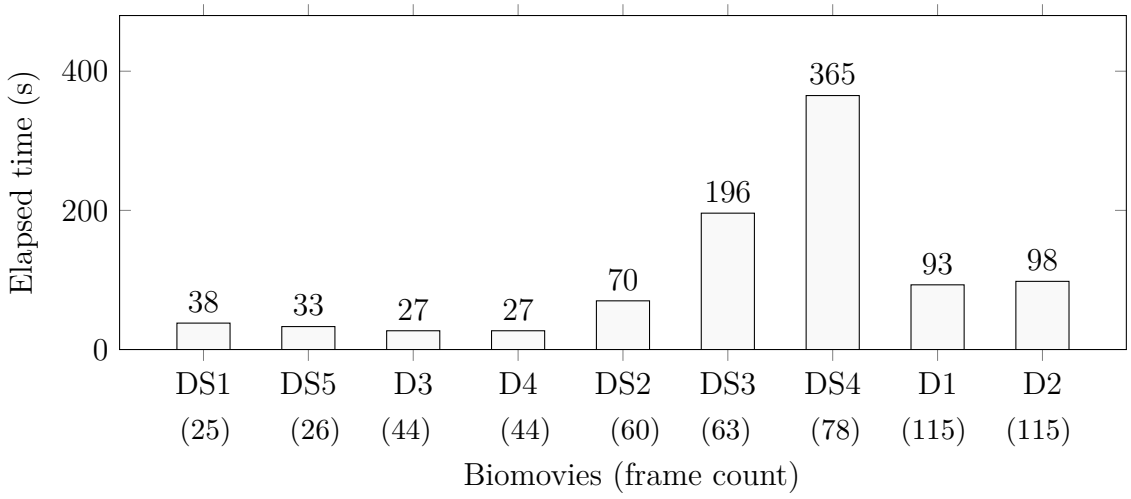



**Figure 5.17:** Average elapsed time of 100 runs of the particle step for all biomovies, in seconds. The particle step includes all three phases of particle finding, linking, and filtering. Biomovies in the x-axis are sorted by frame count, from lowest to highest (as indicated in parentheses). I observe that particle-related computation time is related to the density of the colony in the biomovie, rather than the number of frames. The average time varies with a $\Delta \pm 2$ second(s). The DS4 biomovie is a highly dense special case, where finding and linking in time over 7000 particles takes over 6 min. These procedures can be computationally expensive, given a highly populated colony and a particle diameter set to a low value.

**Summary**: The particle detection and particle trajectory construction step successfully captures the spatial and temporal information in the binary image sequence without computing explicit image segmentation at the level of individual cells. This approach is computationally efficient and requires no manual intervention. It is robust to the transient interactions between neighboring cells that would cause mis-segmentation in attempts to detect individual cells.

### 5.5.4 Particle visualization

As seen in chapter 2, the coupling of microfluidics and time lapse imaging provides functional insight into the biology of cell development. Major biological questions are tackled in the literature. In this immediate work, I retain the example of how bacterial cells develop resistance to antibiotics or adapt to changes in the medium, in small or large populations and in a short period of time or even on a long term basis (e.g. 58 hours experiment). Often, time-lapse image data is displayed on a frame-by-frame basis. Yet such a representation of the data highlights temporal evolution without necessarily displaying spatial changes. Possible questions range from but are not limited to: When did the colony reach a certain biomass? When and how did the colony adapt to the introduction of the antibiotic in the medium? How did the colony survive to particular environmental changes? Were all the cells amassed in one visual field or scattered across the image?

To answer such biological questions, it is necessary to 'play' with the data or its representation. For example, panning, rotating, zooming onto certain cell trajectories, selecting a relevant subset of trajectories, or color mapping the trajectories according to temporal or spatial distances. In this endeavor, particle trajectories $J_k$ can be considered a time series of $x, y$ positions. By using space-time cubes, trajectories are encoded as curves in 3D by mapping the particle positions $(x, y)_{t,p}$ to the x- and y-axis and by mapping time to the z-axis. Space-time cubes are one of the six classes of visualization methods for live cell imaging data[23] that have been proposed on a theoretical level. However, to the best of my knowledge a practical application and discussion of this approach to biomovies (or cell image data in particular) has not yet been reported. By using a space-time cube (see Fig. 3.4), particle trajectories can be highlighted for spatial and temporal investigation.

Three different color mapping methods have been implemented for the data so as to increase the perception of cell lineage growth. This refers to perceiving the extent of a colony in space and/or time, according to different data attributes (e.g. cell ). In this work, they are titled and functionally described as follows:

(1) **Nominal mapping (NM)** highlights single trajectories. Each trajectory is highlighted so as to dissociate neighboring trajectories over space and time. It could help users identify relationships between cell pedigrees.

(2) **Time mapping (TM)** visually promotes the extent of the population growth over

time. It could prove useful in a high-throughput setting where multiple colonies are qualitatively compared. For instance, while holding the time value constant and mapping it to the z-axis, users glance at the different colonies and easily interpret the data. That is to find which colonies grew faster or occupied a larger visual field.

(3) **Progeny mapping (PM)** supports the process of tracing back single trajectories to their parents. It highlights the last known or most recent progeny using the Nominal mapping (NM). This mapping subsets the data to a biologically relevant set. Hence, making it easier to investigate the progeny.

I refer to the display of a particle as a spot. The three visualization encodings map each particle trajectory $J_k$ to a triplet: spot size, spot color, and spot index or $(s, c, f)$, respectively.

Provided $J_k$, the mapping function:

$$\gamma(J_k) = (s, c, f) \tag{5.4}$$

With spot size $s = 3$, the RGBa spot color $c$, and spot index $f$. The size $s$ was chosen arbitrarily in the local coordinate system or scene coordinates. By default, spots scale with the view. Whereas, the alpha channel $a$ of the RGBa spot color varies in $[0, 1]$. By default spots are opaque: $a = 1$. I use two main categories of mappings: particle index based (type 1) and time point based (type 2).

(1) **Nominal mapping (NM)** (type 1):
The particle index $p$ of a particle coordinate $(x, y)_{t,p}$, is treated as a nominal variable, to support pairwise differentiation and contrast of neighboring trajectories. The human perceptual system dictates a strong limit on the amount of categorical colors that can be distinguished[110,66]. My goal is to differentiate between trajectories within local neighborhoods, since it is impossible to have unique colors across the entire image. A set of unique colors $\Upsilon = 10$ were employed by mapping the integer indices $[0, 1, \ldots, \epsilon-1]$ to unique colors from the `Tableau10` color palette[111], as seen in Fig. 5.18. Each color $c$ was chosen randomly for each particle index $p$.

**Figure 5.18:** The `Tableau10` categorical palette[111].

(2) **Time mapping (TM)** (type 2):
To map each particle at a time point $t$ to one spot color $c$, I used the `viridis` color palette (see Fig. 5.19). The TM adapts to the time span of each dataset, by setting its lightest color to the $t_{\max}$ data value.

**Figure 5.19:** The `viridis` color palette[112] is perceptually uniform and with monotonically increasing luminance in multiple hues, ranging from dark purple, through blue and green to the light yellow.

(3) **Progeny mapping (PM)** (type 1):
The idea is to retrieve two trajectory subsets so to use approach (1) NM on the first, and decrease the visibility of the second. The former comprises particle trajectories that are observed at the last time point $t_{\max}$, and the latter remaining trajectories. Let $\{J_k\}$ be the set of all trajectories, subdivided into $J = J_{\max} \cup J'$.

The subset $J_{\max}$ is defined as

$$J_{\max} = \{p \mid \exists\, (x', y')_{t',p'} \text{ with } p' = p \text{ and } t = t_{\max}\} \tag{5.5}$$

That is to denote all 'visible' trajectories in the last frame of the biomovie. $J'$ includes the complement to $J_{\max}$. The latter are visualized using approach (1) NM, and $J'$ are displayed with size $s_{J'} = 1$, and RGBa color $c_{J'} = (255, 255, 255, 0.1)$.

### 5.5.5 An enriched space-time cube

These three mappings are available as part of SEEVIS, a data driven (S)egmentation-fr(EE) and automatic pipeline of methods to (VIS)ualize the growth patterns of a cell population conveyed in a biomovie. Grasping a mental image of a highly dense and ever-growing bacterial population is quite challenging. Especially, when a biomovie holds a low image contrast and a high cell density. In an effort to gain insight at the large scale, the aforementioned high values for the five different properties impeded on proper segmentation results. For these reasons, I opted for a segmentation free based visualization approach, or SEEVIS, that employs particle

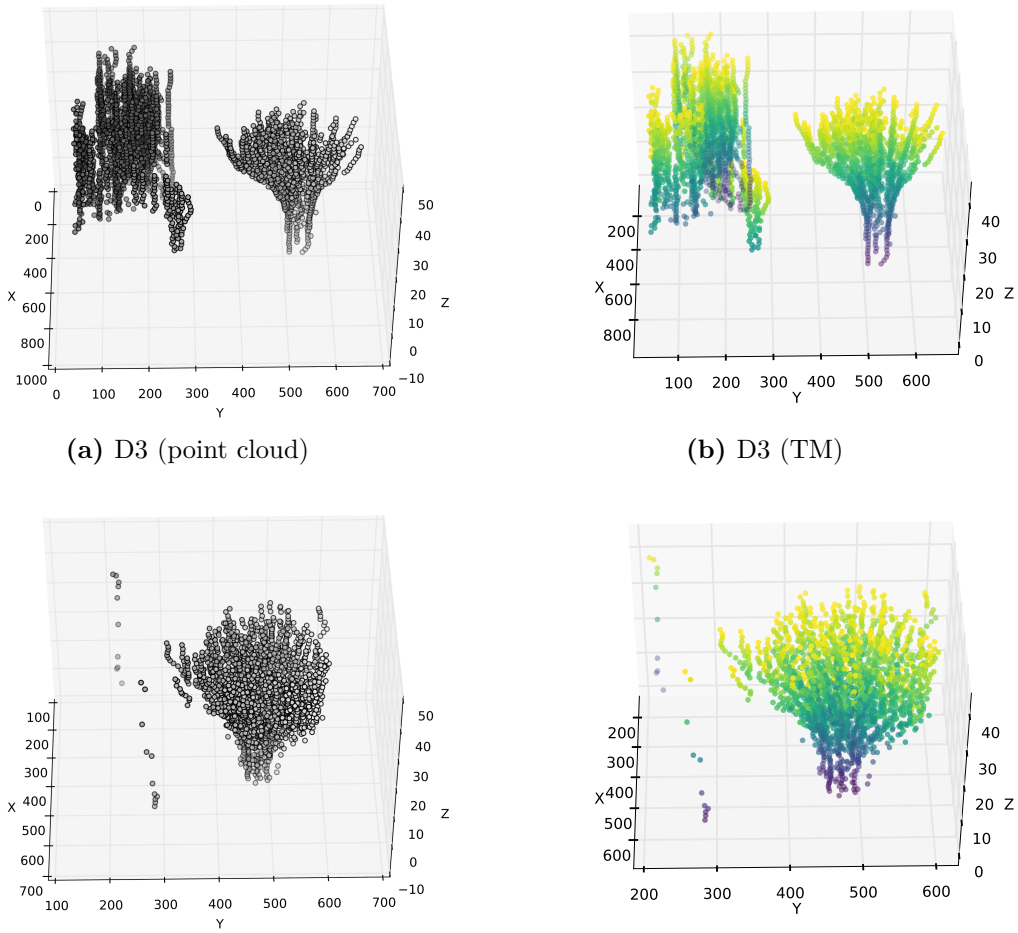
trajectories as a basis for the visualization. I extend the space-time cube by designing color mappings that are adapted to cell growth and enable a rapid investigation of the colony (e.g. visualize the effects of image registration (see chapter 3: Fig. 4.5).

Provided a preferred mapping, users can run the whole pipeline of SEEVIS to render the growth of an entire colony into a visualization. It exports post-processed images and particle positions into image files and a comma separated file (CSV), respectively. SEEVIS ran on both the heterogeneity (D1-D2) and communication experiments (D3-D4), averaging a speed of 1.15 s/image. SEEVIS achieved prompt qualitative results to better appreciate the extent of the colony. This is possible by providing users visual maps with different color mappings while preserving both space and time. Figures 5.20, 5.21, and 5.22 illustrate these results for D1–D4.

The nominal mapping (NM), highlighted all trajectories using ten categorical colors. Yet, being confronted with a large number of data points, this resulted in a cluttered visualization. I justify the use of ten colors so to uniquely identify the different trajectories locally.

The time mapping (TM), delivered a visualization, which promoted the colony growth by displaying the extent of the colony in time. In early times points, the colony was observed in the center of the visualization, it ranged from purple-blue, to a turquoise, then green, reaching the extremities of the colony in yellow. This mapping laid clear emphasis on growth by weighing the factor of time using the `viridis` colormap.

Compared to NM, the progeny mapping (PM) proved to reduce the aforementioned clutter by coloring only particle trajectories pertaining to cells which survived (i.e. present in the last frame of the biomovie). In Figure 5.21, I observed that another colony invaded the initial field of view. The third approach, i.e. PM, clearly shows the temporal shift by color, providing the means to select the time point at which the distance between the initial colony and an invading one is no longer trivial so to prune the particle trajectories. Moreover, in Figure 5.20, I depict another implementation using the `Matplotlib` rendering engine of the space-time cube, including depth shading and the time mapping (TM). This showcases the adaptability of my approach to other libraries or rendering engines (e.g. `Matplotlib`).



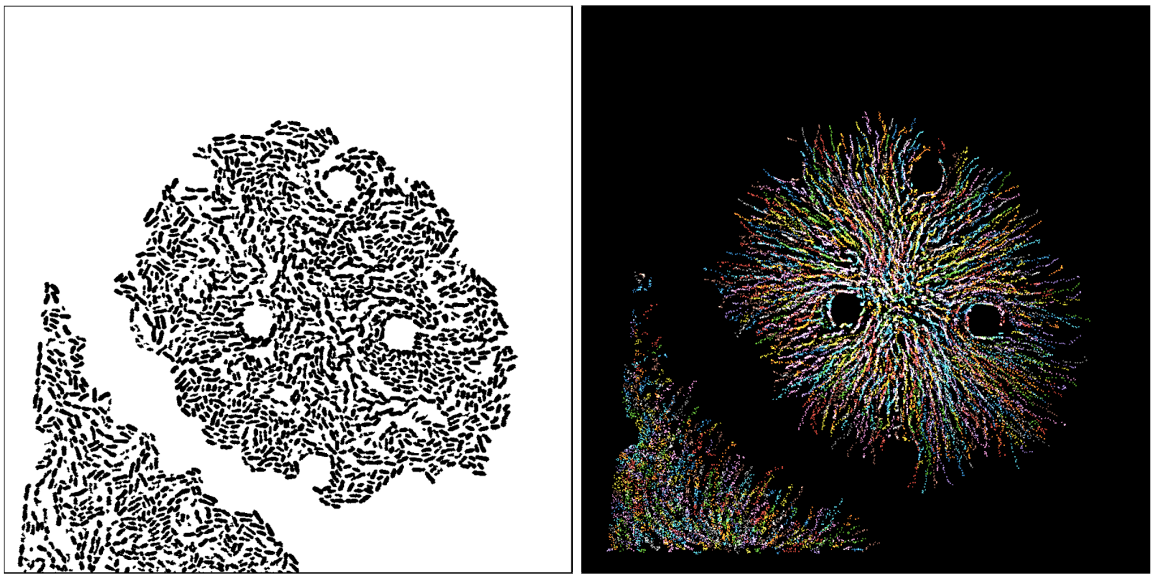

**(a)** D3 (point cloud)

**(b)** D3 (TM)

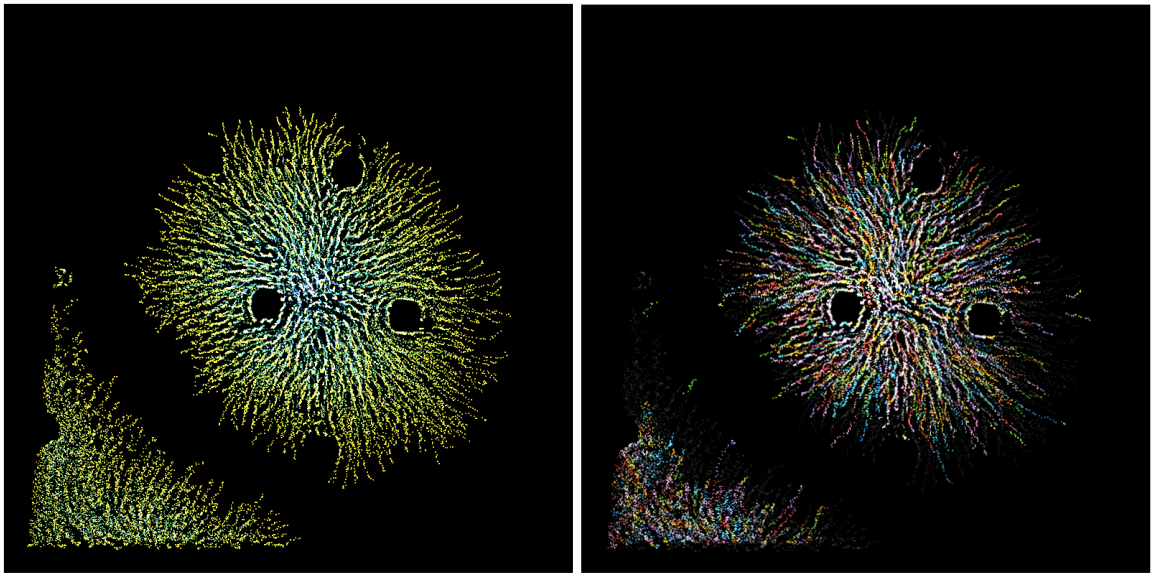
**(c)** D4 (point cloud)

**(d)** D4 (TM)

**Figure 5.20:** The time mapping methodology using the `Matplotlib` rendering engine for biomovies D3 and D4. The space-time cube depicting a depth-shaded point cloud with azimuth = 359°, and elevation = 45°. (a, d). Space-time cube with depth shade and no color encoding of biomovies D3 and D4, respectively. (b, c). Space-time cube with depth shade and the TM color mapping of biomovies D3 and D4, respectively.

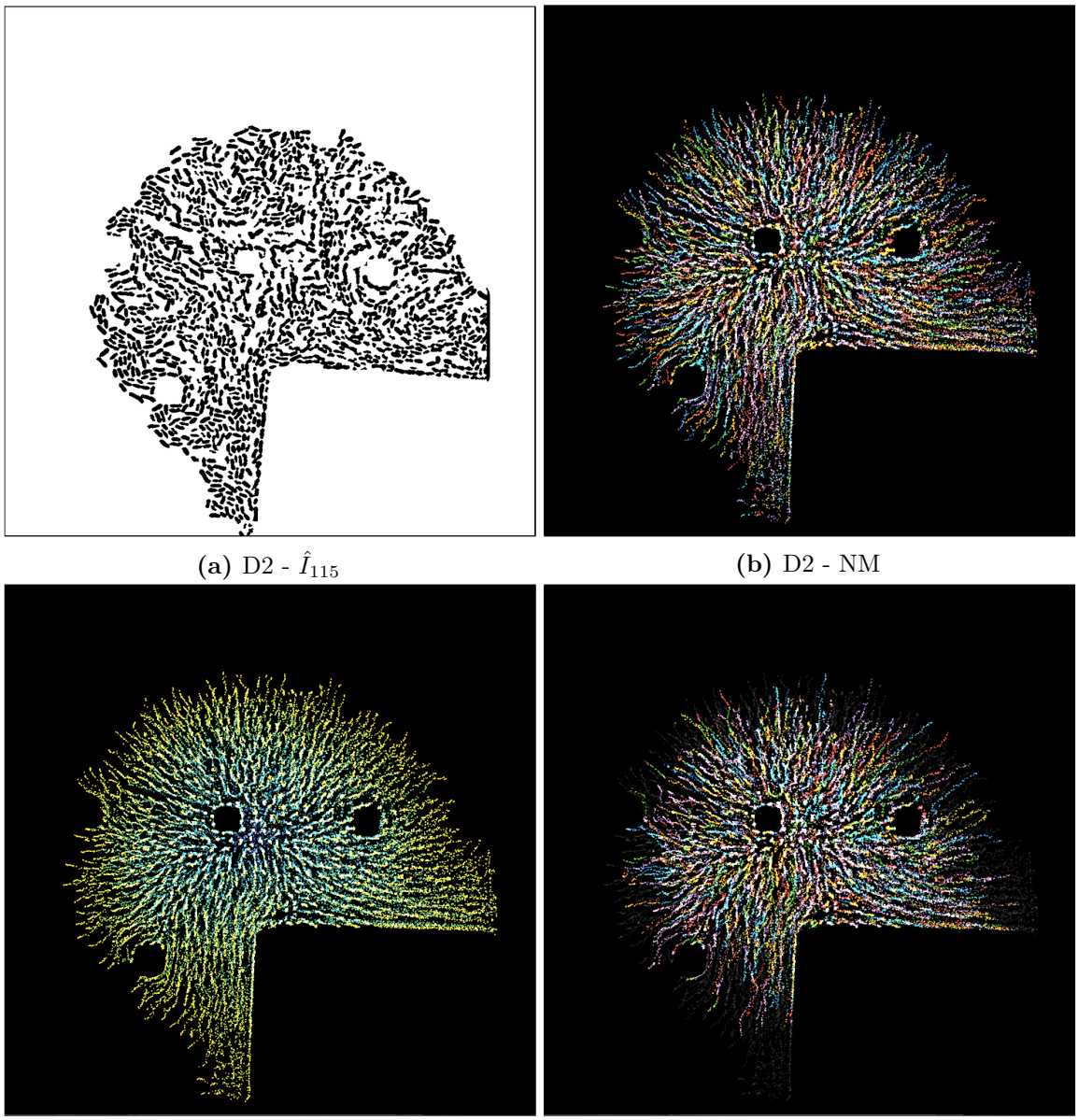


**(a)** D1 - $\hat{I}_{115}$ **(b)** D1 - NM



**(c)** D1 - TM **(d)** D1 - PM

**Figure 5.21:** Color mappings demonstrated for biomovie D1. (a) Biomovie D1 binary image. The space-time cube is displayed with azimuth = 0°, and elevation = 90° for the three mappings. (b) NM, (c) TM, and (d) PM, respectively. D1 comprises a central and an invading colony (from the lower left corner of the biomovie). (b) The nominal mapping displays the corresponding visualization highlighting the single trajectories. (c) The temporal mapping visually promotes the colony growth over time. It is observable that the center of the colony ranges from dark purple to blue, green until reaching light yellow at the colony extremity. (d) The progeny mapping showcases only the surviving particle trajectories while employing NM. There is a clear decrease in the number of observed trajectories.



**(a)** D2 - $\hat{I}_{115}$ **(b)** D2 - NM

**(c)** D2 - TM **(d)** D2 - PM

**Figure 5.22:** Color mappings demonstrated for biomovie D2. (a) Biomovie D2 binary image. The space-time cube is displayed with azimuth = 0° and elevation = 90° for the three mappings. (b) NM, (c) TM, and (d) PM, respectively.

**Summary**: While supplying standard interactive capabilities ranging from panning, rotation, to zooming, our visualization offers three color mapping methods for biomovies. In turn, SEEVIS helps by creating a mental map of the data. These mappings enrich the space time cube visualization and extend it for cell colony growth. The resulting visualization enables users to look into spatial and temporal growth in a timely manner. Moreover, as seen in the end of chapter 4, it is possible to use the graphical engine of SEEVIS to depict the effect of spatial shift and its correction in a biomovie (see Fig. 4.6 and Fig. 4.7). SEEVIS includes all the following methods: the preprocessing, the particle detection and linking, and the visualization.

Thanks to the nature of a particle, the growth is captured at a finer grain. The high density of particles results in visual occlusion, yet supports the next abstraction computationally.

## 5.6 Patch lineages

To move from a particle level to a level of subpopulations, it is necessary to weigh in biological concepts: (a) the natural biological growth of bacteria determines how the colony grows. A mother cell splits into two daughter cells, i.e. doubling. The cell count is highest at the end of a biomovie. (b) Similar cell characteristics reflect a similar behavior. To the naked eye, cells appear to be forming spatially coherent populations. (c) The coherence of such characteristics (e.g. fluorescence) varies throughout time.

To develop a valid approach, it is imperative to bear these concepts in mind. Briefly, my approach relies on both coherence and an algorithm that respects the aforementioned concepts. This coherence includes both space and time. For spatial coherence, previously computed particles that hold to certain spatial and color information are aggregated into a patch. Provided the particle trajectories $\{J_k\}$, the patch information is propagated throughout time. In turn, I obtain patch trajectories. Although temporal coherence is conveyed by particle trajectories, changes in fluorescence may happen and lead to heterogenous populations. This leads to splitting patch trajectories at a time point $t$, so as to homogenize patches. Yet splitting to solely homogenize patches may cause an over-segmentation of the patches. It is possible to observe fluctuations in the fluorescence at different time points. The nature of such fluctuations can either be consistent throughout time or be a time point based event. To

address such an over-segmentation and find consistent fluorescence profiles, a patch merging computation is applied. It consists of verifying if two patches are geometrically intersecting and merging them only and only if they are homogenous in a given time window. A patch lineage results and encapsulates the splitting and joining of all the patch trajectories that descend from a common ancestor patch.

**Algorithm overview**: A particle has visually distinct qualities, features, or attributes. In this work, position and color are considered in a feature vector $\mathbf{v}_{t,p}$ for each particle $(x, y)_{t,p}$. It is of course possible to add texture or other attributes. The algorithm comprises four main steps: patch finding, patch trajectory propagation, patch trajectory splitting, and merging. They are biologically motivated and respect the aforementioned biological concepts (a–c).

Patch finding starts at the last frame of a biomovie to aggregate particles with similar signal characteristics. The motivation to begin the computation from the last time point is biological: the maximum number of cells appears at the end of the growth sequence. As an observer, I am positioned at the last frame since it contains the resulting colony and I am able to look back in time (see Fig. 5.23). This first step evaluates heuristically defined constraints on the feature vectors $\{\mathbf{v}_{t,p}\}$ of particles.

The second step: patch trajectory propagation relies on previously found information which is propagated upstream. As an observer, I rewind to $t_0$ as seen in Fig. 5.26.

Patch trajectory splitting is applied to the patch trajectories $\{J_k\}$, at each time point from last to first. As an observer at time point $t$, I observe differences within different patches and I split the trajectories at $t$ to $t_0$. The split procedure can be interpreted as intra-patch verification, i.e. step 3. Positions are checked at each time point for all particles associated with a patch using the same patch finding. In cases of divergence, particles are split out to a new patch (see Fig. 5.27).

The final step finishes with patch trajectory merging, i.e. step 4. As an observer, I observe neighboring and over-segmented patches that behave similarly; then decide to merge them. This step can be interpreted as an inter-patch verification. It determines whether every possible pair of patches should merge or remain separated (see Fig. 5.28). This occurs from the first to the last time point and mirrors biological cell growth or division. The resulting patch trajectories couple spatial and temporal coherence (see Fig. 5.33).

### 5.6.1 Patch finding

A patch at time point $t$ is the aggregation of spatially contiguous particle trajectories that feature similar signal characteristics; that is, cell subpopulations with similar fluorescence patterns. To create patches from particles, in one image $I_t$, I define a decision function $\Phi(\mathbf{v}_{t,p}, \mathbf{v}_{t,p\prime})$ for the similarity in signal characteristics in the feature space of particles $p$ and $p'$. $\Phi$ could either be one Minkowski metric or a scalar product, joining together multiple particles into a coherent patch. For instance, in some cases, only one color channel might be considered (see Patch lineage graphs).

I am considering features from different domains, i.e. space $(x, y)$ and color $(r, g, b)$. The decision function $\Phi(p, p') = \{1, 0\}$ is defined as a Boolean evaluation of different user thresholds, such as:

$$\Phi(p, p') = \Phi(\mathbf{v}_{t,p}, \mathbf{v}_{t,p'}) = \Pi_j \phi_j(\mathbf{v}_{t,p}, \mathbf{v}_{t,p'}) = \phi_1 \cdot \phi_2 \cdot \phi_3 \cdot \phi_4 \tag{5.6}$$

$$\phi_1(\mathbf{v}_{t,p}, \mathbf{v}_{t,p'}) = \begin{cases} 1 & \text{if } \mathbf{d}((x, y), (x', y')) < t_d \\ 0 & \text{else} \end{cases} \tag{5.7}$$

$$\phi_2(\mathbf{v}_{t,p}, \mathbf{v}_{t,p'}) = \begin{cases} 1 & \text{if } \delta_r = |r - r'| < t_r \\ 0 & \text{else} \end{cases} \tag{5.8}$$

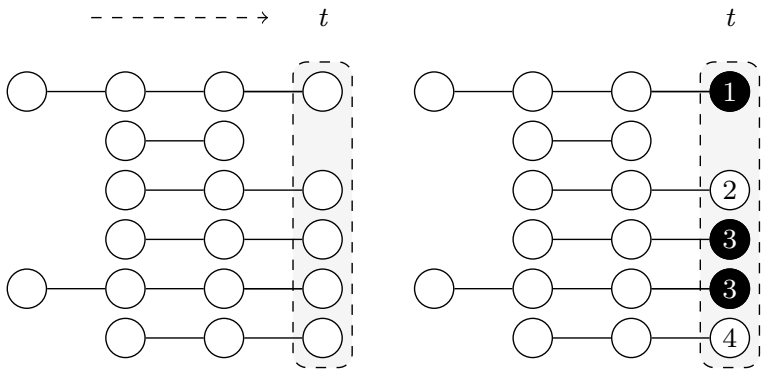
$$\phi_3(\mathbf{v}_{t,p}, \mathbf{v}_{t,p'}) = \begin{cases} 1 & \text{if } \delta_g = |g - g'| < t_g \\ 0 & \text{else} \end{cases} \tag{5.9}$$

$$\phi_4(\mathbf{v}_{t,p}, \mathbf{v}_{t,p'}) = \begin{cases} 1 & \text{if } \delta_b = |b - b'| < t_b \\ 0 & \text{else} \end{cases} \tag{5.10}$$

with user thresholds for space (i.e. distance) $t_d$ and color $t_r, t_g, t_b$, respectively. In principle, other functions can be defined to fit user needs.

The graphical examples in Fig. 5.23, 5.26, 5.27, 5.28, 5.33 show particle trajectories pro-

jected onto rows where time runs from left to right. Particles are colored white, grey, or black to illustrate feature space differences, i.e. particle of the same grey value have similar features ($\Phi(\mathbf{v}_{t,p}, \mathbf{v}_{t,p'}) = 1$). The patch lineage computation begins with an initial patch finding propagation at the last time point, as shown in Fig. 5.23(a). Particles pairs that satisfy the user thresholds are grouped into four patches labelled with distinct patch IDs in Fig. 5.23(b), where patch 3 contains two neighboring particles of the same black color.




(a) Particle traj. at $t$ (b) Find patches $j$

**Figure 5.23:** Graphical description illustrating patch finding in the first step of the patch lineage construction algorithm. Each row shows a temporally coherent particle trajectory that is close to those above and below it in feature space. The dots represent particle positions at each time point and their coloring of white/grey/black represents differences found in feature space provided the user-specified thresholds, respectively. The slice of space-time that is the focus of computation in each subfigure is highlighted by grey boxes with dashed outlines. (a) Biomovies have a naturally occurring temporal direction, represented as a dashed arrow ending at time $t$. The trajectories have a different number of particles, showing that particles can appear at any time point. (b) Particle trajectories are grouped into patches at the last time point.

The patch finding methodology is described in four major computations described in detail below: an all-pairs testing of particles, a particle pairs mapping to vertices in a graph data structure, connected components or patches finding by running a Depth-First Search on the graph, and the computation of their respective boundaries at each time point.

### All pairs-testing of particles

First, the algorithm starts at $t_{\max}$ and finds all particle pairs, from a particle point set $\mathbf{P}$, that hold $\Phi(\mathbf{v}_{t,p}, \mathbf{v}_{t,p'}) = 1$ given their feature vectors $\mathbf{v}_{t,p}$ and $\mathbf{v}_{t,p'}$. This is done by brute

force testing all pairs combinations.

Let $\mathbf{P} = \{(x, y)_{1,1}, \ldots, (x, y)_{1,p}\}$ be a particle point set at one image; with $t$ = time index, $p$ = particle index, and where $m$ is the number of particles in $\mathbf{P}$. Since the algorithm is at $t_{\max}$, the notation can be simplified as follows for each particle position at one time point, $\mathbf{P} = \{(x, y)_1, \ldots, (x, y)_p\}$ with $p$ = particle index, where $(x, y)_1, \ldots, (x, y)_p$ is the sorted list of positions by particle index. The all-pairs testing entails an initialization where three points are addressed:

(a) find ${}^mC_2$ combinations of all non-redundant particle pairs from $\mathbf{P}$ particles with ${}^mC_2 = \frac{m!}{2!(m-2)!}$

(b) compute metrics for each pair: geometrical distance, channel specific differences

(c) evaluate the particle pairs by using the boolean function $\Phi$.

For an example set of particles $m = 5$ with particle indices $[1 : 5]$, the computation in (a) ${}^5C_2$ results in 10 unique pairs: $\{(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)\}$. It is computed in $O(n^2)$ using the `itertools` package. Next in (b), the metrics associated with each pair are computed: the geometrical distance of two given particle coordinates is $\mathbf{d}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ and specific channel differences, e.g. for the red channel I define $\delta_r = |r_1 - r_2|$. This results in a vector for a particle pair $\mathbf{v}_{t,p\prime}$. In (c), the evaluation occurs using the boolean function $\Phi(\mathbf{v}_{t,p}, \mathbf{v}_{t,p\prime})$ by relying on the conjunction (i.e. AND operator) of the user-provided metrics. This process runs in quasilinear time $O(m \log m)$.

### Particle pairs mapping to vertices

Second, if particle pairs hold $\Phi(p, p') = 1$ (eq. (5.6)), particle indices are retrieved and are grouped together. These particles are referred to as interacting, as opposed to non-interacting. Each interaction is iteratively added to an undirected, unweighed, and simple graph $\mathbf{G}$ by mapping each particle index pair, for example $(1, 2)$ to a unique vertex pair $(v_1, v_2)$. This is done so the vertex $v_2$ is reachable from the vertex $v_1$, given an edge $e_1$ from $v_1$ to $v_2$. Once all-pairs testing is done and interacting particle pairs have been added to $G$, the algorithm sorts and finds all non-interacting particles, then adds them as singleton vertices. Provided

an example set of particles $m = 6$ with three interactions for particles 1, 2, 3, one for 4, 5, and no-interactions for particle 6, respectively; see Fig. 5.24. The list of interactions can be written: $((3,2),(3,1),(1,2),(4,5),(6))$ and is supplied as input to the next step.

### Connected components finding

Third, the algorithm finds connected components (i.e. all subsets of interacting particles) by running a Depth-First Search (DFS)[113] on **G**. A connected component of a vertex is the subgraph containing all paths in the graph that visit the vertex. In the case of an undirected graph, a path is defined as a finite and alternating sequence of distinct vertices and edges: $v_1, e_1, v_2, \ldots, v_k, e_k, v_k$, which begins and ends with vertices. Hence, the endpoints of $e_i$ are $v_i$, and $v_{i+1}$. DFS traverses **G** and explores possible vertices, as far as possible, along each path, by marking the current vertex as being visited, and exploring each adjacent vertex that is not included in the visited set. In the context of finding connected components, if one starts from a start vertex DFS marks all the vertices connected to the start vertex as visited. Therefore, if one chooses any vertex in a connected component and run DFS on that node, it will mark the whole connected component as visited.

Given the aforementioned example and based on the nested parentheses representation or the Newick format, running the DFS results in a forest: two trees and one singletons $(1,2,3);(4,5);(6);$ as depicted in Figure 5.24. The step by step search from vertex 2 is separated by a comma and occurs as follows: 2, 2->1, 2->1->3.

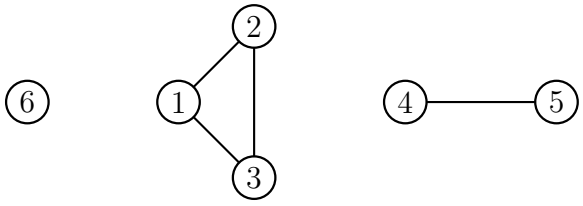



**Figure 5.24:** Each connected component of a graph **G** is a maximal connected subgraph of **G**. Given the disconnected graph **G**, there are three connected components. Each is detailed with the found paths upon visiting each vertex using the DFS. The first is the vertex 6, the second is 2->1->3 as they are linked to each other, and the third is 4->5.

Since the complexity of a search is $O(i_v + j_v)$, for each vertex $v$, let $i_v$ denote the number of vertices in the connected component containing $v$, and $j_v$ for the edges. Let $\mathcal{T}$ be a transversal

of the vertex sets, one from each component. To return all the connected components, the search starts with each vertex, stopping in $O(1)$ time if the vertex's component is encountered. The final time can be expressed by $\sum_{v \in \mathcal{T}} O(i_v + j_v) = O(i + j)$.

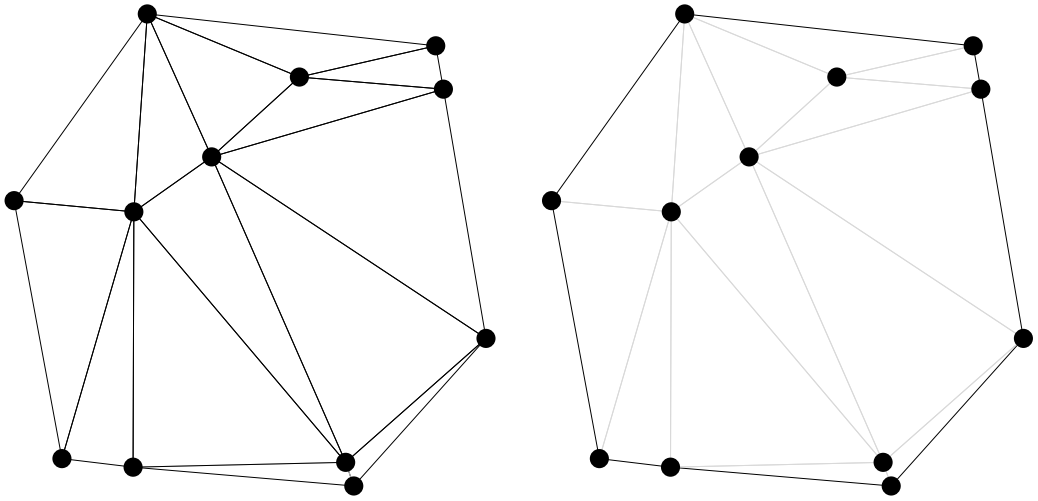
My implementation of the DFS uses a recursive approach to return the invoked located edges. The resulting graph encompasses all the connected components as subgraphs, hence creating the graph $\mathbf{G}$ can be written: $\mathbf{G} = (S_1, S_2, \ldots, S_n) : n =$subgraph index, or patch index, with each subgraph $S_n$ containing vertices $i_v$ that bear unique particle indices. Each subgraph index encodes a unique patch ID for each particle at $t_{\max}$ in a data frame as a 2-dimensional labeled data structure. Particle positions become $\{(x, y)_{t,p,n}\}$, with $t =$ time index, $p =$ particle index and $n =$ patch index which concludes the creation of patches.

### Patch boundary

Fourth and last, one finds the boundary of each connected component, or patch, $S_n$, with $n \geq 6$ at each time point $t$ with the help of the Delaunay tessellation algorithm. Let $\mathbf{P}_1 = \{c_1, \ldots, c_p\}$ be the coordinate point set of patch 1, with $p =$ particle index and $\mathbf{P}_1 \in \mathbf{P}$. To be able to formally define a triangulation of $\mathbf{P}_1$, I first define a *maximal planar subdivision* as a subdivision $U$ such that no edge connecting two vertices can be added to $U$ without destroying its planarity. In other words, any edge that is in $U$ intersects one of the existing edges. A triangulation $\mathcal{T}$ of $\mathbf{P}_1$ is then defined as a maximal planar subdivision whose vertex set is $\mathbf{P}_1$.

Every facet, except the unbounded one, must be a triangle: a bounded face is a polygon, hence can be triangulated. A Delaunay tessellation or Delaunay triangulation in the plane, is a subdivision of a set of coordinate points $\mathbf{P}_1$ into a non-overlapping set of triangles, such that no point in $\mathbf{P}_1$ is inside the circumcircle of any triangle in this triangulation. In practice, such triangulations maximize the minimum angle of all the angles of the resulting triangles.

As observed in Figure 5.25(a), any segment connecting two consecutive points on the boundary of the convex hull of $\mathbf{P}_1$ is an edge in any triangulation $\mathcal{T}$. This implies that the union of the bounded faces of $\mathcal{T}$ is always the convex hull of $\mathbf{P}$, and that the unbounded face is always the complement of the convex hull. In this application, the diversity in colony growth results in variable patch shapes. This implies that if the patch shape is a rectangular area, I have to make sure that the corners of the patch are included in the set of points, so



**(a)** Delaunay triangulation **(b)** Convex hull

**Figure 5.25:** Example patch with a particle point set of size $p = 12$. (a) A Delaunay triangulation, with 14 triangles and 25 edges. (b) The convex hull boundary, with 8 points and 8 edges.

that the triangles in the triangulation cover the patch. Let $\mathcal{T}$ be a triangulation of $\mathbf{P}_1$ with $d$ triangles. The number of triangles is the same for any of the triangulations of $\mathbf{P}_1$, likewise for the edges. The exact numbers depend on the number of points in $\mathbf{P}_1$ that are on the boundary of the convex hull of $\mathbf{P}_1$.

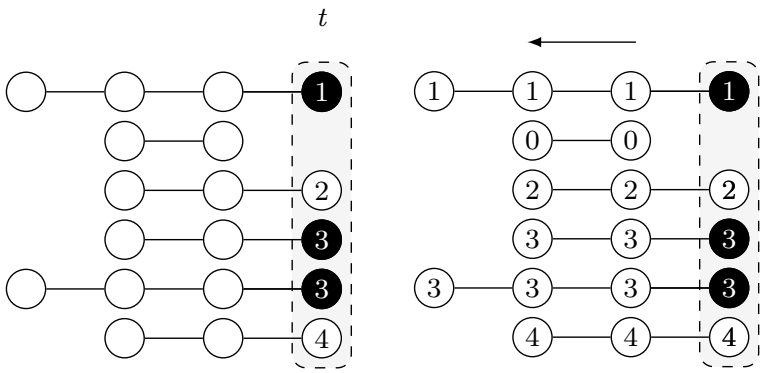
Let $p$ be the number of particle points, and $q$ be the number of points on the convex hull of $\mathbf{P}_1$. Provided the aforementioned properties in the 2-dimensional plane, the Delaunay triangulation contains $O(n)$ simplices. Moreover, provided $q$ vertices on the convex (i.e. $q$ edges on the unbounded face) and based on Euler's characteristic: any triangulation of the points has at most $2p - 2 - q$ triangles, $3p - 3 - q$ edges (i.e. every triangle has three edges and every edge is incident to exactly two faces). This permits me to calculate the number of triangles and edges for the provided example in Fig. 5.25.

By applying the Delaunay triangulation, the algorithm triangulates the irregular grid coordinates using an expected run time in $O(q \log q)$ for $q$ points in the plane. The structure of a triangulation $\mathcal{T}$ is encoded such as the simplices attribute contains the indices of the points $\mathbf{P}_1$. In Figure 5.25(b), the convex hull is represented as a set of 1-dimensional simplices, that is line segments in 2-dimensions. The storage scheme of the convex hull simplices is exactly the same.

In the special case of this work, the convex hull is not sufficiently precise to describe the

irregular shape of patches, or subpopulations in the domain knowledge of biology. Computing the convex hull gives access to the counterclockwise ordered list of its simplical facets. Provided the convex hull and its simplical facets, it is possible to find the non-convex polygon that defines the enclosure of the given set of points (i.e. concave hull or the alpha shape).

In summary, each subgraph of a patch undergoes the following steps: Each facet of the DT is temporarily stored into a graph data structure (i.e. a subgraph). Next, the DFS search is applied to the temporary subgraph (with the particle index as vertices). The result is used to verify that the subgraph is one connected component. Provided the subgraph, the convex hull is computed, and its simplices are stored in a counterclockwise ordered list, respectively. The employed implementation relies on the graph data structure and the `Qhull` library[114]. The library includes the computation of the Delaunay triangulation and the convex hull.




(a) Find patches $j$ (b) Propagate $T_j$

**Figure 5.26:** Graphical description illustrating patch trajectory finding and propagation. Each row shows a temporally coherent particle trajectory that is close to those above and below it in feature space. The dots represent particle positions at each time point and their coloring of white/grey/black represents differences found in feature space provided the user-specified thresholds, respectively. The slice of space-time that is the focus of computation in each subfigure is highlighted by grey boxes with dashed outlines. The black arrow indicates the direction of a propagation. (a) Particle trajectories are grouped into patches at the last time point. (b) The trajectory information is propagated upstream in a run from the last to the first time point.

### Patch trajectory propagation

A patch trajectory reflects the evolution of patches across multiple frames. After a patch is found in the previous step, the decision is propagated upstream by employing the temporal coherence of particle trajectories $J_k$ to patch trajectories $T_j$. The algorithm marches backwards from time $t_{\max} \rightarrow t_0$, inspecting each particle trajectory that appears in the frame. The algorithm either propagates the patch ID from downstream for existing particle trajectories or assigns a new patch ID when a new particle trajectory is first encountered (i.e. a particle trajectory not yet assigned to a patch).

Figure 5.26 shows the result, where the patch trajectory in the second row that has no particle trajectory visible in the last time point has been assigned the patch ID 0. Provided the following example, where $t_{\max} = 10$, and subgraph (1,2,3); let $T_1$ be the patch trajectory of patch ID $n = 1$ such as

$$T_1 = \{(x,y)_{10,p_1,1}, (x,y)_{10,p_2,1}, (x,y)_{10,p_3,1}, \ldots, (x,y)_{1,p_1,1}, (x,y)_{1,p_2,1}, (x,y)_{1,p_3,1}\} \quad (5.11)$$

## 5.6.2 Patch trajectory splitting

Although temporal coherence is conveyed by particle trajectories, changes in fluorescence may happen and lead to heterogenous populations. For example, changes in fluorescence may indicate emerging behaviors in a population or a patch. In that instance, such a patch shows considerable differences in fluorescence, it ultimately requires splitting.

The split computation is a second propagation that verifies a patch in its evolution for spatial consistencies. It runs from $t_{\max}$ to $t_0$, like the first propagation. Unlike the first one, it only propagates the patch information after verifying patches at each time point (or inter-patch verification). If the user-specified distance and color thresholds are surpassed for all particles within a patch, a split is required. Provided many inconsistencies, splitting one patch into multiple patches can occur. Moreover, depending on the patch size, a split may correspond to an emerging behavior within a subpopulation. Figure 5.27(B) shows an example where patch 3 is split when a feature change is noticed at the second to last time point and the particle trajectory is assigned a new patch ID 5.

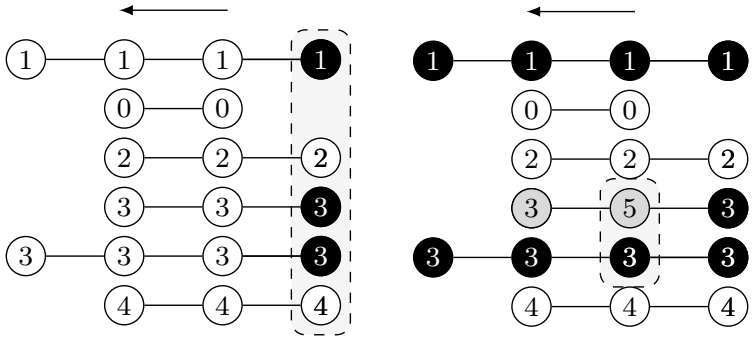
The split computation is divided into two steps: finding non-singleton patches, evaluating

and encoding the patch locally.

### Non-singleton patch finding

While the algorithm iterates from $t_{\max}$ to $t_0$, it first evaluates patches for consistency by mapping all non-singleton patches and their particles onto a temporary graph $\mathbf{G}'$. Then, the DFS is applied to find connected components in $\mathbf{G}'$.

Let $S_1$ be the first patch (or subgraph) of all non-singleton subgraphs $\{S_n\}$ with $n$=subgraph index. Let $i_v$ be the number of vertices in the connected component that satisfy the following condition: $i_v > 1$.



(a) Propagate $T_j$ (b) Split patch traj. $T_3$

**Figure 5.27:** Graphical description illustrating patch trajectory propagation and splitting. The black arrow indicates the direction of a propagation. (a) The trajectory information is propagated upstream in a run from the last to the first time point. (b) The split propagation proceeds from the last to the first time point.

### Patch evaluation and encoding

Second, the subgraph is evaluated to find whether the particle point set $S_1$ remains one connected component, using the patch creation computation. If running the aforementioned DFS algorithm results in two or more subgraphs. Let $S_{1'}$ and $S_{2'}$ be two output subgraphs, with their respective number of vertices $i_{1'}$ and $i_{2'}$, and a maximum patch ID $n = 3$: If $i_{1'} > i_{2'}$, then $S_{1'} = S_1$ and $S_{2'} = S_{n+1}$. That is to say the newly created patch has its

ID assigned incrementally, in respect to the maximum patch ID $n$. Else, the exact same subgraph results, then the computation continues onto the next subgraph.

The example encoding of particle vertices in $S_{1'}$ to patch ID $n = 1$, and of those in $S_{2'}$ to $n = 3 + 1 = 4$ is carried out in linear time $O(m)$ using the aforementioned 2-dimensional data frame structure for $m$ particles.

### 5.6.3 Patch trajectory merging

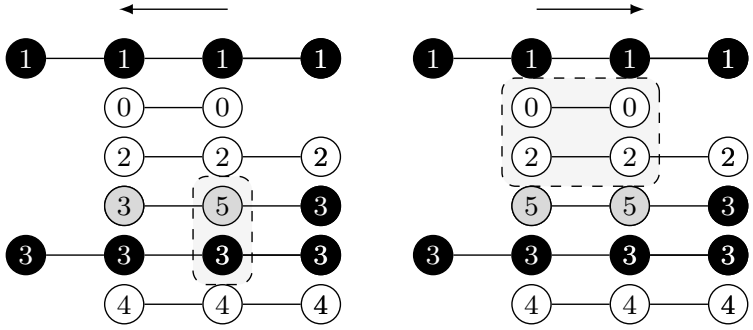
Separating the splitting and merging procedures into separate sequential propagations follows a chunking strategy. The need to merge patches arises from over-segmenting patches using the splitting procedure. To avoid carefully tuning splits to avoid 'over-segmentation' into overly small patch trajectories, this subsequent merge propagation takes care of that.

In this third propagation, patch trajectories are compared iterating over time yet in a forward direction from the first time point $t_0$ to $t_{\max}$. The direction of this final computation matches the biology of patch growth, where previously separate regions touch due to the growth of new cells.

Patch trajectory merging requires checking for intersections between all pairs of patches that exist at each time point. I accelerate it with a fast initial intersection test between the oriented bounding rectangles to rule out patch pairs that have no geometric overlaps. I only evaluate the full set of bounding particles in cases of intersections, which may range from one-point contact to full inclusion of a patch into another.

Fig. 5.28(b) shows an example of how particle trajectories that are absent at the last time point are handled. The second particle trajectory was given patch ID 0 in the propagation phase, it is joined with the third trajectory as patch 2 because it falls within the merge window threshold $\omega_t$. The final set of five patches are enumerated by their patch ID in Figure 5.28(b). The set of patch trajectories is defined as $\{T_j\}$ with $0 \leq j \leq N$, the number of patch trajectories $N$.

The merge computation evaluates neighboring patches. Provided the convex polygon of a patch, I verify whether two patches should be in one by: checking for spatial intersections and verifying if the features of the bounding particles hold $\Phi(\mathbf{v}_{t,p}, \mathbf{v}_{t,p'}) = 1$.



(a) Split patch traj. $T_3$ (b) Merge $T_2$

**Figure 5.28:** Graphical description illustrating patch trajectory splitting and merging. Each row shows a temporally coherent particle trajectory that is close to those above and below it in feature space. The dots represent particle positions at each time point and their coloring of white/grey/black represents differences found in feature space provided the user-specified thresholds, respectively. The slice of space-time that is the focus of computation in each subfigure is highlighted by grey boxes with dashed outlines. The black arrow indicates the direction of a propagation. (a) The split propagation proceeds from the last to the first time point. (b) The merge propagation proceeds from first to the last time point, mirroring biological growth.

It corresponds to an inter-patch evaluation and is divided into two computations: an all-pairs testing of non-singleton patches at a time point $t$ to evaluate whether patch pairs intersect spatially and patch trajectory merging which relies on a user-defined merge window $\omega_t$.

### All-pairs testing of patches

First, the algorithm starts at $t_0$, and runs an all-pairs testing by combinatorially finding all patch pairs (see Patch finding). Let $\mathbf{G}'$ be a graph with $n$ subgraphs representing $n$ resulting patches of the above process: $\mathbf{G}' = (S_1, S2, \ldots, S_n)$. Let $(S_1, S_2)$ be a patch pair for the evaluation, with $P_1$ and $P_2$ their respective particle point set.

To find intersections between patches, I have to first find the smallest-area enclosing rectangle of each patch. Such a problem has received attention in the image processing literature and has many applications (e.g. layout problems). I apply the generalized Rotating Calipers method based on Shamo's algorithm to the minimum-area rectangle problem[97]. The idea of using the Rotating Calipers method establishes a connection between the input polygon's convex hull and the orientation of the resulting minimum-area enclosing rectangle.

It is based on the following theorem, which was proven by Freeman and Shapira[115]: The smallest-area enclosing rectangle of a polygon has a side collinear with one of the edges of its convex hull. This first step of the algorithm is depicted in Figure 5.31.

A pair of vertices $q_k, q_l$ is an antipodal pair if it admits parallel lines of support. An example antipodal pair (or parallel lines of support) is illustrated in Figure 5.29. Provided the convex hull of the input polygon, the algorithm is outlined step by step:

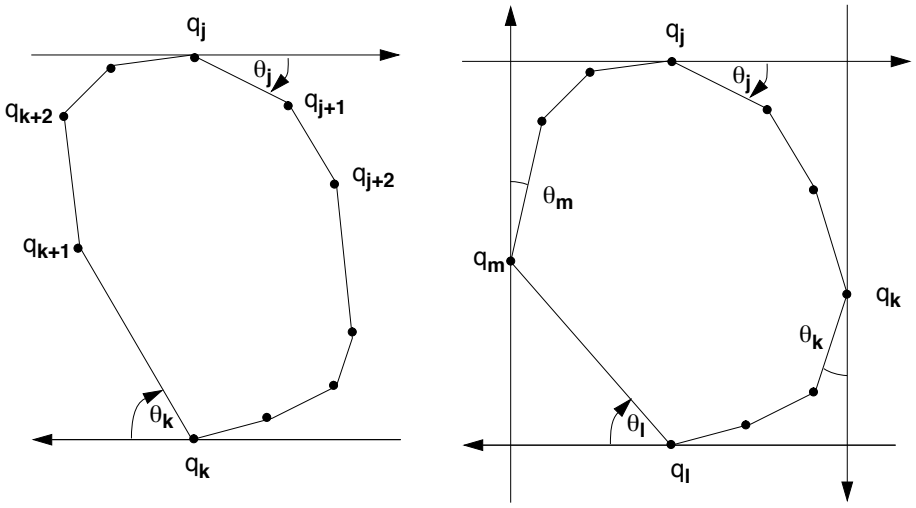



**Figure 5.29:** Illustration of an antipodal pair and the Rotating Calipers method. (Left) Shamo's algorithm generates all antipodal pairs of vertices and selects the pair with largest distance as the diameter-pair. Along the first x-axis, the method is initialized with two antipodal vertices $q_j$ and $q_k$. To obtain the next antipodal vertices, the angles that the lines of support make with edges $q_j q_{j+1}$ and $q_k q_{k+1}$ are $\theta_j$ and $\theta_k$, respectively. To rotate the lines of support, let $\theta_j > \theta_k$. $q_{j+1}$ and $q_k$ becomes the next antipodal pair. (Right) To find the smallest-area enclosing rectangle, two sets of calipers are required. The second set is orthogonal to the first. As in Shamo's diameter algorithm, four angles result: $\theta_j$, $\theta_k$, $\theta_l$, and $\theta_m$. Once the four lines of support are rotated by an angle $\theta_j$, $q_j, q_{j+1}$ forms the base line of the rectangle associated with the edge $q_j q_{j+1}$. This process is repeated until the entire polygon is scanned, i.e. each edge once coincided with one of the four caliper lines. Adapted figure[97].

1. Find the points for the polygon $q_{\min} = (x_{\min}, y_{\min})$ and $q_{\max} = (x_{\max}, y_{\max})$

2. Construct four lines of support for $P$ through $q_{\min}$ and $q_{\max}$. These determine two sets of 'calipers': two vertical supporting lines at $x_{\min}$ and $x_{\max}$ and two horizontal lines at $y_{\min}$ and $y_{\max}$, respectively

3. If one (or more) lines coincide with an edge, then compute the area of the rectangle determined by the four lines, and keep as minimum. Otherwise, consider the current minimum area to be infinite
4. Rotate the lines clockwise until one of them coincides with an edge of its polygon
5. Compute the area of the new rectangle and compare it to the current minimum area. Update the minimum if necessary, keeping track of the rectangle determining the minimum
6. Repeat steps 4 and 5, until the lines have been rotated an angle $\theta > 90°$
7. Output the minimum area enclosing rectangle.

The Rotating Calipers depends on the observation that, in two dimensions, one side of the minimal rectangle must coincide with one edge of the convex polygon it must contain. Its algorithm iterates in the main loop as many times as there are polygon vertices. Hence, the algorithm has a linear time complexity.

Provided one patch with its patch index $n = 1$, let $q_{n,k}, q_{n,k+1}$ be a polygon edge; with the $k^{\text{-th}}$ polygon vertex denoted $q_k$. For simplification, the patch index is omitted in Figure 5.29. The minimum rotated rectangle enclosing a patch is denoted by its vertices set $\{q_{n,k}\}$; such that for the patch pair $(S_1, S_2)$, the two vertices sets are $\{q_{1,k}, q_{1,k+1}, \ldots\}$ and $\{q_{2,k}, q_{2,k+1}, \ldots\}$, respectively.

Next, to find patch pair intersections, the cartesian coordinates of the rectangles vertices are tested. Two scenarios are possible, either no intersection or intersection; the latter includes partial intersection and enclosure of one rectangle in another. A boolean flag is returned for the presence of an intersection. Different intersection examples are illustrated in Figure 5.30.

Both the rotating calipers algorithm and intersection computation have an expected linear run time. In the worst case, all patches need to be verified, the expected run time is then quasilinear.

### Patch evaluation

Second, the patch evaluation is carried out on the set of intersecting patches using the aforementioned patch finding step; particularly the DFS algorithm. At a time point $t$, let $S_t1$ and

$S_t2$ be the two subgraphs of particle point sets $\mathbf{P}_1$ and $\mathbf{P}_2$ from patches 1 and 2, respectively. Let $\mathbf{P}_1$ and $\mathbf{P}_2$ be the particle point set of particles on the hull or the patch boundary. If an intersection is found in the previous step, the DFS algorithm is applied on the union of these subgraphs.

$$DFS(S_{t1} \cup S_{t2}) \Leftrightarrow S_{t1'} \tag{5.12}$$

With $S_{t1'}$ a connected component. If equation (5.12) holds, the patch indices are stored in a candidate merge list, formatted as `(t (1,2))`. Else, the algorithm iterates onto the next pair of patches that intersect. Provided a user-defined merge window $\omega_t$, patches that appear for the length of that window are propagated throughout the merge window.

For instance, provided $\mathbf{P}_1$ and $\mathbf{P}_2$ and a given merge window $\omega_t = 5$; such as patch 1 is larger than patch 2. Then the subset of particle positions in $\mathbf{P}_2$ for particle index 1 can

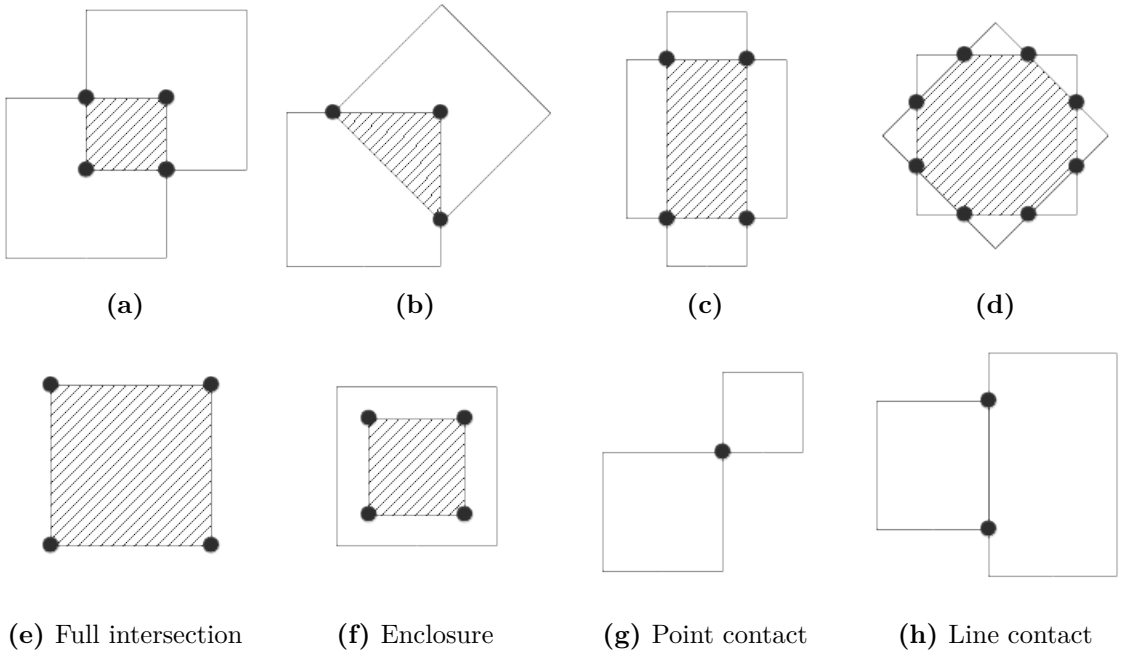
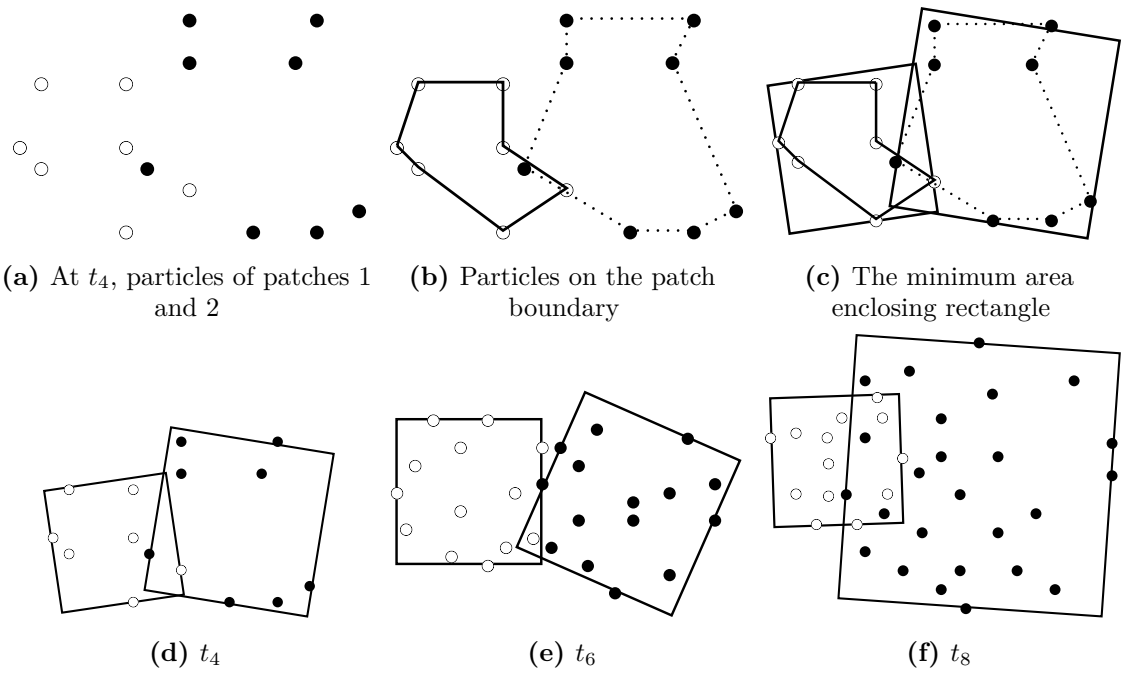



**Figure 5.30:** Example illustrations of intersection configurations for two minimum area rectangles. The textured pattern indicates the intersecting region. Intersecting vertices are indicated in black. (a–d) Cases of partial intersection. (e) Full intersection, where both rectangles share the same vertices. (f) One rectangle is enclosed into the other. (g) A point contact, where two rectangles share one vertex. (h) A line contact, where two rectangles share an edge. Figure adapted from OpenCV. http://docs.opencv.org/

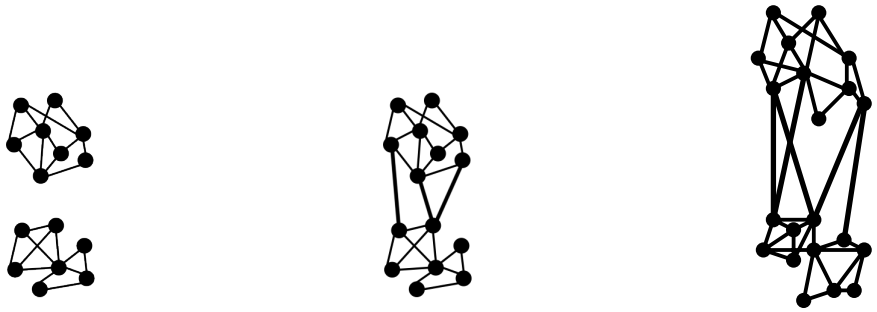


**(a)** At $t_4$, particles of patches 1 and 2 **(b)** Particles on the patch boundary **(c)** The minimum area enclosing rectangle

**(d)** $t_4$ **(e)** $t_6$ **(f)** $t_8$

**Figure 5.31:** Graphical description illustrating the first step of patch trajectory merging (i.e. all-pairs testing of patches). Each dot shows a particle position at a time point. The dot coloring black/white represents particles from different patches. The minimum area enclosing rectangle is depicted in black. All rectangles partially intersect at each time point. The hulls of each respective patch are depicted in full and dotted lines. In this example, the patch merging step of the algorithm has computed the minimum area enclosing rectangles for all patches from $t_4$ to $t_8$. (a) At time point $t_4$, the particles are depicted before the all-pairs testing of patches (b–c). (b) Particles on the patch boundary. These particles are later employed in the patch evaluation step (d–e) (c) The minimum area enclosing rectangle of the patches with their respective hulls. (d) Patch trajectory merging computes the minimum area enclosing rectangle of each patch using the Rotating Calipers algorithm. (e) Both patches show an increasing number of particles throughout time, which is in accordance with biological growth. (f) The split propagation computed a split at this time point. The decision was propagated to (b) then to (c). For temporal consistency, patch trajectory merging evaluates the patches from the first to the last time point. Figure 5.32 depicts an example for patches at $t_8$.

be written $\{(x,y)_{t=1,p=1,n=1},\ldots,(x,y)_{t=5,p=1,n=1}\}$. As depicted in Figure 5.32, the merge results in a reassignment of the patch ID. Otherwise, the algorithm iterates over the next subset of subgraphs (i.e. intersecting patches). A graphical example of a patch lineage result is illustrated in Figure 5.33.

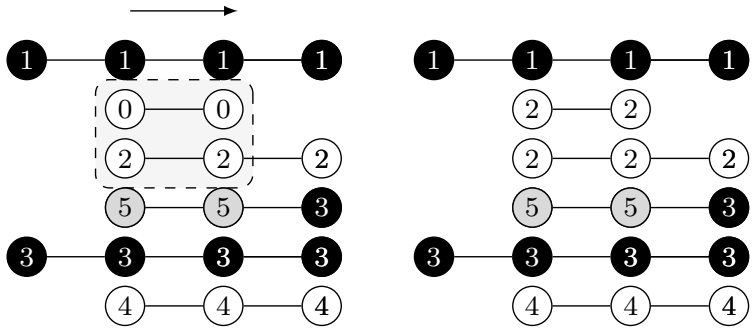


**(a)** At $t_4$, bounding particles are mapped onto graphs

**(b)** DFS is applied

**(c)** $t_8$

**Figure 5.32:** Graphical description illustrating the second step of patch trajectory merging (i.e. patch evaluation). Particles are depicted as vertices. The connected components are provided by the edges between the two subgraphs. (a) At $t_4$, the bounding particles (positions and feature vectors) are mapped to a temporary graph. (b) The patch evaluation step applies the DFS method to identify connected components. Provided a merge window, this step marks the trajectories for merging if the following condition is met: one and only one connected component results. (c) At $t_8$, the evaluation step also finds one connected component. If the algorithm finds that the subgraphs from $t_4$ to $t_8$ are connected components, the patch trajectories can then be merged into one patch trajectory.




(a) Merge $T_2$

(b) Result

**Figure 5.33:** Graphical description illustrating patch trajectory merging and the resulting patch lineage. (a) The merge propagation proceeds from first to the last time point, mirroring biological growth. (b) The resulting patch lineage contains 5 patches.

Moreover, thanks to the 2-dimensional data frame structure, the merge propagation is linear. The data structures that are employed by this framework are detailed in Appendix D.

### 5.6.4 Parameter space

I determined suitable user thresholds $t_d, t_r, t_g, t_b$ for the feature vectors through empirical exploration. I began by considering the basic descriptive statistics of the biomovies in the geometric and color distance channels: the colony diameter in pixels and for each color channel the minimum and maximum values and the standard deviation. I then approached the testing phase by completely discarding homogeneous channels to lower the noise and employing a sensitivity analysis for each threshold. This selective approach allowed for testing the robustness of the results and for increased understanding of the relationships between some thresholds and a desired output. An illustrative example is shown in Fig. 5.34.

To properly run this method, I suggest using more particles than the number of cells, by at least a factor of 2, by setting the particle diameter $d$ smaller than the minimum cell diameter (see Particle detection). There are two parameters that influence particle trajectory linking: the distance radius $\sigma_{\max}$ and the time linking interval $W_{\max}$. The larger the distance radius is, the more particles are evaluated by the neighbor-finding strategy in the particle detection step. Moreover, the larger the size of the time linking interval is, the more memory is allocated for particle positions within that time window. As reported in Section 3.2.2, I chose values in a way so the computational expense is limited. For trajectory filtering (time), it is reasonable to set the default filtering window in accordance to the frame count. Since short trajectories do not necessarily correspond to spurious ones, it is best to change the filtering window on a case by case basis. For example, in specific experimental conditions cells may have a short life span.

### 5.6.5 Patch visualization

Patches are visualized on a frame-by-frame basis, based on the enclosed particles. The visual encoding of a particle position varies in size and color. The particle size is a function of object visibility, that is to highlight relevant change. This is the case in Fig. 5.34 and Fig. 5.35, where the particle size is enhanced to better depict changes in color. The particle color encodes a patch ID and is chosen from the `Tableau10` categorical map. This categorical map helps

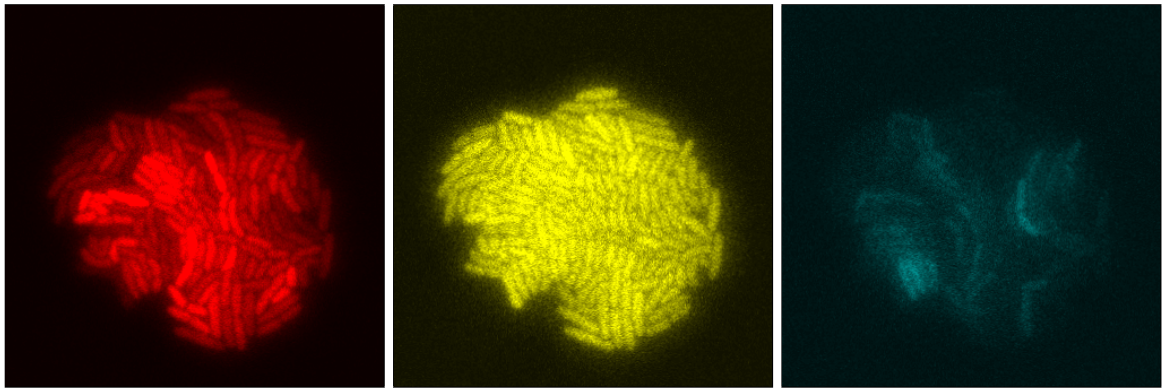
a human observer differentiate between neighboring particle positions of different patches. Such encodings permit the following: parameter tuning (see Fig. 5.34), evaluation of the split/merge propagation, follow the spatial structure of a patch over time (see Fig. 5.38 and Fig. 5.39).

### 5.6.6 Patch lineage graphs

Tuning the thresholds of this approach allows to individually put an emphasis on either the spatial coherence and/or the temporal coherence of features. Figure 5.34 shows some interesting combinations determined through empirical experimentation, where the complex structure of biomovie D3 with high variation across the three channels (top row: a–c) of red, green and blue. This variation is captured in three alternative patch lineages annotated atop binary images as illustrated from Fig. 5.34(d) to Fig. 5.34(f). It also shows two examples from the sensitivity analysis (see Fig. 5.34(g) and Fig. 5.34(h)) benchmarks where a single channel is investigated while the others are ignored. For example, in the case of the red channel: $\mathbf{v}_{t,p} = (r_{t,p})$, near the image size for geometric distance, and near the maximum of 255 for the green and blue color channels.

Fig. 5.38 and Fig. 5.39 provide further illustration of the implications of complex multi-channel and spatial structure of original biomovies. They depict patch assignments before and after the split/merge phase of the computation for biomovie D3. Comparing the patch structure to the fluorescence pattern in the RGB images, Figure 5.38 and Figure 5.39 demonstrate coherent spatial and temporal assignments. This can be seen by looking at location and color of the patches in (i) compared to the spatial distribution and variation of the fluorescence signal across the image space in (c), which ranges from low, to moderate, to high fluorescence signal as seen in the center and across the colony. Temporal coherence alludes to the color consistency of the patches throughout time. It is present in both D3 and D4. Moreover, the results of biomovie D4 in Figure 5.39 depict not only a color consistency but also a spatially structured organization of the patches in (bottom row: g–i).

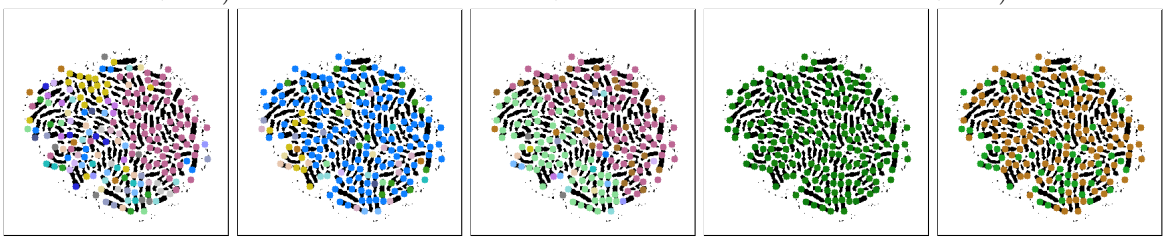
I also carefully validated the algorithm on the simulated biomovie DS5, designed to allow the correct patch structure to be verifiable by the naked eye. Figure 5.35 shows a sequence highlighting the behavior of the split/merge propagations of the algorithm for this biomovie.



**(a)** $I_{33}$: red channel (exposure +60%)

**(b)** Green channel (exposure +90%

**(c)** Blue channel (exposure +90%)



**(d)** d=30, r=15, g=50, b=15

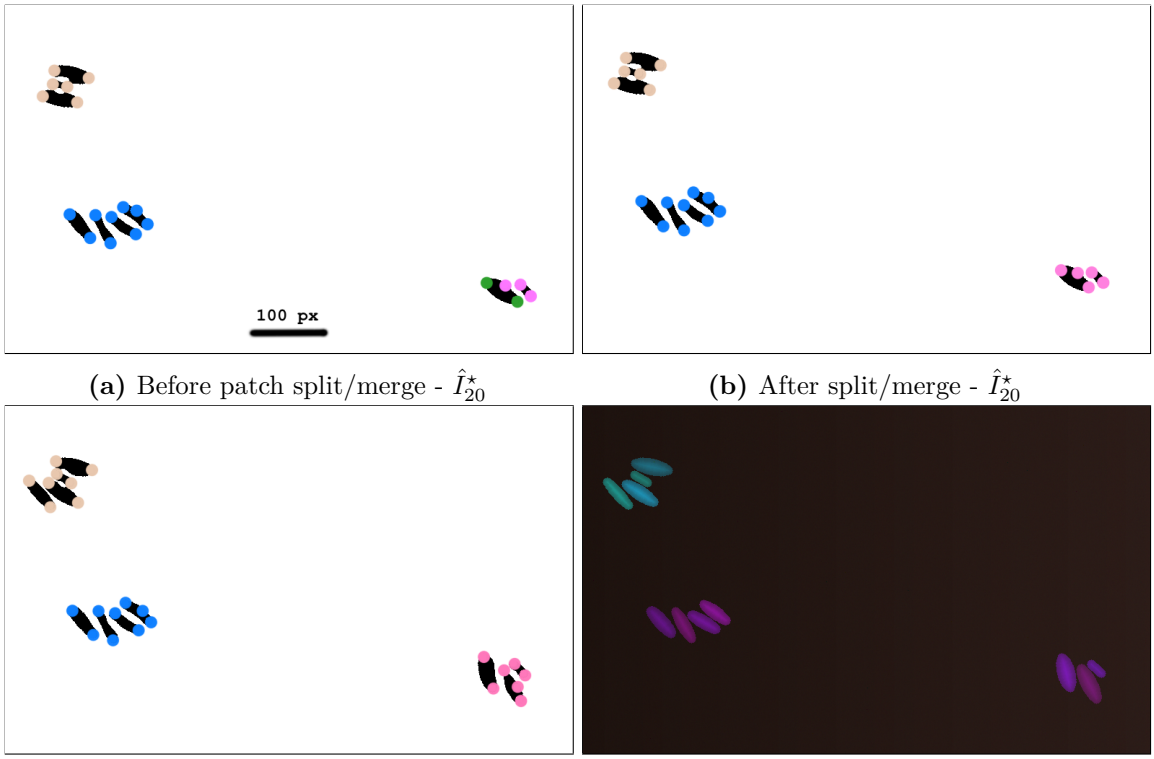
**(e)** d=60, r=30, g=100, b=30

**(f)** d=100, r=20, g=50, b=50

**(g)** g=80

**(h)** r=50

**Figure 5.34:** Example of parameter tuning to emphasize different channels, for time point 33 of biomovie D3. The binary images in the bottom row are annotated with 9-px dots showing particle locations, colored according to their patch IDs. The particle analysis thresholds in the previous computational step were set to 9 px particle diameter, a 5 px distance and 10 frame window for particle linking, and a 3 frame window for time filtering. (a–c) Separate views of red, green, and blue channels show the high structural variation between each channel. (d–f) Three different combinations of settings yield patch structures that capture different combinations of channel features, with thresholds for geometrical distance ($t_d$ denoted as d), and channel specific differences in red, green and blue ($t_r$, $t_g$, $t_b$ denoted as r, g, b, respectively). (g–h) Two examples of sensitivity analysis for individual channel thresholds, where the other channels are ignored by setting thresholds to very high values (geometric distance values near the total image size and color values near the maximum of 255). (g) The threshold $t_g$= 80 for green depicts a homogenous and constant signal across that channel, yielding a single main patch. (h) The threshold $t_r$= 50 for red emphasizes the binary nature of that signal, yielding two major patches. In both (g) and (h), the observed patches are exempt of spatial contiguity due to excluding the spatial dimension (a high value larger than the colony span).



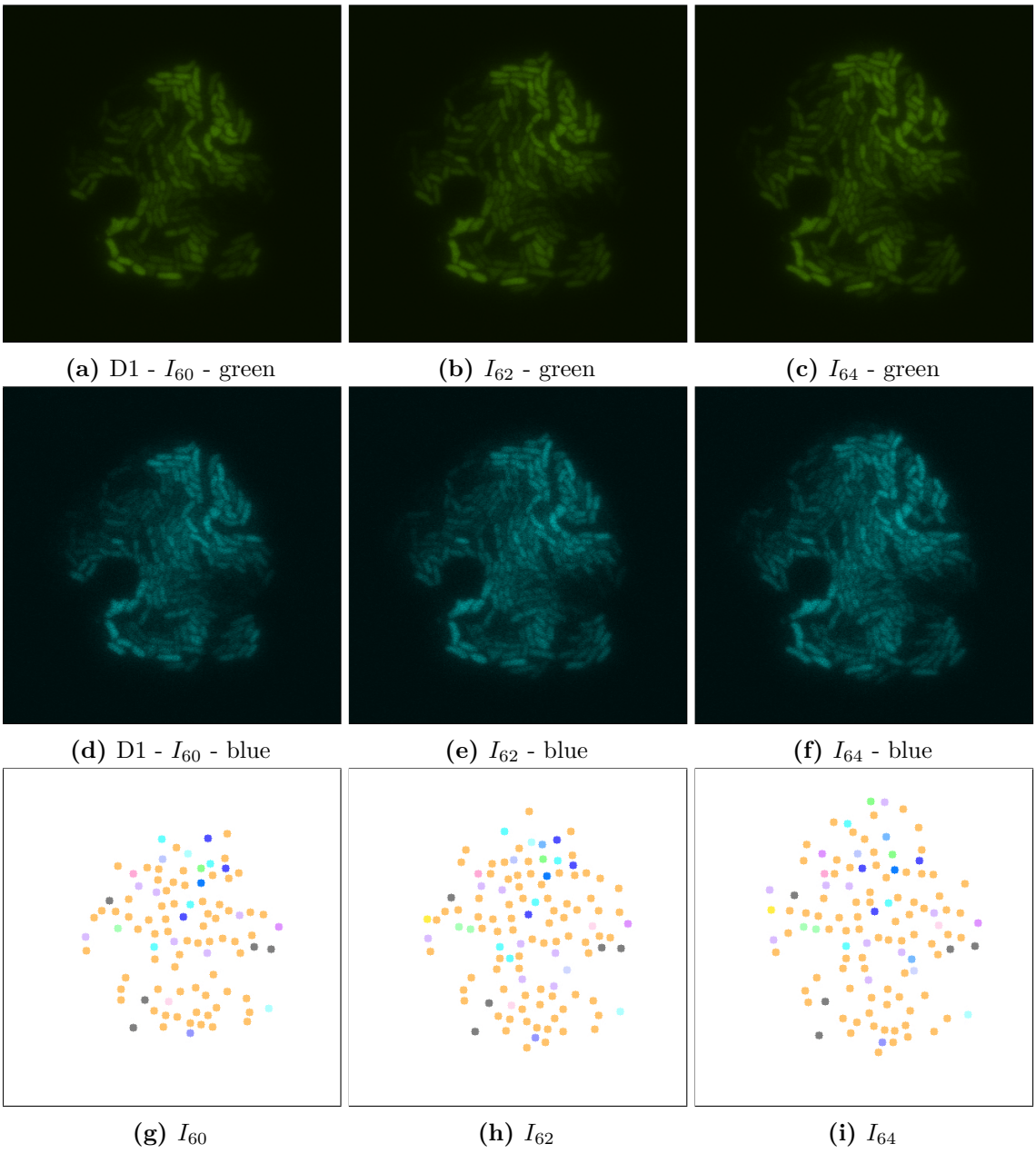

**(a)** Before patch split/merge - $\hat{I}^{\star}_{20}$

**(b)** After split/merge - $\hat{I}^{\star}_{20}$

**(c)** After split/merge - $\hat{I}^{\star}_{23}$

**(d)** RGB $I_{23}$

**Figure 5.35:** Sequence illustrating the split/merge propagations with simulated biomovie DS5, designed to allow patches to be verifiable by the naked eye from the RGB image. Images cropped to a 787x482 px subset. Binary images are marked with colored circles, 16px wide. The color encodes the patch ID. The geometric distance threshold $t_d$ for patch construction is set stringently to 100 px. (a) At time 20, before split/merge computation, showing four current patches. The bottom right quadrant has two neighboring cells with differently colored particles showing current assignments to different patches. (b) After split/merge computation, the particles are indeed the same color, showing that the patches have been merged as the patches are within the threshold distance to each other and have similar fluorescence. (c) At time 23, both the top left patch and the bottom right patch have new cells, and after the split/merge procedure is run for this time point they have correctly been assigned to the correct patch. (d) RGB image at time 23.

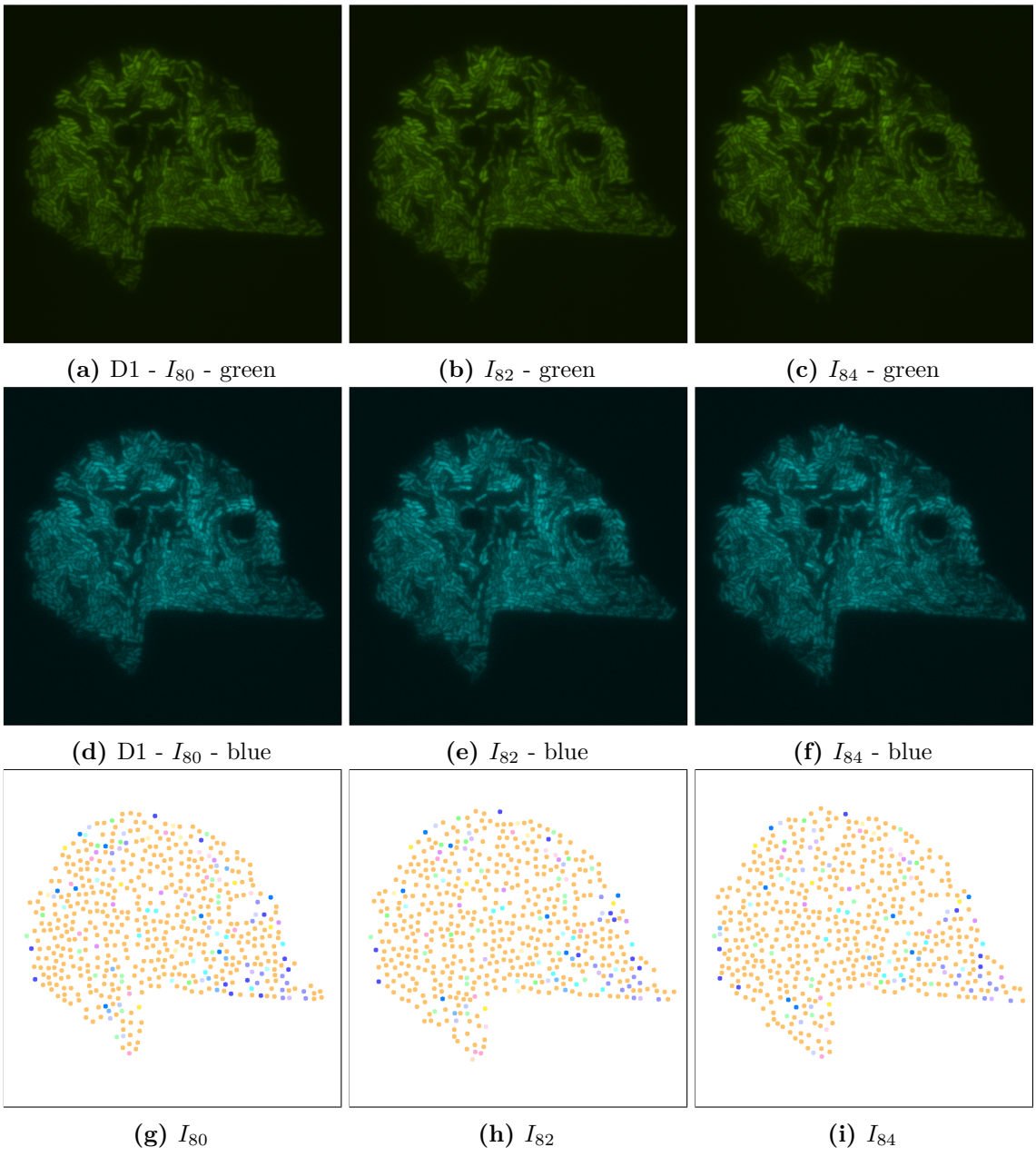


**(a)** D1 - $I_{60}$ - green **(b)** $I_{62}$ - green **(c)** $I_{64}$ - green

**(d)** D1 - $I_{60}$ - blue **(e)** $I_{62}$ - blue **(f)** $I_{64}$ - blue

**(g)** $I_{60}$ **(h)** $I_{62}$ **(i)** $I_{64}$

**Figure 5.36:** Biomovie D1 with RGB channels of image points 60, 62 and 64, and their corresponding patch structure, respectively. Enhanced exposures for green: 80% and blue: 80%. The *S. meliloti* bacterial cells are bio-engineered to fluoresce in a particular way, where each channel encodes a certain trait or behavior. The green (a–c) and blue channels (d–f) show certain behavior in response to changes of conditions; here the bacterial cells are of wild type and exposed to an environmental change, influencing the bacterial growth. The red channel is omitted due to its homogenous fluorescence and is reported in Figure 5.40. The patch structure is found using the following thresholds: geometric distance 100 px, and specific channel differences of red: 100, green: 50 and blue: 50. Main images show 7-px dots at computed particle locations. (g–i) The split/merge computation has been run, and particles are colored by their patch ID.

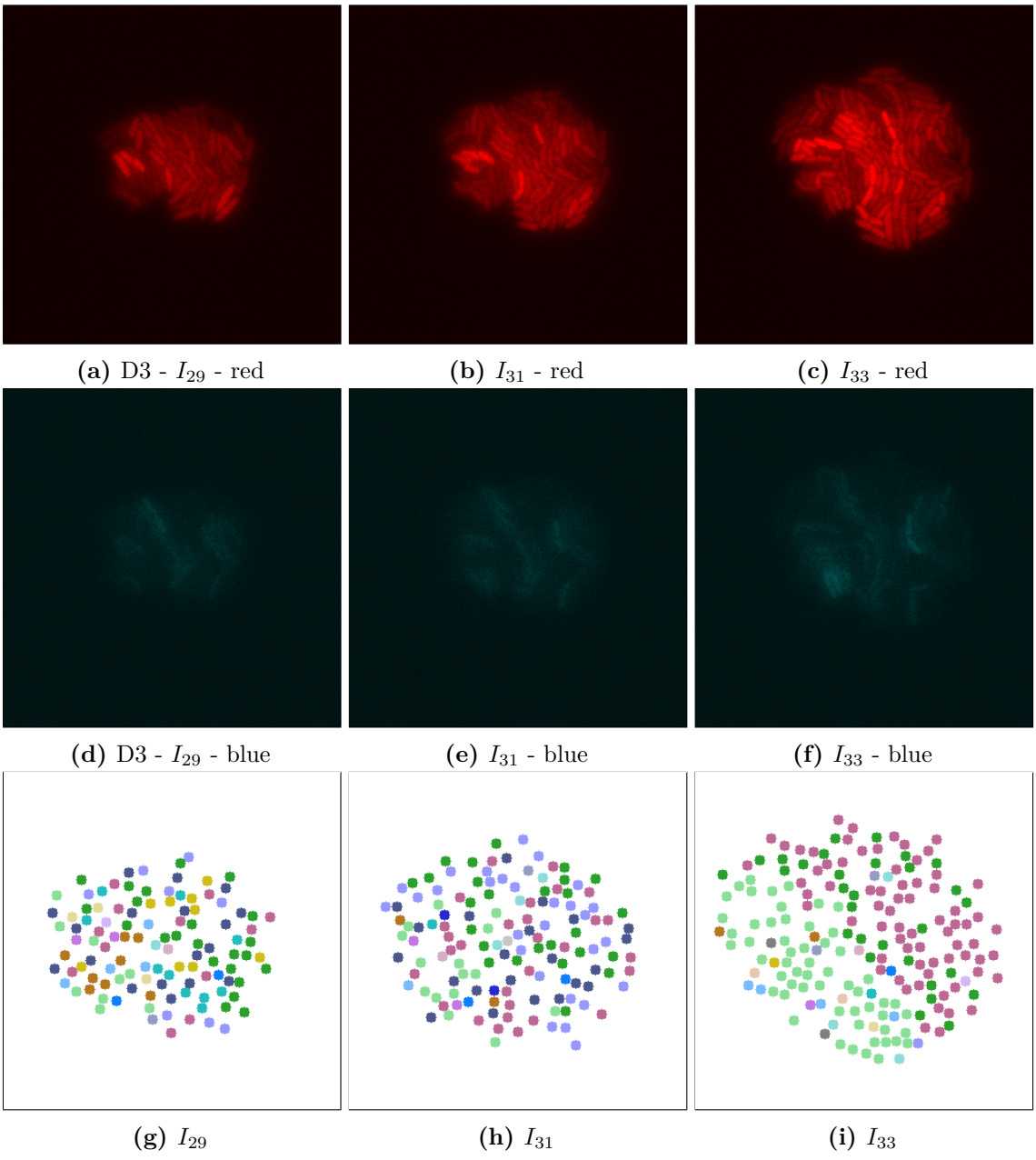


**(a)** D1 - $I_{80}$ - green **(b)** $I_{82}$ - green **(c)** $I_{84}$ - green

**(d)** D1 - $I_{80}$ - blue **(e)** $I_{82}$ - blue **(f)** $I_{84}$ - blue

**(g)** $I_{80}$ **(h)** $I_{82}$ **(i)** $I_{84}$

**Figure 5.37:** Biomovie D2 with RGB channels of image points 80, 82 and 84, and their corresponding patch structure, respectively. Enhanced exposures for green: 80% and blue: 80%. The *S. meliloti* bacterial cells are bio-engineered to fluoresce in a particular way, where each channel encodes a certain trait or behavior. The green (a–c), and blue channels (d–f) show certain behavior in response to changes of conditions; here the bacterial cells are of wild type and exposed to an environmental change, influencing the bacterial growth. The red channel is omitted due to its homogenous fluorescence and is reported in Figure 5.40. The patch structure is found using the following thresholds: geometric distance 100 px and specific channel differences of red: 100, green: 50, and blue: 50. Main images show 7-px dots at computed particle locations. (g–i) The split/merge computation has been run, and particles are colored by their patch ID.

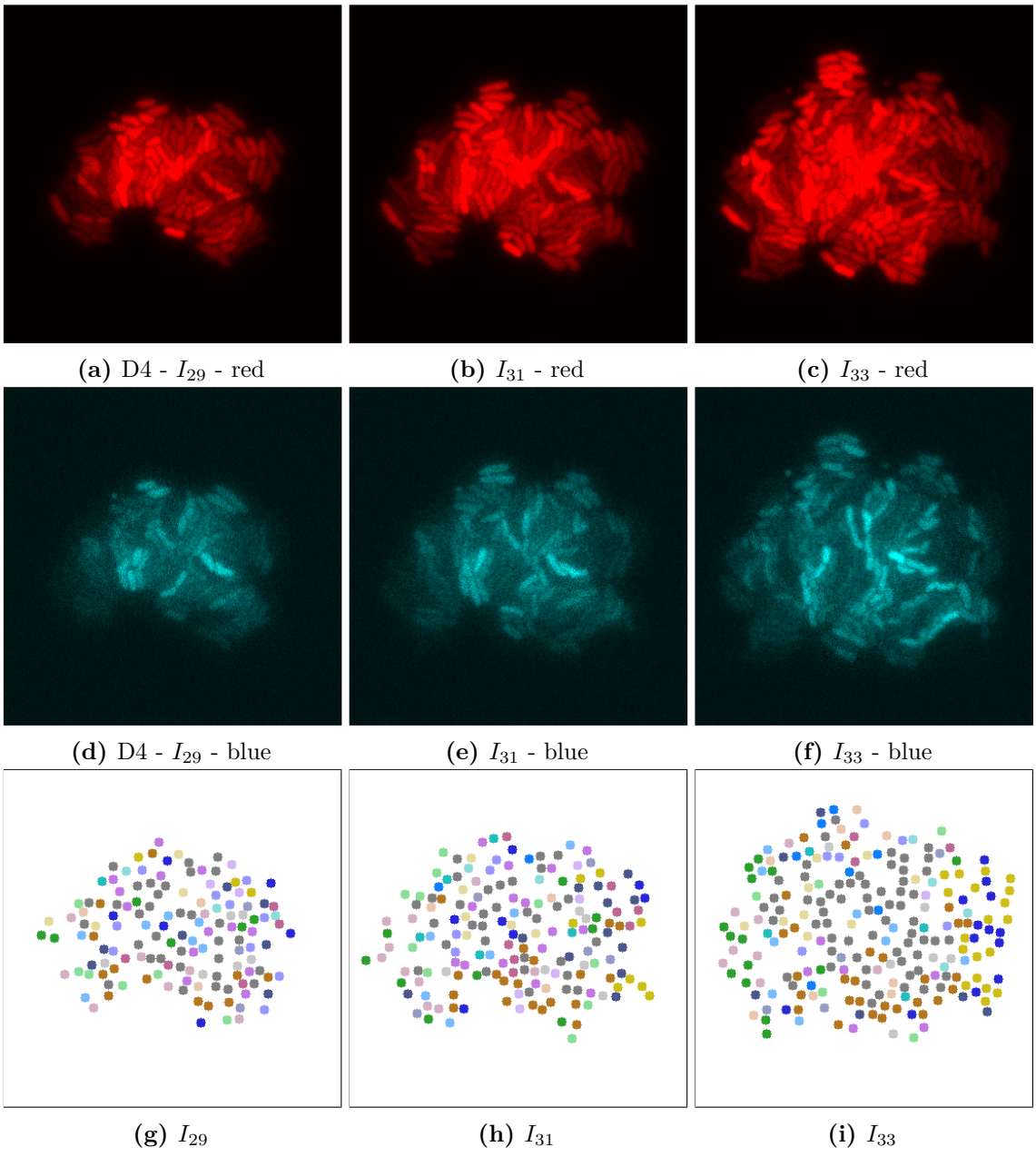


**(a)** D3 - $I_{29}$ - red **(b)** $I_{31}$ - red **(c)** $I_{33}$ - red

**(d)** D3 - $I_{29}$ - blue **(e)** $I_{31}$ - blue **(f)** $I_{33}$ - blue

**(g)** $I_{29}$ **(h)** $I_{31}$ **(i)** $I_{33}$

**Figure 5.38:** Biomovie D3 with RGB channels of image points 29, 31 and 33, and their corresponding patch structure, respectively. Enhanced exposures for red: 60% and blue: 90%. The *S. meliloti* bacterial cells are bio-engineered to fluoresce in a particular way, where each channel encodes a certain trait or behavior. The red (a–c) and blue channels (d–f) show certain behavior in response to changes of conditions; here the bacterial cells are of wild type and exposed to high concentrations of phosphate, influencing bacterial communication. The green channel is omitted due to its homogenous fluorescence and is reported in Figure 5.41. The patch structure is found using the following thresholds: geometric distance 100 px and specific channel differences of red: 20, green: 50, and blue: 50. Main images show 7-px dots at computed particle locations. (g–i) The split/merge computation has been run and particles are colored by their patch ID.



**(a)** D4 - $I_{29}$ - red **(b)** $I_{31}$ - red **(c)** $I_{33}$ - red

**(d)** D4 - $I_{29}$ - blue **(e)** $I_{31}$ - blue **(f)** $I_{33}$ - blue

**(g)** $I_{29}$ **(h)** $I_{31}$ **(i)** $I_{33}$

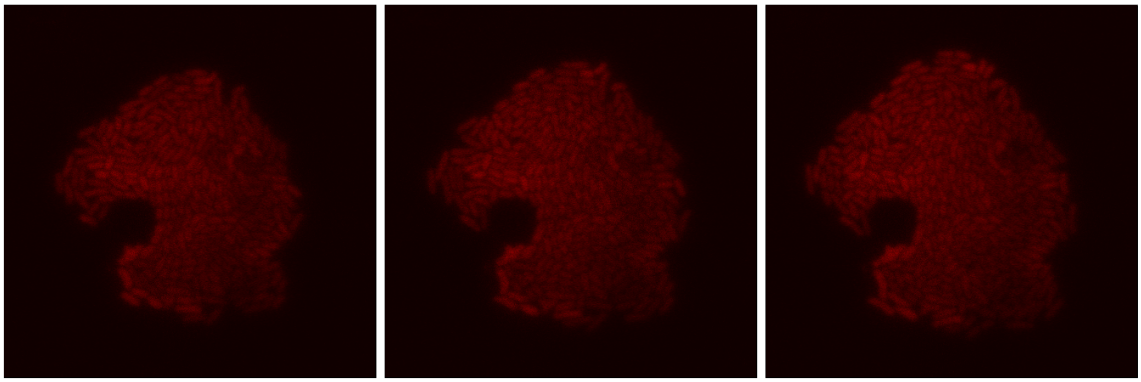
**Figure 5.39:** Biomovie D4 with RGB channels of image points 29, 31 and 33, and their corresponding patch structure, respectively. Enhanced exposures for the blue channel: 90%. As seen in Fig. 5.38, the biomovie showcases bio-engineered *S. meliloti* bacterial cells fluorescing in a particular way: The red (a–c) and blue channels (d–f) show certain behavior in response to changes of conditions; here the bacterial cells are of wild type and exposed to high concentrations of phosphate, influencing bacterial communication. The green channel is omitted due to its homogenous fluorescence and is reported in Figure 5.41. The patch structure is found using the following thresholds: geometric distance 100 px, and specific channel differences of red: 20, green: 50, and blue: 50. Main images show 7-px dots at computed particle locations. (g–i) The split/merge computation has been run and particles are colored by their patch ID.

**Computational performance**: The benchmarks show that the time required to create patch trajectories and patch lineage graphs primarily varies according to the user-settable thresholds for geometric and color channel distances that define patch boundaries. Using user thresholds that favor aggregation into a smaller number of patches yields faster computation, whereas tuning these thresholds to create a fine-grained structure of many patches increases the time spent computing splits and merges. Satisfactory results can be achieved with 2 min computation time.

**Summary**: Although a cell lineage is clearly a tree rooted from an ancestor cell that divides into its descendants as the colony grows, a patch lineage is in fact a directed acyclic graph (DAG). Smaller patches of similar fluorescence that are spatially separated in an earlier frame can end up merging together into a single larger patch in a later frame. This occurs as the cells continue to divide and respond to their environment. Biomovies with large colonies may contain multiple patch lineages that arise from multiple ancestor cells.

I demonstrate the success of CYCASP for colony-scale extraction of lineages of over 300 cells with automatic methods for the first time. The patch lineage construction algorithm aggregates and simplifies the spatial-temporal changes that take place within a biomovie into a unified data structure, with a small number of parameters that can be tuned to control the level of detail represented. The multi-propagation algorithm runs both forward and backwards in time. It takes advantage of knowledge about the last time point of biomovie to reap more profound benefits from temporal coherence than previously proposed methods.

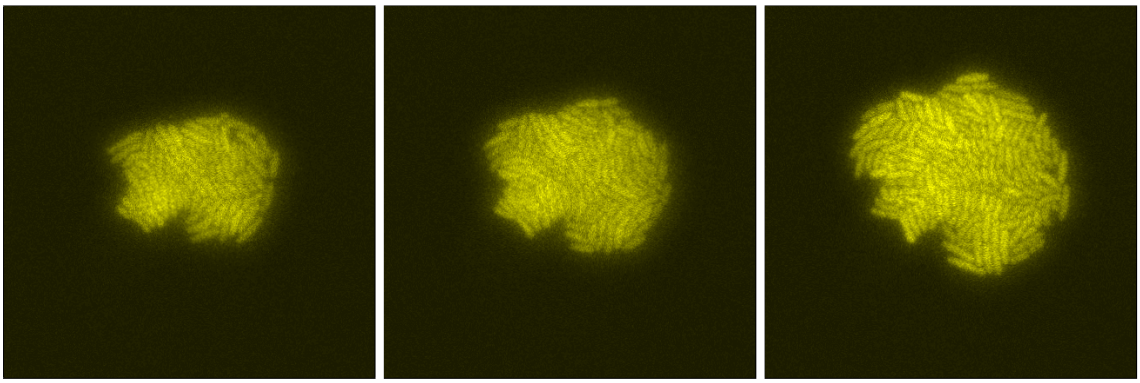
**Biological interpretation**: The simulated biomovies used in this paper are designed as minimal working examples that serve as understandable examples to both test and illustrate the CYCASP algorithm. They contain objects that mimic cell morphology and to a certain extent also mimic cell behavior. The original data sets or biomovies showcase the response of bacterial colonies to experimental disruptions compared to normal development, for both the heterogeneity experiments (D1, D2) and the communication experiments that are focused on quorum sensing (D3, D4), respectively. In the heterogeneity experiments, CYCASP was successfully applied to D1 and D2 which is also reflected by their analysis. Both resulting colonies exhibit distinct and non uniform fluorescence signals resulting in an even greater multitude of patches. As seen in Figure 5.36 and Figure 5.37, I investigate particles that stand out when provided user thresholds that favor a smaller number of patches. The different time points display the patch lineage graph on a frame-by-frame basis and these time points are selected based on colony growth events: For D1, shortly before another colony invaded



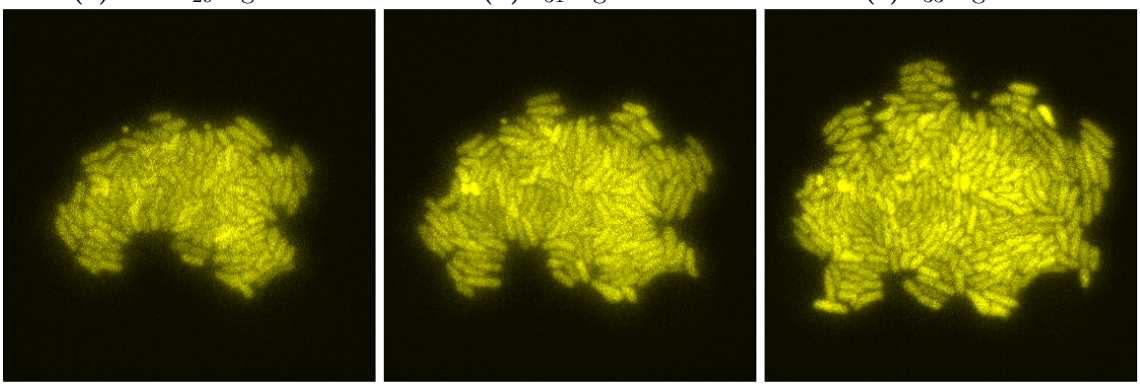
**(a)** D1 - $I_{60}$ - red **(b)** $I_{62}$ - red **(c)** $I_{64}$ - red

**(d)** D2 - $I_{80}$ - red **(e)** $I_{82}$ - red **(f)** $I_{84}$ - red

**Figure 5.40:** The red fluorescence channel as control in biomovies D1 and D2. The red channel shows a homogeneous fluorescence for cells that are alive throughout the colony growth using the mCherry fluorophore. Enhanced exposures for both D1 and D2: 90%.



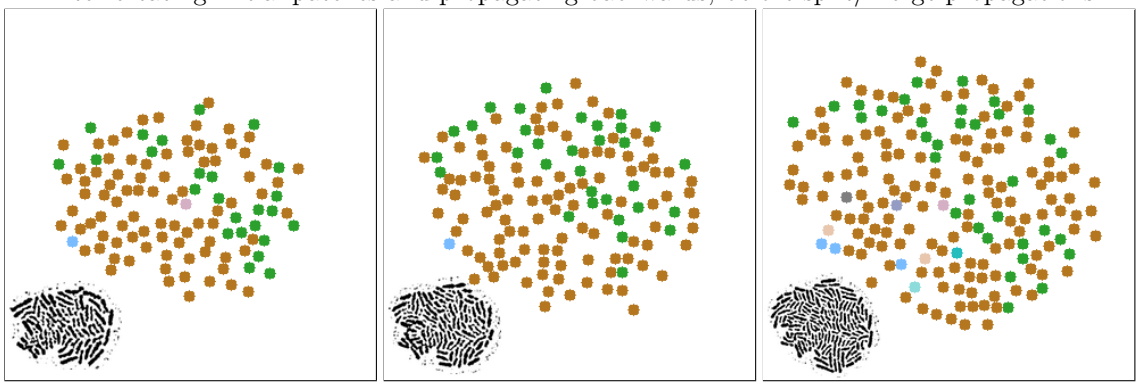
**(a)** D3 - $I_{29}$ - green **(b)** $I_{31}$ - green **(c)** $I_{33}$ - green



**(d)** D4 - $I_{29}$ - green **(e)** $I_{31}$ - green **(f)** $I_{33}$ - green

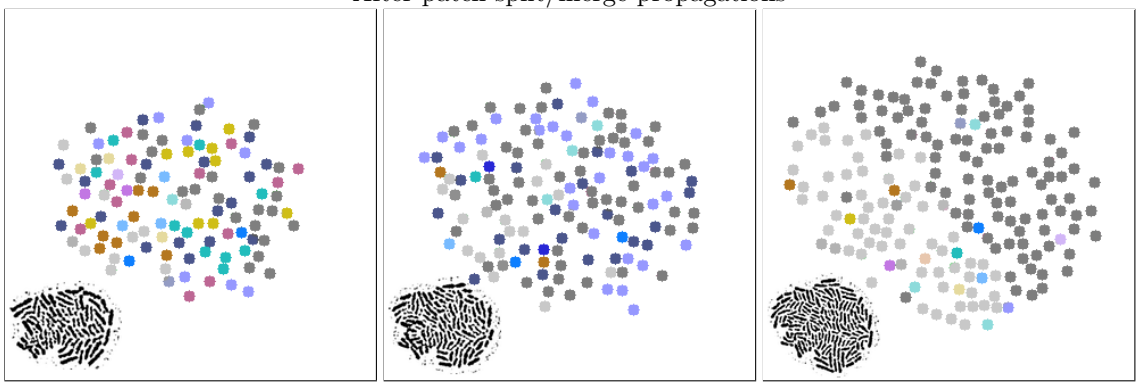
**Figure 5.41:** The green fluorescence channel as control in biomovies D3 and D4. The green channel shows a homogeneous fluorescence for cells that are alive throughout the colony growth. Enhanced exposures for both D3 and D4: 90%.

After creating initial patches and propagating backwards, before split/merge propagations



**(a)** D3 - $I_{29}$ **(b)** $I_{31}$ **(c)** $I_{33}$

After patch split/merge propagations



**(d)** $I_{29}$ **(e)** $I_{31}$ **(f)** $I_{33}$

**Figure 5.42:** Patch structure before and after splits/merges occur in biomovie D3 at time 29 and 31, corresponding to the RGB images in Fig. 5.38. Thresholds are $t_d$= 100 px and specific channel differences: $t_r$= 20, $t_g$= 50, and $t_b$= 50. Main images show 7-px dots at computed particle locations; lower left corner shows original binary image. Top row shows three time points before the split/merge computation and bottom row shows those time points after the computation. (a–c) Before the split/merge computation, all three time points have highly similar patch assignment patterns where particles are colored by their currently assigned patch ID. The initial patch creation computation has been run at the final time point (frame 44 in this case) and patch IDs have been propagated backwards to previous time points. (d–f) The split/merge computation has been run, and only particles with changed patch assignments are colored by their patch ID; unchanged particles with assignments matching the top row are grey. (d) Many colored new patch assignments reflect the fact that the previously propagated patch information was not valid at most of the particle positions for time 29; splits and merges updated these assignments. (e) A moderate number of new patches reflect the difference in fluorescence patterns between the middle and right columns of the RGB images in Fig. 5.38, leading to updated assignments in the split/merge propagation. (f) Only a few singletons stand out with new assignment colors against the mostly grey points reflecting a globally unchanged patch pattern, showing that most trajectories visible at time 33 maintained correct assignments from the initial computation.

the field of view (see Fig. 2.1(c)). For D2, shortly before the colony mass reached the right edge of the imaged field of view (see Fig. 2.1(f)). In Figures 5.37(g–i), I chose to display the biologically relevant patches in contrast to the orange colored particle positions that structure the largest patch of this colony. Indeed, the biologically relevant patches are mostly positioned at the periphery, where bacterial cells are showing different signal characteristics, i.e. fluorescence, which corroborates with changes in the environment. At the core of the colony, I observe bacterial cells behaving similarly. On the contrary, I observe an heterogenous bacterial behavior either in contact with the microplate, or in few instances proximal to dead bacterial cells (i.e. in the vicinity of empty areas in the image, see Fig. 5.36(g–i)).

In the communication experiments, I could identify different subpopulations in similar conditions in both D3 and D4. High phosphate concentrations in the medium disrupts cell communication by repressing quorum sensing signaling, hence fluorescence signals. In this stressful condition, I found subpopulations that adapted to such levels by setting user thresholds to favor variation in the red channel. In biomovies D3 and D4, splitting and merging of patches visible in the resulting patch lineage DAG highlighted regions that showed changes in reporter gene activity indicating a switch in cell state. In Fig. 5.38, the patch structure depicts a clear delineation of three main patches at time point 33. This suggests that the colony grew into coherent subpopulations, which may either be a result of a stochastic event or an adaptive event to changes in the medium.

To investigate biomovie D4 and find similarities and dissimilarities in colony growth, I used the same thresholds for the algorithm. Compared to Fig. 5.38, I observe more patches in Fig. 5.39, where subpopulations grew into different local regions of the colony. This suggests a more important disruption of bacterial colony growth, yet triggered other cells to enter the quorum sensing state. For both biomovies, I observed an homogenous activity of the mVenus gene reporter in the green channel where the yellow fluorescence is homogeneously distributed between the bacterial cells. This indicates that the older the colony is, the higher the quorum sensing signal is. And as seen in the blue channel (mCerulean), the heterogenous activity of the exopolysaccharide gene reporter is captured by the patch lineage results. Moreover, according to particle numbers at time point 33 and throughout the biomovie D4, the colony grew faster than expected. For D3 and D4, I found at time point 33: 253 vs. 356 particles and 5679 vs. 8207 particles, respectively. This suggests that the colony in D4 grew 1.4 times faster than in D3. By setting the same thresholds for both biomovies, favoring variation in the red channel, I was able to find that both colonies were similarly able to adapt to changes in the

medium. It is reflected by the spatial coherence, or the patch structure found in Fig. 5.38(i), and Fig. 5.39(i). Temporal coherence is demonstrated by the color consistency of the patches throughout time in Figure 5.38(g–i) and Figure 5.39(g–i). Furthermore, compared to D3, the patch lineage results of biomovie D4 depict a spatially structured organization of the patches over the different time points (g–i), yet showing a rather fragmented view of the colony suggesting the growth of a dozen of subpopulations occurred in the early stages of the colony growth. I hypothesize that these subpopulations result from stochastic events, which set apart this biomovie. By using the patch concept, I identify subpopulations, find dissimilarities between data sets, follow the diversity, and how quickly colonies grew in biomovies.

The patch lineage graph concept is biologically motivated: the automatically computed graphs are intended to help microbiologists understand how and when changes in cell state occur in microbial populations. The patches, namely contiguous regions that are bounded by similar fluorescence patterns, do indeed provide insight into bacterial cell colony development. Moreover, the particle abstraction that I proposed is successful in handling cell division and exponential bacterial growth. In general, multiple observations can be formulated: the particle positions spatially describe the colony growth, the patch lineage graph reflects changes in signal characteristics depending on the user thresholds and temporal coherence is respected.

## 5.7 Implementation

This framework is available for download at `http://github.com/ghattab/cycasp`. The space-time cube vis. of particles is available at `http://github.com/ghattab/seevis`. Both are implemented in Python under the MIT license and supported on UNIX-based operating systems.

## 5.8 Discussion

Grasping a mental image of a highly dense and ever-growing bacterial population proves to be quite challenging. The CYCASP framework handles the five-fold challenge of high cell count, high cell density, high cell shape diversity, strong noise, and high resolution by using the abstractions of particles, patches, and the patch lineage DAG. The presented results are the first automatic solution for the problem of efficient comparative analysis of an arbitrary

number of biomovies. Since a full manual annotation of a biomovie can last one to two full working days, a computational approach evaluating particles and patches instead of single cells may provide at least a worthy and additional view of the data. It offers an alternative so the bottleneck in the analysis can be overcome. In a high-throughput environment, the particle based visualization (SEEVIS) demonstrates its capabilities by optimizing user time, and by providing a mental map to speed up the acquisition of the data space. In turn, this permits users to gain valuable insight and validate whether a dataset is worthy of further analysis. The proposed abstractions succeed in exploiting and qualitatively integrating spatial and temporal coherence without explicit segmentation at the cell level, and were demonstrated to be successful on both biological and simulated biomovies, respectively.

The methodology of SEEVIS presents the first color mappings adapted to cell colony development. These color mappings have strengths, but also weaknesses. Larger colonies with many data points did suffer greatly from visual occlusion, especially when using the nominal and progeny mappings. The employed categorical colors are being repeated multiple times. Although, I did not observe any proximity between identically colored trajectories; such a phenomenon could happen for even larger datasets. Moreover, the space-time cube requires an array of guidelines in the three-dimensional space in such a way that important data elements and data patterns can be quickly perceived[116]. Representing the particle point cloud using another rendering engine using the time color mapping and a depth shading function, enabled me to test the methodology in a different visual environment, respectively. Provided these guidelines, SEEVIS could be further improved. For example, a filtering approach to highlight a trajectory subset deemed relevant by the user.

CYCASP is the first attempt to study subpopulations at the level of patches that have similar behavior with common ancestry. The results show that patches and patch trajectories are an intuitive, flexible, and powerful concept. They reflect different cell behaviors for subpopulations that split off from each other at some times and merge together in others. CYCASP implements a modular and automatic patch lineage algorithm. It succeeded in constructing patch trajectories across all of the time points of a biomovie. With appropriate parameter settings, these trajectories can be assembled into a patch lineage DAG that captures the high-level behavior of interest. I argue that the goal of understanding this high-level behavior was the underlying motivation behind the previous manual analysis that reconstructed very low-level views of the ancestry relationships between individual cells. The innovation with CYCASP is to support this level of analysis both directly and automatically.

I have shown the effectiveness of the particle abstraction in handling the complex biomovies targeted in this chapter, where there is a difficult combination of high cell density, exponential growth, and a low temporal resolution.

As most algorithms and state-of-the art methods, it is possible to improve the computational efficiency of this framework. I discuss possible improvements for both abstractions: the particle and the patch. At the particle level, particle linking into particle trajectories could be more performant by employing predictive methods to relieve the tuning of the (dis-)appearing particles parameter. Moreover, since the RGB channels encode fluorescence signals from a molecular standpoint, they do carry the relevant information (i.e. living cells). Hence, I argue that extracting particle positions from the RGB space makes more sense and is less prone to image noise. In the current implementation, the RGB value of a particle is a point-based value, as opposed to a median- or mean-fluorescence value. Additionally, it is possible to computationally estimate the particle diameter so the user has less parameters to worry about. At the patch level, multiple steps could be improved concerning computational speed and the computation of certain metrics. Namely, patch finding could be sped up by 'parallelizing' the combinatorial aspect of the all-pairs testing. Moreover, it is possible to argue that the Delaunay triangulation is slow compared to other algorithms, such as the gift wrapping or divide and conquer algorithms. Withal, computing further descriptors for each patch is at hand, for example establishing if particles of a patch are strongly connected (i.e. connectedness of a subgraph), or computing the KDTree for nearest-neighbor point queries, or other distance computations for various metrics using the `Qhull` library. I argue that such implementations exceed the necessary requirement for the method's functionality, although it is possible to extend the implementation, and tailor it to the needs of the user. In the patch finding step, the alpha-shapes or concave hulls computation returns the boundary of each patch. It is possible to only use the convex hull, since the algorithm relies on the convex polygon of a patch for the computation of its minimum-area rectangle. I argue the concave hull computation is an extra step, where in future implementations it would be beneficial to visually denote particular patches which have an intrinsically non-convex shape.

The computational efficiency of this framework hinges on processing far fewer elements in far more depth at each stage. A number of $m$ particles are extracted very efficiently using only spatial coherence, then $k$ particle trajectories are constructed to exploit temporal coherence. Finally, $j$ patch trajectories are computed using a multi-propagation algorithm with bidirectional traversal or propagations of a biomovie. The three quantities generally

obey $m >> k >> j$, typically multiple particles are detected within each cell, so the particle count $m$ is larger than the cell count $c$ by at least a factor of 2. However, identifying particles is much more efficient than detecting single cells. The benchmark results show that CYCASP can automatically extract the entire forest of patch lineages from biomovies in under 5 minutes for biological data sets of over a hundred frames and approximately three hundred cells, in contrast to the two full working days of manual analysis previously required of my collaborators. I also discuss the parameter settings required to correctly track particles across space and time, and aggregate them into patches.

## 5.9 Outlook

In the case of other experimental setups (e.g. petri dish) and different resolutions, CYCASP is generalizable yet would require fine parameter tuning so as to handle for instance overlapping cells (i.e. $W_{\max}$ for disappearing particles). Moreover, this algorithm could be employed to empower specialists in other imaging domains. These imaging domains range from crowd analysis, through astrophysics, to stem cell research, to cancer imaging.

### 5.9.1 Astrophysics

Almost a decade after the Hubble telescope took the Deep Field image, astrophysicists went ahead, and took another long exposure over a period of four months obtaining a long exposure. In the resulting images, they observed 10 000 galaxies. Half of these galaxies have since been analyzed in what is known the XDF: e(X)treme (D)eep (F)ield images. These images are for the first time composed of the full range of ultraviolet, to near-infrared light[117]. By combining over ten years of photographs, the XDF shows galaxies so distant that they are only one ten-billionth the brightness that the human eye can perceive. Since space and time are inextricably linked, the Deep Field images are like 'cosmic' time machines to the ancient universe. This enables astrophysicists to observe galaxies that existed over thirteen billion years ago. This means when we are looking at such long exposures, we are in fact looking at the universe as it was less than a billion years, after the Big Bang. Such images allow scientists to research galaxies in their infancy. The XDF have also shown that the universe is homogenous. That is to say, images taken at different spots in the sky look similar. Provided this method, it would be possible to characterize the behavior of galaxies

by coupling space, time, and the different channels (from ultraviolet to infrared light). Once galaxies are identified, each galaxy position $(x, y)_t$ is mapped to a feature representation: $(x, y)_t \mapsto \mathbf{v}_t[0, 1]^D$ that encodes the signal characteristics of interest in a $D$-dimensional space. For example, shedding light into the clumping behavior of galaxies, due to the presence of dark matter halos. Because dark matter, like galaxies, has gravity and will pull galaxies towards it, causing them to clump[118,119]. Achieving such a task opens up the door to further analyze and understand such events.

### 5.9.2 Stem cell research

In stem cell research, researchers study the reprogramming of somatic cells to induced pluripotency. Such studies help analyze and understand pluripotent cells, that are able to differentiate into different cell types. Live-cell time-lapse imaging of somatic cells undergoing that process raises interesting questions about its mechanism. The main bottleneck is the very low efficiency of such a process. In many experiments, the interest shifts to subpopulations where researchers are unable to distinguish between an early stochastic event versus the existence of a predetermined subset of cells that are in some way primed for cellular reprogramming[44]. This shift is motivated by the realization of failing to trace the origin of a subset of a colony or particular cells that detach from other colonies. Such biological questions could be tackled by the methodology presented herein.

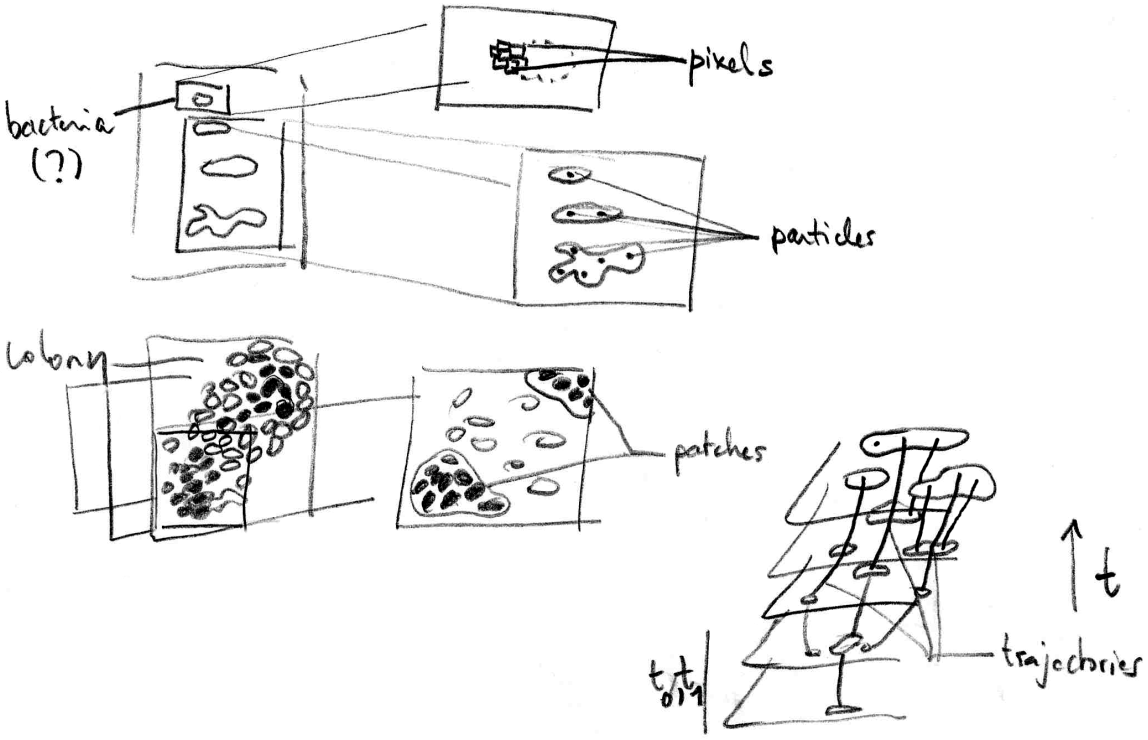
### 5.9.3 Cancer imaging

Metastasis is the spread of a cancer, or another disease from one organ, or part of the body to another without being directly connected with it. The metastases occur for example using the blood circulation system. A tumor cell or an aggregate of tumor cells circulate in the blood stream to reach a target tissue or organ, hence leading to a spread of the cancer in another area of the organism. There exists assays to detect such circulating tumor cells in human blood using imaging systems[120,121,122]. Adapting this framework to detecting and following cell aggregates in blood samples is manageable.

Another motivation resides in cancer diagnosis using imaging, also referred to as tumor detection[123]. In the early stages of disease and at the cellular level, no two cancers are identical. This leads to the impossibility of devising a universal strategy to differentiate tumors from

normal tissues. Conversely, at the macroscopic level of biological organization (e.g. tissue), patterns emerge within the local environment and for different cancers (e.g. gastric carcinoma[124]). Such observations mean that it is possible to use this framework to investigate where and when the patterns emerged, or how they changed.

In this chapter, I motivated and defined two data abstractions: the particle and the patch. Then, I described the methodology behind SEEVIS and CYCASP. The former provided rapid qualitative insight into the growth of bacterial colonies. The latter succeeded where previous automatic methods failed because I avoid the bottleneck of needing to achieve a segmentation for each individual single cell. Moreover, it creates a patch lineage DAG that employs both the particle and the patch. CYCASP supports the understanding of cell-to-subpopulation, subpopulation-to-subpopulation interactions, and the reasoning about the behavior of entire cell colonies at the biologically relevant level of subpopulations with similar behavior, rather than needing to infer it from the overwhelmingly complex branching structure of individual cell lineages. This method paves the way towards a more manageable way of analyzing biomovies starring nanoscale organisms.


pixels
bacteria
(?)
particles
colony
patches
t
trajectories

# 6

# Conclusion

In this thesis, I investigated questions that span from the time-lapse image data that depict the growth of isogenic bacterial populations, to the scales of cell colony development, to an example of task-oriented visualization, to preprocessing steps, to a novel lineage reconstruction. Based on a given experimental setup (i.e. microfluidics) and the resulting image data (i.e. biomovies), I developed integrative approaches that include, but are not limited to, state of the art computer vision, image registration, and data abstractions for cell colony growth. The main question that I formulated and that motivated this thesis was: How do we reliably take into account the cellular context to follow cell-to-subpopulation, subpopulation-to-subpopulation events within a colony?

In chapter 1, I explored the intersection of domains at which this thesis resides, that is synthetic biology and bioimaging. I defined a range of data properties that arise from acquiring time-lapse image data, a set of five data properties: cell count, cell shape diversity, cell density, image noise, and resolution. Then, I presented their diversity across different species and the dynamics that influence their corresponding values.

In chapter 2, I introduced the particular biomovies that are at the core of this project, the high values of the aforementioned data properties and their incidence on their analysis. This defined a bottleneck that hinders the data analysis for both humans and computers. On

one hand, humans observers were not able to achieve high inter-/intra-observer agreement, even with computational support. On the other hand, none of the known computer based approaches worked automatically. These approaches looked into every quantitative variable, pertaining to every single cell, without a preliminary overview of the colony or population. Their analysis focused and relied on single cells. Moreover, no reports of studies that have tackled isogenic bacterial populations with high values for all of the aforementioned five properties have been found. I argued that tackling the main question cannot be accomplished for such colonies, neither quickly nor efficiently by using only the general paradigm as is (i.e. segmentation and tracking of individual cells, and cell lineage construction).

In chapter 3, I presented the domain field of visualization and explained the nested model of visualization. I explored related spatiotemporal visualizations ranging from aggregate plots, to space-time cubes in cell imaging visualizations, to spatiotemporal visualizations in functional magnetic resonance. Next, I integrated the methodology and employed it for a task-oriented visualization applied in biochemistry and infographics (i.e. the visualization of amino acids molecules and their properties). This example is described in appendix B and resulted in a set of cards that made the molecular structure accessible to the untrained eye. Moreover, provided the nested model of visualization and design, I was presented with the means to devise data abstractions to address the task of identifying subpopulations in the biomovies.

Nevertheless, all the different biomovies suffered from spatial shift and it was necessary to adjust each frame of each biomovie by using the first frame as reference for the alignment. This task of image registration is detailed in chapter 4, I addressed this problem by designing an adaptive and performant computational approach. This approach led me to confidently align the biomovies depicting the growth of *S. meliloti*, so to further advance in my analysis. Singularly, to describe the context of colony growth and eventually depict cell-to-subpopulation, subpopulation-to-subpopulation interactions, I reconsidered the problem from a visualization stand point by following the methodology of the nested model. In chapter 5, I conceived data abstractions that are able to handle the high values for all five properties and help answer the formulated question.

I adapted the first abstraction, i.e. *the particle*, from the field of fluid mechanics. After preprocessing the averaged RGB channels and based on the resulting binary images at every time point of a biomovie, the particle permitted to identify foreground regions that contain signal characteristics (e.g. fluorescence, edge, etc). Moreover, the particle abstraction per-

mitted to bypass the problem of single cell segmentation, which was initially hindered by the high values of the different properties. Hence, I was able to extract the colony and its extent in the spatial domain. To include the temporal domain, I used particle linking to obtain particle trajectories. Next, I designed two types of color mappings adapted to colony growth, either a particle index- or time point-based color mapping. The particle index based type comprised a nominal and a progeny mapping. The former was used to differentiate neighboring particle trajectories (using the `Tableau10` categorical palette). The latter employed the nominal mapping and consisted of highlighting only particle trajectories that were visible at the last time point of a biomovie. The time point based type consisted of a time mapping, where subsets of particle positions were colored in respect of the time index with monotonically increasing luminance ranging from dark purple, to light yellow, from the first, to the last time point $t_{\max}$, respectively. This mapping showcased the extent of the colony in time. To compare multiple biomovies qualitatively, it is possible to set a fixed value for $t_{\max}$. These color mappings are implemented and are available under SEEVIS. This methodology offered a quick render of the colony to help users conceive a mental map where both time and space are preserved. This entailed using a visualization from the aforementioned six classes, the space-time cube and representing the feature space in which bacterial cells grew.

Next, I moved away from the abstraction of a cell, part of a cell, or an aggregate of cells towards subpopulations. I defined *the patch* abstraction as an aggregation of spatially contiguous particle trajectories that feature similar signal characteristics (i.e. similar fluorescence patterns). Based on user thresholds and a modular algorithm, a DAG of each colony or biomovie was constructed. Its construction relied on first finding the patches at $t_{\max}$, then computing three propagations: 1. patch trajectory propagation, relied on found information at $t_{\max}$, then 2. patch trajectory splitting, or inter-patch evaluation, where the algorithm looked at spatial inconsistencies within a patch trajectory throughout the spatiotemporal domain, and 3. patch trajectory merging, or intra-patch evaluation, where the algorithm looked for possible merges between patches and throughout time according to a merge window threshold. By design, a patch lineage DAG has a dramatically simpler structure than a cell lineage tree because it has far fewer branches. The frame-by-frame visualization of the patch lineage and its structure supported the reasoning about the behavior of entire cell colonies at the biologically relevant level of subpopulations with similar behavior, rather than needing to infer it from the overwhelmingly complex branching structure of individual cell lineages. By defining *the particle*, *the patch*, and a modular algorithm that resorts to using

both, I presented a novel, elegant, and efficient solution that favors coherence over single cell delineation to locate and follow subpopulations. The methodology pioneered herein relied on the spatiotemporal qualities of the conceived data abstractions. Moreover, this methodology illustrated the connection between data visualization methodology and cell colony growth in microplates. As a means to an end, my methodology answered the aforementioned main question by employing CYCASP's modular algorithm and explicitly presenting the argument of spatiotemporal coherence. Last but not least, while conceiving this framework, I came across various possible alternatives and applications, that I qualify as perspectives. I detail a couple below.

For the specific task of cell segmentation, there exists many ways to tackle the problem. Even though this thesis described a feature extraction approach, i.e. the particle, I also conducted a short study using deep learning for this task (c.f. appendix C). Deep learning unites the process of feature extraction and classification or regression into one system so to be optimal to the task. Its usage was motivated by the fact that higher level features are derived from lower level features to form a hierarchical representation. As a means to an end, deep learning can be quite robust. For the task of cell segmentation, as seen in Raza et al., large cells are easily detected with a moderate cell density[125]. In my case, the high cell density of a fluorescing colony impedes on the task, making it inconclusive. In the following, I discuss few pointers.

For the purpose of discriminating bacterial cells from background and noise, the network's architecture is quite important (i.e. the number of input neurons and layers). Ideally and as seen in recent studies, deep learning could also learn to optimize the structure of the network[126]; yet there exists ways to potentially ameliorate the performance. Typically, researchers start with a network structure that has performed well on a similar task, then they test it against their current problem and make refinements to address whether the network's structure under- or over-fits the data. Such refinements are either possible using a parameter search approach or going back to the data for analysis. The former, i.e. parameter search, initiates a testing of different values for the filter size to find the best performing one. The latter relies on the specifics of the image data, where the researcher thinks about the filter size or the feature size that could be most discriminative in the image. I argue that in my case, the task of delineating single cells is difficult even for neural networks since the initial input data has high values for all of the five properties. This means that if the input data does not permit the neural networks to find enough low-level pixel features, the more

complex and higher-level features will not be as easily perceptible by the neural networks. In the domain knowledge of cell lineage analysis, the particle abstraction could either help bypass the correspondence problem or help solve it. Provided an experimental setup that produces data with high values for all data properties and the need to delineate every single cell; the particle could be used in a semi-manual approach. This would entail users verifying the when and the where two or more particles ought to merge. The over-segmentation of the spatial domain provided by the particle abstraction is ultimately yielding further data points to work with, in contrast to very few or none in the case of a cell segmentation based approach. Provided researchers use the CYCASP framework, they would be able to build patches intuitively, yet requiring to conduct a sensitivity analysis. Such a task can be time consuming. I could well imagine an automatic fine tuning of the user thresholds so as to either maximize or minimize the amount of patches. In light of the biology, neither scenarios are sufficient and conducting such an analysis is a better alternative.

Another perspective deals with a quick investigation of colony growth, where the methodology of the space-time cube from SEEVIS and the resulting patch lineage from CYCASP would be coupled. In general, the patch abstraction involved herein motivates visualization ventures so as to depict whole colony events. Assuming researchers are in a screening setting, a large number of experiments would be conducted with different cellular treatments (e.g. gene knockouts, antibiotics, etc). Being able to convey a quick and global overview of the colony growth is of paramount importance. This would entail analyzing the different conditions for each experiment to discover the potential links between the different growth patterns to such treatments.

Besides the state-of-the art approaches, the methodology herein presents a notable advancement to study cell-to-subpopulation and subpopulation-to-subpopulation interactions. An alternative to explore would be probabilistic approaches as for finding cell edges, for instance by defining a maximum-likelihood based objective. Such probabilistic solutions might be interesting, yet would fall short if not enough information is present across the different imaging channels.

All in all, the perspectives for integrating cell information from the images are driven by the availability of signal, or a good SNR. From a methodological point of view, developments such as the methods herein occur when confronted with important bottlenecks and pushes us to integrate as much information as possible, eventually leading to further the possible research avenues.

# A

# Synthetic data

Synthetic biomovies were created by employing: cell simulation, shape, texture, channels with noise, and artifacts. While the cell simulation software from Wiesmann et al. 2013 has been extended for biomovie simulation, the steps for simulation are similar to image simulation[65]. First, the cell shape is calculated. Second, the cell position on the image. In the third step, the cell texture is added. In the fourth and last step, imaging artifacts, and noise are added. For the synthetic biomovies, bacterial shapes are modeled as ellipses with varying length of semi-major and semi-minor axis. Bacterial cell positions are determined on a frame by frame basis by minimizing an energy function. For the first frame the first bacterium is placed in the image centre. After it has divided, the new bacterium is placed next to the bacterium of which it originates from. After all bacteria have been calculated in one frame, the bacteria are input to the following energy equation:

$$E^*(bacteria) = \sum_{o}^{N_o} \sum_{p \in o} I_{dist}(p) + k * \sum_{o_1 \neq o_2}^{N_o} \sum_{p_1 \in o1} \sum_{p_2 \in o2} \delta(p_1, p_2) \tag{A.1}$$

Where:

$N_o$ = number of bacteria
$I_{dist}$ = distance transformed mask of the bacterial cell shape
$\delta(p_1, p_2) = 1$ if the condition below is fulfilled.

If $p_1 == p_2$ pixels of bacterium $o_1$ and $o_2$, respectively. Else, $\delta(p_1, p_2) = 0$.

The first energy summand keeps bacteria sticking together in the image centre. The second energy summand prevents overlap between bacteria. The factor k weighs energy summand one against the second energy summand. The gradient descent method is applied to iteratively minimize the energy equation to find the positions of bacterial cells on the current frame. Their positions at the previous frame are the starting point for energy minimization at the current frame. Three channels are simulated with varying appearance modeling the properties of various real fluorophores. The bacterium's texture is calculated with the sigmoid function as written below.

$$f_{sigmoid} = \frac{I_i}{1 + e^{\kappa * v}} \tag{A.2}$$

$I_i$ = maximum intensity of the texture of a bacterium $i$ (Gaussian distributed)
$v$ = distance transformation value for the corresponding pixel on the mask
$\kappa$ = controls the slope of the intensity at the bacterium's edges

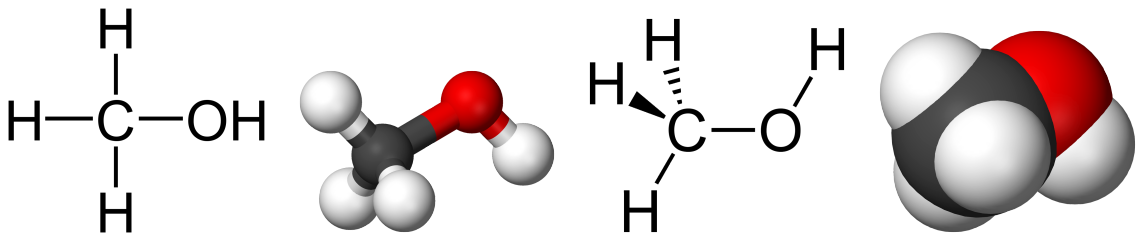
The bacterial intensity is highest in the blue channel with lowest variability. The green channel has medium intensity level and variability. The red channel has the lowest intensity and the highest variability. Each channel depicts linearly increasing background intensity from the left to the right side of a modeled cell to simulate illumination inhomogeneity. This slope of the intensity ramp is chosen to be increasing from the blue channel over the green channel to the blue channel. Gaussian noise is added with increasing levels from blue channel over the green channel to the red channel.

# B

# Amino Acids

Biomolecules represent a huge collection of objects with individual structural, geometrical, qualitative and quantitative features. Although the feature representations are standardized to some extent depending on the used structural formula (e.g. skeletal formula, Fischer projection, etc); learning to navigate in this knowledge domain takes years. This is rendered possible by using the graphical standards of the chemical nomenclature[127].

There exists many different ways of representing the structure of a molecule. I list five different ones: the molecular formula, where only the number of each kind of atom are presented, the structural formula shows which atoms are connected, the ball-and-stick model represents the atoms as sphere and the bonds as sticks in 3D, the perspective drawing, or a wedge-and-dash shows the three-dimensional structure of the molecule, and the space filling model or the representation of van der Walls forces, shows the atoms and molecule but not the bonds. Figure B.1 showcases four of these representations for the example of Methanol with the molecular formula $CH_4OH$.




**(a)** The structural formula (Lewis representation)

**(b)** The ball-and-stick model

**(c)** The wedge and dash projection

**(d)** The space-filling, van der Waal's based representation

**Figure B.1:** Different representations for the example molecule Methanol ($CH_4OH$).

Whether it is the letter H or a white ball, the hydrogen atom is represented by a data abstraction. The design space of a molecule comprises a multiplicity of molecular representations based on different data modalities. Each representation heightens a particular feature of the molecule. In the following section, I address the design space of a special group of molecules, the amino acids.

In this endeavor, I report a particular design I developed to aid retain the molecular formulae of a special category of organic molecules: the amino acids. Amino acids are the building blocks of proteins. They have different features, which are often shared among more than one amino acid. Current specialist representations have shortcomings for less expert target audiences or the public. For an untrained eye, it is difficult to spot the distinct part(s) of a molecule or a particular feature (see Figure B.2). I address the representation of the twenty amino acids, by employing a simplified molecular representation, a novel visual encoding, and a flash cards system to help perceive differences among the amino acids. As seen in other attempts of scientific vulgarization, flash cards, and card games have been used to educate the public. Two prominent examples are 'Phylo' and 'Molecules', to inform the notion of biodiversity and the building of chemical compounds, respectively[128,129].

The task at hand is to identify molecular features, simplify the structure, and visualize the molecules. To do so, I classify the amino acids based on these features, then design an abstraction that simplifies the structure based on its shape, the number of atoms, etc.

The data comprises the different molecular properties which permit to categorize the biomolecules. I report the amino properties and the structural formula representations. First, I take into account four data attributes, which describe the physicochemical properties of

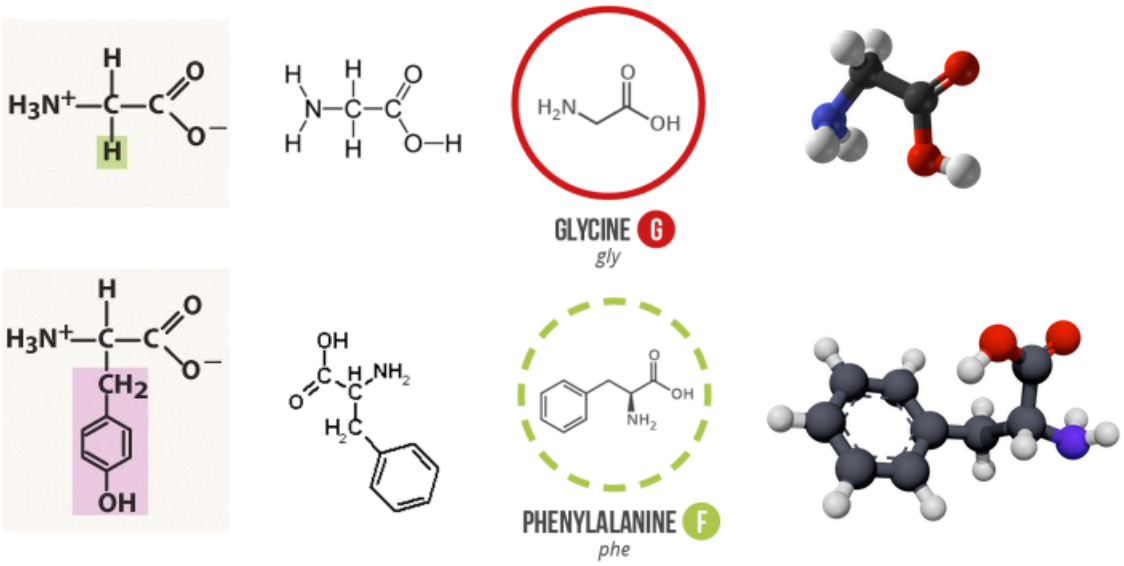



**Figure B.2:** Current specialist representational forms, and their shortcomings for less expert target audiences. Two amino acids were considered: Glycine, and Phenylalanine. These representations were taken from Wikimedia Creative Commons, and Compound Interest. The representations range from the Fischer representation (left) to the cyclohexane conformation (right). The differences between the molecules are easy to spot for the trained eye of a specialist. Yet, the first representation on the left is the clearest for less target audiences. This is due to simply highlighting differences (i.e. the side chain), and leaving the redundant part (i.e. common skeleton) in the background. All the other different representations are correct yet provide no means for easy recollection or to guide the reader's eye and retain their attention.

each amino acid: (a) the molar mass $[g/mol]$: the given mass of a compound divided by its amount, (b) the isoelectric point [no unit]: pH at which a molecule is neutral or does not migrate in an electric field, (c) the solubility in water at 20°C $[g/L]$: the ability of a solute in $g$ to be dissolved in one Liter of solvent, (d) the frequency in proteins (%): for vertebrates in the Protein Data Bank[130]. These properties reflect the chemistry of an amino acid, and help determine its state given a certain environment. Second, the structural formula representations provide multiple solutions. These range from but are not limited to the molecular surface of the side chain, to representations of the covalent radii of the atoms[131], to the unspecified stereochemistry representation. I choose the latter representation, where a mixture of both enantiomers is present and is indicated explicitly as a wavy line (i.e. each of a pair of molecules that are mirror images of each other). Such a wavy line is shown in the Lysine structure as part of the common skeleton (c.f. Figure B.4). The wavy line simplifies the structure of amino acids and is indicative of their sequential assembly into a protein.

To represent each amino acid, I opted for the unspecified stereochemistry after the design process of multiple iterations. Figure B.3 illustrates the major iterations that led to the adopted representation. The representations of the amino acids were documented, and validated from previous Biochemistry knowledge[132]. Both the amino acids representations and the card design were created using LaTeX and TikZ[133]. In this endeavor, I define two different tasks:

(1) Find similarities/differences between all the amino acids molecules

(2) Represent a simplified structure for each molecule.

To address them, I first classify the amino acids based on molecular features, then I design an abstraction that simplifies the structure based on its shape, the number of atoms, etc.

The solution is formulated in respect to each task, as follows. The first task is addressed using sketching and biochemical knowledge of the atomic composition of each molecule. Sketching in Figure B.3 permits to go through multiple iterations to refine the parts of a molecule that make it peculiar and different than the rest.

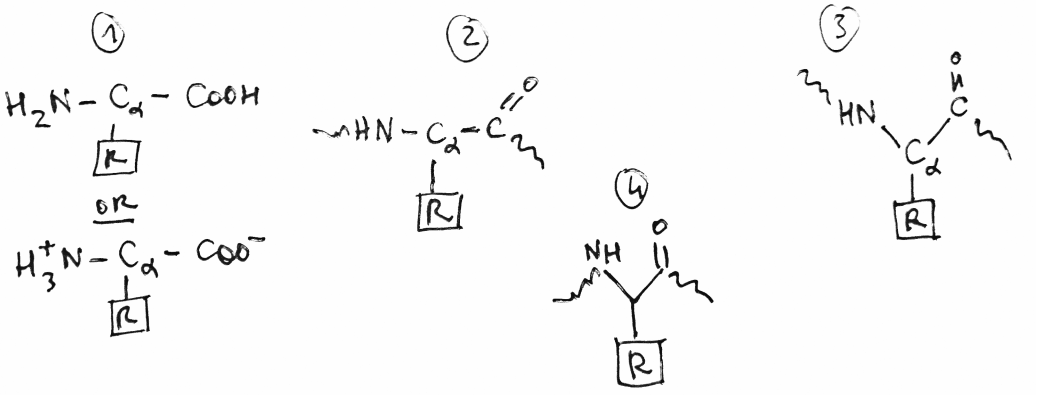



**Figure B.3:** Sketches of the design process of the main molecule representation. The redundant part of the molecule (i.e. common skeleton) is shared between the amino acids. Subfigures (1) to (4) showcase these iterations. The squared R group is for the side chain. (1) The positively charged parts of the common skeleton: amino and carboxyl groups, are included. (2) The simplification and the adoption of the snake-like shapes using the unspecified stereochemistry representation. (3) The process of finding a correct representation. (4) The final design without the stylized differences: the snake-like shapes are in a correct position, symmetrical as is the Y-shape of the common skeleton.

The second task of visually representing the molecules relies on visual encoding and the use of a data abstraction: the canopy.

To help memorize the structural formula of amino acids, molecular features are encoded into shapes, colors, and textures as reported in Figure B.4. I first detail the visual encoding as follows. Once molecular features are identified, the amino acids are grouped (1). Each molecule has its unique name encoded in three different ways (2). Each resulting molecule design is unique and is accompanied with data attributes that describe its physicochemical properties (3).

(1) Amino acids groups. Three main classifications exist: one that targets whether an amino acid is essential, and two others that depend on the side chain structure (i.e. where differences occur). There exists multiple ways to group amino acids based on the side chain. To adopt a compact grouping, and support easy memorization, I chose four categories: acidic, basic, polar, and non-polar. The groups were visually represented by nominal colors, and glyphs: blue – circle with minus sign, red – circle with plus sign, purple – empty circle, and green – full circle, respectively. I chose saturated hues of these colors for a more vibrant card set[134]. Each group has a corresponding category card which explains the main physicochemical properties of the grouped amino acids (see Figure B.6).

(2) Amino acids name encoding. They were reported at the top of each card: the full name, the three letters code, and the one letter code (example of Lysine, Lys, K)[135].

(3) Amino acid properties. Each card bears the properties under four card attributes: molar mass, isoelectric point, solubility, and frequency. They are represented by symbols: a scale, an electric sign, an erlenmeyer flask, and a pie chart, respectively.

Second, to simplify the molecular representation, I define a data abstraction: the canopy. It overlays the molecular structure, and comprises stylized differences: emphasis, de-emphasis. The canopy employs the Gestalt principle of symmetry to easily perceive similarities, and differences in the structural representation. Emphasis is given to changing parts of the molecule, i.e. in the foreground. On the contrary, de-emphasis is employed for the common part of the molecule, i.e. in the background (see Fig. B.4). De-emphasis is brought by layering wave-like lines on top of the common part. It creates the effect of a texture. Only one amino acid among twenty, Proline, is exempt of the common part. Hence, only emphasis is employed. In the case of emphasis, the visual encoding employed to represent the data abstraction is more elaborate. Emphasis of the canopy is based on two instances: the amount of carbon

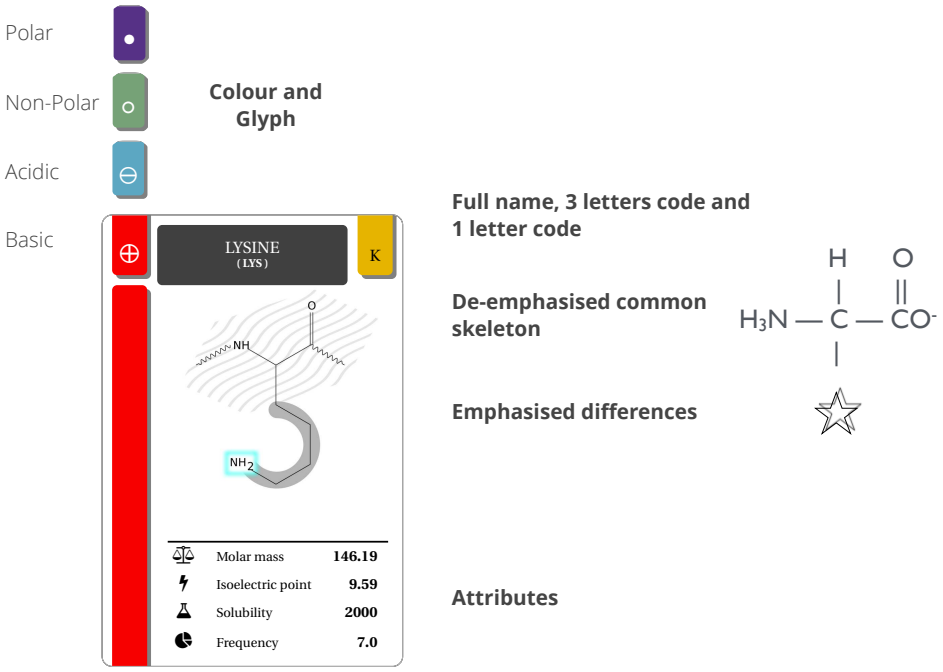
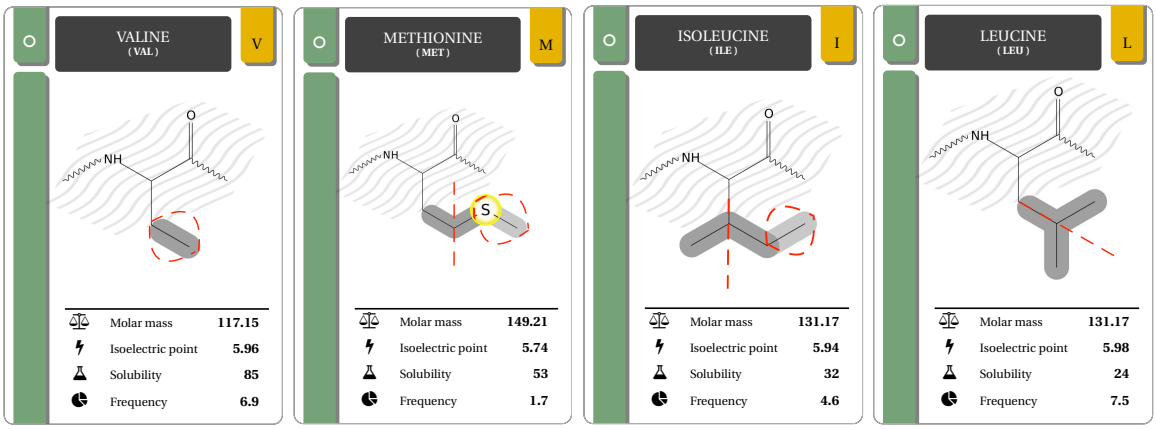



**Figure B.4:** The four major steps to visually encode each amino acid flash card. Color and glyph category encoding, the name encoding, the stylized differences encoded (emphasis, de-emphasis), and the attributes.

bonds and the presence of special atoms. In both instances, this concerns only the molecule's side chain. In the first instance, the canopy is represented by light-, and gray-shapes. The use of light and dark gray helps perceive differences by using the gestalt principles of similarity, and proximity (e.g. axial/central symmetry in Figure B.5)[137,138,139]. A change in luminance reflects an asymmetry. The second instance addresses the presence of peculiar atoms, that is Sulphur (S) and Nitrogen (N). Their visual encoding changes by using a unique color and shape, respectively. Sulphur and Nitrogen atoms are highlighted using a yellow circle and teal blue rectangles, respectively (c.f. Fig. B.4 and Fig. B.5). Coupling the canopy abstraction and appropriate visual encoding, twenty four cards result for twenty amino acids, with additional four group cards. The full set of flash cards is available at `http://bit.ly/aa-cards`.




**(a)** Valine (V) **(b)** Methionine (M) **(c)** Isoleucine (I) **(d)** Leucine (L)

**Figure B.5:** Stylized differences explained for a subset of amino acids pertaining to the non-polar group. De-emphasized common skeleton in the background using wave-like lines. Emphasized differences in the foreground are depicted in luminance (i.e. two different grays: light, dark). This emphasis depends on the amount of carbon bonds. Whereas light and dark grays are chosen to perceive symmetries (i.e. axial symmetry: dotted red line), asymmetries (denoted in a red polygon) and special atoms (e.g. zoomed-in and encircled Sulphur (S) atom).

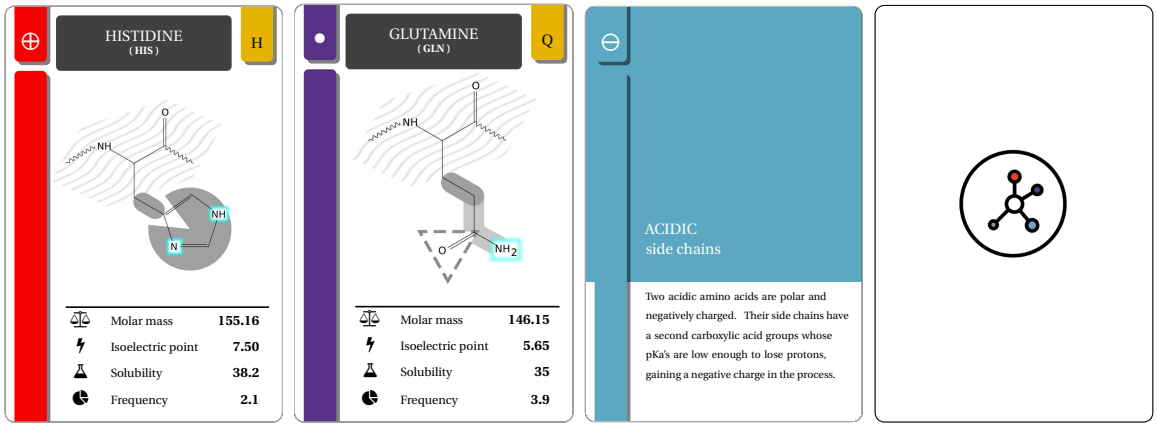



**Figure B.6:** Example playing cards included in the cards game. (a) The Histidine (H) amino acid card. With its respective formula, the category color, and symbol (left, and top left of card) which is in this case the basic category (or positively charged, in red). (b) The Glutamine (Q) amino acid card, from the polar group (in purple). (c) The acidic category card, it explicates the amino acids properties pertaining to this category. (d) Back design for each card. Logo adapted with the colors of the four categories, courtesy of Ed Harrison [136].

# C

# Cell segmentation task

It is difficult to extract features that are both reliable for detecting objects of interests (e.g. bacterial cells), and robust to natural variations in the image data (e.g. changes in luminosity). To discriminate bacterial cells from background, machine learning and in particular deep learning has proven conclusive in the field of bioimage informatics. As opposed to task specific algorithms, deep learning is part of the broader machine learning methods based on learning data representations and has brought about breakthroughs in processing images, video, speech and audio[140,141]. Deep learning permits the computer to learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts. The deep convolutional networks rely on neural networks, a system that is modeled on the human brain and nervous system. The neural networks rely on input data (i.e. training data) to effectively learn to recognize the presence of important discriminative information.

In this appendix, I present two experiments where neural networks are employed in the task of single cell segmentation using the architecture of the Network in Network (NIN) and the Fully Convolutional Network (FCN), respectively. The computation of this study was conducted using the `Caffe` framework[142] and `TFLearn`[143]. Neural networks are most often represented as directed graphs and are referred to as network graphs[144]. Each unit is repre-

sented by a labeled node according to its output and the networks units are interconnected by directed edges. I illustrate a single processing unit in Figure C.1 with the external input $w_0$.

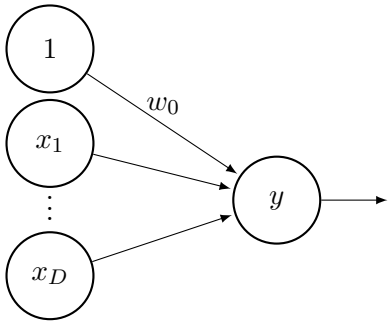



**Figure C.1:** Network graph for a single processing unit. A processing unit consists of a propagation rule mapping all inputs $w_0, x_1 \ldots, x_D$ to the actual input $z$, and an activation function $f$ which is applied on the actual input to form the output $y = f(z)$. Here, $w_0$ represents an external input called bias and $x_1, \ldots, x_D$ are inputs from other units of the network. In a network graph, each unit is labeled according to its output. Therefore, to include the bias $w_0$ as well, an example unit with value 1 is included.

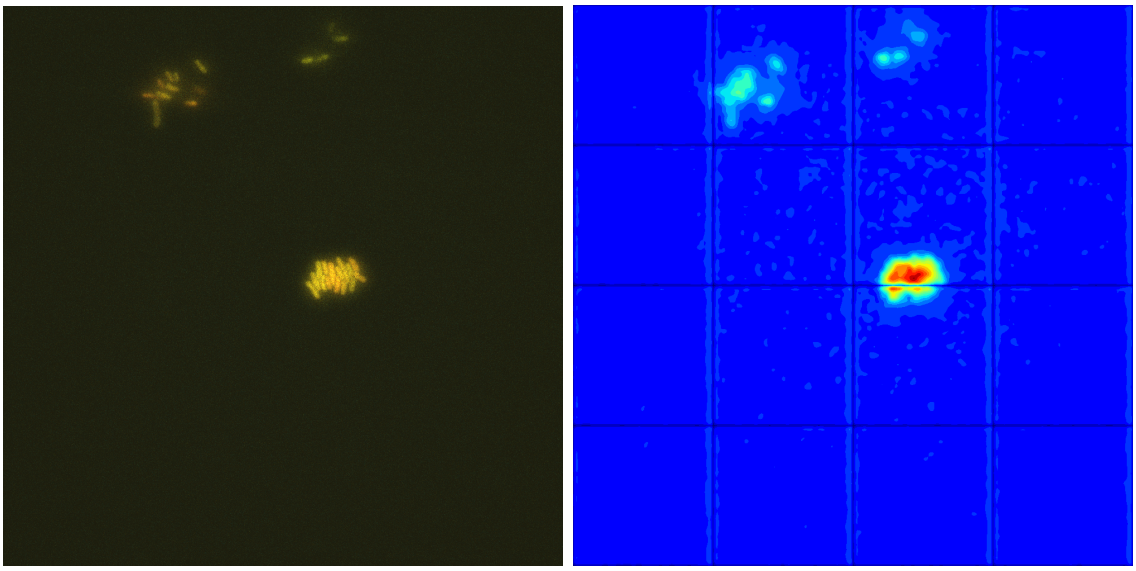
This study comprises three steps: 1. creating the image tiles and clustering them, 2. conducting the first experiment using the NIN architecture, and 3. using a FCN for the second.

A preprocessing step consists of preparing a subset of the data. I select three frames from biomovies D1 and D3 comprising: few observable bacterial cells, a larger count of observable cells, a fully crowded bacterial colony. All images are then supplied across the RGB channels (i.e. TIRF). Image tiles of size $n * n$ are generated with $n = 32$. Then, I apply a vector quantization method, i.e. k-means clustering, to partition $n$ observations into $k$ clusters in which each observation belongs to the cluster with the nearest mean. This helps dissociate the different observations where image tiles contain solely the background, or only foreground signals from the bacteria, etc. Provided $k = 128$, the image tiles are stored with their cluster number, and are then used for each of the supervised experiments.

In the first experiment, I use the NIN architecture which has been successfully demonstrated with state-of-the-art classification performances[145]. It relies on a multilayer structure, in which a micro network (MLP) is introduced within each convolutional layer to compute more abstract features from the local image tiles. Moreover, it is less prone to overfitting than traditional fully connected layers. As described in Lin et al., I employ proper initializations for the weights and learning rates are set manually, as described in Krizhevsky et al.[140]. The network is trained using mini-batches of size $k$. The training process starts from the initial weights and learning rates, then continues until the accuracy on the training set stops improving. As detailed in Lin et al., this procedure is repeated once such that the final learning rate is one percent of the initial value[145]. In the second experiment, I employ a FCN.

As Shelhamer et al. demonstrated it, FCNs can make dense predictions for per-pixel tasks like semantic segmentation[146].

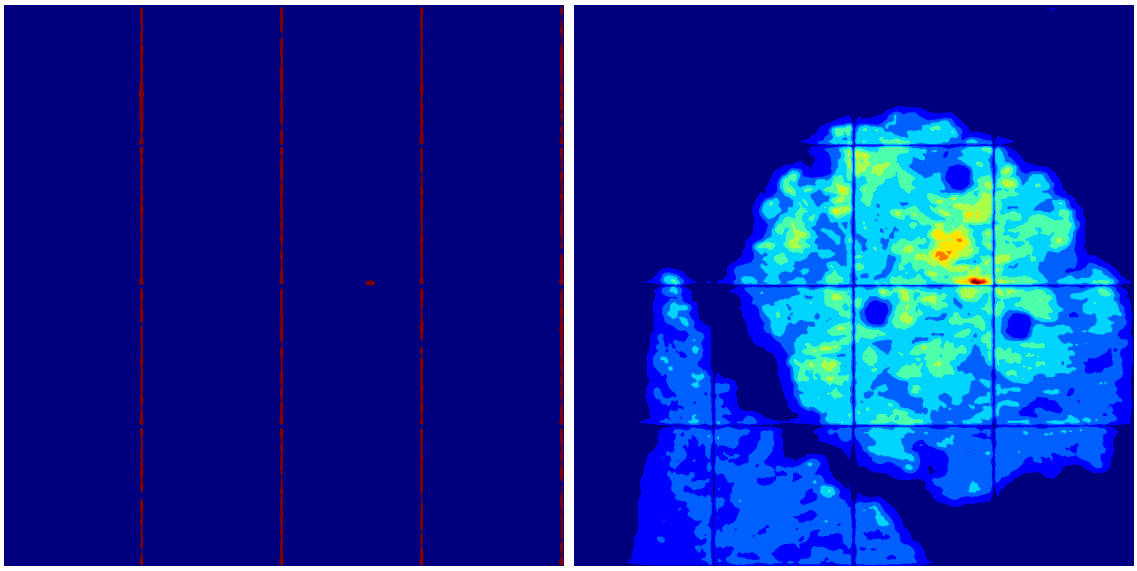
The feature maps are visualized as heat maps for the last activation layer of the network in Figure C.2 and Figure C.3. I report for each experiment one best result, where a more or less good prediction is visually encoded from low prediction rates (in blue) to high prediction rates (in red). For reproducibility, the network structure of both experiments, the amount of input layers that are being used, and the image patches are freely provided at `http://github.com/ghattab/ml-seg`.



**(a)** D3 - RGB $I_{19}$ **(b)** FCN Feature map - $I_{19}$

**Figure C.2:** Feature map of the last layer of the FCN for biomovie D3 - $I_{19}$. (a) RGB image $I_{19}$ (enhanced image exposure: 80%) (b) Heat map.

As seen in both Figure C.2 and Figure C.3, the neural networks are more or less able to discriminate bacterial cells from background and noise, yet are unable to successfully accomplish the task of single cell segmentation in the particular case of this data.



**(a)** FCN Feature map - $I_{115}$ **(b)** NIN Feature map - $I_{115}$

**Figure C.3:** Feature maps of the last layer for the FCN and the NIN models, respectively. Test frame 115 from biomovie D1 (c.f. Figure 2.1). (a) Heat map (FCN). (b) Heat map (NIN)

# D

# Data Structures

The presented framework in chapter 5 employs two main data structures to compute, access, and store particle and patch information. On one hand, particle tracking results are stored using a 2-dimensional data frame structure. The storage and dynamic encoding of the patch indexes is achieved with the data analysis library `pandas`. An excerpt of an example data frame is presented in Table D.1. The data frame is exported in the CSV format; other formats are also available: Excel, SQL, HDF5, etc.

On another hand, I store the patch lineage graph using the Python high-productivity library: `networkxx`. It contains different structural analyses and measures. In this particular endeavor, its usage mainly relies on a dictionary of dictionaries with two distinct levels: (a) time index based (i.e. $t$) and (b) key based (e.g. patch index, patch boundary, etc). The graph's internal data structure is based on the adjacency list representation and I justify its usage for rapid querying and updating of the patch lineage. For the first level, let $\mathbf{G}$ be the patch lineage graph. I denote the following general format: `G.node[t]` which returns all the information at the node $t$, including the patch index and other keys. For the second level, provided a key, the format changes to `G.node[t]['key']`. As in the case of a dictionary in Python, the command `G.node[t].keys()` returns all the *keys* that could be queried at each node. The implemented keys include but are not limited to: the patch index 'pids', the parti-

| ecc | patch | particle | signal | size | x | y | z |
|---|---|---|---|---|---|---|---|
| 0.0234 | 1 | 2 | 189.73 | 22160 | 484.116943439 | 457.772918373 | 0 |
| 0.0555 | 1 | 2 | 187.12 | 22160 | 485.620440404 | 456.667551941 | 1 |
| 0.0931 | 1 | 2 | 184.31 | 22160 | 488.966926345 | 455.997947074 | 2 |
| 0.0380 | 1 | 2 | 190.26 | 22160 | 491.463650296 | 456.621354238 | 3 |
| 0.0465 | 1 | 2 | 184.24 | 22160 | 492.046998513 | 457.658365366 | 4 |
| 0.0417 | 1 | 2 | 168.47 | 22160 | 492.782556827 | 458.680769203 | 5 |
| 0.0899 | 1 | 2 | 171.64 | 22160 | 493.12536365 | 458.516956741 | 6 |

**Table D.1:** Excerpt of an example 2-dimensional data frame. Various characterizations of a particle's appearance are computed, as seen in Crocker and Grier centroid-finding algorithm: the `size` is the radius of gyration of particle's Gaussian, `ecc` is its eccentricity (0 is circular), etc. Other important measures include: the RGB fluorescence signal per channel (`r, g, b`), as raw image values and normalized with an 8-bit encoding (i.e. from 0 to 255).

cle index 'p', the particle coordinates 'c', the bounding particles indices 'pb'. The following example call `G.node[33]` in the case of biomovie D1 outputs `{'c': [[(553.92050275650547, 520.01613202063277)]], 'b': [[(553.92050275650547, 520.01613202063277)]], 'pw': [[0.0]], 'pb': [[0.0]], 'pids': [[0.0]], 'w': [[(553.92050275650547, 520.01613202063277)]], 'id': [155]}`. Whereas `G.node[33]['pb']` outputs: `[[0.0]]`. In this particular case, this patch has an ID of 155 and comprises one particle with particle index 0. Such a patch is referred to as a singleton patch.

# Acronyms

**CLAHE** Contrast Limited Adaptive Histogram Equalization 37, 44, 55, 58, 59

**CYCASP** (C)olon(Y) growth and (C)ell (A)ffect in (SP)atiotemporal experiments 54, 64, 104, 109, 110, 112, 114, 120, 121

**DAG** Directed Acyclic Graph 104, 108–110, 114, 119

**DFS** Depth-First Search 83, 84, 86, 88, 92, 93, 95

**DNA** Deoxyribonucleic Acid 10

**NM** Nominal Mapping 71–74, 76, 77

**PHT** Probabilistic Hough Transform 41

**PM** Progeny Mapping 72–74, 76, 77

**SEEVIS** (S)egmentation-fr(EE) (VIS)ualization 64, 73, 74, 78, 110, 114, 119, 121

**SIFT** Scale-Invariant Feature Transform 41

**SNR** Signal to Noise Ratio 6, 16, 29, 32, 49, 121

**TIRF** Total Internal Reflection Fluorescence 16, 134

**TM** Time Mapping 71, 73, 74, 76, 77

**ViCAR** (Vi)sual (C)ues (A)daptive (R)egistration 36, 47, 49

# Glossary

**biomovie** a particular movie resulting from time lapse imaging of cell colonies in microfluidics chambers; pp. iii, v, 7, 8, 15–18, 20–23, 25, 26, 29, 33, 35, 47–49, 51–54, 56, 57, 61, 62, 71, 73, 75–79, 96, 97, 110–112, 114, 117–119, 134–136, 138

**colony** a community of cells of one kind living close together or forming a physically connected structure; pp. iii, 11, 17, 22, 56, 96, 118

**depth penetration** a measure of how deep light or any electromagnetic radiation can penetrate into a material; pp. 4

**embryogenesis** the formation and development of an embryo; pp. 2

**enantiomer** each of a pair of molecules that are mirror images of each other; pp. 127

**fluorophore** a fluorescent chemical compound that can re-emit light upon light excitation; pp. 16, 17, 19, 124, 142

**galactoglucan** a polysaccharide composed of alternating glucose and galactose units; pp. 18

**gene marker** a broader term than gene. It is a segment of DNA with an identifiable physical location on a chromosome whose inheritance can be followed; pp. 141, 142

**genotype** the genetic constitution, or genetic material of an individual organism. It is often contrasted with phenotype; pp. 2, 18, 141

**lineage** an unbroken chain of ancestors and descendants; pp. 2, 4, 7, 52, 118

**linkage** genotype-phenotype linkage is obtained by analyzing the heritability of certain genes, and other gene markers based on their location. By following the inheritance of

genes, or gene markers, such an analysis serves as a way of genetic testing (e.g. drug screening, diagnosis of genetic diseases, etc); pp. 2

**mCherry** a fluorophore used in biotechnology as a tracer to follow the flow of fluids, and as a marker when tagged to molecules, and cell components; pp. 18, 19, 105

**microplate** a microtiter plate, or microplate, or microwell plate, or multiwell, is a flat plate with multiple 'wells' used as small test tubes for biological experimentation. It is a standard tool in analytical research, and clinical testing. A microplate typically has 6, 24, 96, 384, or 1536 sample wells arranged in a 2:3 rectangular matrix; pp. 16, 108, 120

**particle** an intuitive geometric abstraction that results from considering whether the neighborhood around a pixel falls within a cell by checking for signal characteristics such as signal intensity, edge orientation, fluorescence signals, or texture; pp. iii, 8, 62, 110, 118–121

**particle trajectory** assembled by tracking a particle over time, exploiting temporal coherence to filter out spurious signals that do not persist across multiple frames; pp. 63, 64

**patch** the aggregation of spatially contiguous particle trajectories that feature similar fluorescence patterns; pp. iii, 8, 51, 78–81, 87, 89, 95, 97, 98, 104, 111, 112, 119

**patch lineage** encapsulates the splitting and joining of all the patch trajectories that descend from a common ancestor; pp. iii, 8, 79, 137

**patch trajectory** reflects the evolution of patches across multiple frames; pp. 119

**phenotype** the set of observable characteristics of an individual resulting from the interaction of its genotype with the environment; pp. 2, 18, 141, 142

**quorum sensing** a system of stimuli and responses correlated to population density. Quorum sensing, or QS, allows bacteria to restrict the expression of specific genes at the high cell densities to prioritize the most beneficial phenotypes; pp. 4, 18, 108

**segmentation** referred to as spatial coherence, entails delineating individual cells in each frame; pp. iii, 2, 3, 6, 7, 51, 52, 54, 73, 114, 118

**side chain** a group of atoms attached to the core part of a molecule called 'main chain', or backbone; pp. 127–130

**tracking** referred to as temporal coherence, entails following identified cell positions throughout a biomovie; pp. iii, 3, 7, 52, 63, 118

# References


[1] RJ. Petri. Eine kleine modification des kochschen plattenverfahrens. *Centralbl Bacteriol Parasitenkunde*, 1:279–280, 1887.

[2] R. Bockrath, D. Harper, and S. Kristoff. Crowding depression of UV-mutagenesis in E. coli. *Mutat. Res.*, 73(1):43–58, Nov 1980.

[3] R. Reski. Development, genetics and molecular biology of mosses. *Plant Biology*, 111(1):1–15, 1998.

[4] A. T. Soldo and S. A. Brickson. A simple method for plating and cloning ciliates and other protozoa. *J. Protozool.*, 27(3):328–331, Aug 1980.

[5] GK. Batchelor. The effect of brownian motion on the bulk stress in a suspension of spherical particles. *Journal of fluid mechanics*, 83(01):97–117, 1977.

[6] E. Klein. *Micro-organisms and Disease: An Introduction Into the Study of Specific Micro-organisms*. Macmillan, 1885.

[7] G. M. Whitesides. The origins and the future of microfluidics. *Nature*, 442(7101):368–373, Jul 2006.

[8] A. Groisman, C. Lobo, H. Cho, J. K. Campbell, Y. S. Dufour, A. M. Stevens, and A. Levchenko. A microfluidic chemostat for experiments with bacterial and yeast cells. *Nat. Methods*, 2(9):685–689, Sep 2005.

[9] L. J. Millet and M. U. Gillette. Over a century of neuron culture: From the hanging drop to microfluidic devices. *Yale J. Biol. Med.*, 85(4):501–521, Dec 2012.

[10] F. Amat and JP. Keller. Towards comprehensive cell lineage reconstructions in complex organisms using light-sheet microscopy. *Develop. Growth Differ.*, 2013.

[11] A. McMahon, W Supatto, SE Fraser, and A. Stathopoulos. Dynamic analyses of drosophila gastrulation provide insights into collective cell migration. *Science*, (322), 2008.

[12] J. Swoger, M. Muzzopappa, H. Lopez-Schier, and J. Sharpe. 4D retrospective lineage tracing using spim for zebrafish organogenesis studies. *J. Biophotonics*, 4, 2011.

[13] B. Okumus, S. Yildiz, and E. Toprak. Fluidic and microfluidic tools for quantitative systems biology. *Curr. Opin. Biotechnol.*, 25, 2014.

[14] Q. Wang, J. Niemi, C.-M. Tan, L. You, and M. West. Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. *Cytometry A.*, 1(77), 2010.

[15] E. Andrianantoandro, S. Basu, D.K. Karig, and R. Weiss. Synthetic biology: New engineering rules for an emerging discipline. *Molecular systems biology*, 2(1), 2006.

[16] M. Tehranirokh, A.Z. Kouzani, P.S. Francis, and J.R. Kanwar. Microfluidic devices for cell cultivation and proliferation. *Biomicrofluidics*, 7(5):051502, 2013.

[17] J. Klein, S. Leupold, I. Biegler, R. Biedendieck, R. Münch, and D. Jahn. TLM-Tracker: Software for cell segmentation, tracking and lineage analysis in time-lapse microscopy movies. *Bioinformatics*, 28(17), 2012.

[18] J. Huth, M. Buchholz, J.M. Kraus, K. Mølhave, C. Gradinaru, G.v. Wichert, T.M. Gress, H. Neumann, and H.A. Kestler. Timelapseanalyzer: Multi-target analysis for live-cell imaging and time-lapse microscopy. *Computer methods and programs in biomedicine*, 104(2):227–234, 2011.

[19] F. De Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, et al. Icy: An open bioimage informatics platform for extended reproducible research. *Nature methods*, 9(7):690–696, 2012.

[20] M. A. Bray and A. E. Carpenter. CellProfiler Tracer: Exploring and validating high-throughput, time-lapse microscopy image data. *BMC Bioinformatics*, 16:368, 2015.

[21] J-Y. Tinevez, N. Perry, J. Schindelin, G.M. Hoopes, G.D. Reynolds, E. Laplantine, S.Y. Bednarek, S.L. Shorte, and K.W. Eliceiri. Trackmate: An open and extensible platform for single-particle tracking. *Methods*, 2016.

[22] 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. van den Berg, M. Endele, M. Etzrodt, C. Marr, F.J. Theis, and T. Schroeder. Software tools for single-cell tracking and quantification of cellular and molecular properties. *Nat. Biotechnol.*, 34(7):703–706, Jul 2016.

[23] A. J. Pretorius, I. A. Khan, and R. J. Errington. A survey of visualization for live cell imaging. *Computer Graphics Forum*, 2016.

[24] P. Charoenpanich, M.J. Soto, A. Becker, and M. McIntosh. Quorum sensing restrains growth and is rapidly inactivated during domestication of *Sinorhizobium meliloti*. *Environ. Microbiol. Rep.*, 2015.

[25] E. Krol and A. Becker. Rhizobial homologs of the fatty acid transporter fadl facilitate perception of long-chain acyl-homoserine lactone signals. *Proc. Natl. Acad. Sci.*, 29(111), 2014.

[26] J.-P. Schlüter, P. Czuppon, O. Schauer, P. Pfaffelhuber, M. McIntosh, and A. Becker. Classification of phenotypic subpopulations in isogenic bacterial cultures by triple promoter probing at single cell level. *J. Biotechnol*, 198:3:14, 2015.

[27] A. Wang, L. You, and M. West. Celltracer: Software for automated image segmentation and lineage mapping for single-cell studies. *Systems Biology*, 2005.

[28] K. Zagorovsky and W. Chan. Bioimaging: Illuminating the deep. *Nature materials*, 12(4):285–287, 2013.

[29] H.P. Klug, L.E. Alexander, et al. X-ray diffraction procedures. 1954.

[30] R.H. Ackerman, J.A. Correia, N.M. Alpert, J-C. Baron, A. Gouliamos, J.C. Grotta, G.L. Brownell, and J.M. Taveras. Positron imaging in ischemic stroke disease using compounds labeled with oxygen 15: Initial results of clinicophysiologic correlations. *Archives of Neurology*, 38(9):537–543, 1981.

[31] E.M. Haacke, R.W. Brown, M.R. Thompson, R. Venkatesan, et al. *Magnetic resonance imaging: Physical principles and sequence design*, volume 82. Wiley-Liss New York:, 1999.

[32] C. Riccardi and I. Nicoletti. Analysis of apoptosis by propidium iodide staining and flow cytometry. *Nature protocols*, 1(3):1458–1461, 2006.

[33] D.J. Heeger and D. Ress. What does fMRI tell us about neuronal activity? *Nature Reviews Neuroscience*, 3(2):142–151, 2002.

[34] J. Yao, L. Wang, J-M. Yang, K.I. Maslov, T. TW Wong, L. Li, C-H. Huang, J. Zou, and L.V. Wang. High-speed label-free functional photoacoustic microscopy of mouse brain in action. *Nature methods*, 12(5):407–410, 2015.

[35] L.V. Wang and S. Hu. Photoacoustic tomography: In vivo imaging from organelles to organs. *Science*, 335(6075):1458–1462, 2012.

[36] B. Neumann, T. Walter, J-K. Hériché, J. Bulkescher, H. Erfle, C. Conrad, P. Rogers, I. Poser, M. Held, U. Liebel, et al. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature*, 464(7289):721–727, 2010.

[37] C. Li. A targeted approach to cancer imaging and therapy. *Nat Mater*, 13(2):110–115, Feb 2014.

[38] Y. Wang, K. Zhou, G. Huang, C. Hensley, X. Huang, X. Ma, T. Zhao, B. D. Sumer, R. J. DeBerardinis, and J. Gao. A nanoparticle-based strategy for the imaging of a broad range of tumours by nonlinear amplification of microenvironment signals. *Nat Mater*, 13(2):204–212, Feb 2014.

[39] B. Huang, W. Wang, M. Bates, and X. Zhuang. Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy. *Science*, 319(5864):810–813, 2008.

[40] T. Schroeder. Long-term single-cell imaging of mammalian stem cells. *Nat. Methods*, 8(4 Suppl):S30–35, Apr 2011.

[41] M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D.M. Balak, et al. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30(11):1609–1617, 2014.

[42] E. Meijering. Cell segmentation: 50 years down the road [life sciences]. *IEEE Signal Processing Magazine*, 29(5):140–145, 2012.

[43] E. Meijering, O. Dzyubachyk, and I. Smal. Methods for cell and particle tracking. *Meth. Enzymol.*, 504:183–200, 2012.

[44] Z.D. Smith, I. Nachman, A. Regev, and A. Meissner. Dynamic single-cell imaging of direct reprogramming reveals an early specifying event. *Nature biotechnology*, 28(5):521–526, 2010.

[45] C.M. Ho and Y.C. Tai. Micro-electro-mechanical-systems (mems) and fluid flows. *Annu. Rev. Fluid Mech.*, 30(1):579–612, 1998.

[46] H. A. Stone, A. D. Stroock, and A. Ajdari. Engineering flows in small devices: Microfluidics toward a lab-on-a-chip. *Annu. Rev. Fluid Mech.*, 36:381–411, 2004.

[47] D. J. Beebe, G. A. Mensing, and G. M. Walker. Physics and applications of microfluidics in biology. *Annu. Rev. Biomed. Eng.*, 4:261–286, 2002.

[48] C. S. Chen, M. Mrksich, S. Huang, G. M. Whitesides, and D. E. Ingber. Geometric control of cell life and death. *Science*, 276(5317):1425–1428, 1997.

[49] R. A. Jain. The manufacturing techniques of various drug loaded biodegradable poly(lactide-co-glycolide) (PLGA) devices. *Biomaterials*, 21(23):2475–2490, Dec 2000.

[50] J. Voldman, M. L. Gray, and M. A. Schmidt. Microfabrication in biology and medicine. *Annu. Rev. Biomed. Eng.*, 1:401–425, 1999.

[51] J. El-Ali, P.K. Sorger, and K.F. Jensen. Cells on chips. *Nature*, 442(7101):403–411, 2006.

[52] J. Happel and H. Brenner. *Low Reynolds Number Hydrodynamics with Special Application to Particulate Media*. Prentice-Hall, 1965.

[53] W.B. Russel, D.A. Saville, and W.R. Schowalter. *Colloidal dispersions*. Cambridge university press, 1989.

[54] G. Velve Casquillas, C. Fu, M. Le Berre, J. Cramer, S. Meance, A. Plecis, D. Baigl, J-J. Greffet, Y. Chen, M. Piel, and P. T. Tran. Fast microfluidic temperature control for high resolution live cell imaging. *Lab Chip*, 11:484–489, 2011.

[55] P. Wang, L. Robert, J. Pelletier, W. L. Dang, F. Taddei, A. Wright, and S. Jun. Robust growth of Escherichia coli. *Curr. Biol.*, 20(12):1099–1103, Jun 2010.

[56] A. Grunberger, C. Probst, S. Helfrich, A. Nanda, B. Stute, W. Wiechert, E. von Lieres, K. Noh, J. Frunzke, and D. Kohlheyer. Spatiotemporal microbial single-cell analysis using a high-throughput microfluidics cultivation platform. *Cytometry A*, 87(12):1101–1115, Dec 2015.

[57] E.K. Sackmann, A.L. Fulton, and D.J. Beebe. The present and future role of microfluidics in biomedical research. *Nature*, 507(7491):181–189, Mar 2014.

[58] H. Peng, A. Bateman, A. Valencia, and JD. Wren. Bioimage informatics: A new category in *Bioinformatics*. *Bioinformatics*, 28, 2012.

[59] B. Obara, AJM. Roberts, PJ. Armitage, and V. Grau. Bacterial cell identification in differential intereference contrast microscopy images. *BMC Bioinformatics*, 14, 2013.

[60] P. Vallotton, C. Sun, D. Wang, L. Turnbull, C. Whitchurch, and P. Ranganathan. Segmentation and tracking individual pseudomonas aeruginosa bacteria in dense populations of motile cells. In *Image and Vision Computing New Zealand, 2009. IVCNZ '09. 24th International Conference*, pages 221–225, Nov 2009.

[61] T. Kanade, Z. Yin, R. Bise, S. Huh, S. Eun Eom, M. Sandbothe, and M. Chen. Cell image analysis: Algorithms, system and applications. In *IEEE Workshop on Applications of Computer Vision (WACV) 2011*, January 2011.

[62] J-P. Schlüter, P. Czuppon, O. Schauer, P. Pfaffelhuber, M. McIntoch, and A. Becker. Classification of phenotypic subpopulations in isogenic bacterial cultures by triple promoter probing at single cell level. *J. Biotechnol.*, (198), 2015.

[63] J.-P. Schlüter, M. McIntosh, G. Hattab, T.W. Nattkemper, and A. Becker. Phase contrast and fluorescence bacterial time-lapse microscopy image data. Bielefeld University, 2015.

[64] Matthew McIntosh and Vera Bettenworth. Onset of quorum sensing and exopolysaccharide production in single cells within growing microcolonies. Philipps University of Marburg, 2017.

[65] V. Wiesmann, T. Sauer, C. Held, R. Palmisano, and T. Wittenberg. Cell Simulation for Validation of Cell Micrograph Evaluation Algorithms. *Biomed. Tech. (Berl)*, Sep 2013.

[66] C. Ware. Visual thinking: For design. Morgan Kaufmann series in interactive technologies. Elsevier Science, 2010.

[67] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, Dec 2013.

[68] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009.

[69] H.R. Sankey. The thermal efficiency of steam-engines. (including appendixes). In *Minutes of the Proceedings of the Institution of Civil Engineers*, volume 125, pages 182–212. Thomas Telford-ICE Virtual Library, 1896.

[70] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986.

[71] S. Helfrich, C. E. Azzouzi, C. Probst, J. Seiffarth, A. Grunberger, W. Wiechert, D. Kohlheyer, and K. Noh. Vizardous: Interactive analysis of microbial populations with single cell resolution. *Bioinformatics*, 31(23):3875–3877, Dec 2015.

[72] J. Klein, S. Leupold, I. Biegler, R. Biedendieck, R. Munch, and D. Jahn. TLM-Tracker: Software for cell segmentation, tracking and lineage analysis in time-lapse microscopy movies. *Bioinformatics*, 28(17):2276–2277, Sep 2012.

[73] K. Li, E. D. Miller, M. Chen, T. Kanade, L. E. Weiss, and P. G. Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Med Image Anal*, 12(5):546–566, Oct 2008.

[74] A. J. Pretorius, I. A. Khan, and R. J. Errington. Cell lineage visualisation. *Computer Graphics Forum*, 34(3):21–30, 2015.

[75] N.K. Logothetis. What we can do and what we cannot do with fMRI. *Nature*, 453(7197):869–878, Jun 2008.

[76] W. M. Abdelmoula, R. J. Carreira, R. Shyti, B. Balluff, R. J. van Zeijl, E. A. Tolner, B. F. Lelieveldt, A. M. van den Maagdenberg, L. A. McDonnell, and J. Dijkstra. Automatic registration of mass spectrometry imaging data sets to the Allen brain atlas. *Anal. Chem.*, 86(8), 2014.

[77] G. Bonmassar, D. P. Schwartz, A. K. Liu, K. K. Kwong, A. M. Dale, and J. W. Belliveau. Spatiotemporal brain imaging of visual-evoked activity using interleaved EEG and fMRI recordings. *Neuroimage*, 13(6 Pt 1):1035–1043, Jun 2001.

[78] A. K. Liu, J. W. Belliveau, and A. M. Dale. Spatiotemporal imaging of human brain activity using functional MRI constrained magnetoencephalography data: Monte Carlo simulations. *Proc. Natl. Acad. Sci. U.S.A.*, 95(15):8945–8950, Jul 1998.

[79] A.H. König, H. Doleisch, E. Gröller, et al. Multiple views and magic mirrors-fMRI visualization of the human brain. 1999.

[80] R. Oostenveld, D.F. Stegeman, P. Praamstra, and A. van Oosterom. Brain symmetry and topographic analysis of lateralized event-related potentials. *Clin Neurophysiol*, 114(7):1194–1202, Jul 2003.

[81] W. F. Bodmer. The public understanding of science. 1985.

[82] A. Irwin and B. Wynne. *Misunderstanding Science?: The Public Reconstruction of Science and Technology*. Cambridge University Press, 1996.

[83] G. Hattab, G.B. Brink, and W.T. Nattkemper. A mnemonic card game for your amino acids. Paper presented at the Information+ conference, Vancouver, Canada, 2016.

[84] P. Thévenaz, U.E. Ruttimann, and M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing*, 7(1):27–41, January 1998.

[85] M. Tektonidis, I.H. Kim, Y.C. Chen, R. Eils, D.L. Spector, and K. Rohr. Non-rigid multi-frame registration of cell nuclei in live cell fluorescence microscopy image data. *Med Image Anal*, 19(1), 2015.

[86] S. Yang, D. Kohler, K. Teller, T. Cremer, P. Le Baccon, E. Heard, R. Eils, and K. Rohr. Nonrigid registration of 3-d multichannel microscopy images of cell nuclei. *IEEE Transactions on Image Processing*, 17(4):493–499, Apr 2008.

[87] L. Cooper, S. Naidu, G. Leone, J. Saltz, and K. Huang. Registering high resolution microscopic images with different histochemical stainings - a tool for mapping gene expression with cellular structures. In *Proc. Workshop on Microscopic image analysis with applications in Biology*, 2007.

[88] A. Hakkinen, A. B. Muthukrishnan, A. Mora, J. M. Fonseca, and A. S. Ribeiro. CellAging: A tool to study segregation and partitioning in division in cell lineages of Escherichia coli. *Bioinformatics*, 29(13):1708–1709, Jul 2013.

[89] I. Mekterović, D. Mekterović, and Z. Maglica. Bactimas : A platform for processing and analysis of bacterial time-lapse microscopy movies. *BMC Bioinformatics*, 15(251), 2014.

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

[91] R.H.J. Fick, D. Fedorov, A.H.K. Roeder, and B.S. Manjunath. Simultaneous cell tracking and image alignment in 3d clsm imagery of growing arabidopsis thaliana sepals. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium*, pages 914–917, April 2013.

[92] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Proceed. IEEE Computer Vision, 6th Int. Conf.*, 1998.

[93] MS. Pizer, PE. Amburn, DJ. Austin, R. Cromartie, A. Geselowitz, T. Greer, TB. Haar Romeny, BJ. Zimmerman, and KJ. Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision Graphics*, (39), 1987.

[94] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, New York, 1982.

[95] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing (CVGIP)*, 1(30), 1985.

[96] S. E. Raza, A. Humayun, S. Abouna, T. W. Nattkemper, D. B. Epstein, M. Khan, and N. M. Rajpoot. RAMTaB: Robust alignment of multi-tag bioimages. *PLoS ONE*, 7(2):e30894, 2012.

[97] G.T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE Melecon*, volume 83, page A10, 1983.

[98] S. Yam and LS. Davis. Image registration using generalized hough transform. In *Proc. IEEE Conf. Pattern Recognition and Image Processing*, 1981.

[99] B. Sun, W. Kong, J. Xiao, and J. Zhang. A hough transform based scan registration strategy for mobile robotic mapping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference*, 2014.

[100] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[101] G. Hattab, J-P. Schlüter, A. Becker, and T.W. Nattkemper. Vicar: An adaptive and landmark-free registration of time lapse image data from microfluidics experiments. *Frontiers in Genetics*, 8:69, 2017.

[102] J. Ashburner, P. Neelin, D.L. Collins, A. Evans, and K. Friston. Incorporating prior knowledge into image registration. *NeuroImage*, 6(4), 1997.

[103] Y. L. Zhang, S. J. Chang, X. Y. Zhai, J. S. Thomsen, E. I. Christensen, and A. Andreasen. Non-rigid landmark-based large-scale image registration in 3-D reconstruction of mouse and rat kidney nephrons. *Micron*, 68, 2015.

[104] K. Mkrtchyan, A. Chakraborty, and A.K. Roy-Chowdhury. Automated registration of live imaging stacks of arabidopsis. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium*, 2013.

[105] Z. Bao, I.J. Murray, T. Boyle, L.S. Ooi, J.M. Sandel, and H.R. Waterston. Automated cell lineage tracing in *Caenorhabditis elegans*. *PNAS*, 103(8), Feb 2006.

[106] D. Allan, T. Caswell, N. Keim, and C. van der Wel. Trackpy: Trackpy v0.3.0, November 2015. Full list of contributors: Aron Ahmadia, Continuum Analytics; François Boulogne, Princeton University; Rebecca Perry, Harvard University; Al Piszcz, The Mitre Corporation Jan Schulz, TU Freiberg; Leonardo Uieda, Universidade do Estado do Rio de Janeiro.

[107] J.C. Crocker and D.G. Grier. Methods of digital video microscopy for colloidal studies. *Journal of Colloid and Interface Science*, 179(1):298 – 310, 1996.

[108] G. Ji, H-W. Shen, and R. Wenger. Volume tracking using higher dimensional isosurfacing. In *Proceedings of the 14th IEEE Visualization 2003 (VIS 03)*, pages 209–216, 2003.

[109] Georges Hattab, Veit Wiesmann, Anke Becker, Tamara Munzner, and Tim W Nattkemper. A novel methodology for characterizing cell subpopulations in automated time-lapse microscopy. *Frontiers in bioengineering and biotechnology*, 6:17, 2018.

[110] T. Munzner. Visualization analysis and design. A.K. Peters visualization series. A K Peters, 2014.

[111] V. Setlur and M.C. Stone. A linguistic approach to categorical color assignment for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):698–707, Jan 2016.

[112] S. Van der Walt and N. Smith. A better default colormap for matplotlib, 2015.

[113] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[114] C.B. Barber, D.P Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[115] Herbert Freeman and Ruth Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, 18(7):409–413, 1975.

[116] C. Ware. *Information visualization: Perception for design*. Elsevier, 2012.

[117] H. Atek, J-P. Kneib, C. Pacifici, M. Malkan, S. Charlot, J. Lee, A. Bedregal, A. J. Bunker, J. W. Colbert, A. Dressler, et al. Hubble space telescope grism spectroscopy of extreme starbursts across cosmic time: The role of dwarf galaxies in the star formation history of the universe. *The Astrophysical Journal*, 789(2):96, 2014.

[118] G. Bertone, D. Hooper, and J. Silk. Particle dark matter: Evidence, candidates and constraints. *Physics Reports*, 405(5):279–390, 2005.

[119] M. Markevitch, A.H. Gonzalez, D. Clowe, A. Vikhlinin, W. Forman, C. Jones, S. Murray, and W. Tucker. Direct constraints on the dark matter self-interaction cross section from the merging galaxy cluster 1e 0657–56. *The Astrophysical Journal*, 606(2):819, 2004.

[120] J. Y. Pierga, F. C. Bidard, C. Mathiot, E. Brain, S. Delaloge, S. Giachetti, P. de Cremoux, R. Salmon, A. Vincent-Salomon, and M. Marty. Circulating tumor cell detection

predicts early metastatic relapse after neoadjuvant chemotherapy in large operable and locally advanced breast cancer in a phase II randomized trial. *Clin. Cancer Res.*, 14(21):7004–7010, Nov 2008.

[121] T. Kojima, Y. Hashimoto, Y. Watanabe, S. Kagawa, F. Uno, S. Kuroda, H. Tazawa, S. Kyo, H. Mizuguchi, Y. Urata, N. Tanaka, and T. Fujiwara. A simple biological imaging system for detecting viable human circulating tumor cells. *J. Clin. Invest.*, 119(10):3172–3181, Oct 2009.

[122] I. Baccelli, A. Schneeweiss, S. Riethdorf, A. Stenzinger, A. Schillert, V. Vogel, C. Klein, M. Saini, T. Bauerle, M. Wallwiener, T. Holland-Letz, T. Hofner, M. Sprick, M. Scharpff, F. Marme, H. P. Sinn, K. Pantel, W. Weichert, and A. Trumpp. Identification of a population of blood circulating tumor cells from breast cancer patients that initiates metastasis in a xenograft assay. *Nat. Biotechnol.*, 31(6):539–544, Jun 2013.

[123] D. Ling, M. J. Hackett, and T. Hyeon. Cancer imaging: Lighting up tumours. *Nat Mater*, 13(2):122–124, Feb 2014.

[124] H. M. T. El-Zimaity, H. Ota, D. Y. Graham, T. Akamatsu, and T. Katsuyama. Patterns of gastric atrophy in intestinal type gastric carcinoma. *Cancer*, 94(5):1428–1436, 2002.

[125] Shan E Ahmed Raza, Linda Cheung, David Epstein, Stella Pelengaris, Michael Khan, and Nasir M Rajpoot. Mimo-net: A multi-input multi-output convolutional neural network for cell segmentation in fluorescence microscopy images. In *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium*, pages 337–340. IEEE, 2017.

[126] Jiashi Feng and Trevor Darrell. Learning the structure of deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2749–2757, 2015.

[127] J. Brecher. Graphical representation of stereochemical configuration (IUPAC Recommendations 2006). *Pure and Applied Chemistry*, 78(10):1–74, September 2006.

[128] N. David, T. Derek, and al. The PHYLO(MON) project, 2010. http://phylogame.org/.

[129] D. Daniel. Molecules – a chemistry card game, 2015. http://playefg.com/.

[130] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, T. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Res*, 28:235–242, 2000.

[131] R. Heyrovska. Atomic Structures of all the Twenty Essential Amino Acids and a Tripeptide, with Bond Lengths as Sums of Atomic Covalent Radii. *ArXiv e-prints*, April 2008.

[132] A. Lehninger, D. L. Nelson, and M. M. Cox. *Lehninger Principles of Biochemistry*. W. H. Freeman, fifth edition edition, June 2008.

[133] Tikz based card template: Creating playing cards using TikZ. https://tex.stackexchange.com/questions/47924/creating-playing-cards-using-tikz/.

[134] Mike B. d3 categorical colors & Google colors. http://bl.ocks.org/aaizemberg/raw/78bd3dade9593896a59d/.

[135] M. O. Dayhoff, W. R Eck, M.A. Chang, and M.R. Sochard. *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, 1965.

[136] The Noun Project. http://thenounproject.com/.

[137] A. Rudolf. The Gestalt theory of expression. *Psychol Rev*, 56(3):156–171, May 1949.

[138] A. Rudolf. *Art and Visual Perception: A Psychology of the Creative Eye, The New Version, Second edition, Revised and Enlarged*. University of California Press, 1974.

[139] D. Chang, K. V. Nesbitt, and K. Wilkins. The gestalt principles of similarity and proximity apply to both the haptic and visual grouping of elements. In *Proceedings of the Eight Australasian Conference on User Interface - Volume 64*, AUIC '07, pages 79–86, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.

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

[141] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013.

[142] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[143] Aymeric Damien et al. Tflearn. https://github.com/tflearn/tflearn, 2016.

[144] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[145] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[146] E. Shelhamer, J. Long, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, Apr 2017.