Leonardo Pedro Donas-Boto de Vilhena Martins

Mestre em Engenharia Biomédica

Image Processing and Simulation Toolboxes of Microscopy Images of Bacterial Cells

Dissertação para obtenção do Grau de Doutor em
Engenharia Electrotécnica e de Computadores

Orientador: José Manuel Matos Ribeiro da Fonseca, Professor Associado com Agregação, FCT-UNL

Co-orientador: André Sanches Ribeiro, Professor, Faculty of Medicine and Health Technology Tampere University

Júri:

Presidente: Prof. Doutor Luís Manuel Camarinha de Matos

Arguentes: Prof. Doutor Pedro Miguel Dinis de Almeida
Prof. Doutor Fernando Jorge Coutinho Monteiro

Vgais: Prof. Doutor Luís Manuel Camarinha de Matos
Prof. Doutor José Manuel Matos Ribeiro da Fonseca
Prof. Doutor Arnaldo Joaquim Castro Abrantes
Prof. Doutor André Teixeira Bento Damas Mora

Monte da Caparica, Março de 2020

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Image Processing and Simulation Toolboxes of Microscopy Images of Bacterial Cells

Acknowledgements

First, I would like to thank my supervisor Prof. Dr. José Manuel Fonseca, who offered me this great opportunity of doing my PhD Thesis project and gave me great support over all this years and is an inspiration to my future academic endeavours.

To Prof. Dr. André Sanches Ribeiro, who welcomed me in his group and provided a great assistance during my stay in Finland, both at supporting and providing me with all the tools necessary for this work.

To all my friends and especially that started our studies in Biomedical Engineering, more than 12 years ago, my own Biofofos: Joaquim Horta, Nuno Fernandes, Hugo Pereira, Pedro Martins, Luís Mendes, Fernando Mota, Bernardo Azevedo, João Martins, Pedro Cascalho, Sérgio Mendes, João Santinha, Filipe Rodrigues, Sara Gil, Mafalda Fernandes, Rita Carvalho, Ana Marques, Ana Valente and Milene Bação, and I also want to thank also want to thank Rita Narciso, Gabriela Pereira, Mafalda Oliveira and Sara Coutinho, who have also been a pillar of support for me. It has been a great journey, and hopefully we will continue to enjoy the ride.

I want to show my greatest gratitude to everyone that I met during my time spent in FCT for all the good times spent with you guys and hopefully we shall continue this great friendship after completing this journey together, and my friends from the PTCG community. A picture is worth a thousand words, but sometimes we just need 4 to express our feelings: I love you guys!!!

I also want to thank to all the friends that I have met in Finland and especially everyone from the Laboratory of Biosystem Dynamics (LBD) who welcomed me into their group.

Finally, I want to thank my family, especially my parents and my brother who always gave me strength to overcome any problem and who gave me the support to make this important move for my education.

This work was supported by the Portuguese Foundation for Science and Technology (FCT/MCTES) through a funded PhD Scholarship (ref. SFRH/BD/88987/2012).

This work was developed in the CA3 group (CA3 - Computational Intelligence Research Group) of CTS, UNINOVA in cooperation with Laboratory of Biosystem Dynamics (LBD) from Tampere University of Technology. This work is integrated in project SADAC – Study of the kinetics of asymmetric disposal of aggregates in cell division and its correlation to functional aging from in vivo measurements, one event at a time – with the reference PTDC/BBB-MET/1084/2012, funded by FCT - Fundação para a Ciência e a Tecnologia.

Abstract

Recent advances in microscopy imaging technology have allowed the characterization of the dynamics of cellular processes at the single-cell and single-molecule level. Particularly in bacterial cell studies, and using the E. coli as a case study, these techniques have been used to detect and track internal cell structures such as the Nucleoid and the Cell Wall and fluorescently tagged molecular aggregates such as FtsZ proteins, Min system proteins, inclusion bodies and all the different types of RNA molecules. These studies have been performed with using multi-modal, multi-process, time-lapse microscopy, producing both morphological and functional images.

To facilitate the finding of relationships between cellular processes, from small-scale, such as gene expression, to large-scale, such as cell division, an image processing toolbox was implemented with several automatic and/or manual features such as, cell segmentation and tracking, intra-modal and intra-modal image registration, as well as the detection, counting and characterization of several cellular components.

Two segmentation algorithms of cellular component were implemented, the first one based on the Gaussian Distribution and the second based on Thresholding and morphological structuring functions. These algorithms were used to perform the segmentation of Nucleoids and to identify the different stages of FtsZ Ring formation (allied with the use of machine learning algorithms), which allowed to understand how the temperature influences the physical properties of the Nucleoid and correlated those properties with the exclusion of protein aggregates from the center of the cell. Another study used the segmentation algorithms to study how the temperature affects the formation of the FtsZ Ring.

The validation of the developed image processing methods and techniques has been based on benchmark databases manually produced and curated by experts. When dealing with thousands of cells and hundreds of images, these manually generated datasets can become the biggest cost in a research project. To expedite these studies in terms of time and lower the cost of the manual labour, an image simulation was implemented to generate realistic artificial images.

The proposed image simulation toolbox can generate biologically inspired objects that mimic the spatial and temporal organization of bacterial cells and their processes, such as cell growth and division and cell motility, and cell morphology (shape, size and cluster organization). The image simulation toolbox was shown to be useful in the validation of three cell tracking algorithms: Simple Nearest-Neighbour, Nearest-Neighbour with Morphology and DBSCAN cluster identification algorithm. It was shown that the Simple Nearest-Neighbour still performed with great reliability when simulating objects with small velocities, while the other algorithms performed better for higher velocities and when there were larger clusters present.

Keywords: Image Processing; Image Simulation; Cell Segmentation; Cell Tracking,

Resumo

Os recentes avanços nas tecnologias imagiológicas utilizadas em microscopia proporcionaram a caracterização das dinâmicas de processos celulares ao nível celular e molecular. Particularmente em estudos com bactérias, e tendo a *E. coli* como caso de estudo, essas técnicas têm sido utilizadas para detetar e monitorizar estruturas celulares como o Nucleoide, a Parede Celular, e também agregados moleculares marcados com fluorescência, como as proteínas FtsZ e do sistema Min, corpos de inclusão e moléculas de RNA. Estes estudos têm sido realizados utilizando microscopia multi-modal, processual e com séries temporais, produzindo tanto imagens morfológicas como funcionais.

De modo a facilitar as descobertas de ligações entre processos celulares, com diferentes escalas, como a expressão genética a divisão celular, foi implementada uma plataforma de processamento de imagem com diversas funções (automáticas e/ou manuais), como a segmentação e monitorização celular, registo de imagens intra-modal e inter-modal, e também a deteção, contagem e caracterização de vários componentes celulares.

Dois algoritmos de segmentação de componentes celulares foram implementados, tendo o primeiro sido baseado na Distribuição Gaussiana e o segundo baseado em funções de limitação e estruturação morfológica. Ambos os algoritmos foram utilizados para segmentar Nucleoides e para identificar os diferentes estágios de formação do Anel de FtsZ (aliado com a utilização de algoritmos de aprendizagem automática). Estes passos permitiram perceber como é que a temperatura influencia as propriedades físicas do Nucleoide e permitiram correlacionar essas propriedades com a expulsão do centro da célula de agregados proteicos. Um outro estudo utilizou os mesmos algoritmos de segmentação para estudar como é que a temperatura influencia a formação do Anel de FtsZ.

A validação dos métodos de processamento de imagem tem sido baseada em bases de dados de referência, produzidas e curadas manualmente por especialistas. Quando se lida com milhares de células e centenas de imagens, essas bases de dados podem tornar-se o maior custo num projeto de investigação. Um simulador de imagens foi implementado para gerar imagens artificiais e realistas, de modo a diminuir o custo do trabalho manual, a acelerar esses estudos em termos de tempo.

O simulador de imagens proposto, pode gerar objetos biologicamente inspirados, sendo estes capazes de imitar a organização espacial e temporal de células bacterianas, tal como imitar os seus processos, como o crescimento celular, divisão, motilidade, e também a sua morfologia (forma, tamanho e organização em aglomerados). O simulador de imagens mostrou ser útil na validação de três algoritmos de monitorização celulares: Simples Vizinho-Mais-Próximo, Vizinho-Mais-Próximo com Morfologia e o algoritmo de identificador de aglomerados, DBSCAN. Foi demonstrado que o Simples Vizinho-Mais-Próximo ainda teve um desempenho de grande fiabilidade quando foram simulados objetos com velocidades baixas, e que os outros algoritmos tiveram melhores desempenhos para velocidades maiores e para aglomerados maiores.

Palavras-chave: Processamento de Imagem; Simulação de Imagem; Segmentação Celular; Monitorização Celular

Table of Contents

List of Figures

List of Tables

Glossary – Acronyms, Abbreviatures and Definitions

This section gives an alphabetical list of Acronyms and Abbreviatures, but also comprehensive definitions of some biological terms, due to the multi-disciplinary nature of this work.

3D-PALM	Three Dimensional Photoactivation Localization Microscopy
3D-SIM	Three-Dimensional Structured Illumination Microscopy
3D-STORM	Three-Dimensional Stochastic Optical Reconstruction Microscopy
API	Application Programming Interface
ATP	Adenosine Triphosphate
Bacteriophage	Virus that infects bacterial cells
BFP	Blue Fluorescent Protein
CA3	Computational Intelligence Research
CART	Classification and Regression Tree
CCD	Charge-Coupled Device
CLSM	Confocal Laser Scanning Microscopy
CME	Chemical Master Equation
DAPI	4', 6-diamidino-2-phenylindole
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNA	Deoxyribonucleic acid
DT	Decision Tree
E. coli	Escherichia coli
FCT	Fundação para a Ciência e a Tecnologia
FCT-UNL	Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa
FISH	Fluorescence in situ hybridization
Fts proteins	Filamenting temperature sensitive proteins
Functional Images	Images with Cellular Functional Information (Spatial and Temporal)
GFP	Green Fluorescent Protein
GPL	Gradient Path Labelling
GUI	Graphic User Interface
HILO	Highly Inclined and Laminated Optical Sheet
IbpA	Small heat shock protein - IbpA
in vitro	Latin for "within glass"
in vivo	Latin for "within the living"
iPALM	Interferometric Photoactivation Localization Microscopy
JPDAF	Joint Probabilistic Data Association Filter
KLT	Kenade-Lucas-Tomasi

LB	Lysogeny broth
LB	Lysogeny Broth
LBD	Laboratory of Biosystem Dynamics
LSFM	Light Sheet Fluorescence Microscopy
M1 and M2	Manders Coefficients
mCherry	member of the mFruits family of red fluorescent proteins
MHT	Multiple Hypothesis Tracking
Min Proteins	Proteins from the MinCDE system
Morphological images	Images with Cellular Morphological Information
MreB	Cell shape-determining protein - MreB
mRNA	Messenger Ribonucleic acid
NaCl	Sodium chloride
NAP	Nucleoid Associated Proteins
NN	Nearest-Neighbour
OAA	One-Against-All
OAO	One-Against-One
ODE	Ordinary Differential Equation
OpgH	Cell Envelope Biogenesis Glucosyltransferase Enzyme - OpgH
PCA	Principal Component Analysis
PCC	Pearson Correlation Coefficient
PDAF	Probabilistic Data Association Filter
Project SADAC	Project Study of the kinetics of asymmetric disposal of aggregates in cell division and its correlation to functional aging from in vivo measurements, one event at a time
RGB System	Color Space of Red, Green and Blue
RMLR	Regularized Multinomial Logistic Regression
RNA	Ribonucleic acid
RNAp	Ribonucleic acid polymerase
rRNA	Ribosomal Ribonucleic acid
SDCM	Spinning Disk Confocal Microscopy
sfGFP	superfolder Green Fluorescent Protein
SlmA	Nucleoid occlusion factor SlmA
SSA	Stochastic Simulation Algorithm
SVM	Support Vector Machines
TIFF	Tagged Image File Format
TIRF	Total Internal Reflection Fluorescence Microscope
TPM	Two-Photon Microscopy
tRNA	Transfer Ribonucleic acid
YFP	Yellow Flurescent Protein
ZipA	Cell division protein – ZipA

Chapter 1. Introduction

This section introduces to the field in which this research work aims to be developed, specifically the main motivation behind this research work and open questions related to the field of work. In this section, the Main Research Question and the main Hypothesis are presented, together with supplementary research questions and *the Research Methodologies that were implemented and tested to validate the main research Hypothesis. Finally, a succinct description of each chapter that composes the structure of the Dissertation is also provided.*

1.1. Motivation

Recent advances in microscopy imaging technology has allowed the detection of single molecules at the live-cell level, due to biochemical techniques that are able to highlight the targets responsible for gene expression, such as the Deoxyribonucleic acid (DNA), Ribonucleic acid (RNA) and proteins (mainly with the use of fluorescent fusion proteins) [1] but also due to the capability of acquiring multidimensional images with better quality and higher resolution, which have advanced imaging capabilities of single-molecule microscopy [2].

The above-mentioned revolutionary techniques have been especially useful in the detection and tracking of single RNA molecules in *Escherichia coli* (*E. coli*), one of the most studied organisms, by fusing Green Fluorescent Proteins (GFP) with the RNA bacteriophage MS2 coat protein [3]–[5], a technique which have also been used recently by the Laboratory of Biosystem Dynamics (LBD) from Tampere University of Technology to produce time-lapsed microscopy images of *E. coli* cells.

The main objective of the LBD group is to study the processes of segregation and polar retention of cellular aggregates [6] and how the morphological symmetry of those processes can be broken due to different environmental conditions [7], as these asymmetries between sister cells can be indicative of cell aging, as unwanted protein aggregates tend to concentrate at the older pole of the mother cell and accumulation can cause a slower division rate of the daughter cells [8].

These state-of-the-art experimental studies prompted the LBD to start a collaboration with the Computational Intelligence Research (CA3) Group of UNINOVA / FCT-UNL (Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa) resulting in project SADAC (Study of the kinetics of asymmetric disposal of aggregates in cell division and its correlation to functional aging from in vivo measurements, one event at a time), which has been funded by Fundação para a Ciência e a Tecnologia (FCT), with the reference PTDC/BBB-MET/1084/2012, and was one of the driving forces to start this Doctoral Work, also funded by FCT (reference SFRH/BD/88987/2012).

The major components of the SADAC project were the development of image processing techniques (i.e. image registration, cell segmentation, segmentation of cellular components, cell tracking) and the establishment of automated statistical methods to extract information from time series of confocal microscope images. The last objective of this project was to use these developed tools and methods to detect cell divisions and characterize partitioning of aggregates by daughter cells.

One of the most important steps in the development of these computational tools and methods is their validation. Nowadays, most of these tools are still validated by benchmark data of manually annotated images. In high-throughput experiments (with enormous amounts of data),

manual validation is a very time-consuming task, which prompted the development of artificial image generators to create new "gold-standard" images. Such artificial images need to be as close as possible of images acquired in the laboratory, so they should be based on mathematical models of the cell biophysical behaviour and empirical information acquired from experiments.

Most of the developed solutions have isolated applicability, particularly automatic and semi-automated methods, which biases the comparison of segmentation methodologies based on images acquired in different conditions [9]. These comparisons should be done on Contests and open challenges, based on benchmark data (acquired by an independent laboratory or created by artificial image simulators), which prevents abuses of method comparison in the literature [9]. The use of computational modelling to create artificial deformable benchmark images to simulate biological models is an alternative to create a "ground truth" for quantitative evaluation of the image processing algorithms and has been one of the growing trends in microscopy imaging in the last years [10].

Taking into consideration the biophysical modelling of prokaryotic cells (and specifically to bacteria such as *Escherichia coli*), it is necessary to reproduce the cellular spatial and temporal organization by modelling the cell morphology (shape, size and spatial arrangement), cell growth, division, motility and internal functions and structure organization.

Due to the above-mentioned factors, the motivation for this work is divided into three distinct parts: The first is to contribute to the advances of several image processing techniques, especially related to the characterization of the dynamics of cellular processes at the single-cell level. The second part is related to the use of new statistical methods to extract and describe new biophysical models of cellular processes (based for example on Machine Learning). The last part is to be responsible for the creation of new biophysical models, which can be able to reproduce morphological and functional (spatially and temporally) features of the cell (by implementing models coming from the new analysed data in the second step or using the existing mathematical or empirical models from the literature). The ongoing process of the research work, starting with the designing new experiments, which leads to the development of new image processing techniques, new statistical methods and new models and simulators is shown in Figure 1.1.

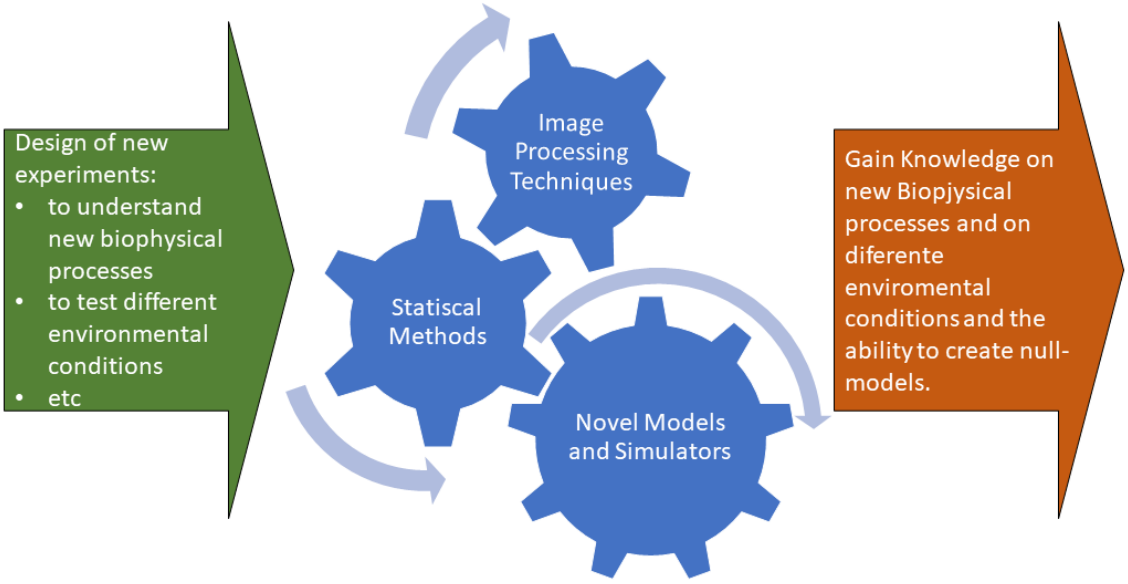


Figure 1.1 – Schematic representation of the ongoing process of the SADAC project *research work.*

1.2. Open Questions in the Area

The development of Artificial Image Generators (that can create reliable gold-standard benchmarks that can be used to validate image processing tools) is one of the emerging fields in Biomedical Engineering. Taking into consideration Generators of microscopy images, these tools have focused on the simulation of morphological features of the cellular biophysical models.

The morphological information can be enough to create multidimensional images but it is not enough to simulate time-lapsed functional images, where important time-dependent processes are present such as cell growth, cell division and cell movement. The main questions in this specific topic are: 'which processes are enough to create a realistic simulation of the cellular spatial and temporal organization?' and 'how to derive new biophysical models?' (if existing models are not able to provide all the necessary data).

One of the open topics in the area is the development of a Generator of functional and morphological time-lapsed simulated images. Mathematical and empirical realistic biophysical models need to be implemented to create simulated images as close as possible of real images acquired in the laboratory. Such a Generator should be made first on simple organisms such as bacteria and then could be fittingly adapted to more complex organisms. When a fully operational Generator of Time-Lapsed Microscopy Images is developed, new questions related to the creation of null models will arise and how they will be able to shed light about some biological processes.

As seen in Figure 1.1, the creation of new biophysical models (e.g. due to simulation of new cellular processes or in simulation of different environmental conditions), can lead to the necessity of developing novel statistical methods and novel image processing techniques which can be used to characterize the dynamics of cellular processes of the new biophysical models and create new simulators of such models.

1.3. Main Research Question

The problematic behind this research work was introduced alongside the emerging challenges in the area that are still open research questions. The main challenge emerges from the necessity of creating a benchmark ("gold-standard") of microscopy images for validation of newly developed image processing tools, as nowadays most benchmark data are produced manually, which is an unfeasible task for high-throughput experiments. A proper system for creating such a benchmark is to use simulated data, using realistic mathematical and empirical cell models, which need to be thoroughly studied and implemented into the simulator.

These models should be able to reproduce time-lapsed experiments by simulating time-dependent processes such as cell growth, cell division, and cell movement. In the initial simulation framework, the main focus will be directed towards bacterial cells, more specifically using E. coli cells models. Simulation of temporal and spatial modelling of external factors/stress conditions can also be done to produce even more realistic results. Simulation of different acquisition systems should also be done to generate the unique features of morphological and functional microscopy images.

After validation of the image generation tool, it is possible to begin the validation of image processing tools and expand the image generation to other bacterial models or to simple cell

organisms such as yeast. Other studies could be made by developing null-models that can assist in research about biological processes, such as cellular aging.

From the above information, the main research question is subsequently written:

How to design a toolbox capable of simulating models and reproducing realistic morphological and functional experiments of bacterial time-lapsed microscopy images?

In addition, a second research questions is proposed with the aim of better solidifying the main research question, related to the design of an image processing toolbox capable of extracting information about cellular processes in different environmental conditions, which can be used to create the new models that are deployed in the simulation toolbox:

Which models of biological processes need to be extracted using an Image processing toolbox, in order to create a realistic simulation of the cell spatial and temporal organization?

1.4. Hypothesis

Following the Main Research Question, a main Hypothesis was devised:

An Artificial Image Generator capable of replicating realistic bacterial time-lapsed experiments can be developed if the produced images consider the characteristics of the different image acquisitions systems and environmental conditions in the laboratory and by reproduce the spatial and temporal cell morphological and functional features.

The main Hypothesis can be solidified by devising the secondary Hypothesis, which responds to the secondary research questions.

In conjunction with the Artificial Image Generator, a novel Image Processing Toolbox can be developed in order to characterize the dynamics of cellular processes (division, growth, motility and gene expression), which can then be used to create novel biophysical models that can be introduced in the Simulator, namely they can establish a correlation between these processes and cellular aging.

1.5. Research Methodologies

To answer the research questions, the biological processes and environmental conditions that need to be included in the image generator must be outlined. The first step is to search the literature for the state-of-the-art mathematical and empirical models of bacterial cell modelling. These studies should include the temporal and spatial features of bacterial growth and division (which is linked to its morphological features of cell size and shape).

It is also important to study how these processes and cell motility are connected to the spatial arrangement into clusters. If some of these functions and connections cannot be described mathematically, it should be possible to use machine learning techniques to reproduce the empirical data. The external environmental conditions should also be considered, such as the bacterial response to external factors such as temperature (heat-shock, cold-shock), pH stress, oxidative stress, nutritional stress or even exposure to antibiotics.

The extraction of data from internal cellular structures should be done initially by applying existing methods and finally by complementing it by the development of new image processing techniques if the existing methods don't provide satisfactory results.

To test and validate the research hypothesis, several groups of microscopy and biotechnology experts have been approached to provide manually segmented benchmarks which will be used as a gold-standard to validate the image processing techniques. Secondly, these groups will be asked to provide a qualitative analysis of the generated simulated images, compared to the ones acquired in the laboratory. Quantitative analysis will be done by direct comparison with real *E. coli* images acquired in various image acquisitions systems and various environmental conditions. Then it is possible to compare the simulated distributions of the model parameters indicators such as cell sizes and shapes (distributions of bacterial spatial organization), motility velocity, division and growth rates (distributions of bacterial temporal organization), and the production and localization of subcellular structures (fluorescent proteins, nucleoid, etc).

The main objective of developing this image generator is to create time-lapsed microscopy image benchmarks that can be used to validate the newly developed image processing tools. There can be other applications to the image generator such as creating null-model that can be used to investigate how the removal or the insertion of features can affect the bacterial behaviour (e.g. to study the effects of the nucleoid by removing it from the cells or changing the bacterial size distribution) or sampling some parameters (e.g. evaluate the effects of adjusting the growth rates to unrealistic values).

1.6. Structure of the Dissertation

This Dissertation is structured into 7 chapters. This first chapter served as an Introduction to the present work and its main motivation, while giving emphasis to the open research questions in the area. Two main research questions were presented, alongside with the two main Hypothesises. An introduction to the main Research Methodologies was also given and finally this chapter ends with the description of the Dissertation structure.

The second chapter introduces the main biological and bioinformatics topics that will be the foundation of the research work, which can benefit from the implementation of novel Electrical and Computer Engineering techniques. The third chapter contributes for a comprehensive summary of the state of the art, namely focusing on the available image processing techniques and the simulation of biophysical cell models. In the fourth chapter, implementation and development of the Image Processing Framework is detailed along with the development of the Image Generator Framework and all the necessary modelling features.

The fifth chapter provides a high-level description of the laboratory experiments that were used to validate the computational frameworks. In the sixth chapter, a compilation of all the results is provided, which can be used to validate the implementation of the toolboxes and thus validate the hypothesis. The closing chapter presents the main conclusions of the work, while also providing future development perspectives and directions.

Chapter 2. Background Information

This section gives a brief overview of the multi-disciplinary nature of this work, which combines the knowledge from biological and bioinformatics studies and the application of techniques related to Electrical and Computer Engineering into those studies. Useful information about the spatial and temporal organization of bacteria is provided, namely regarding the models of cellular processes such as cell growth, division and motility. Information on bacterial cell morphology is also presented, such as its shape, size and spatial arrangement and on the internal cell functions and some of the important structures that are found inside bacterial cells.

2.1. Cell Modelling – Spatial and Temporal Organization of Bacteria

In order to create realistic models of bacterial cell behaviour, it is necessary to understand the available information on bacterial spatial and temporal organization, namely the cell shape and size, kinetic models of cell motility, division and growth and models of location and functionality of cellular structures [11].

As aforementioned, *E. coli* is potentially the most studied organism, making it the basis for an impressive number of scientific breakthroughs, even in the medical field. *E. coli* is an organism that lives symbiotically in the intestines of other organism, although some strains may cause gut diseases and sepsis [12].

E. coli also has significant information in orthologous genes, which are present in various organisms such as humans, animals, plants and other bacteria. This suggests that this is an important model organism to be studied and will be kept being adopted in various experimental laboratories [12], making the *E. coli* K-12 strain and B strain the perfect candidates to study cellular structures and cellular processes, such as cell growth and division through computational and mathematical modelling of spatial and temporal bacterial organization [11], [13] along with the advances in microscopy and sequencing techniques.

Previous efforts to tackle the *E. coli* cell modelling problem have been extensively reviewed [13] and include the creation of a common language to represent biological models, namely the Systems Biology Markup Language [14], the development of numerous mathematical and empirical models found in the literature will have to be researched along with accessing specialized information stored in databases, such as the International *E.coli* Alliance Database Portal [15] or the advances in the computational cell modelling. The next sub-chapters focus on these topics, with the *E. coli* species as the pivotal example, but also making analogies with other bacterial species.

2.2. Cell Morphology - Shape, Size and Spatial Organization

Bacterial cells can be classified by their shape and by spatial organization. As can be observed in Figure 2.1, *E. coli* has a rod-shape (bacillus), while other bacteria have shown a vast diversity of shapes, such as spherical (coccus), intermediate shapes (coccobacillus) or curved/corkscrew shapes (spirochete, spirillum and vibrio), each of them with its specific purpose [16]–[18].

Bacteria can also have a wide range of cell sizes (volumes that range from 0.02 to 400 μm^3), where even a vast variability can be observed within the same species [19], [20]. These variations can be explained due to cell adaptation to external factors, such as lack of nutrients leading to starvation, situations of extreme temperatures (low and high) or of extreme dryness [20].

It has been shown that the lower bound for cell size is maintained by the cellular mechanisms that cope and adapt to the environment, while the higher bound is normally limited by diffusion of nutrients along the cell. For example, studies using *E. coli* as a model organism have shown how temperatures between 22 °C and 42 °C affect two *E. coli* strains in different growth media [21].

In terms of spatial arrangement, bacteria can be organized in single forms or be grouped in pairs (diplo prefix), in chains (strepto prefix), as can be seen in Figure 2.1. Cocci bacteria can also organize in groups of 4 (tetrad), 8, 16 or 32 (sarcinae) or in grape-like clusters (staphylo prefix). Bacilli bacteria can organize in palisade structures (side by side) or can be in unstructured spatial clusters [18]. An example of different *E. coli* spatial arrangements is shown in Figure 2.1 (namely in (2-A) single bacillus; (2-B) diplobacilli; (2-C) streptobacilli and (2-D) palisade).

A typical bacterial cell envelop is mainly composed by a cytoplasmic membrane and peptidoglycan (also known as murein) cell wall. As can be seen in Figure 2.2-A, bacteria can also be divided in two groups regarding a fundamental difference in the cell envelope: Gram-negative and Gram-positive bacteria. In the first group (which is the case of *E. coli*) a bacterial outer membrane is also present (with intercalating pore-forming proteins, called porins), with lipopolysaccharides connected to the exterior of that outer wall.

The interior of the outer wall is then connected to a very thin murein wall by a lipoprotein [22]. On the other hand, in the second group (which is the case of human pathogenic bacterium Streptococcus pneumonia), the cell envelope consists of a very thick murein wall (sometimes more than 10 times thicker than the first group) with teichoic acids spread across the murein. The Gram-positive bacteria also have a cytoplasmic membrane as the Gram-negative [22].

The shape in bacterial cells (see Figure 2.1 and Figure 2.2-B) is maintained and determined by the way murein is incorporated during cellular elongation, especially in rod-shape organisms, such as *E. coli* [23] and B. subtilis [24], as the murein is the main cell wall structure that supports the stress from the outside [25], as computational physical models have been developed to study how defects in the murein can affect *E. coli* shape (and the shape robustness to murein damage) and how different murein defect patterns can build bacterial shape patterns such as curved rods and spirochetes [26].

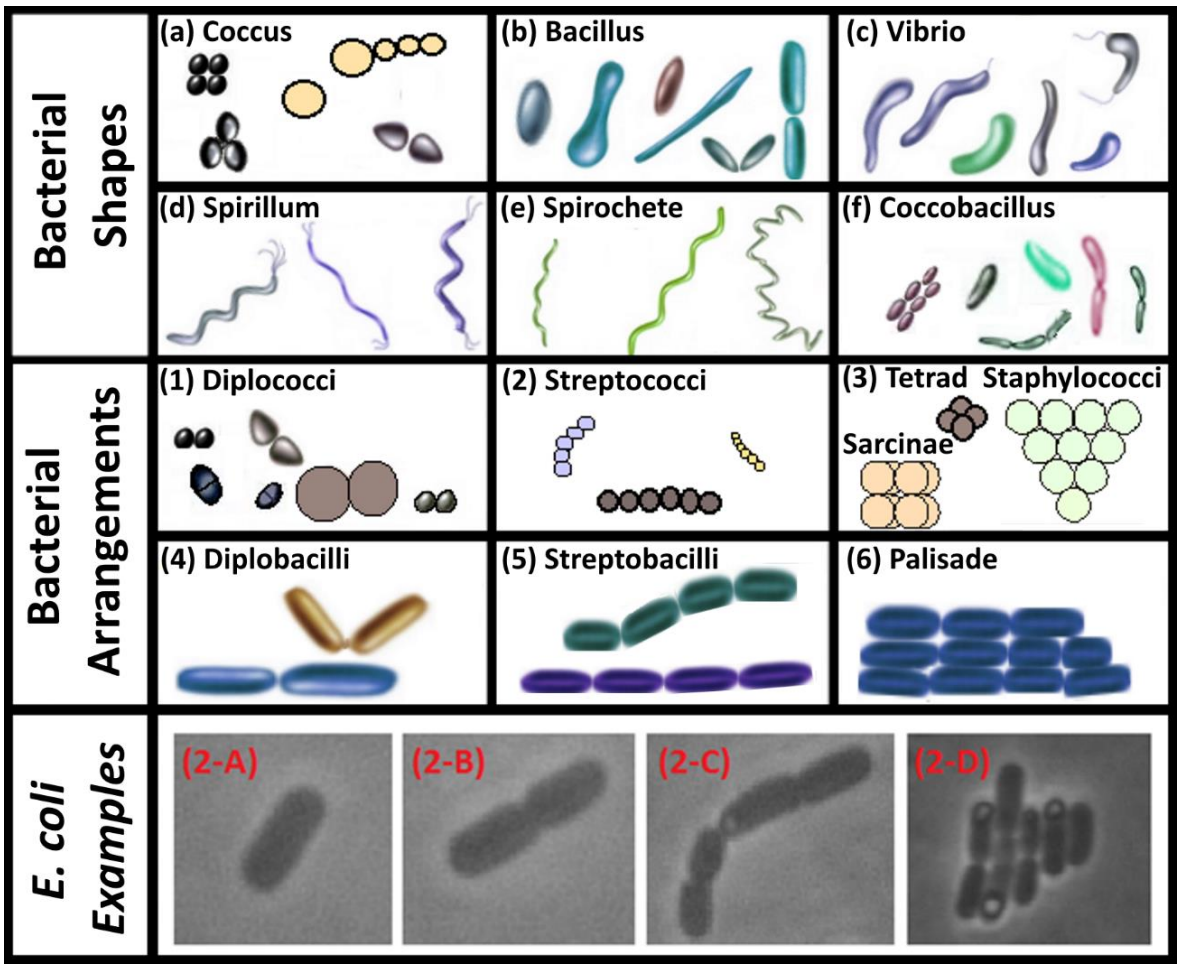


Figure 2.1 – Bacterial Cell Morphologies. Bacterial Shapes (a, b, c, d, e and f) and Bacterial Spatial Arrangements (1, 2, 3, 4, 5 and 6). E. coli examples: (2-A) Single bacillus; (2-B) diplobacilli; (2-C) streptobacilli; (2-D) palisade. The visualized cells are from the *E. coli* BW25993 strain (lacIq hsdR514 ΔaraBADAH33 ΔrhaBADLD78) [27] and were acquired with a Nikon Eclipse (Ti-E, Nikon) inverted microscope with a 100x Apochromat TIRF (Total internal reflection fluorescence - 1.49 NA, oil) objective, and an external Phase-Contrast system and DS-Fi2 CCD (Charge-coupled device) camera (Nikon) at the Laboratory of Biosystem Dynamics at 37 °C.

Along with the cell wall, other cytoskeleton proteins are associated with bacterial shape, such as FtsZ (Filamenting temperature sensitive), MreB and crescentin (with similar activities as tubulin and actin in eukaryotic cells) [16], [24]. These proteins influence how the cell wall is created and hydrolysed during cell growth and division, respectively, influencing their sizes, shapes and spatial and temporal organization [24]. The role of cell size in bacterial growth has been discussed alongside a model for cell growth control in different nutrients, as observed in Figure 2.2-C [28].

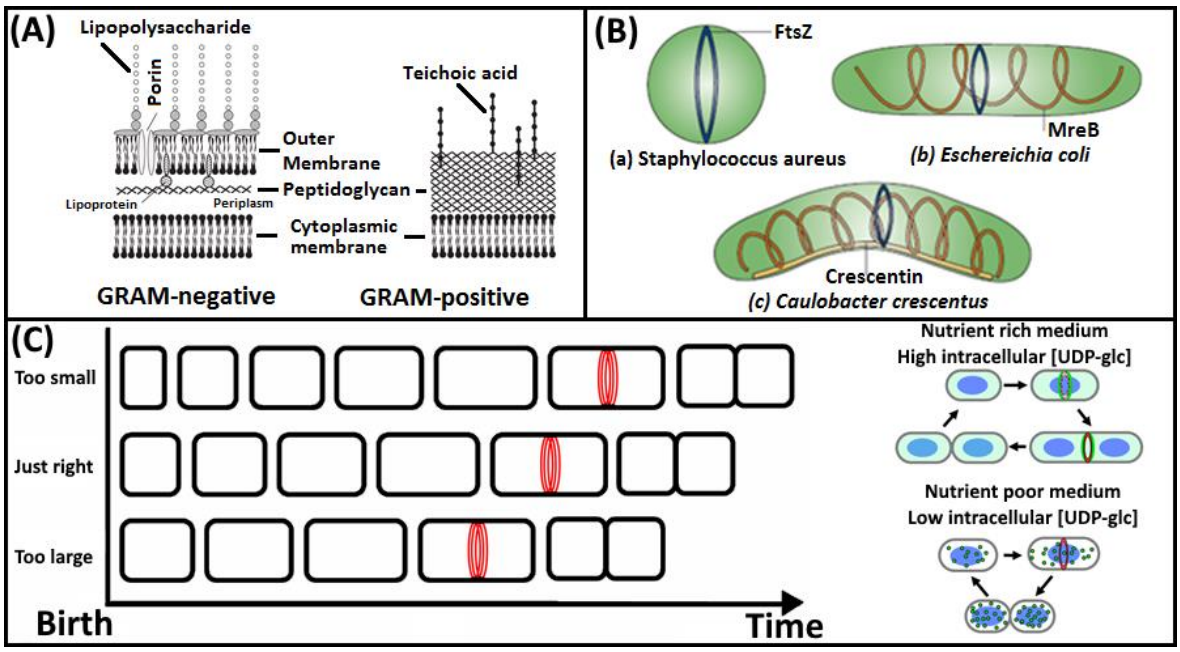


Figure 2.2 – Representation of the cell walls and cellular growth in *E. coli*. (A) Structure of the cell envelope of Gram-negative (left) and Gram-positive (right) bacteria and their differences. Adapted from [22]. (B) Bacteria cell shapes and cytoskeletal elements. Adapted from [16]. (C) Cell size control by growth and division processes and how they can cope with a poor nutrient medium. Adapted from [28]

Bacterial morphology is closely related to important mechanisms to the bacterial cell activity, such as cell growth/elongation and cell division, so it is also important to understand how these mechanisms are regulated in the time and space [16], [24], [29].

2.2.1. Cell Growth and Division

Bacterial cell cycle is normally divided in three stages, specifically the period between its "birth" and the initiation of DNA replication (the biological process of assembling two identical replicas of DNA from one original DNA molecule), the replication period where the cell increases its mass and size (Cell Growth) and finally the binary fission process into two new daughter cells (Cell Division), which will be repeated over the next generations [30], as shown in Figure 2.3-1.

Cell growth in spherical cells is done through the creation of new murein polymer at the division septum, in the middle of the cell which then leads to a division event, as cell elongation does not occur in this type of cells (as observed in Figure 2.3-2-A), where two daughter cells are created [16]. In other bacterial cells, elongation trough the creation of new murein polymer leads to cell growth, as murein can inserted in the sidewalls at the middle of the cell (see Figure 2.3-2-B) or at the poles (see Figure 2.3-2-C).

Each of those processes (Division and Elongation) have their own protein and enzymatic apparatus, which work in specific places of the cell wall [16], [24], as can be observed in Figure 2.3-3. The FtsZ cytoskeleton protein (see Figure 2.3-3-A) along with various other proteins create the division septum at the middle of the cell (as two proteins MinC and SlmA that are present in the rest of the cell, inhibit the assembly of the FtsZ ring required for division [31]. The MreB cytoskeleton system proteins (see Figure 2.3-3-B) are mainly associated with the elongation apparatus, as aforementioned [31].

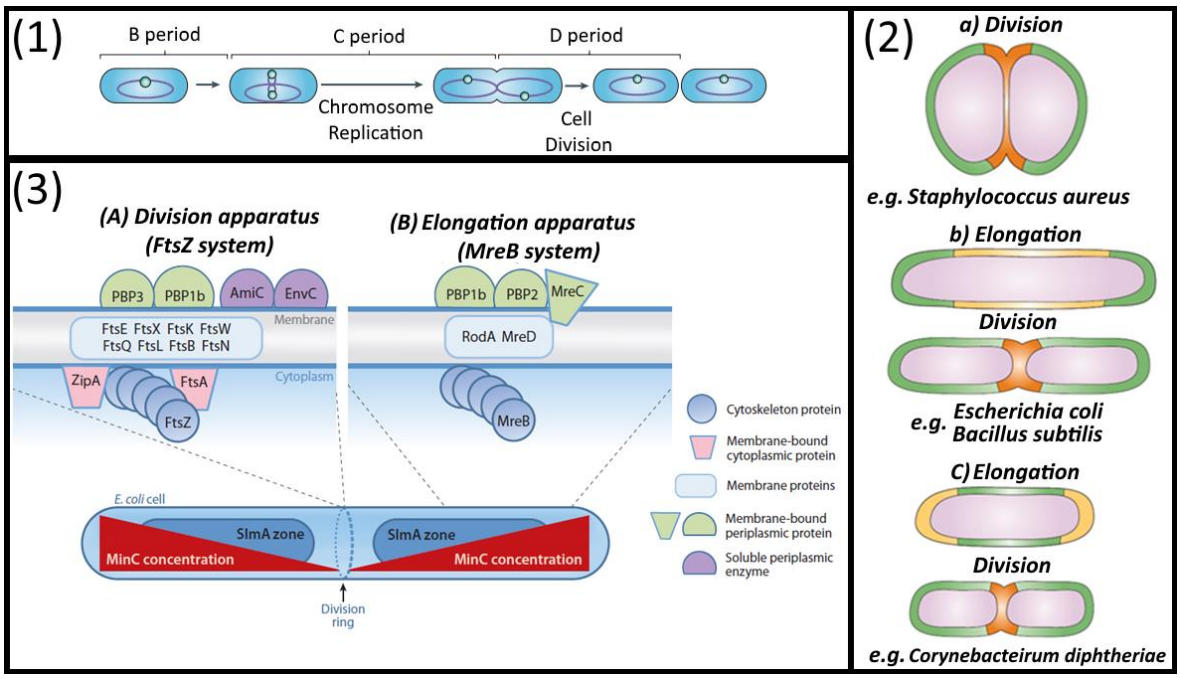


Figure 2.3 – Apparatus and Mechanisms of Cell growth and Cell Division. (1) Bacterial Cell Cycle. Adapted from [30]. (2) Division and Elongation processes in different bacterial organism. Adapted from [16]. (3) Protein apparatus for the (A) Division and (B) Elongation processes. Adapted from [31].

Mathematical models of the temporal and spatial organization of the bacterial cell cycle [32]–[34] are required to model cell elongation and cell division. These mathematical models arise from numerous experimental studies at the single cell level, especially using *E. coli* as a model organism [11].

A mathematical model showed how the FtsZ ring can act as force generator to predict the contraction speed and force and how the cell shape arises, as shown in Figure 2.4-1 [32]. Another model showed how the chromosome can be segregated during cell elongation and cell division, implying an influence of the MreB cytoskeleton protein that is involved in the bacterial elongation apparatus. This process modifies the membrane pressure and influences the DNA segregation, as shown in Figure 2.4-2-A [33]. It is important to note that they use cell shape assumptions where the width of the cell doesn't change over time, and that cells do not deform, as shown in Figure 2.4-2-A and Figure 2.4-2-B. Another model of cell growth, based on *Bacillus subtilis,* showed that the rate of cell division can be dependent not only on cell size but also on the its age [34].

Bacterial growth as a colony can also be dependent on the capability to move in the direction of more favourable conditions, which at its basic form is normally associated with Brownian random movement or active movement towards a specific gradient, e.g. chemicals (chemotaxis) and temperature (thermotaxis) [35], but can also be influenced by the availability of nutrients [36].

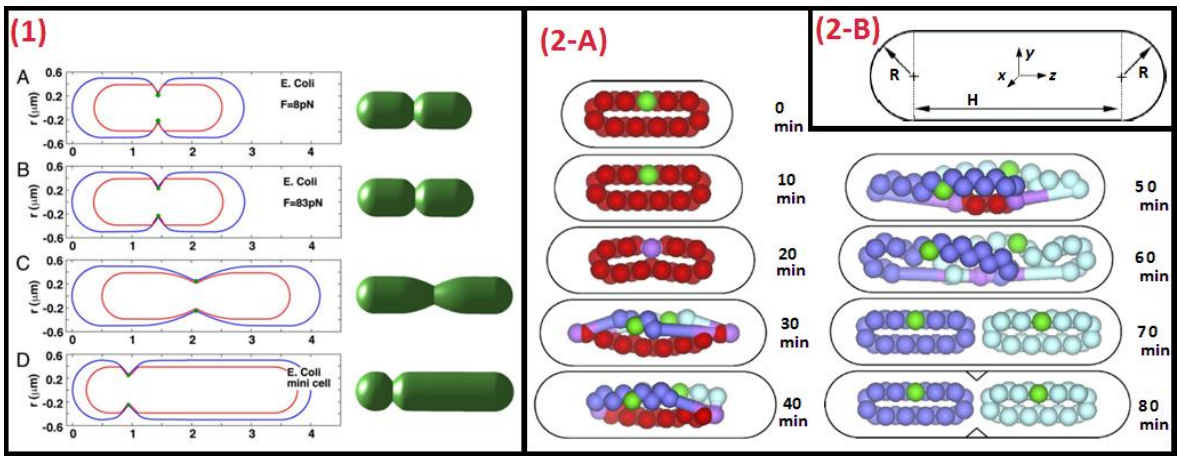


Figure 2.4 – Mathematical and computational models of cell shape and growth. (1) E. coli shapes due to FtsZ force generation. Adapted from [32] (Copyright (2007) National Academy of Sciences, U.S.A.). (2-A) Models of DNA Replication and Segregation. (2-B) Computational model of the *E. coli* rod shape. Adapted from [33].

2.2.2. Cell Motility

Each individual bacteria cell performs a random walk in the absence of external factors, but when these factors are present, active bacterial cell movement is activated, e.g., towards nutrient sources or by moving away from certain toxins or stress conditions [35]. The random walk is characterized by a smooth swim/run in a determined direction for a few seconds followed by a tumble (a change in direction, which last tenths of a second [37]

Bacterial cells such as *E. coli* have developed external structures for motility purposes (the flagella) that provides the ability to propel themselves by acting as cellular motors [37], [38]. The active motility mechanism is controlled by a distinct biochemical network (see in Figure 2.5-A) that transmits information from the extracellular environment, gathered by the membrane receptors to the flagella [39].

While each cell tends to behave independently, bacterial populations also display collective behaviour, as bacteria are shown to be spatially arranged in various configurations (see Figure 2.1). Bacterial cells can even organize in large clusters, due to its high rate of division, having specific macroscopic motility properties, which have been studied using individual *E. coli* cells were tracked inside those clusters, using a fluorescence microscopy [40]. The boundaries of the cluster is maintained by supressing the direction change (tumble) of individual bacteria in the centre of the cluster, as observed in Figure 2.5-B, while it is restored for cells at the edge of the cluster [40]. These experimental findings were confirmed by using a computational simulation, which confirmed that the tumble rate and the cluster morphology are determined by the sensory memory of cells [40]. As previously stated, to develop computational models of these mechanisms, it is essential to have mathematical models or empirical models based on empirical observations that can describe these mechanisms.

In the case of cell motility, such mathematical models have been extensively studied and reviewed [38]. These models incorporate how an individual bacterial cell behaviour affect the population by interacting with other cell and they also incorporate how bacteria interact with the environment [38].

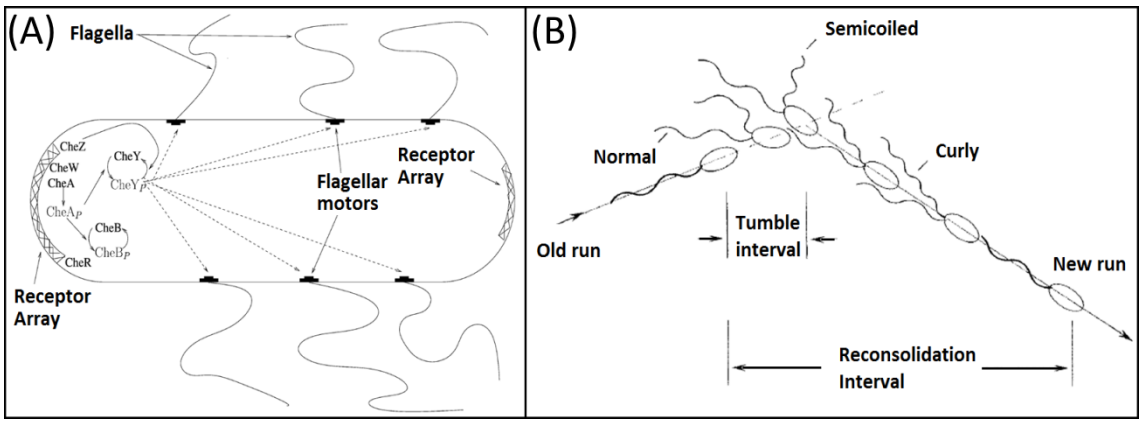


Figure 2.5 – Motility behaviour in *E. coli* cells. (A) Motility biochemical network and anatomy of the flagellar systems. Adapted from [38]. (B) Run and Tumble. Adapted from [37].

Unlike bacteria motility, the mechanism by which eukaryotic cells can migrate is still a problem that isn't totally solved, especially as the mathematical models of cell migration are still being envisioned [41], which are essential for the simulation tools of temporal cellular activity of eukaryotic cells.

All the previously described morphological processes are controlled by functional processes that occur inside the bacteria, which can be studied by observation of specific gene expression products such as Ribonucleic acids (RNAs) and proteins, which can be labelled with fluorescent probes.

2.2.3. Gene Expression

Gene expression is the process of synthesizing a functional gene product (e.g. proteins), by using the specific gene information. This process starts with Transcription, where a Messenger RNA (mRNA) transcript is synthesized by copying the genetic information contained in a determined region of the DNA, which is executed by the RNA polymerase (RNAp) and transcription factors, followed by the translation of the RNA transcript into a protein, which is executed by the ribosome. These two mechanisms are part of the general steps in the central dogma of molecular biology [42], which also includes the DNA replication, which is carried a by complex group of proteins called the replisome and occurs before the division event (see Figure 2.6), as described in Section 2.2.1. Other special mechanisms include RNA replication, Reverse Transcription from RNA to DNA and direct translation from DNA to proteins, as they occur specially in virus or in special conditions (see Figure 2.6).

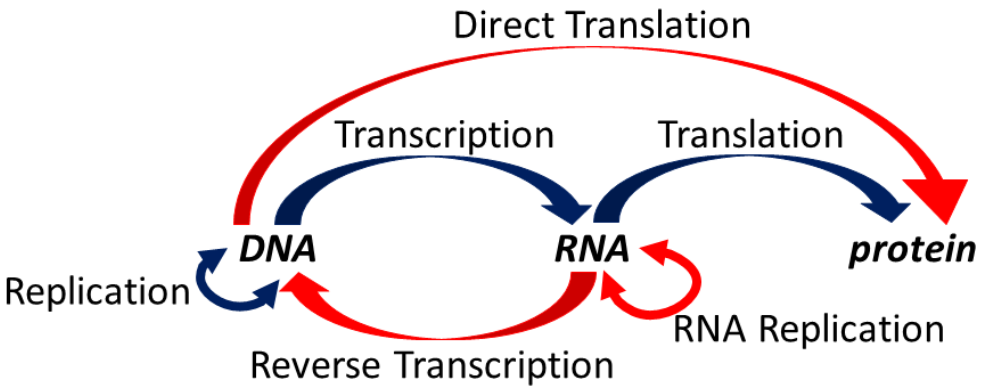


Figure 2.6 – Information flow in biological systems. Blue arrows represent the general mechanisms stated by the central dogma of molecular biology and red arrows represent special mechanisms.

The two DNA strands are composed of simpler monomer units called nucleotides [43]. Each nucleotide is composed by a sugar called deoxyribose, a phosphate group and finally one of four nitrogen-containing nucleobases (cytosine, guanine, adenine or thymine, respectively abbreviated as C, G, A, T) [43]. Each nucleotide is joined together by a chain of covalent bonds between the sugar of previous nucleotide and the phosphate group of the following nucleotide, which results in an alternating sugar-phosphate backbone [43]. Each of the opposite DNA strands are bound together according to specific pairing rules (A always binds with T and C with G) [43]. The RNA, unlike the DNA, is normally found in a single-strand conformation (RNA from virus can be found in double-strands), and its information is stored with the same nitrogenous bases as the DNA, but replacing the thymine with uracil (represented by letter U) [43].

If a cell needs to produce a certain protein, it activates the gene that expresses that protein, which is subsequently transcribed by the RNAp to produce a mRNA molecule, with the help of a specific transfer RNA which has a distinctive folded structure with three hairpin loops, with one of these hairpin loops containing a sequence called the anticodon and being able to recognize and decode a mRNA codon, which is a specific sequence of 3 nucleotides [43].

Since each tRNA (Transfer RNA) has a corresponding amino acid (out of the 22 genetically encoded amino acids), it will transfer that amino acid to the peptide chain that is being translated based on the specific codon of the mRNA [43]. This process is controlled by the ribosome (with the help of the so called rRNA or ribosomal RNA, which moves along the mRNA molecule and binds to tRNAs and various accessory molecules necessary for protein synthesis.), which continues to decode the mRNA molecule, until a STOP codon is translated, forming an amino acid sequence, which is the forming a polypeptide, which can be further arranged to form a functional Protein [43]. If, for example, the coding pattern on the coding strand of the DNA is TAC (Thymine, Adenine, Cytosine), based on the pairing rules the opposite strand is ATG (Adenine, Thymine, Guanine) and the messenger RNA codon is UAC (Uracil, Adenine, Cytosine) and the transfer RNA anti-codon is AUG (Adenine, Uracil, Guanine), which is then finally translated into a Tyrosine Amino Acid.

This research work focuses especially in the first steps of the transcription mechanism, as in prokaryotes this is the mechanism where gene regulation takes place [44]. The main structural components at the DNA level are the promoter, operator(s) site and the actual structural gene(s) [43].

Transcription starts with the promoter search process, where the RNAp localizes the promoter (a specific location in the DNA) by diffusing over the non-specific areas of the DNA. In bacterial cells, the promoters have specific hexameric motifs [45], which are special sequences of 6 nucleotides normally centred close to the -10 and -35 positions relative to the Transcription Starting Site (TSS), as analysed in *E. coli* cells [46]. When a polypeptide (called σ factor) binds to the RNAp core enzyme, it forms a holoenzyme, reducing the affinity of the RNAp for nonspecific DNA and increasing the affinity to the hexameric motifs [47]. A third component of the bacterial promoters localized upstream of the -35 hexamer was identified as the “UP element” [48] and is able to increase transcription due to the interaction with the α subunits present in the RNAp [49] (as detailed in Section 2.3.3).

Regarding the cellular modelling of gene expression, one of the first models considered transcription to be an instantaneous process [50]. This initial approximation was corrected in following models based on E. coli cells (implemented by the LBD group), modelling transcription at the single nucleotide level using time delays, based on a Gaussian distribution [51], [52]. These models already included events in elongation such as transcriptional pauses, error correction, arrests, premature

termination and collisions between elongating RNAps, with following models coupling both the transcription and translation events [53].

As mentioned, from all the major mechanisms in transcription (initiation, promoter escape, elongation and termination), this research work focuses on the dynamics and models of transcription initiation until promoter escape. Transcription initiation was first detailed at the nucleotide level in a subsequent model [54], specifically created to simulate closely spaced promoters in different configurations (divergent and convergent).

This model [54] included the observed rate-limiting steps of transcription initiation [55] were separated (instead of modelled as a single delayed event) to account for diffusion process during promoter search [56], [57]. The binding of the RNAp to the σ factor was not detailed at the single molecule level, but this step was included in the promoter search mechanism [54].

Following the localization of the promoter, the RNAp forms a closed complex (where the RNAp and the σ-factor binds to the promoter region of the DNA), which is followed by the isomerization of the closed complex, which is proceeded by the open complex formation (where the DNA strand are separated to form an unwounded DNA) [58].

Before starting the elongation process, the model also considered the abortive initiation, where the RNAp needs to accumulate energy to be able to escape the promoter and start the elongation process [59], [60]. This model [54] did not explicit elongation events, except for the collision between elongating RNAps and RNAps at the transcription start site, which are named as 'sitting ducks' [61]. Finally, a repression mechanism was modelled as steric occlusion due to a binding competition to the DNA molecule, between the RNAp and the repressor molecules, preventing the RNAp to start transcription [62].

In prokaryotes, as opposed to eukaryotes, the produced RNA doesn't need to be processed or transported to other areas of the cell, which means that the RNA is ready for Translation, the process in which ribosomes present in the cell produces a specific sequence of amino acids using the information stored from the previously transcribed messenger RNA (mRNA), which is processed 3 nucleotides at a time (codons) to produce a specific amino acid [63].

The simulation of these processes requires the explicit modelling of the interaction between the chemical species, which in the case of bacterial gene expression both the reactants and the products of the reactions are present in the cell in small numbers [64], which in turn makes the deterministic modelling problematic, as it cannot explain the noisy and stochastic nature of the gene expression and regulation mechanisms [65].

The Stochastic Simulation Algorithm (SSA, as detailed in Section 3.2.1) was created to solve the problem as instead of calculating the exact moment that all collisions take place and track each molecule in the cell space, since in contrast to the deterministic approach it estimates the distribution probability $P(x, t|x_0, t_0)$ that the system will evolve into x at time t based on an initial system of x_0 and t_0 respectively. This approach is the underlying foundation for development of the first version of the Stochastic Genetic Networks Simulator (SGN Sim) [66] and of the second version [67], allowing the implementation and simulation of the gene expression and gene regulation models and even coupling those effects with cell division at the LBD group (e.g. [53], [68]–[73]).

A new method was also developed at the LBD that can dissect the dissect the in vivo kinetics of transcription initiation in live E. coli cells [74] which is based on techniques previously implemented in vitro studies [55], [58], [75].

By changing the concentrations of free RNAp, this methodology allows the estimation of the fraction of time spent prior and after the commitment to the open complex formation, as all the steps prior to this commitment are affected by the concentration of RNAp, while the steps after this commitment are independent of the RNAp concentration [74]. This estimation is possible by using a linear extrapolation between the inverse of the RNAp concentration and the inverse of RNA production rates, which have been shown to be valid within a determined range of media richness [74], [76], which allows one to extrapolate the inverse of the RNA production rate when RNAp is considered to be infinite, which correspond to the total time spent by all the steps that occur after the commitment to open complex formation (due to the independence from the RNAp concentration [74].

This methodology [74] has also been recently used, with the $LacO_3O_1$ promoter, to study the time spent in a repressed state by changing the concentration of inducers (instead of RNAp) and extrapolating the RNA production rate when infinite inducer concentration is considered [77] and inducer intake kinetics as a function of temperature, with the Lac-ara-1 promoter [78].

In this research work, the aforementioned methodology [74] has been used to dissect the transcription initiation kinetics as a function of temperature, both in with the T7 Phi10 (Φ10) promoter and with the $LacO_3O_1$ promoter, integrated both in a plasmid and in the chromosome as detailed in Section 4.3.3.

The bacterial processes described in this Section, are governed by several molecular apparatuses, which have specific coordinated activities and spatial organizations inside the cell. The following Section presents the details of the structures of interest for this research work and how to visualize these structures with microscopy imaging.

2.3. Cellular Structures and Molecules

The observation of the internal cellular structures at the single live-cell level, such as the cell membrane, genetic material (DNA and RNA), cytoskeleton proteins, and other organelles allows the integration of data coming from studying the spatial and temporal organization of these molecules with the dynamics of the processes that they control [1], [79], [80]. This is especially relevant when doing *in vivo* studies, as the internal environment of the cell has drastic differences from the in vitro studies, where molecules are normally in a homogenous and well-mixed state [81].

Each of these molecules can have a specific spatial distribution or even change its spatial distribution along the cell cycle. In E. coli cells, molecules related to transcription [81] are normally associated with cytoplasmic, clustered, pole, membrane, specific, helical, and nucleoid spatial distributions (see Figure 2.7-A). Some molecules, especially the ones from the division apparatus (see Figure 2.3-3-B) tend to organize at the mid-cell (see Figure 2.7-B) as they are involved in the creation of the division ring. From all the proteins involved in the division process, the FtsZ and the MinD proteins are highlighted in this research work, as these were used to study the performance of the structure segmentation algorithms (see Section 6.1).

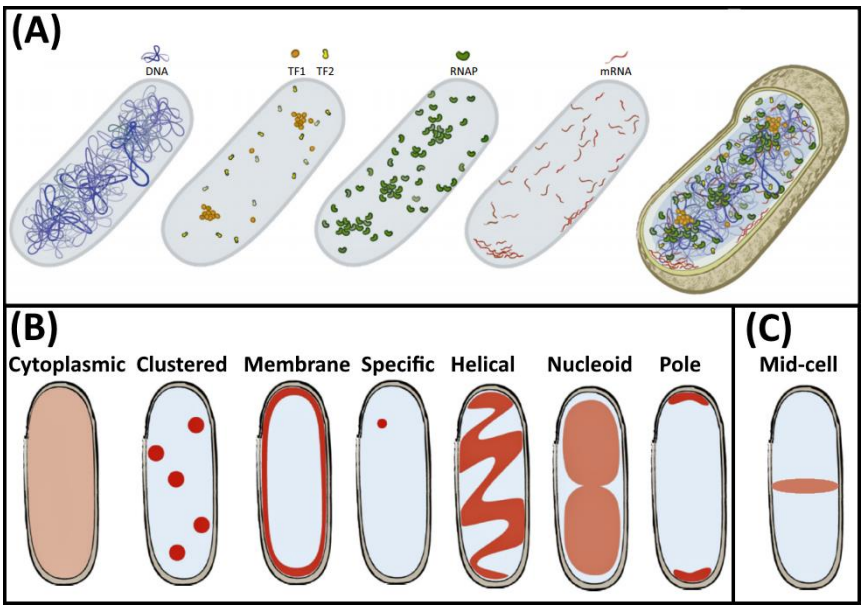


Figure 2.7 – Molecular spatial distribution inside *E. coli* cells. (A) Inhomogeneous spatial organization of transcription molecules in an E. coli cell, with a visual representation of Genes (DNA), transcription factors (TF1 and TF2), RNA polymerase (RNAP), and mRNAs Adapted from [81] (B) Spatial distribution patterns of transcription molecules. Adapted from [81]. (C) Distribution of molecules within the mid-cell, namely responsible for the division apparatus, such as the FtsZ, not directly involved in transcription. Drawn using [81] as a template.

To visualize the cellular structures of interest, at the single live-cell level, there are two strategies which were able to progress several biological studies (and have been intensively used in the past decades): the use of genetically modified fluorescent proteins able to form aggregates with the structures of interest and the use of fluorescent dyes capable of staining the structures of interest [82], [83].

The selection of fluorescent proteins, dyes, media and solvents is mostly associated with both the compatibility to the structure of interest, but also based on the absorption and emission of light outside of the spectrum of the organism auto-fluorescence (a phenomenon that occurs naturally and that leads to the detection of background fluorescence) [84], [85].

Although most of the structures used in this research work are based on the use of fluorescent proteins, some fluorescent dyes are also used (e.g. DAPI to dye the Nucleoid). The following Sub-Sections provide a brief description of the structures of interest and of the specific fluorescent proteins and dyes that are used here to visualize these structures, including several examples of images taken by the LBD group and that were used to test the Image Processing Techniques that were developed for this work.

2.3.1. Fluorescent Proteins

The use of genetically modified fluorescent proteins initiated with the use of the jellyfish *Aequorea victoria*, when the green fluorescent protein (GFP) was discovered [86]. This protein was only applied *in vivo* [87] when gene that allowed the fusion between the fluorescent labels and other structures of interests was discovered [88], which the allows highlighting of such structures.

The impact of fluorescent protein labelling on the structure, function and stability of the fused aggregate has been recently discussed and reviewed [83], [89]. Based on general guidelines [83], one can select the fluorescent protein based on the fusion sites with the structure of interest and using a range of excitation and emission wavelengths that span over the entire visible spectrum (from violet

to red), allowing the use of different optical methods to provide quantitative measurements of the structures of interests [90].

Photoactivatable fluorescent proteins are a distinct class of fluorescent proteins, as their properties can be switch on or off (reversibly or irreversibly) using a laser of a specific wavelength [91]. The use of PAFPs allows the precise temporal and spatial activation and consequent visualization of certain molecules, while non-activated molecules remain invisible [91].

The use of these fluorescent probes has pushed the development of new image acquisition and image processing tools, as detailed in Sections 2.4 and 3.1 respectively, that are able to handle the enormous amounts of data that are generated by fluorescence imaging based laboratories [92] and have been pushing the limits of structure visualization, e.g. with super-resolution microscopy and temporal analysis of cellular processes in live cells [91]. The newly developed acquisition systems have also allowed the simultaneous study of different structures of interest, by using distinct lasers, filters and fluorescent proteins [93], [94].

In this research work, several fluorescent proteins are used, such as GFP (e.g. to be fused with RNA, FtsZ and MinD proteins), BFP (Blue Fusion Protein, fused with the Nucleoid) and mCherry (also fused with the Nucleoid). Figure 2.8 shows *E. coli* cells under the expression of the MS2-GFP fused protein reporter with 0.4% of L-Arabinose (Sigma-Aldrich, USA) at 30 °C, without induction with IPTG (leading to no observable MS2-GFP-RNA clusters, contrary to Figure 2.13, where 1000 µM IPTG was used and MS2-GFP-RNA clusters are observable). As expected, each cell has a significant stochastic variability in the background fluorescence, due to the GFP maturation process [95].

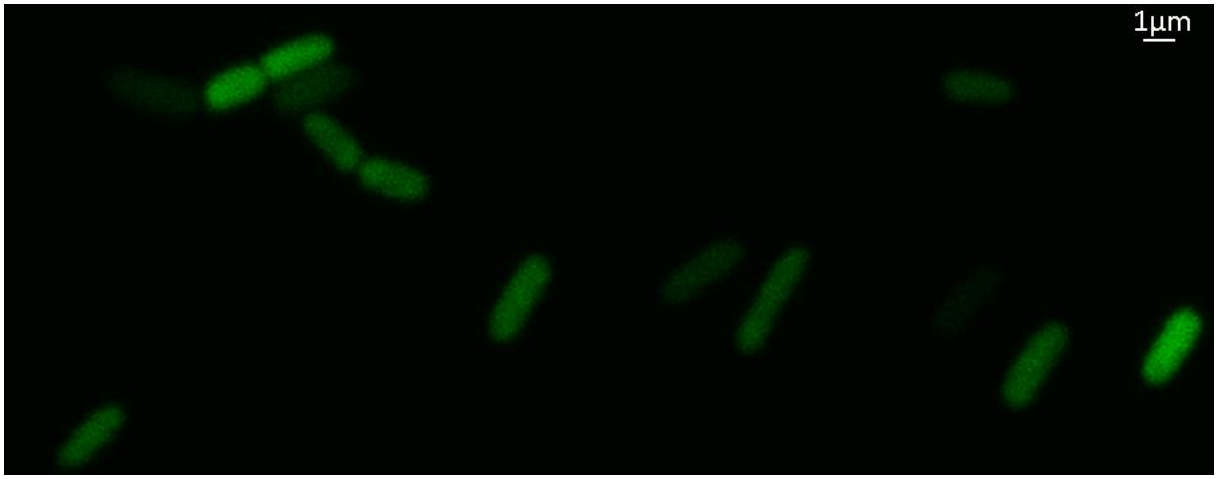


Figure 2.8 – Visualization of *E. coli* cells expressing GFP proteins at 30 °C. Original image has been acquired by a 488-nm laser (Melles-Griot) and a HQ514/30 emission filter (Nikon) and is saved as a grayscale image of the green channel, and has been transformed into an RGB image by concatenating an image containing only zeros on the red and blue channels.

2.3.2. Nucleoid

In bacterial cells, contrary to eukaryotic cells, the chromosomal DNA is not enclosed in the nucleus, but is localized in a large region called the nucleoid, which occupies around 75% of the volume in *E. coli* cells [96]. The nucleoid is responsible for several of the cellular processes described in the previous Section: gene expression, DNA replication during cell growth and the transfer of two identical nucleoids to the daughter cells during cell division [97].

The Nucleoid in bacterial cells is normally associated with circular, double-standard DNA pieces, generally with containing thousands of coding genes and at least 1 to 10 million base pairs (e.g. *E. coli* K-12 strain has 4,639,221 base pairs and 4288 protein-coding genes [98]).

Although the Nucleoid is a large structure, it must be compacted into a smaller size to fit inside bacterial cells. This Chromosomal DNA compaction starts with a global super coiling effect, which twists the relaxed circular DNA contorting it into a more packaged shape [99]. Over twisting leads to positive supercoiling, while under twisting leads to negative supercoiling, with each state able to distinctively affect transcription events [100]–[103].

The mechanisms of changing the topological state of the DNA is regulated both by nucleoid associated proteins (NAPs) [104], [105] and by the so called topoisomerases enzymes (e.g. Gyrase and Topoisomerase I [106]–[108]). Each topoisomerase affects differently the DNA, e.g. Gyrases can release positive, but not negative supercoiling, while Topoisomerase I releases negative, but not positive supercoiling [103], [106], [108].

Even with its compacted nature, the genome is still able to be accessed by several enzymes, such as the DNA polymerase or the RNA polymerase (RNAp) to perform the replication, transcription and translation processes. This is made possible by the aforementioned NAPs, such as histone-like Heat unstable proteins (HU), histone-like nucleoid structuring proteins, factor-for-inversion stimulation and integration host factors [104], [105] and by the unwinding of the DNA, trough supercoiling [99], [102], [109].

As previously mentioned, the strategies for structure visualization are divided in the use of fluorescent proteins (FPs) and fluorescent dyes. For the visualization of the Nucleoid both strategies are widely used. The use of fluorescent proteins to visualize the Nucleoid is linked with the use of the aforementioned NAPs, which are able to fuse with FPs like GFP, YFP or mCherry [96]. This research focuses on the mCherry protein and fuses with the α subunit of the Nucleoid-associated protein HU (HupA), which is encoded by hupA gene, thus can be used to study how the nucleoid is distributed along the cell and its role in the organization of other cellular structures [96]. The HU protein fused with GFP has also shown the ability to co-localize the E. coli nucleoid [110].

Both DAPI stain (4', 6-diamidino-2-phenylindole) and ethidium bromide are the most used fluorescent dyes to visualize the Nucleoid [111]. The first one can bind to the minor groove present in the DNA helix while the second one is able to insert between the planar bases of the DNA (DNA intercalation) [112]. DAPI is the dye that has been extensively used in this research work and has also been shown to co-localize the E. coli nucleoid is convenient [110]. DAPI is associated with a maximum emission wavelength of 450 nm, even when bound to nucleic acids (which can be visualized with a blue/cyan filter) allowing for example the simultaneous visualization of other structures with GFP, due to the low spectrum overlap [113].

In terms of limitations, DAPI can be used in time-lapsed in vivo studies [114], its use is normally associated with fixed cell studies as these, as the structure visualization efficiency is greater when cells are fixed [115] and the concentrations of DAPI required to use in live-cells studies can reach high toxicity levels [116]. In a recent study [117], the LBD showed that the size detection of the Nucleoids by HupA-mCherry tagging match the results obtained with DAPI staining at different temperatures (24 °C, 37 °C and 43°C), where DAPi showed a slight larger size, which agrees with studies that showed slight expansion due to UV lamp perturbations [114]. At 10°C, the HupA-mCherry signal was found to

be too weak to be able to be analysed, showing a limitation to the use of the mCherry as a co-localization tool at extreme low temperatures [117].

In this research work, both the DAPI fluorescent dye and the mCherrry fluorescent protein are used, as detailed in the Experimental Development Section 5.1. While the DAPI fluorescent dye is used to test a co-localization tool with the RNAp signal, the mCherry data is used to test and validate the Structure Segmentation Algorithms (see Section 4.1.5 and 4.1.6). An example of the Nucleoid visualization with DAPI and mCherry at 30 °C is shown in Figure 2.9 (A and B respectively).

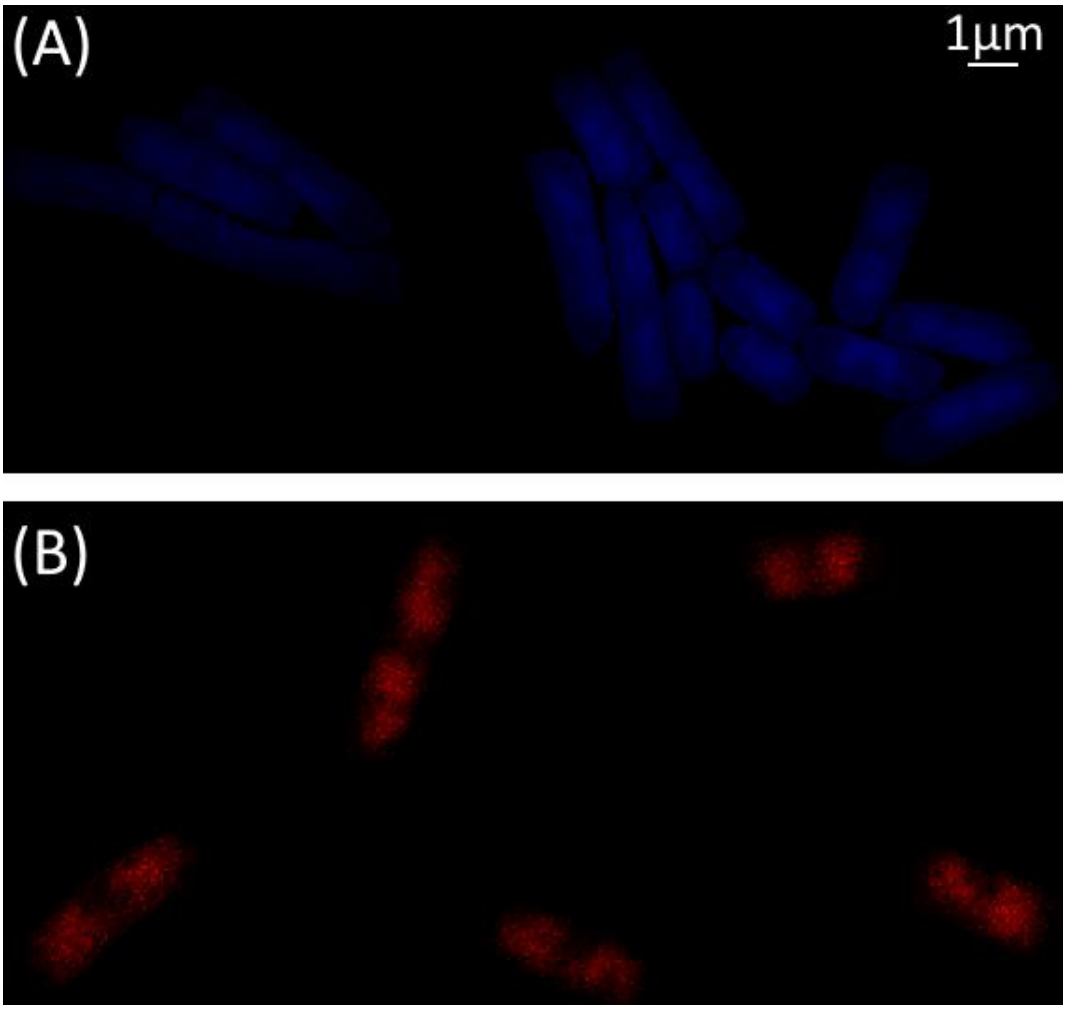


Figure 2.9 - Visualization of Nucleoids in *E. coli* cells at 30 °C (A) with DAPI staining and (B) with mCherry fused proteins tagging. (A) Original image has been acquired by epifluorescence microscopy using a mercury lamp with DAPI filter (Nikon) and is saved as a grayscale image of the blue channel, and has been transformed into an RGB image by concatenating an image containing only zeros on the green and red channels. (B) Original image has been acquired by 543 nm He-Ne laser (Melles-Griot) and HQ585/65 filter (Nikon) and is saved as a grayscale image of the red channel, and has been transformed into an RGB image by concatenating an image containing only zeros on the green and blue channels.

2.3.3. RNAp

The RNA polymerase (RNAp) is the main enzyme involved in the transcription mechanism (as detailed in Section 2.1.4). The bacterial RNAp (with a mass of around 400 kDa) is composed of 5 core subunits [118], [119]: the β' and the β subunits are the largest subunits and are encoded respectively by the rpoC and the rpoB genes, with both of them being responsible for the RNA synthesis and the interactions with the non-specific DNA, especially using the "jaws" (see Figure 2.10) to detect downstream DNA [118], [119]; the α^{I} and α^{II} subunits are encoded by the rpoA gene and are equal

subunits that are responsible for the assembly of the RNAp and are also able to detect and interact with specific DNA sequences (the α subunits are divided into two terminals, C and D, with the first one normally associated with upstream DNA segments rich in Adenosine and Thymine, such as the "UP element", and the second Terminal binds with the other subunits [45], [48], as seen in Figure 2.10), interact with regulatory transcription factors [118], [119]; the ω subunit is the smallest, is encoded by the rpoZ gene and is responsible for the stabilization of the RNAp by facilitating its assembly [120].

A sixth subunit is called the σ factor, with its variants designated by its molecular weight [44]. The housekeeping sigma factor (σ70), encoded by the RpoD gene, is the most important σ factor, increasing the affinity to the core bacterial promoter consensus sequences (the hexameric motifs present in several promoters, as detailed in Section 2.1.4) and decreasing the affinity to nonspecific DNA [121], [122]. Other σ factors (e.g. σ24, σ32, σ38, σ54) are normally associated to genes that respond to stress situations like heat shock (σ32) and nitrogen limitation (σ54), recognize less common promoter motifs, as they are only needed in special conditions [123]. When the σ factor is bound to the rest of the core subunits it forms the so called RNAp holoenzyme.

The interaction between the various subunits of RNAp and the promoter motifs can be visualized with a graphical representation in Figure 2.10.

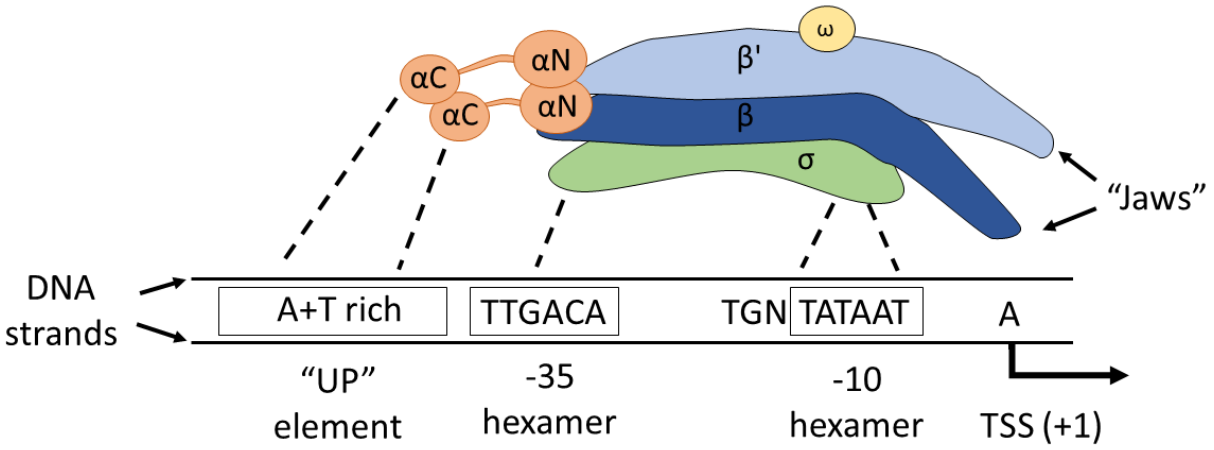


Figure 2.10 – Graphical representation of the RNAp interaction with the promoter. Each RNAp subunit is identified (α^{I}, α^{II}, β, β', ω and σ) by different colouring and name tags. Dashed lines represent the specific contacts between each subunit and specific DNA elements of the promoter. TSS represents the Transcription Start Site, with the numbers corresponding to the distance of the upstream elements to the TSS. In the DNA strand, T corresponds to Thymine, G to Guanine, and A to Adenosine. The image is based on the information in [47] and [120] and does not represent true scales and shapes.

Different organisms have distinct types of RNAps, which can be divided into single-unit or multi-subunit RNAps. The single-unit RNAp has been associated to viruses with one of the most studied examples coming from the T7 bacteriophage RNAp (which is known to infect most strains of E. coli cells) [124]. Multi-subunit RNAps have been associated to eukaryotes, bacteria and archaea, with major differences in the RNAps across those domains, although there exists evidence of homologous structure and function of some of the subunits found in archaea and eukaryotes when compared to bacteria [125], [126].

Although the *E. coli* RNAp is the most commonly used structure for gene expression studies, other bacterial organisms have been used to gather information on the *E. coli* RNAp, by being able to use higher or similar image resolutions, e.g. *Thermus aquaticus* at resolution of 4 Å [121], *Thermus thermophilus* at 2.6 Å [127] and Bacteriophage T7 at 3.3 Å [128], while the core *E. coli* RNAp was initially visualized at a resolution of 19 Å and the α subunit N-terminal at a 2.5 Å [129]. A recent study was

finally able to study the entire holoenzyme with the σ70 factor at a resolution of 3.7 Å [130]. Some studies detected a large structural and functional conservation between viral and bacterial RNAps while also observing the small conformational changes that allow each species to adapt to their environment [131], [132]

In this research work, both the T7 RNAp and the *E. coli* RNAp have been used to study the gene expression kinetics in each specific promoter, while only the *E. coli* RNAp has been visualized by fusing it with a green fluorescent protein, as visualized in Figure 2.11, as detailed in Section 5.1.

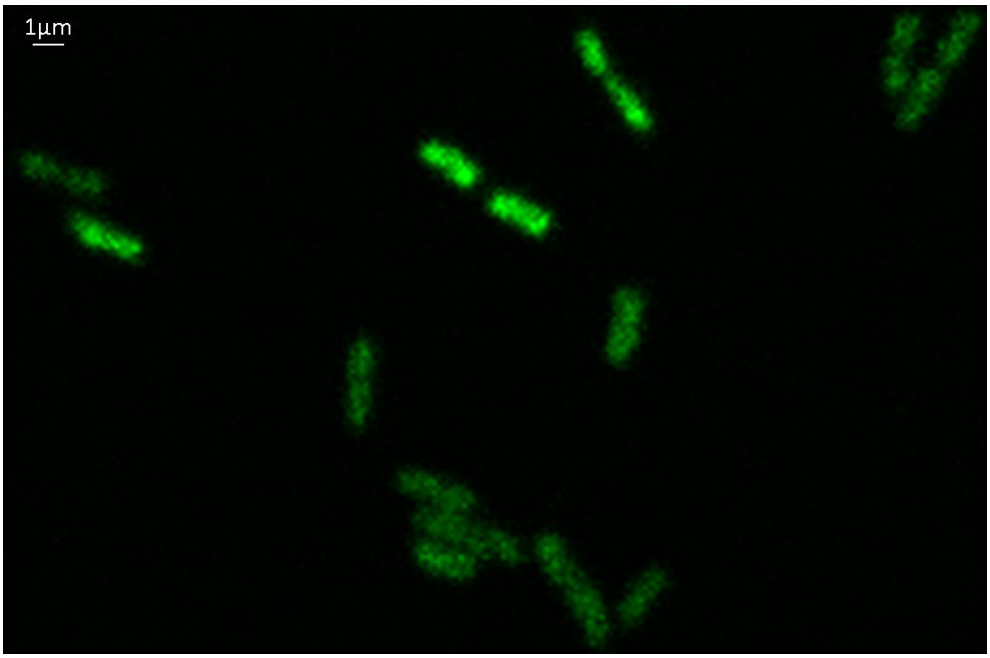


Figure 2.11 - Visualization of *E. coli* cells expressing RNAp-GFP fused aggregates. Original image has been acquired by a 488-nm laser (Melles-Griot) and a HQ514/30 emission filter (Nikon) and is saved as a grayscale image of the green channel, and has been transformed into an RGB image by concatenating an image containing only zeros on the red and blue channels.

2.3.4. RNA

As mentioned in Section 2.1.4, RNA particles are essential to the flow of information between the stored data in the DNA and ending with the production of the proteins based on that information (see Figure 2.6). RNA's can appear in three different configurations, mRNA, tRNA and rRNA, [133] although in this research work, only the first one is used (so every appearance of just "RNA" in this dissertation will be related to mRNA). Not only the visualization of RNA particles can be applied to understand the transcription mechanisms but can also be used to understand other roles of the RNA, such as regulating cellular processes, including cell division, growth and differentiation, and can also act as an enzyme to speed chemical reactions, and for example in virus, it is the RNA that carries the genetic information (instead of the DNA) [133]–[136].

The *in vivo* visualization of RNA molecules can be achieved by fusing it with fluorescent proteins, which can bind to the RNA when it is produced [81], [137]. In *E. coli* cells, one of most used binding systems between proteins and RNAs is based on the interaction of the MS2 bacteriophage coat protein with its own genome [138] and was found to be able to localize and detect RNA particles in yeast cells [139] and was later improved in 2003 [3] and adapted to *E. coli* cells in 2004 [4]. Other RNA tagging techniques have also derived from other bacteriophages, such as the PP7 bacteriophage [140] λ bacteriophage [141], and since each binding site is specific to each tagging system [141], this property

allows the visualization of up to three different RNA targets or different regions of a single RNA, using different fluorescent proteins [141].

In this research work, only the MS2-tagging technique has been used to visualize single RNA particles in living cells, similarly to what has been described in [4], [5], [142]. This technique is based on the fusion of the target gene with several copies of the untranslated RNA region of the target gene encoding the target mRNA and then fusing the RNA binding protein with a fluorescent protein (e.g. GFP) [4], [5], [142]. This technique allows the visualization of the single RNA particle when multiple fluorescent probes bind to the RNA (as a single bright spot), and multiple RNA particles in the same region (which still appear as a single but brighter spot) can be identified using an estimation based on the total fluorescence of the spot, as developed in [143] and detailed in Section 4.1.7.

This technique can be either chromosomally-integrated or be engineered into a plasmid. In this research work, both conditions were used, by constructing two different sequences, both with the same gene of interested, PLacO3O1-mCherry (as it is controlled by LacO3O1 and it was engineered from the *E. coli* native lac promoter, by removing the O2 repressor binding site downstream of the transcription start site [144]), fused a tandem array of MS2 binding sites (48 for the plasmid and 33 for the chromosome) with random sequences between each array to increase stability and a second sequence for the expression of the RNA binding MS2 coat protein in dimerized configuration (see Figure 2.12 for a schematic representation of both the chromosome and the plasmid constructs).

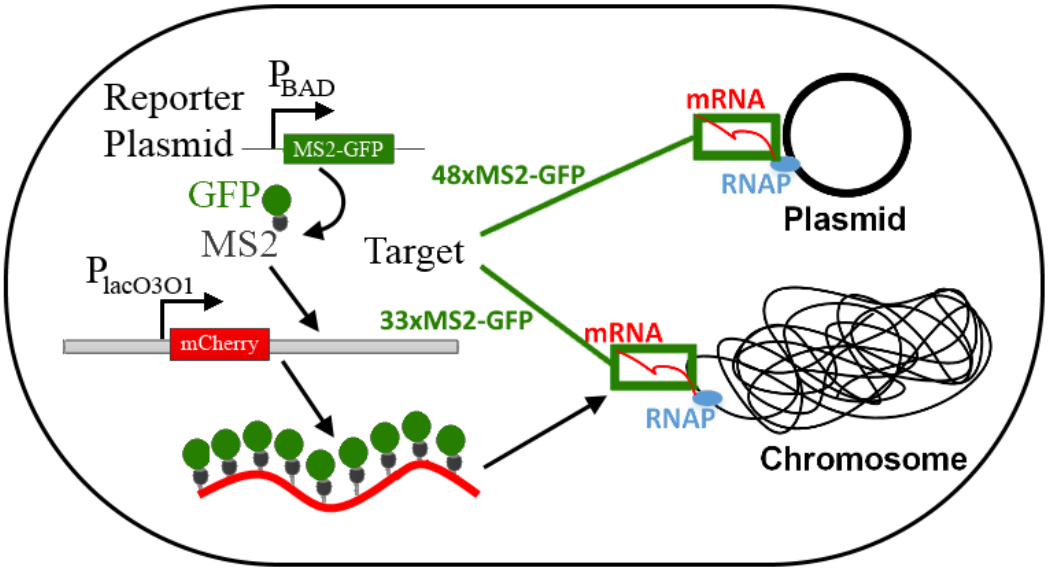


Figure 2.12 - Single-RNA detection system schematic. The production of the MS2-GFP reporter proteins is controlled by the P_{BAD} promoter. For the chromosome integration, a mCherry-MS2 with 33 binding sites (BS) RNA is produced under the under the control of the $P_{LacO3O1}$ promoter, while with the integration into a single-copy plasmed a mCherry-MS2 with 48BS RNA is produced under the control of the same promoter**.** Using this system, individual target RNA molecules are produced and then are rapidly tagged by MS2-GFP proteins produced by the reporter plasmid, making the tagged target RNA visible under the microscope as a fluorescent "spot".

Figure 2.13 presents an example of *E. coli* cells under the expression of the reporter with 0.4% of L-Arabinose (Sigma-Aldrich, USA) at 30 °C, with an induction of 1000 µM IPTG), which allows us to observe MS2-GFP-RNA clusters, as the abovementioned "bright spots" (see white arrows in Figure 2.13).

The observation of the spatial distribution of these molecules at the single cell level, and studying the kinetics of segregation to the cell pole and partitioning during division can establish a correlation

between those events and cellular aging (see Section 2.5), i.e., loss of reproductive vitality, which is one of the secondary objectives of this research work, as it allows us to create realistic models capable of simulating the spatial and temporal organization of bacteria.

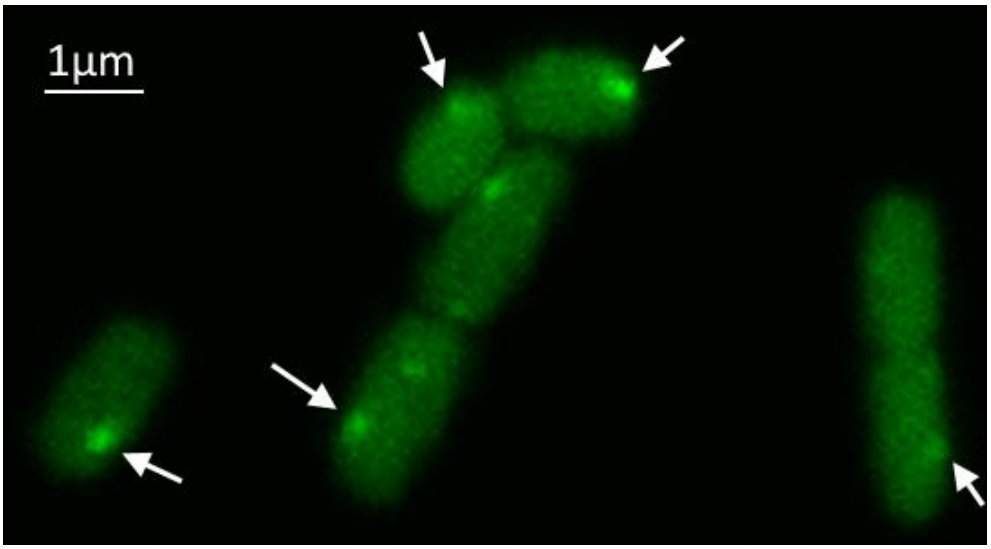


Figure 2.13 - Visualization of *E. coli* cells expressing MS2-GFP-RNA aggregates at 30 °C. Original image has been acquired by a 488-nm laser (Melles-Griot) and a HQ514/30 emission filter (Nikon) and is saved as a grayscale image of the green channel, and has been transformed into an RGB image by concatenating an image containing only zeros on the red and blue channels. White arrows indicate the observable clusters.

Other visualization techniques of RNA molecules like single molecule fluorescent in situ hybridization (smFISH) or stochastic optical reconstruction microscopy (STORM) usually require the fixation of the cells [81], [137], so are not suitable for studies with live cells.

It should be noted that the MS2-GFP tagging technique also has some hindrances, since it required the construction of artificial genes that contain the fusing of the MS2 stem-loops and the binding of MS2-GFP might affect the motility and function of the tagged RNAs [145]. This means that studies with based on this system might can require a case-by-case study to show that these RNAs have equivalent properties as the non-tagged ones, as for example the binding with the MS2-GFP proteins protects the RNA particles from natural degradation [3], [5].

Additionally, it is necessary to check if all the binding sites are working properly, as non-functional binding sites can diminish the observed fluorescence intensity of the "spot" [3], [145], which affects the quantification of the 'integer-valued absolute number' of RNA molecules in a single cell [4], [74], [146]. If all the binding sites are working properly, then this quantification be used to track single RNA molecules over a time period, as the fluorescence of tagged RNAs will not decrease significantly over time (gradually or abruptly), in agreement with previous reports of a mean half-life of over 140 minutes [146].

2.3.5. FtsZ protein

As described in Sections 2.1.2, the Fts (Filamenting temperature-sensitive) proteins (e.g. FtsZ, FtsA, FtsE, FtsX) are mainly associated with the division apparatus (see Figure 2.3-3-A) with most of those proteins located in the membrane, while the FtsZ (a prokaryotic homologue of the eukaryotic tubulin) and the FtsA proteins are located within the bacterial cytoplasm [31], [147].

Not only the FtsZ protein is responsible for the division process, but it also helps the coordination of the chromosome replication and segregation [148]–[150] and is also associated with bacterial shape, along with proteins such as MreB (actin homologue and mainly associated with the elongation apparatus [31]) and crescentin [16], [24]. Although not directly responsible for elongation, a recent study showed that the inhibition of the FtsZ polymerization by the OpgH enzyme allows *E. coli* to regulate growth size based on the availability of nutrients [36], in agreement to what was previously reviewed both for *E. coli* and *B. subtilis* cells [28].

Another important protein present in the cytoplasm is the ZipA, which has been found to protect the FtsZ protein from degradation [151]. Both the ZipA and FtsA proteins have been found to be required for the assembly of the divisome, which is the complex protein structure that forms the division apparatus and creates the division septum by causing the invagination of the cell envelope. This process is followed by the constriction of the ring produced by the FtsZ proteins (also called the Z-ring), until the mother cell completely divides into two equally sized daughter cells (with a high precision), succeeding with the Z-ring disassembly in both daughter cells layers [152]–[154].

It is believed that in *E. coli* cells, the placement of the divisome at the centre of cell is controlled by two negatively regulated and independent mechanisms, namely the Min system proteins and Nucleoid Occlusion (NO) [149], [150], [155], [156].

The Min system (also named the MinCDE system) is composed of three Min proteins, MinC, MinD, and MinE and prevents the FtsZ polymerization at the cell poles (a detailed description of this process is provided in the next Section) [150], [155]. By oscillating between the poles with a period of around 40s at room temperature [157], the Min System does not allow the assembly of the FtsZ ring in each of the poles, as the FtsZ Ring takes 60s to 180s for total assembly and 60s for disassembly, during the cell cycle [158], which can prevent the formation of mini-cells [159].

The NO mechanism [156] prevents the FtsZ polymerization over the Nucleoid through the action of the DNA binding protein, SlmA [160], which prevents the assembly of the FtsZ ring until the chromosome is completely replicated and separated into two Nucleoids [155], [161]. Due to the NO mechanism, the FtsZ ring can only be assembled in the space between the two nucleoids, and this space has been shown to be stochastically generated, as both the distance between replicated nucleoids and the location of each nucleoid can vary from cell to cell, although with a high-level of precision at optimal conditions [7]. The same study reported that that at suboptimal temperatures, the relative distance between nucleoids is increased (prior to cell division), decreasing the probability of symmetric division [7], which have been found to related to cellular aging, as a functional asymmetric division can result in unwanted protein aggregates, concentrating in the older pole of the mother cell and causing a slower division rate of the daughter cells [8] (as detailed in Section 2.5).

Studies with fts *E. coli* mutant cells showed that ftsA, ftsI, and ftsQ mutants (without fully functional genes), could form the Z-rings but the final ring contraction was blocked, suggesting that it is the FtsZ ring that allows the localization of the fts gene products, which then allow the ring to finalize the division septum [162].

Additional studies with *E. coli* mutant cells lacking functional Min and the NO systems, showed that these cells still divided preferentially at mid-cell, although in these conditions the division at around the quarter of the cell increased significantly (resulting in the cells with multiple FtsZ rings and FtsZ rings placed near the poles, leading to the creation of asymmetric divisions and unviable mini-cells) [159]. The same study [159] found that additional Z-ring localization systems independent from

the NO and the Min systems are present in *E. coli*, with similar studies finding that in other rod-shape cells (e.g. Bacillus subtilis [163]) the NO and the Min System also ensures the full precision of the ring placement mechanism at the mid-cell. Both of these studies [159], [163] also found that bacterial cells have developed other independent mechanisms to ensure that the division process maintains a high precision, safeguarding the survivability of the daughter cells, even when both the NO and the Min systems are simultaneously removed [159], [163].

The in vivo visualization of the FtsZ protein spatial dynamics has been normally associated with the fusing FtsZ with GFP (FtsZ-GFP) to visualize its spatial dynamics [147], [164]. With the fused FtsZ-GFP proteins, the FtsZ proteins have been found to assume three different maturation stages over the cell cycle [165], [166]. In the first stage, cells no rings are visible and most of the FtsZ-GFP proteins are either located at the cell poles or are spread along the cell in a cloudy formation (with no visible structure) [166]. In the second stage, the FtsZ-GFP proteins start to form a ring structure at mid-cell, which is visualized as two bright dots located near the cell border and a less bright band that connects the dots (this is due to the 2D top-projection of the ring in a rod-shaped cell) [166]. In the third and final stage, the ring starts to contract and be fully closed, so the 2D top-projection changes to a full bright band that goes touches both cell borders at the centre of the cell [166]. An example of the visualization of these three stages using the FtsZ-GFP system can be observed in Figure 2.14-A (with the three different maturation stages [165], [166]), while an example of the FtsZ-mCherry system is presented in Figure 2.14-B.

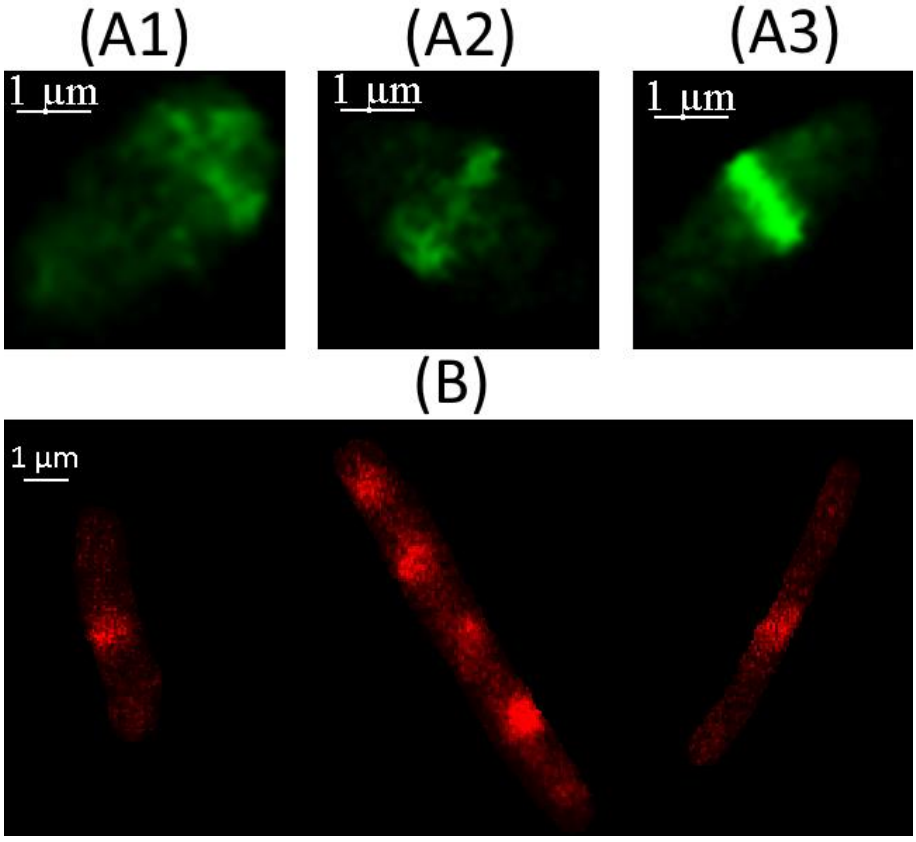


Figure 2.14 - Visualization of FtsZ proteins in *E. coli* cells at 30 °C (A) with FtsZ-GFP and (B) with FtsZ-mCherry tagging. Original image has been acquired by a 488-nm laser (Melles-Griot) and a HQ514/30 emission filter (Nikon) and is saved as a grayscale image of the green channel, and has been transformed into an RGB image by concatenating an image containing only zeros on the red and blue channels, (A1), (A2) and (A3) are respective representations of the three different maturation stages of the FtsZ proteins [165], [166]. (B) Original image has been acquired by 543 nm He-Ne laser (Melles-Griot) and HQ585/65 filter (Nikon) and is saved as a grayscale image of the red channel, and has been transformed into an RGB image

by concatenating an image containing only zeros on the green and blue channels. Cell in the middle is not dividing so it shows multiple FtsZ rings, similar to examples where both the NO and the Min systems are simultaneously removed [159], [163].

A recent study [167] has made use of automatic Machine Learning methods to classify the FtsZ ring formation into the three possible stages based on the microscopy images as observed in [166]. In this research work, additional Machine Learning methods are studied to classify the FtsZ ring formation, focusing also on the binary classification of cells into "stage 3" and "non-stage 3", which improved the statistical scores of the classification and allow us to use cells in stage 3 for future studies, as detailed in Section 4.3.3. Additionally, the FtsZ-GFP system (see details in Section 5.1) is also used to test and validate the Structure Segmentation Algorithms (see Section 4.1.5 and 4.1.6).

2.3.6. Min System Proteins

As mentioned in the previous Section, *E. coli* cells have a mechanism that allows the precise localization of the FtsZ proteins (to be polymerized) to the mid-cell and consequently of the divisome, which is the Min System (also called MinCDE) and composed of three proteins MinC, MinD, and MinE [150], [168], and that this mechanism is coupled together with the Nucleoid Occlusion system (NO) in order to ensure that the FtsZ ring can only be formed when the cell completes the chromosomal segregation and the two Nucleoids inside the mother cell are completely separated (which allows the FtsZ to polymerize in the space between the two Nucleoids) [156].

The Min System (the discovery of the three proteins involved in this mechanism) was initially discovered using *E. coli* mutant cells that produced minicells, due to being incapable of produce the septum correctly localized at mid-cell [169]. These cells mostly contained RNA and proteins, but almost no chromosomal DNA (no Nucleoid) [169].

The FtsZ polymerization is prevented by the MinC protein [168], [170], which is only active when bound to the MinD protein [168], [171]. This prevention is done near the poles, as the MinD protein localizes to the membrane at the poles and contains an ATPase and an ATP-binding domain, which forces the binding to the membrane when in the ATP-bound conformation [168], [170]. Clusters of MinD are produced when enough proteins bind to the protein, activating the MinC proteins and preventing the FtsZ polymerization [168], [170]. The MinE protein prevents the formation of the bound clusters of MinC and MinD proteins at mid-cell by forming a ring near the cells poles, which acts a catalyser in the release of the MinD from the poles by activating MinD's ATPase, hydrolysing the MinD proteins that are in the ATP-bound conformation [168], [170].

Since the concentration of the MinC and MinD proteins is minimal on the mid-cell, especially in the ATP-bound conformation [31] (see Figure 2.3-3 and Figure 2.15-A) the FtsZ ring can only be formed in the mid-cell, which happens only when two nucleoids are fully separated and the SlmA protein that is bound to the Nucleoid no longer inhibits the FtsZ polymerization [160], [161]. A schematic of the binding and unbinding of the MinCDE proteins is shown in Figure 2.15-B.

In *E. coli* cells, this process is completed by the oscillation of the MinCDE proteins (see Figure 2.15-A) from one pole to the other, pushing the non-polymerized FtsZ proteins to the opposite pole, until they start to from the FtsZ ring [168]. Different studies observed that the *in vivo* periodicity of this oscillation is approximately of 50 s [170], [172], [173], while this oscillation is temperature-dependent as observed in [157], where it was observed to be between 60 s and 10 s respectively for temperatures of 20°C to 40°C [157]. This oscillation is not necessary for all bacterial cells, as it has been shown that in *B. subtilis*, a static concentration of gradient of MinC and MinD is present at the cell poles [168],

[174]. It is important to note that it is the MinE rings that push the MinD and MinC to travel to the other pole [168], as studies with mutant cells (preventing the formation of MinE rings) showed that in those cells, the MinCD clusters were able to be formed at mid-cell, preventing the FtsZ to be polymerized into a FtsZ-ring configuration and cell division to be completed [170].

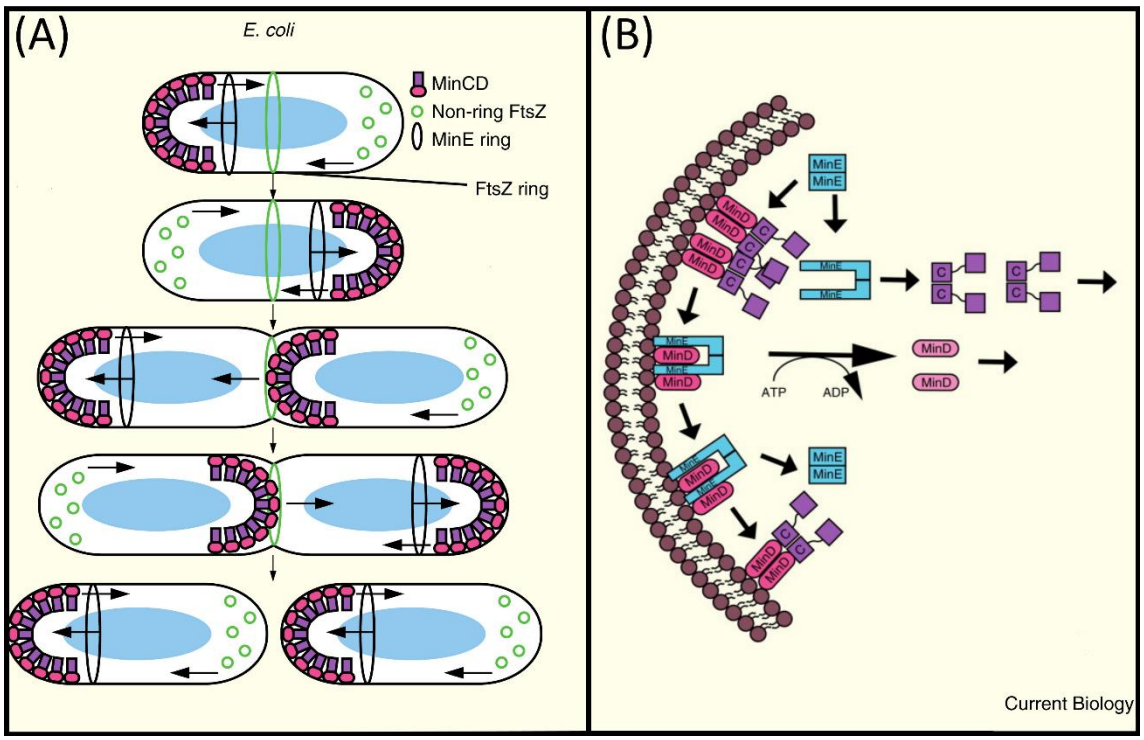


Figure 2.15 – Schematic representation of the MinCDE system in *E. coli* cells. (A) Visualization of MinCD proteins oscillating from pole to pole and the MinE ring preventing the activity of MinCD at mid-cell. Adapted from [168]. (B) Visualization of the binding and unbinding of the MinCDE proteins and the consequent activation and inactivation of the Min system mechanism that prevents the polymerization of the FtsZ ring. Adapted from [168].

In this research work, the Min System was visualized to test and validate the Structure Segmentation Algorithms (see Section 4.1.5 and 4.1.6), by fusing the MinD protein with the superfolder GFP protein (sfGFP), as detailed in Section 5.1. An example of a 6-minute timeseries (each image is taken 1-minute apart) is shown in Figure 2.16, showing the oscillation of the MinD protein from one pole to the other and co-localizing to the cell membrane, as represented in the schematic of Figure 2.15-A.

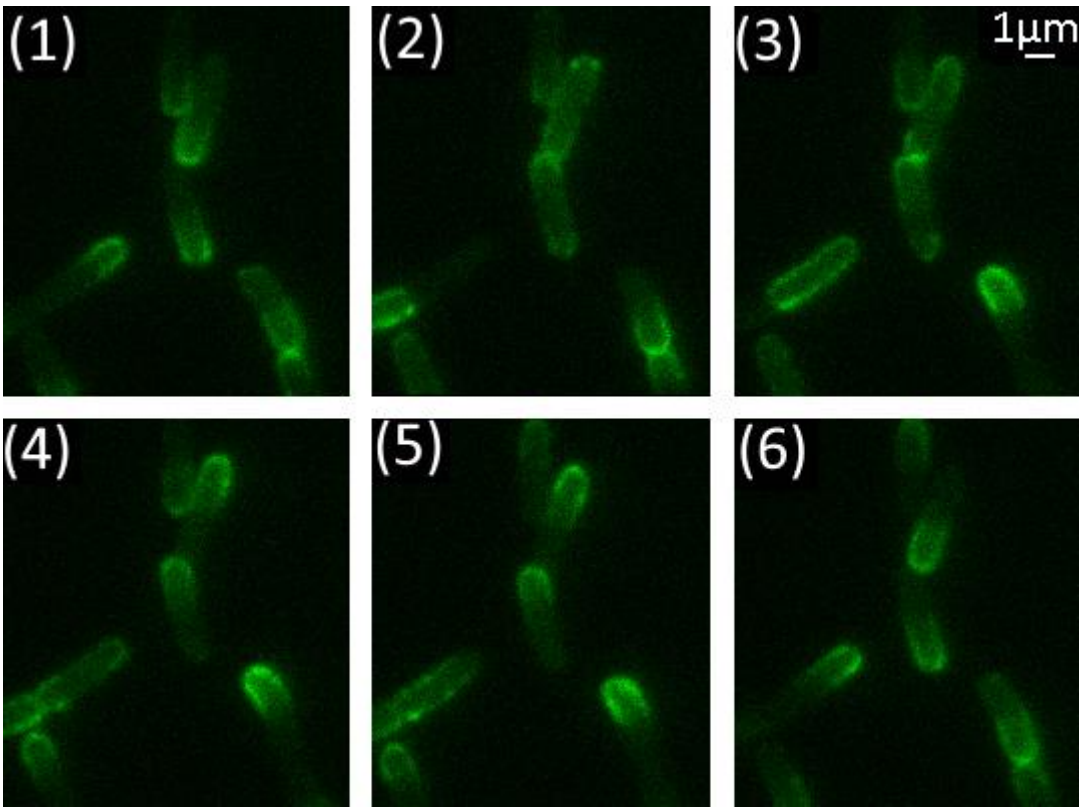


Figure 2.16 – Visualization of MinD system proteins fused with superfolder GFP protein (sfGFP), oscillating from pole to pole. Each image is 1 minute apart. Images were acquired with Highly Inclined and Laminated Optical sheet (HILO) microscopy [175] using a EMCCD camera (iXon3 897, Andor Technology) with 488nm laser, along with the HQ515/30 filter and the Texas Red filter (Nikon, Tokyo, Japan).

2.3.7. Inclusion Bodies

Inclusion bodies are usually formed by dense packs of overly expressed aggregates of proteins [176]. Inclusion bodies can be used to identify diseased cells also be hallmarks of genetic diseases, as in the case of Neuronal Inclusion bodies in disorders like frontotemporal dementia, Parkinson's and Huntington's diseases [177], [178] while in bacterial cells, the asymmetric segregation of protein aggregates has been associated with cellular aging, cells without any visible inclusion bodies, exhibit a larger reproductive ability [179].

In *E. coli* cells, most studies related with inclusion bodies are trying to actively study techniques to improve the solubilization and refolding procedures of such proteins aggregates, since the refolding of inclusion body proteins into the bioactive form is energetically costly [176], [180]. This is especially important, as around 75% of recombinant proteins expressed in *E. coli* cells are present in inclusion bodies [181]. These inclusion bodies are normally extracted and are subsequently solubilized, which has been proved to be important in the extraction of cloned human insulin, produced by *E. coli* cells [182].

Although inclusion bodies are mainly associated with misfolded proteins, other findings have also been able to identify green fluorescent proteins aggregates as co-localized in inclusion bodies [183]. A similar conclusion was found in [117], where 91% of the inclusion bodies were co-localised with the observed green fluorescent synthetic aggregates and 83% of the synthetic aggregates were co-localised with an inclusion body, similarly to what was found with IbpA-YFP fused aggregates [179]. The same study showed that osmotic stress increased the amount of visible inclusion bodies in the cells [117].

In this research work, an inclusion body detection algorithm is presented in Section 4.1.7, based on the acquisition of Phase-Contrast images. In such images, inclusion bodies are characterized by bright round objects inside the cells. Three example images are shown in Figure 2.17-A, Figure 2.17-B, respectively with no addition of NaCl (low stress condition) and the addition of 125 mM of NaCl (medium stress condition), as described in Section 5.1 and as used in in Section 6.1.6 to test the inclusion body detection algorithm (where a third high stress condition was used with the addition of 300 mM of NaCl. An algorithm with high accuracy could be used to automatically separate diseased cells from healthy cells, similarly to what was done manually in [184].

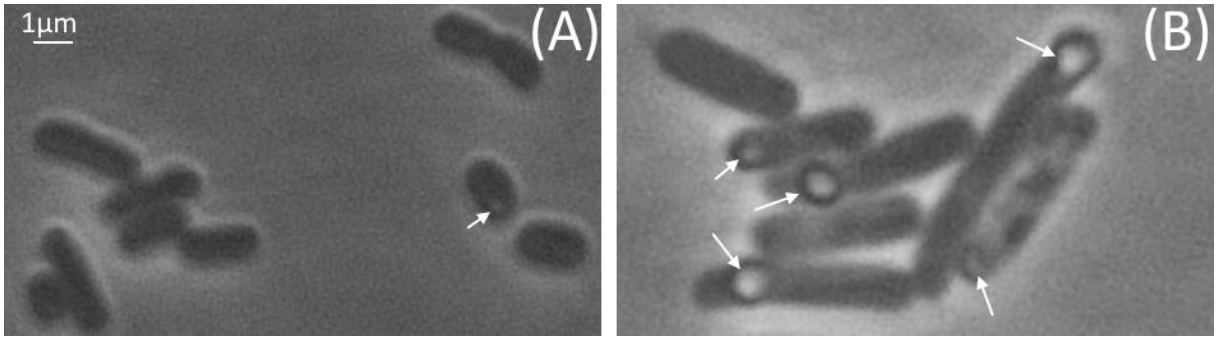


Figure 2.17 - Visualization of *E. coli* cells containing inclusion bodies in three different stress conditions. (A) low stress (no NaCl); (B) medium stress (125 mM of NaCl added). Phase-Contrast images were acquired with a CCD colour camera (DS-Fi2, Nikon).

2.4. Microscopy Imaging

There has been an impressive progress over the recent years in the microscopy technology, resulting in multidimensional images with better quality and resolution. Associated with these progresses, the fusion of data coming from different microscopy techniques led to the development of several computational approaches in the last decade to deal and analyse image-based studies in Cell and Molecular Biology. Microscopy images and particularly sets from time-lapsed series can contain information about the cell dynamics, subcellular constituent distribution, such as the cell membrane, the cytoskeleton, genetic material (DNA and RNA) and various organelles [1], [79], [80].

Three major improvements have been developed in parallel and driven the breakthroughs in the production of high quality microscopy images, which have played an important role in cellular studies [79].

The first advancement allowed the tracking of the activity of a great diversity of molecules using bright and genetically encoded fluorescent probes inside the cell. The second improvement was based on the optimization of the optical sensors and the usage of hardware controlled by feedback which permitted an efficient acquisition of large, high-quality microscopy image datasets. Third, an accelerated progress in the electronic detection technology enable the generation of high sensitivity datasets of microscopy images. In the latest years, single-molecule detection at the single-cell level grew into a conventional technique in the microbiology laboratories [79].

Time-lapsed fluorescence microscopy imaging is also being used in live single bacterial cells to study the *in vivo* activity of transcription and translation, and also the protein interactions using the previously described fluorescent probes [1], [185]. These techniques have been used to study genetic circuits such as the Toggle Switch [186] and the Repressilator [187].

A revolutionary technique, capable of detecting and tracking single RNA molecules in *Escherichia coli* (*E. coli*) by fusing the RNA bacteriophage MS2 coat protein with GFP [3], [5] have also been used recently by the Laboratory of Biosystem Dynamics (LBD) from Tampere University of Technology to produce time-lapsed microscopy images of *E. coli* cells. The RNA-MS2-GFP complexes were tracked to study to study the activity of the lar promoter [188], the activity of the arabinose promoter [189] in *E. coli* and to study the partitioning of RNA [190] and proteins [191] in cell division.

2.4.1. Main Challenges and Limitations in Live-cell imaging

The main challenges in live-cell imaging can be divided between occurrences during the image acquisition and the post-acquisition processing [192]. For the first part, it is mostly related to the microscope components (e.g. shutter, lens, camera, stage) and can be solved by using better components, such as LED illumination, increasing the speed of the stage, using better filters and cameras. Improving all these solutions will make the system more expensive, so there is a necessity to compromise [192].

The post processing limitations start with the data storage and archiving, which can be solved by using archiving software and by having dedicated databases that can be easily accessed [192]. Using the safely stored data, there still needs be a correct cell tracking analysis in order to make accurate signal quantifications, sometimes requiring image correction, image registration and other image processing techniques, such as the fusion of multimodal microscopy images or the volume rendering of multi-dimensional data [192].

A summary of the use of multimodality and multidimensionality in microscopy is presented in Section 2.4.2, while a detailed description of the Literature Review in microscopy image processing, statistical analysis and the simulation of microscopy images in Biological studies is presented in Section Chapter 3, since these topics cover the main trends in field of microscopy imaging [10] and the main topics of this research work.

2.4.2. Multimodal and Multidimensional Microscopy

The establishment of novel biological studies that depended on the detection of fluorescent aggregates in live cells led to the development of data fusion techniques coming from different microscopy techniques has become a necessity in order to integrate and correlate functional (coming from fluorescent methods of microscopy) and morphological information (coming from illumination and contrast methods of microscopy), which can be combined to provide new information about biological processes [193].

For live cell imaging, these microscopy modes include the use of bright-field and dark-field imaging, Phase-Contrast, differential interference contrast, fluorescence microscopy, total internal reflection fluorescence microscopy, single and multiple photon excitation and a multitude of super-resolution microscopy techniques, such as the stimulated emission depletion and scanning near-field optical microscopy [10], [193]–[195]. Information integration based on the image fusion of multimodality microscopy images for the study of co-localization of internal cellular structures became a common strategy in many biotechnology studies [196].

One of the first biotechnology applications of microscopy image fusion was the study of double labelled DNA via the fusion of dual colour fluorescence (specifically the red and green filter components) from a three-dimensional confocal microscopy in order to study the temporal and spatial organization of DNA inside the interphase nuclei of eukaryotic cells [197], [198].

The initial studies with multimodal image integration were based on the fusion of fluorescence and Phase-Contrast images, which was used in studies of the partitioning of F-plasmids molecules (see Figure 2.18-A) during the cell division of *E. coli* cells [199] and the localization of DNA segments on the chromosome of *E. coli* cells [200]. Other studies also used super imposed Phase-Contrast images with fluorescence microscopy (see Figure 2.18-B) to investigate the stimulation of the proliferation and differentiation of endothelial progenitor cells by the erythropoietin darbepoetin alfa [201]. Similarly in another study, researchers explored the kinetic dynamics of the genetic circuit responsible for the utilization of lactose in *E. coli*, green fluorescence proteins were used to fuse fluorescence microscopy with inverted Phase-Contrast images (see Figure 2.18-C) of the cells [202].

The kinetic dynamics of protein production in *E. coli* cells at the single-molecule and single-cell level by fusing differential interference contrast and fluorescence microscopy images and using yellow fluorescence proteins [203] and to study the protein and mRNA copy numbers in *E. coli* [64] by fusing fluorescence and Phase-Contrast microscopy images (see Figure 2.18-F) .

To study the differential protein expression in *Colletotrichum acutatum* and its impacts in the pathogenicity of the strawberry, one group [204] used the superposition of differential interference contrast and fluorescence microscopy (using green fluorescent probes and Red Nile staining of lipid bodies) images (see Figure 2.18-D). Finally another group overlaid Phase-Contrast and fluorescence microscopy (using green and cyan fluorescent proteins) in order to study the chromosome segregation in the bacterium *Caulobacter crescentus* by using a partitioning apparatus, similar to the existing spindles in eukaryotes [205].

As can be seen in Figure 2.18, in some cases, the simple superposition of multimodal images will result in fused images where both images are correctly aligned (see Figure 2.18-B, C, D and F), while in some cases, possible registration misalignments can be observed, as several fluorescent F-Plasmids, appear to be outside of the cell contours (see Figure 2.18-A and E), observable in the fused image, which is more than expected from diffraction effects. Intra-model registration can be used in images taken at different time frames, while inter-model registration can be used if images are taken from different sensors, as detailed in Section 3.1.1.

Since these type of misalignments can affect statistical analysis such as the calculation of the plasmids spatial distribution along the cell, various image processing algorithms started to be developed, such as image registration, image segmentation, which proved to be required to perform better statistical analysis for the qualitative and quantitative characterization of the processed data [80].

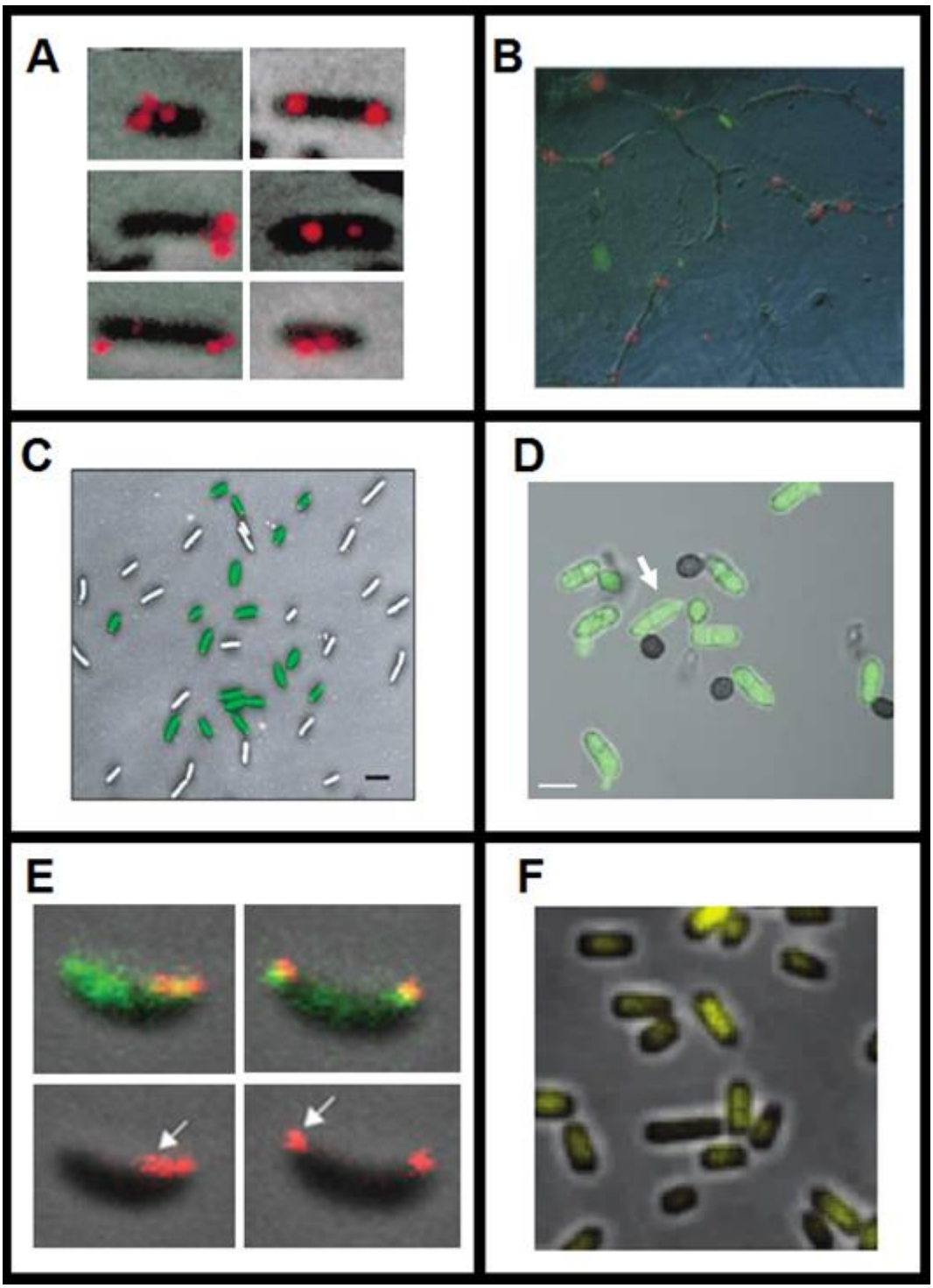


Figure 2.18 - Examples of multimodal image fusion. (A) Fluorescence and Phase-Contrast images of *E. coli* cells [199]. (B) Fluorescence and Phase-Contrast images of endothelial progenitor cells [201]. (C) Fluorescence and inverted Phase-Contrast images of *E. coli* cells [202]. (D) Differential interference contrast and fluorescence images of *Colletotrichum acutatum* cells [204]. (E) Fluorescence and Phase-Contrast images of *Caulobacter crescentus* cells [205]. (F) Fluorescence and Phase-Contrast images of *E. coli* cells [64]. All images were adapted with permission from the respective reference.

In terms of multi-dimensional microscopy, the main challenge is to extract spatial and temporal information at the maximum resolution possible while trying to minimize the damage associated with photobleaching [206]. The resolution along the Z-axis of the microscope is normally smaller than the X and Y-axis, due to the three-dimensional diffraction pattern of the point spread function [206], which can be an obstacle in the production of several Z-stacks, especially while imaging bacterial cells (due to their small sizes), as they normally require Z-axis micro-scale resolutions [19], [20].

Several 3D-microscopy modes have been extensively reviewed [206]: Confocal Laser Scanning Microscopy – CLSM; Two-Photon Microscopy – TPM; Spinning Disk Confocal Microscopy – SDCM; Light Sheet Fluorescence Microscopy – LSFM; Three- Dimensional Structured Illumination Microscopy – 3D-SIM; Three Dimensional Stochastic Optical Reconstruction Microscopy - 3D-STORM; Three-Dimensional Photoactivation Localization Microscopy - 3D-PALM; And Interferometric Photoactivation Localization Microscopy – iPALM; Three-Dimensional Single-Molecule Localization Microscopy 3D-SMLM [206].

The FtsZ ring and the Nucleoid has been one of the most studied bacterial structures in 3 dimensions. For example, 3D-SMLM was used in combination with immunofluorescence labelling to visualize Z-ring in fixed *E.coli* cells [207]. The 3D-PALM and iPALM techniques were also employed (see Figure 2.19-A) to dissect the rate-limiting steps in each process of the Z-ring synthesis [208], while 3D-SIM was used to compare the localization patterns of FtsZ, FtsA, and ZipA in *E. coli* cells, showing that FtsZ localized in patches within a ring structure and that FtsA and ZipA also colocalize in identical patches [209].

Wide-field epi-fluorescence microscopy was able to provide a 4D visualization (3D and temporal) of the Nucleoid revealing a dynamic helical ellipsoid structure [96] (see Figure 2.19-B, C and D), while each component of the segregation apparatus within the Nucleoid were also visualized by applying the 3D-SIM technique [210]. A high-throughput 3D bacterial tracking method for use in standard Phase-Contrast microscopy was also proposed for the characterization of the structure motility patterns in several bacterial cells (including *E. coli*), which showed better tracking accuracy than just 2D projection and 2D slicing [211].

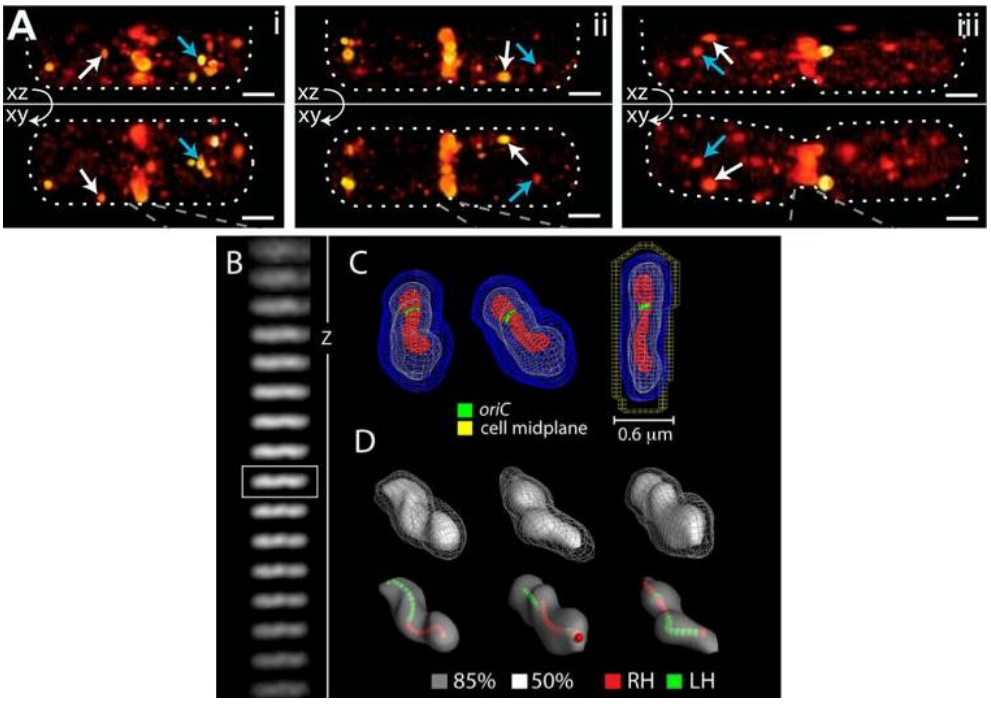


Figure 2.19 – Three-dimensional visualization of FtsZ Rings and Nucleoids. (A) Two-dimensional projections (XZ plane on top and XY plane on bottom) of iPALM images that are used to make a 3D reconstruction of three fixed *E. coli* DH5α cells expressing FtsZ-mEos2. The arrows represent membrane-proximal (white) and cytoplasmic (cyan) clusters of FtsZ-mEos2, with cell outlines drawn with white dotted lines. Adapted from [208] (B) Z-stack of HupA- mCherry images. Adapted from [96]; (C) 3-D reconstruction of a G1 nucleoid. Adapted from [96]; (D) 3-D Reconstruction of three nucleoids and representation of the longitudinal density centroid paths. Adapted from [96].

Each of these microscopy modes have specific advantages, such as high speed (LSFM and SDCM), very high resolution (3D SIM, 3D STORM, 3D PALM and iPALM) and good region of interest manipulation (CLSM and TPM) but also each technique have specific limitations such as damage from photobleaching (CLSM, TPM and 3D SIM), slow speed (CLSM, TPM, 3D SIM, 3D STORM, 3D PALM and iPALM), necessity of cell fixation (3D STORM, 3D PALM and iPALM) and lower resolution (LSFM) and no control on the region of interest (SDCM) [206].

2.5. Cellular Aging

Prokaryotic organisms, such as bacteria, appear to be functionally immortal, when in suitable environments, as each cell perpetuates itself by dividing into two daughter cells with the same genotype as the mother cell.

For that, the stockpiling of unwanted substances or degradation of internal components needs to be managed effectively with mechanisms such as the deliberate asymmetry in the partitioning of intracellular material during division [8]. This was first encountered in unicellular organisms that exhibit highly asymmetric divisions, such as yeast and *Caulobacter crescentus* [212], [213].

Other unicellular organisms, such as *Escherichia coli*, have apparently a morphologically symmetrical division, making the aging process less straightforward to comprehend. Studies of individual cell lineages have shown that two morphologically identical sisters can exhibit functional asymmetries [8]. The detected asymmetry can be indicative of aging, as unwanted protein aggregates tend to concentrate at the older pole of the mother cell and that accumulation can cause a slower division rate of the daughter cells [8].

Another study using *E. coli* also investigated how a different complex construct (Tsr-GFP fusion protein) accumulated at the old poles, which can be used to identify the old and the new poles, when no information about the cell ancestry is available [214]. One related study also concluded that the asymmetric deposition of aggregates can increase the bacterial population fitness and also resulting in higher rates of growth in the daughter cells that have less damaged aggregates, due to the misfolding of proteins [215].

The link between cellular aging and many age-related diseases such as Alzheimer's disease, spongiform encephalopathies [216], Huntington's disease, Alexander's disease, cataracts [217], Motor Neuron disease and Frontotemporal Dementia [218] further enhances the importance of studying the aging process at a simpler level, as it will then shed light how these mechanisms work in complex organisms.

To make accurate conclusions about all the processes described above, even at a simple organism level such as the *E. coli*, it is necessary to use reliable tools and methods for image processing, capable of detecting and tracking individual molecules at the single-cell level. To validate such image processing tools, one needs to apply the developed image generator to realistic spatial and temporal models of bacterial organization, including the cellular structures that were described above. Since the core of this research work was to produce such image processing and image simulation tools, a more detailed literature review of these topics is provided in the next Section.

Chapter 3. Literature Review

The following section gives a comprehensive literature review of core areas related to the research work in the areas of Electrical and Computer Engineering. These areas include image processing techniques such as image registration, cell segmentation and tracking. A literature review on simulations tools is also presented, starting with a review of biological image simulations toolboxes and presenting the Stochastic Simulation Algorithm. Finally, a review on the application of statistical methods and machine learning algorithms to biological studies is also provided. Several examples of these techniques are shown in this revision, based both on images acquired by the LBD and other laboratories.

A typical workflow in a live-cell imaging setup (see Figure 3.1) starts with experimental and microscopy preparations followed by the process of image acquisition. After the image is acquired, several post-acquisition image processing processes are required (such as Image Registration, Cell Segmentation and Tracking), which can be followed by the usage of Statistical techniques to extract valuable information and derive cellular models or by applying Machine Learning techniques to classify the obtained data. Finally, one can use these models to simulate different conditions and validate the developed image processing techniques. The main outcome of this research work is to produce two toolboxes that contain all the developed image processing techniques and the simulation strategies.

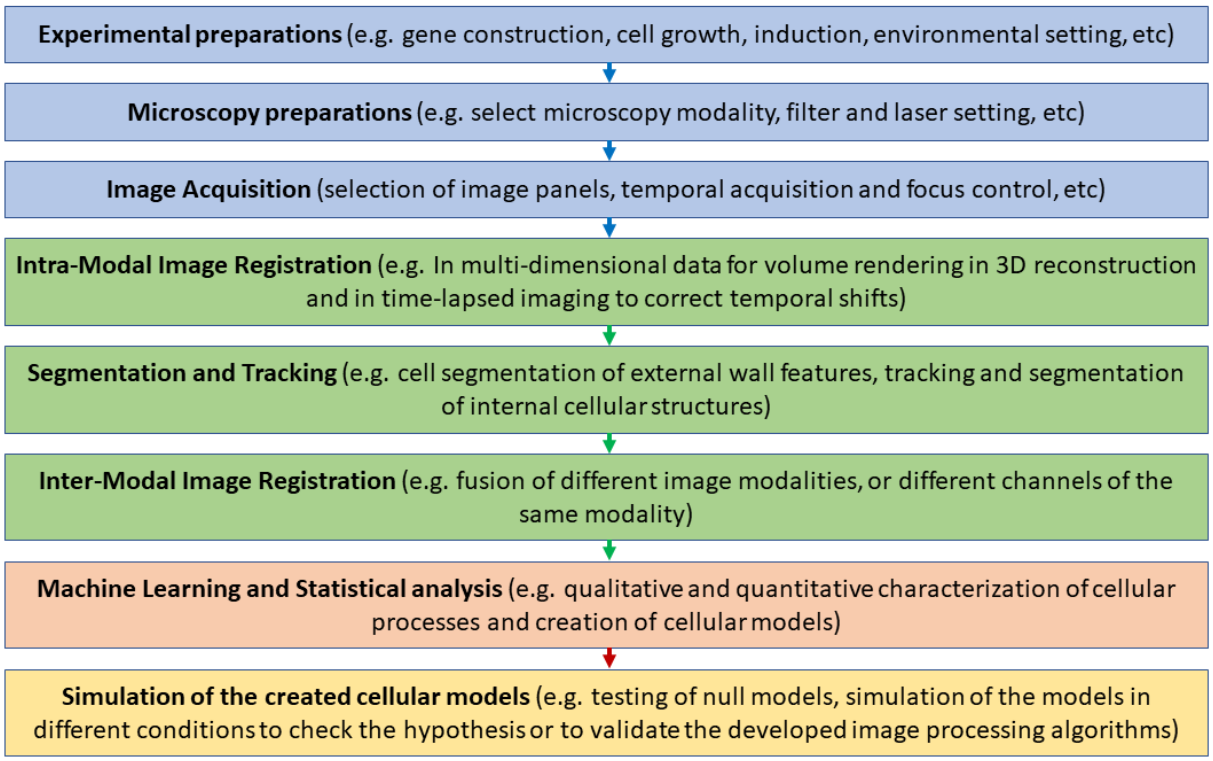


Figure 3.1 - Typical workflow in live-cell imaging, focusing on computer vision techniques related to the planned research work. In this research work, the first three steps (in blue) are described in Section 5.1 and were performed by experimentalist colleagues. Background information about these steps is detailed in Chapter 2. The following steps are related to all the post-acquisition steps that were performed and developed in this research work. Namely, the fourth to sixth steps (in green) are part of the image processing steps (including image registration, segmentation and tracking). The seventh step is related to the application of machine learning and statistical analysis techniques to obtain valuable data and extract biophysical models from the acquired image. Finally, the eighth step is associated with the simulation of the developed models, which can be used to test other experimental conditions and validate the image processing algorithms.

This Literature Review provides both fundamental knowledge of the techniques that stem from core areas of Electrical and Computer Engineering, but also several biological applications of these techniques, while providing the necessary connections to this research work. Section 3.1 provides a literature review of the image processing techniques (Image Alignment, Segmentation and Tracking). Section 3.2 focuses on the literature review of image simulation tools and the Stochastic Simulation Algorithm (SSA). Section 3.3 provides a literature review of several Machine Learning Techniques and statistical techniques that were applied in this research work.

3.1. Microscopy Image Processing

This Section is divided into three main image processing techniques that have been developed and applied in this research work, namely, Image Registration (also known as Image Alignment), Cell Segmentation and Cell Tracking. For the past decades, these three subjects have been extensively surveyed and reviewed [9], [219]–[221]. Other groups also reviewed these techniques specifically applied to biological studies [222], [223].

In this research work, the chosen programming platform for the development and implementation of the image was MATLAB®, as this is a platform that already has integrated almost all of the necessary image processing tools into its own Application Programming Interface (API) [224], [225], without requiring the download of third party libraries, making MATLAB® API a good candidate to develop an academic image processing toolbox tailored for the analysis of microscopy images.

The majority of the previously developed image processing solutions have been tailored to specific problems, since designing an algorithm that could achieve a high specificity and sensitivity for an extensive range of cases is still one of the biggest challenges in Image Processing [9]. Due to this situation, each algorithm needs to be validated and evaluated for each specific problem, which normally requires the use of benchmark datasets, which were usually created using manual processing procedures, but have been substituted with data provided by artificial image simulators, especially in Contests and Open Challenges that have been recently organized such as the 'Cell Tracking Challenge', which is already in its 2nd Edition and is part of the 'Grand Challenges in Biomedical Image Analysis' – (http://grand-challenge.org/Home/), where a benchmark of artificial and real datasets was created in order to measure six segmentation and tracking algorithms [226]. These contest can provide unbiased comparisons between methods, especially prevent abuses in the literature where particular methods are claimed to be superior to other methods [9].

In this work, it was decided that no direct comparison with other tools should be done, due to the inexistence of any tool that executes the entirety of the developed and implemented analysis techniques, unfair comparisons with partial software modules of the pipeline because as they "may be easily abused to prove superiority of their own methods" [9].

3.1.1. Image Registration

The process of image registration is done by properly overlaying two or more images of the same location taken at different time frames and/or from different viewpoints and/or by different sensorial devices. The biggest advances in Image Registration methods have been mainly associated with registration of land images acquired by satellites, the matching of stereo images acquired by several cameras, which allow the measurement of depth and finally the alignment of different medical

modalities, such as magnetic resonance imaging, computerized tomography, ultrasound and different imaging microscopy modalities (see Section 2.4 for a review on Microscopy Imaging) [227]–[229].

When microscopy images are acquired by different camera sensors, the simple superposition of multimodal images fusion normally results in the aforementioned misalignments (see Figure 2.18-A), requiring reliable image registration techniques to properly align such images [227]–[229].

Image registration methods have been divided in four steps: "(i) feature detection (ii) feature matching (iii) transform model estimation (iv) image resampling and transformation" [219]. Based on these steps and the nature of the images the registration methods have been classified as area-based or feature-based [219] (area-based algorithms can also be named as intensity-based, since the feature-based algorithms do not work directly with image intensity values [219], [230]). In this process, one image is used as the reference for the registration, called the 'fixed' image, while the transformed image that is registered is called the 'moving' image [231]. Figure 3.2 shows a temporal analysis (similarly to the analysis related to Tracking Methods performed by [232] and presented in Figure 3.10) of published papers related to image registration methods, along with an intersection with the terms 'area', 'intensity' and 'feature', showing a growing and stable interest in this area in recent years.

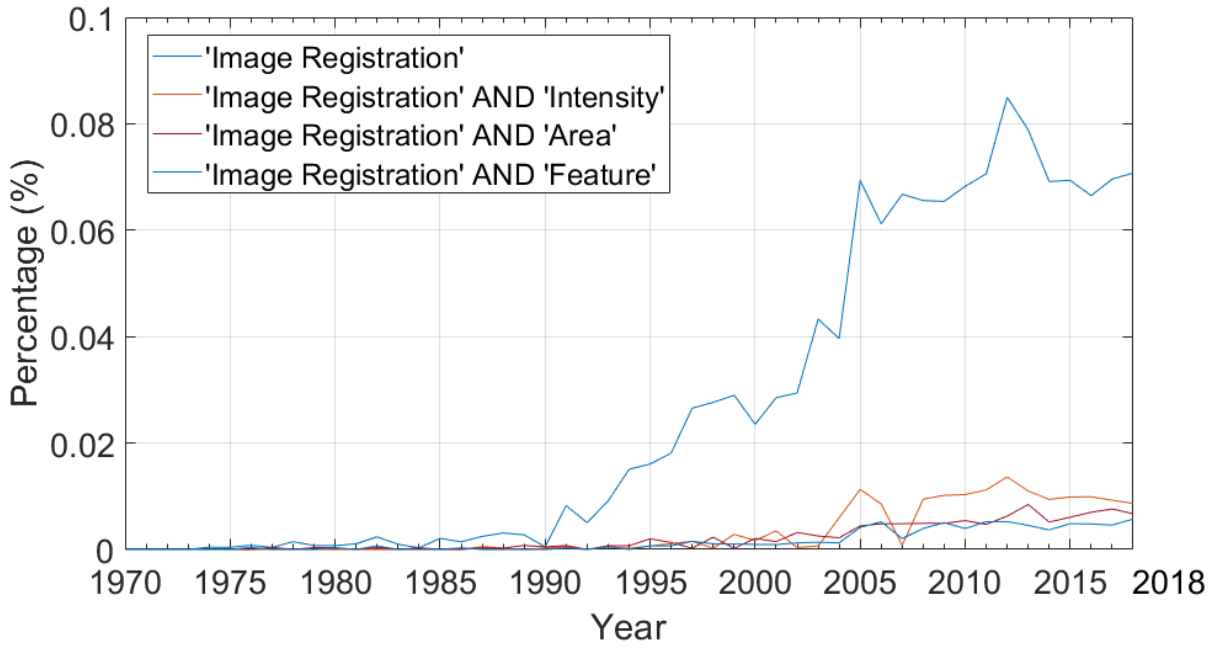


Figure 3.2 – Temporal analysis of publications in the PubMed database (National Library of Medicine, National Institutes of Health, Bethesda, MD, USA) for the indicated combinations of words in the title and/or abstract in the area of Image Registration. The plot shows an increase in the percentage of published papers in image registration until 2010 in the biomedical (and related) literature, with a stable number of papers in the recent years (note that this corresponds to an increase in the total of papers published due to the intrinsic growth of the number of publications in the database). Also plotted is the intersection between 'Image Registration' and three other terms: 'Area', 'Intensity' and 'Feature'. This analysis was done on the 20th of December using https://www.ncbi.nlm.nih.gov/pubmed/advanced.

Area based methods give more significance on the feature matching step rather than on their detection while featured based methods give more emphasis to the detection step [219]. Examples of area-based methods are correlation-like methods (e.g. normalized cross-correlation and its modifications), Fourier methods (e.g. phase-correlation and its modifications to add rotation and scale factors to the transformation), mutual information methods and search techniques based on the sum of squared intensities [219], [228], [233], [234].

The main disadvantage of area-based methods is that they are normally generalized to produce small shifts and rotation, as large scale transformations lead to big computational costs [219].

The aim of the feature-based methods is to find the pairwise correspondence of local structural information (rather than information carried by the image intensities), allowing to even register images of different natures and with large distortions [219]. The main idea behind the feature-based methods is the use of discriminative and robust feature descriptors, that should be invariant to all registered images [219], [228]. These descriptors can be represented by control-points on the images (literal points, end points or centres of line features, centres of gravity of regions, etc.), using their spatial relations invariant descriptors [235]. Other descriptors can be obtained by relaxation methods, pyramids and wavelets [227]. Multispectral/multisensory image registration can raise challenging problems due to different grey level characteristics, making inadequate the application of simple techniques such as those based on area correlation [227].

The main disadvantage of feature-based methods is that in images acquired by a microscopy normally don't have high quality intrinsic features, resulting in a difficulty and a high computational cost of feature detection and feature matching between the images, which is normally solved by using extrinsic features such as fiducial markers). Another drawback is that in a timeseries, these features (both intrinsic and extrinsic) can become unstable over the time of image acquisition [219].

As observed in Figure 3.2, there exists a significant number of published papers that aren't classified as area-based or feature-based (or don't use these terms as keywords), which has prompted the use of a more detailed classification of image registration techniques [234] (based on a survey on medical image registration techniques [229]) separating these algorithms on several parameters:

- **Automatization level**: Automatic or semiautomatic. This can be automatic or semiautomatic depending on user intervention level.
- **Dimensionality:** (from 2D to 2D; from 2D to 3D; 3D to 3D and aligning time-displaced images, using time as a fourth dimension);
- **Domain of the transformation:** (local or global transformations, respectively depending if only parts of the image are registered, or the transformation is applied to the entire image).
- **Method of parameter determination**: Use of direct or search oriented methods to determine the parameters of the image registration transformation.
- **Mode of registration**: Intra-modal registration, when the registered images are from the same modality (e.g. images are captured by the same camera sensor at a different time) or inter-modal, when the registered images are not from the same modality (e.g. images are capture by different camera sensors at the same time).
- **Nature of the transformation basis or Source of Features:** (extrinsic, when it is based on foreign objects that can be introduced in the sample, such as fiducial markers and control-points; intrinsic, when it is based on information provided by the image and non-image based, when for example the two image acquisition devices use spatial coordinates to match the images).
- **Tightness of feature coupling**: The transformation method can be based on the interpolation of the features of previous transformations or can be based on feature approximation.
- **Type of data**: The registered data can be based on raw images, on the features obtained from the image or based on fiducial markers introduced into the images.
- **Type of transformation**: The image registration methods can be based on rigid, affine, projective or curved (nonlinear) transformations (see Figure 3.3 for a representation of these transformations in 2D).

In 2D-2D transformations, rigid transformations only include translation and rotation, while affine transformations include scaling, rotation, translation, shear, and all the combinations of these transformations (all rigid transformations are also affine), while projective transformations include all affine transformations, but also allows the correction of perspective distortions [229], [234].

Rigid, affine and projective transformations can be represented by a Transformation Matrix:

$$T_{matrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} \quad \textbf{(3.1)}$$

with the a1, a2, and a3 and a4 indexes involved in the rotation of the image, a1 and a4 involved in the scaling, a3 and a2 involved in the shear process, b1 and b2 in the translation and c1 and c2 in the projection [229], [234].

Each transformation process can be decomposed into specific matrices (where the indexes not involved in the transformation are mapped to 0 if they are not in the identity line, or to 1 if they are in the identity line). These decomposed matrices can be multiplied to obtain the final Transformation Matrix (T_{matrix}), which can finally be multiplied by each point in the image to obtain the coordinates of the transformed point [229], [234].

Curved transformations cannot be generally described with the same type of matrices, being normally represented by a displacement field (see example in Figure 3.4-B and Figure 3.4-C) or using a specific polynomial function to map each pixel onto the new coordinates al to map [229], [234]. Figure 3.3 shows a visual representation of all the above-mentioned transformations in 2D.

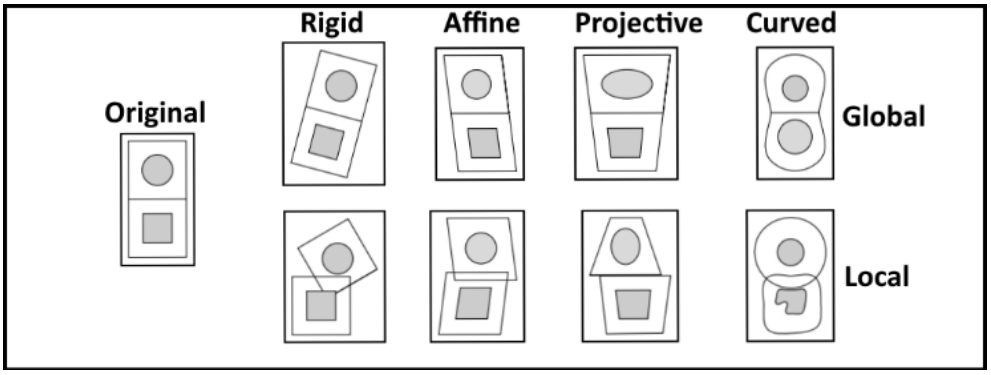


Figure 3.3 – Visual representation of different transformations types. Rigid, affine, projective and curved transformations are presented on a global and local level domain. Adapted from [229].

Other methods based on non-parametric diffeomorphic transformations can also be applied [236]. These methods perform non-rigid registrations by warping the moving image according to the displacement field and has been adapted and optimized from a previously developed method (called Thirion's demons) [237] to provide non-parametric diffeomorphic transformations. This registration method evolves into several localized curved transformations and results in local curved deformations of the image (as seen in Figure 3.4).

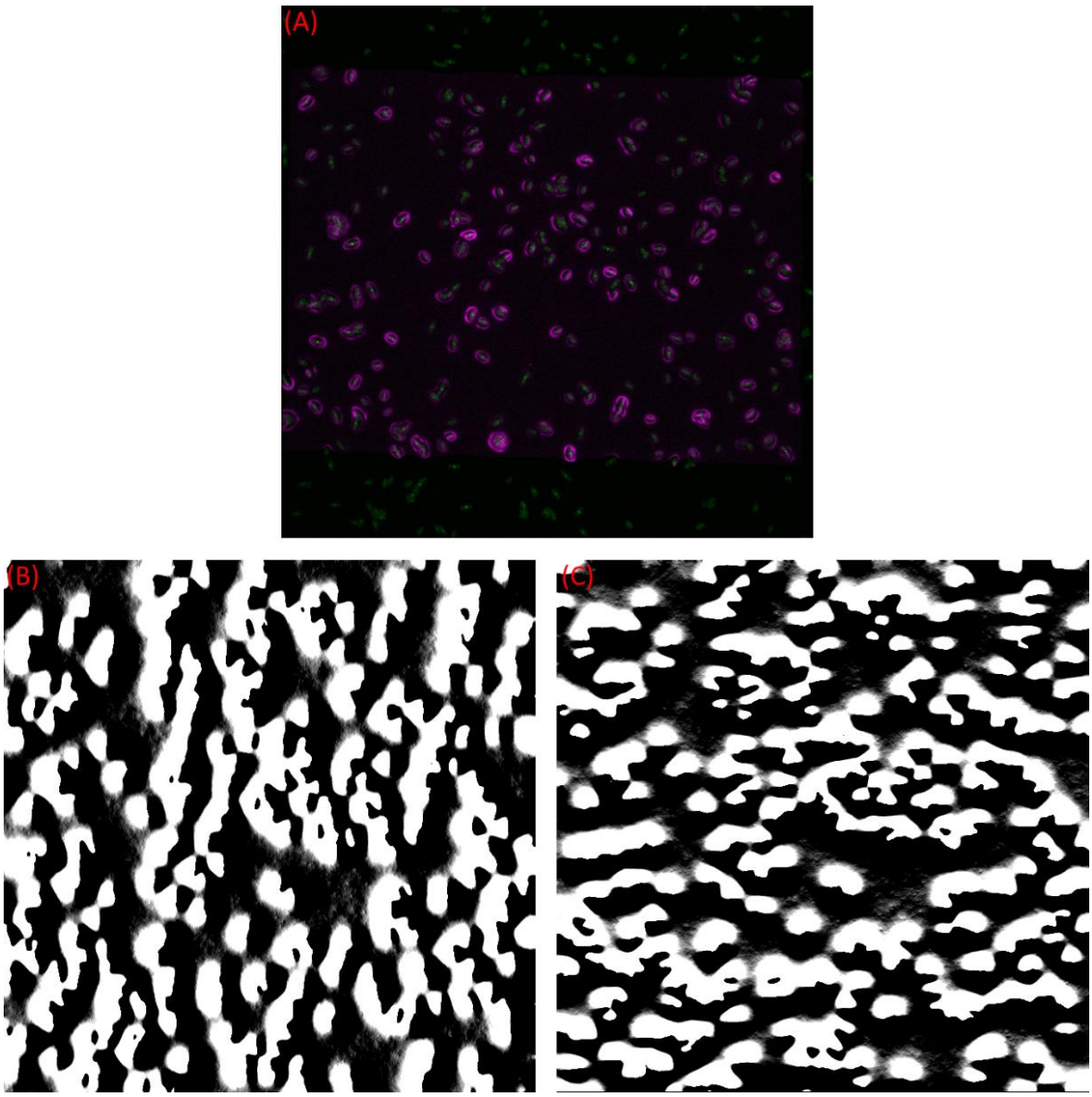


Figure 3.4 - Example of the application of non-parametric diffeomorphic transformations [236]**.** (A) Final Registration based on the estimation of the displacement fields, represented by displacement vectors (B) in the x-axis and (C) in the y-axis.

When the automatic algorithms fail to register correctly the images, it is necessary to extrinsic features, such as the use of fiducial landmarks in live-cell imaging has two major problems: (i) having several fiducials too close to cellular constituents can affect the imaging and (ii) the fiducials on the image might show divergent drift patterns, either due to movement within the sample or due to the variability of the drift within the sample [238], [239]. In bacterial cells, fiducial markers have been especially useful in Super-Resolution Imaging by allowing the drift correction that occurs during image acquisition [240], [241], but some drawbacks have been reported, as the presence of large number of fiducial markers can degrade the quality of the signal that originates from the studied fluorescent molecules, due to the strong fluorescence emission that emanates from the fiducials [241].

If the acquired images lack both intrinsic and extrinsic features, it is required to do a manual adjustment of the image registration, by a manual implementation of extrinsic features, by placing corresponding control points in the fixed and the moving images [242], and using different interpolating functions for the image resampling [243] (linear, near-neighbour and cubic), with the near-neighbour giving the image sampling results.

It should be noted that at least 4 non-collinear control-points are required to create a global projective transformation, with additional points being used for mapping of local errors [219], [244]. It is also recognized that at least N non-collinear control points are necessary to create a global polynomial transformations, with N=(n+1)(n+2)/2, so N=6 and N=10, respectively for 2nd (n=2) and 3rd (n=3) degree polynomials (n=2 and n=3) [245]. The increase in complexity of higher order polynomials and in the minimum number of control-points, has prevented the usage of such polynomials in practical applications, as they may also unnecessarily warp the moving image in the areas not covered by the control-points [219].

Manual processing techniques were the first gold standard for validation of the methods included in the automatic tools for cell parametric measurements (Cytometry) [229], [246]. The ground-truth established by the manual processing can be challenged as they are expert-dependent (repeatability of results depends on the user, and even intra-user variability can be very high) and can become unfeasible for large data-sets (the case of high-throughput microscopy studies), due to becoming a non-viable and time-consuming task [247], providing an opportunity for the use of fiducial landmarks or of simulated data to become the new gold standard validation methodologies [229], [246].

Simulations of biological processes using computational modelling is a viable alternative to create a "ground truth" by producing artificial deformable images that can be used for quantitative validation of image processing algorithms [248], specifically for the validation of image registration techniques, [249], which has been one of the growing trends in microscopy imaging in the last years [10] (see Section 3.2 for the Literature Review of this topic). An image generator capable of simulating artificial images of *E. coli* cells was built in this research work (see Section 4.2), which will be able to provide future validations of the image registration techniques.

To compare each of the described image registration techniques, one must perform a quantitative performance evaluation, which requires the use of reference datasets for validation, using statistical analysis such as Precision, Accuracy or 2D Correlations, as shown in Section 6.1.1. [229], [250].

3.1.2. Cell Segmentation

Image segmentation is the process of separating a digital image into various segments, where the segments of interest are called the foreground and the rest of the image is the background, by attributing a corresponding label (foreground or background) to every pixel in that image [221].

In live-cell imaging studies, the use of segmentation techniques is associated both with the detection of cell boundaries and sub-cellular structures and has been the focus of numerous scientific works in the past decades [9]. This type of techniques allow the study of the cellular morphology and of intracellular processes and structures, which can provide a meaningful integration between morphological and functional cellular features [9], [79], [251], [252].

Various cell segmentation techniques have appeared over the years (see a temporal analysis of the development of cell segmentation techniques in Figure 3.5-A), such as intensity thresholding, region accumulation, feature identification, morphological filters, deformable model fitting and other miscellaneous approaches. The temporal analysis (similarly to the analysis related to Tracking Methods performed by [232] and presented in Figure 3.10) in Figure 3.5-B shows a linear increase in published papers related to cell segmentation in the last 60 years.

A detailed analysis of intensity thresholding, morphological filtering and region accumulation is presented in this section, as these algorithms are the basis of the novel cell and structure segmentation algorithms presented in this research work and have also been extensively in other cell segmentation platforms applied to live-cell microscopy imaging of bacterial cells [253]–[256]. The other identified cell segmentation techniques (see Figure 3.5) are described here in less detail.

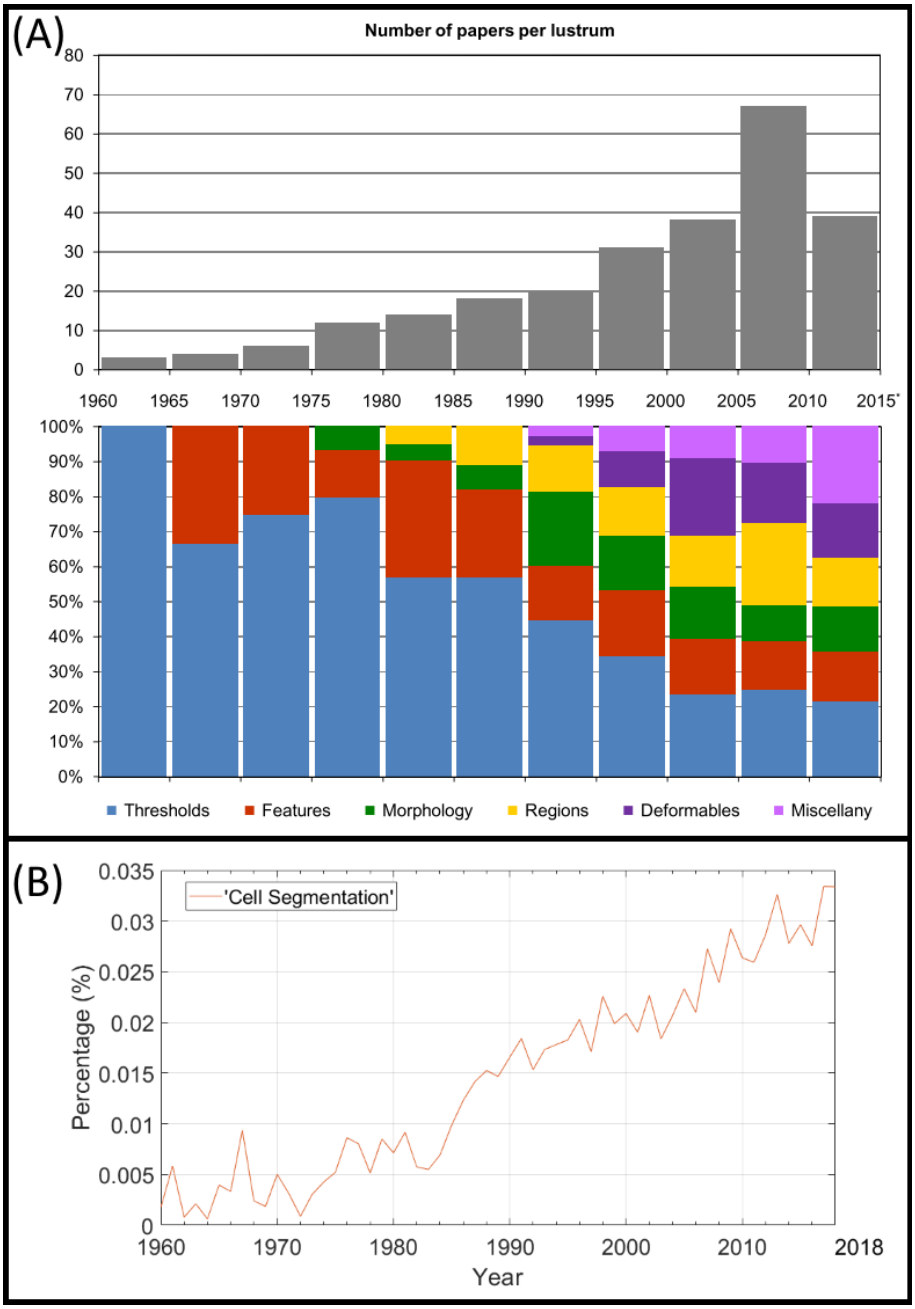


Figure 3.5 – Temporal analysis of cell segmentation techniques. (A) Top graphic shows the number of articles in the area per 5 years, and the bottom graphic shows the evolution of the used cell segmentation methods. These include intensity thresholding (in blue), feature detection (in red), morphological filtering (in green), region accumulation (in yellow), deformable model fitting (in violet) and techniques that were not classified as any of the former (in magenta). Taken with permission from [9], © 2012 IEEE. (B) Percentage of publications in cell segmentation, from the PubMed database (National Library of Medicine, National Institutes of Health, Bethesda, MD, USA), with an increase in published papers over the last decades (note that this increase takes into consideration the intrinsic growth of the total number of publications) This analysis was done on the 20th of December using https://www.ncbi.nlm.nih.gov/pubmed/advanced.

Intensity thresholding (binarization of the image into black and white pixels based on the pixel intensity values) was the first cell segmentation methodology to be developed and is still one of the

most used techniques in cell segmentation, as the background of the images is normally associated with significantly different intensities from the studied cells [9].

A review on thresholding methods has classified them as [257]:

- **Clustering-based methods**: where the grey-level pixels are always clustered into two sets samples (background and foreground). Most of the clustering-based approaches in cell segmentation are based on two different concepts originated from Otsu's thresholding technique [258]. The first, the Global Otsu method [258], which calculates a global threshold value that separates the background from the foreground pixels, by minimizing their intra-class variance. The second is based on the Multi-level Otsu thresholding [258], [259], which is a special case of this methodology because the two-class segmentation problem is transformed into a multi-class segmentation, where the background is still separated from the foreground, but then the foreground can be divided into various clusters. Another main strategy in clustering-based methods was to model that the intensity levels as a combination of two Gaussian curves [260]–[262], with each algorithm using different iterative search methods to determine the threshold, which can be the midpoint between the two peaks of the Gaussian curves [261], [262] or the point that minimizes the misclassification error [260]. Finally, another cluster-based algorithm has been developed by assigning fuzzy clustering memberships to pixels depending on the Euclidean distance to each of the set's (background and foreground) mean intensity [263].
- **Entropy-based methods**: where the maximization of the entropy from the thresholded image is optimized by considering that the foreground and background signals are obtained by distinct sources followed by a maximization of the sum of both signals entropies [264], [265]. A cross-entropy approach, which measures the data consistency between the original and the final binary image, can also be used to calculate a threshold value by minimizing the Kullback-Leibler divergence [266], [267]. Finally, another entropy-based algorithm was also developed by selecting a fuzzy region of the membership function so that the image can be thresholded with maximum fuzzy entropy [268].
- **Histogram shape-based:** where the iterative search for the peaks and valleys of the histogram are used to separate the cells from the background or to separate internal cellular components, without the assumption that the intensity levels are modelled as two Gaussian curves [269]. Other algorithms use different shape-based assumptions, e.g. they approximate the histogram to several rectangles [270], [271] or use autoregressive modelling to force the histogram into a smoothed two-peaked representation [272], [273]. The subtraction of the convex hull of the histogram with the actual histogram can also be used to calculate the optimal threshold value, which should lie on the points with the deepest concavities (different methods can be used to calculate competing points) [274]–[276].
- **Local-based methods:** where a different threshold value is calculated for each pixel, depending on localized statistics (in the neighbourhood of each pixel) like mean, median, variance, range or surface-fitting parameters. One of the most widely used algorithms is based on the computation of a locally adaptive threshold based on first-order statistics (the local mean intensities) [277], which was an extension of a previously developed algorithm, which calculated the moving average based on a specific number of previously scanned pixels (one-eighth of the image width) [278]. The new algorithm added a relevant feature, as it used a fixed window to make an initial scan of the neighbouring pixels resulting in the computation of the average of the fixed window and a second scan to compare that value to the evaluated pixel, setting the value of that pixel to black if it is the evaluated pixel is lower than a specific percentage value of the average (this value can be changed based on a sensitivity factor) [277]. Similar algorithms have implemented analogous scanning algorithms but their decision process has been the calculation of the median [279] or the Gaussian weighted mean [280] of the fixed

window, the calculation of second-order statistics (the local variance) [281] and finally the calculation of a mid-range value (the mean between the minimum and maximum values in the fixed window) [282]. Another algorithm calculates several fixed windows centred around the evaluated pixel and measures the local contrast of each window to perform the binarization process [283]. Finally, a local threshold can be calculated based on the edge and grey intensity levels, which can be used to construct a threshold surface by first thinning the gradient magnitude to yield local gradient maxima, and then fitting several surface functions using a successive overrelaxation method and obtaining the threshold by iteratively applying a discrete Laplacian [284] or by applying a variational methods [285].

- **Object Attribute-based methods:** where attributes found in both the grey-level and the binarized images are matched in terms of quality and similarity. One algorithm has matched a thinned edge field obtained in both the grey-level and the binarized image by applying the Sobel operator for the edge detection, and obtaining the global threshold by computing the value that maximizes the coincidence of both edge fields [286]. A second algorithm has been developed based on fuzzy similarity thresholding, by measuring the distance between the grey-level and the binary images, using different entropy measurements (Shannon entropy, logarithmic entropy, and exponential entropy), and obtaining the optimal threshold value by minimizing this measurement (index of fuzziness) in terms of the fuzzy membership functions of the foreground and background [287], [288]. Other attributes that have been used to compute deterministically the threshold calculation are the matching of the first three grey-level moments with the first three moments of the binary image [289]. The measurement of the size stabilization of foreground objects, using a size-threshold function, has also been used to calculate the optimal threshold based on how many objects possess at least a fixed number of pixels [290], while another algorithm tried to compute the optimal threshold by preserving the connectivity (maximizing the local information within the binarized image) of the segmented regions [291].
- **Spatial-based methods:** where the spatial distribution of the pixels intensity values is studied in the context of correlation with the neighbour pixels, cooccurrence probabilities and models of local linear dependence. The first developed algorithms that explored the pixels spatial information, calculated the threshold value based on the local mean and mode of the grey intensity values inside a three-by-three matrix [292], which was an extension of the implementation of second-order grey level statistics [293] and was later optimized by the use of co-occurrence probability matrixes as an indicator of spatial dependence [294]. One algorithm has been able to capture the localized spatial pixel dependence using binary block patterns of a pre-determined pixel size, calculating the spatial correlation of the pixels using the entropy of these block configurations as their symbol source [295], while another algorithm calculates the threshold value by estimating the probability that a pixel belongs to the foreground or background based on both the joint grey level values of the studied pixel and of its neighbouring pixels, with an optimal threshold value being obtained when this posteriori spatial probability is maximized [296]. Finally, another technique has been developed based on the idea that each grey-scale image generates a random set and that the binary distance transform of the thresholded image can mimic the average distance transform when different threshold values are considered [297].

Morphological Filtering is a technique that has also been used in several cell segmentation algorithms (see Figure 3.5), both as the main segmentation technique or in conjunction with other techniques in Miscellaneous algorithms [9]. Morphological Filtering is based on Mathematical Morphology theories of topology, geometry and group theory, using concepts such as size, convexity, geodesic distances, connectivity, shape, lattices, random sets, graphs, surfaces and other spatial structures [298]. These concepts started to be implemented in the 1960s for image processing purposes, by applying a set of operators that were able to initially perform transformations on binary

images [299]. A similar set of morphological operations was later extended and optimized for applications in grayscale images [300], [301].

An overview on Morphological Filtering [299] catalogued dilation and erosion as the two basic image processing nonlinear morphological operations. Both operations are based on the application of a probe, which is known as a '*flat morphological structuring element*'. These structuring elements are based on small sets of binary valued neighbourhoods with pre-defined shapes and have the same dimensions as the applied images (2-D or 3-D), where a centre pixel (called the origin) of the structuring element is matched with the pixel in the image that will be processed. An example of different types of structuring elements is shown in Figure 3.6.

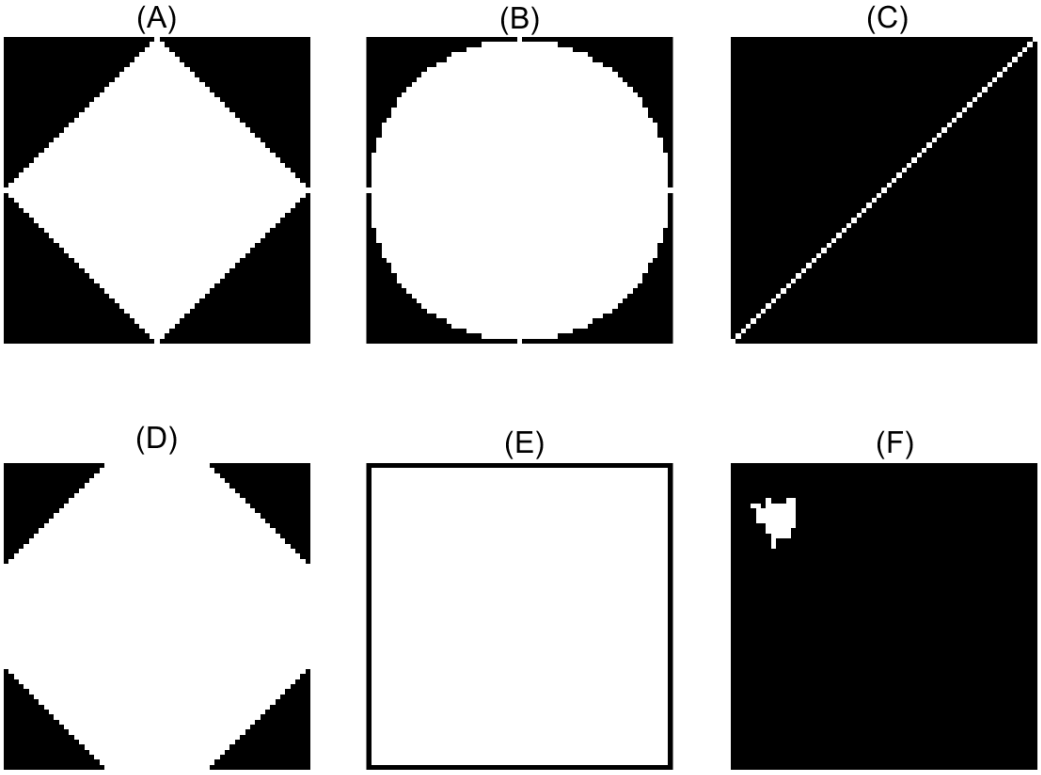


Figure 3.6 – Different 2D Representations of '*flat morphological structuring elemenst*'. (A) diamond-shaped (B) disk-shaped (C) line (D) octagon (E) square (F) arbitrary.

Erosion is the process of removing pixels from the boundaries of the objects inside the image, by applying a rule where pixels with value 1 are changed to 0 if any of the pixels in the neighbourhood have the value 0, while dilations is the process of adding pixels to the boundaries of the objects inside the image by applying a rule where pixels with value 0 are changed to 1 if any of the pixels in the neighbourhood have the value 1 [299]. The successive application and combinations of different methods can be used to implement numerous image processing operations such as [299]:

- **Image opening:** where an image is eroded and then dilated using the same structuring element, which can be used to remove small objects from the image while still able to preserve the size and shape of the large objects.

- **Image closing:** where an image is dilated and then eroded using the same structuring element, which can be used to fill small holes inside an object, while still able to preserve the size and shape of the objects.

- **Image skeletonization:** where all objects are eroded to their centrelines, while preserving the Euler number (so it doesn't remove holes and branches)

- **Object perimeter finding:** where a pixel is part of the object perimeter if t is connected to at any zero-valued pixels and its own value is nonzero.

- **Top-hat transform:** where an image is opened and then subtracted from the original image. The top-hat transform allows the contrast enhancement of grayscale images with nonuniform illumination and can isolate small bright objects in an image

- **Bottom-hat transform:** where an image is closed and then subtracted from the original image. The top-hat transform allows intensity troughs to be perceived in grayscale images

Distinct results are obtained if these operation are done in either binary or grayscale images, since in the first, these operations are mostly used to polish and smooth the segmentation results, while in the second, they are used to amplify or remove features from the image, which is used as a prepossessing step during segmentation [300].

As previously stated and observed in Figure 3.5, another alternative method is the use of region accumulation algorithms [9], with the one of the most popular algorithms is the Watershed Transform [302]. The Watershed Transformation mimics the idea of a water source that starts flooding the image from each regional minima of the image [302]. This process continues until the algorithm finds the so called "watershed ridge lines", which the correspond to the highest peak in intensity levels (normally light pixels are represented by high elevations and dark pixels represented by low elevations) [302]. An example of the implementation of the watershed algorithm on the segmentation of bacterial colonies [303], is shown in Figure 3.7, representing the "watershed ridge lines" that can split the regions of two touching bacterial cells.

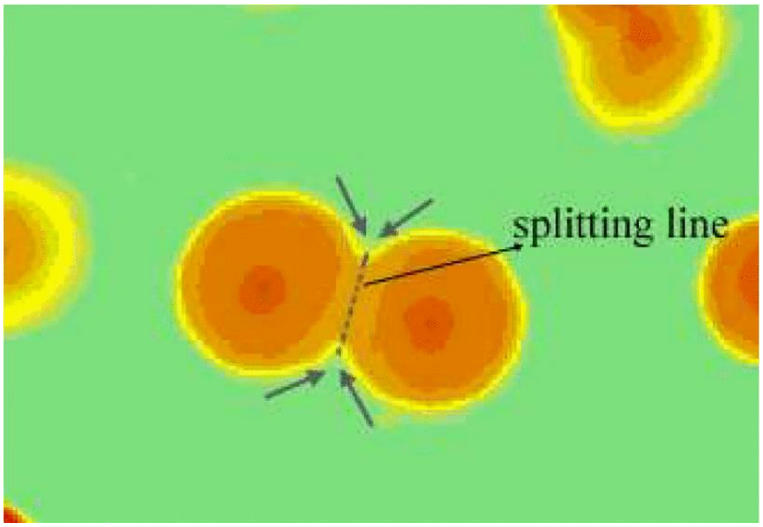


Figure 3.7 – Segmentation of bacterial colonies with the watershed Algorithm. The splitting line is where the segmentation algorithm stopped its execution and divided the region into two cells. Taken with permission from [303], © 2009 IEEE.

Another region accumulation algorithm is the one initially developed by Mora *et al.* (from the CA3-UNINOVA group) for the segmentation of Drusen's in Retinal images [304]. This method (called Gradient Path Labelling or GPL) was later adapted to segment both the external border of *E. coli* cells and also to the segmentation of cellular structures such as the Nucleoid [305]. This method starts by labelling each pixel based on their gradient azimuth, and propagates these labels based on their

gradient paths. Afterwards, the labels are reduced by applying identity rules and forming segmented regions (e.g. two labels are considered equivalent and joined in the same segment when both belong to the same maximum gradient).

It has been reported that both the Watershed [9] and the GPL algorithm [306] have a tendency for over-segmentation when acquisition conditions are not perfect, which is common in most of the microscopy modes in high-throughput experiences, due to the high variability of the illumination and contrast conditions, even within the same time series [307]. The over segmentation problem has been shown to be corrected by using Machine Learning Algorithms (which will be reviewed in Section 3.3), like Classification Trees [308], that were previously trained in to merge or not merge these over segmented areas [306], as can be observed in Figure 3.8.

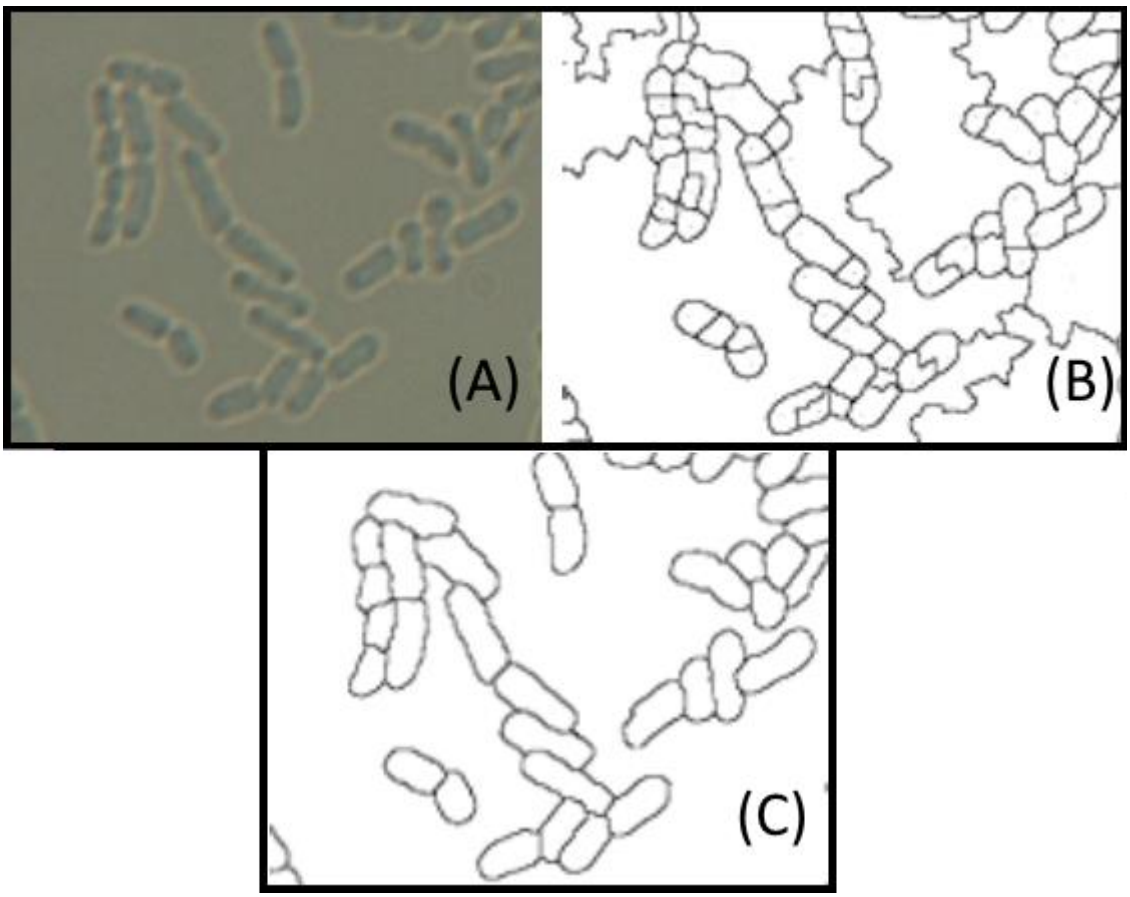


Figure 3.8 – Correction of the GPL over-segmentation. (A) Original DIC image with *E. coli* cells (b) Final step of the GPL segmentation showing over-segmented regions (C) Corrected segmented regions after the implementation of Machine Learning algorithms. Modified with permission from [306].

Other similar methods are the so-called Region-Growing algorithms or the Region-based Active Contour model (also called snakes) [309]–[313], which work by manual or automatic placement (based on certain parameters) of seeds in the image. After this placement, the segmentation is grown based on the local or global intensity of the neighboring pixels of the seed, e.g. large differences in the pixel intensity of one of the 4 or the 8 neighbor pixels (depending on the used connectivity) will halt the growing of the segment in the neighbor direction. An example of the implementation of the Active Contour model on cocci bacterial cells is shown in Figure 3.9.

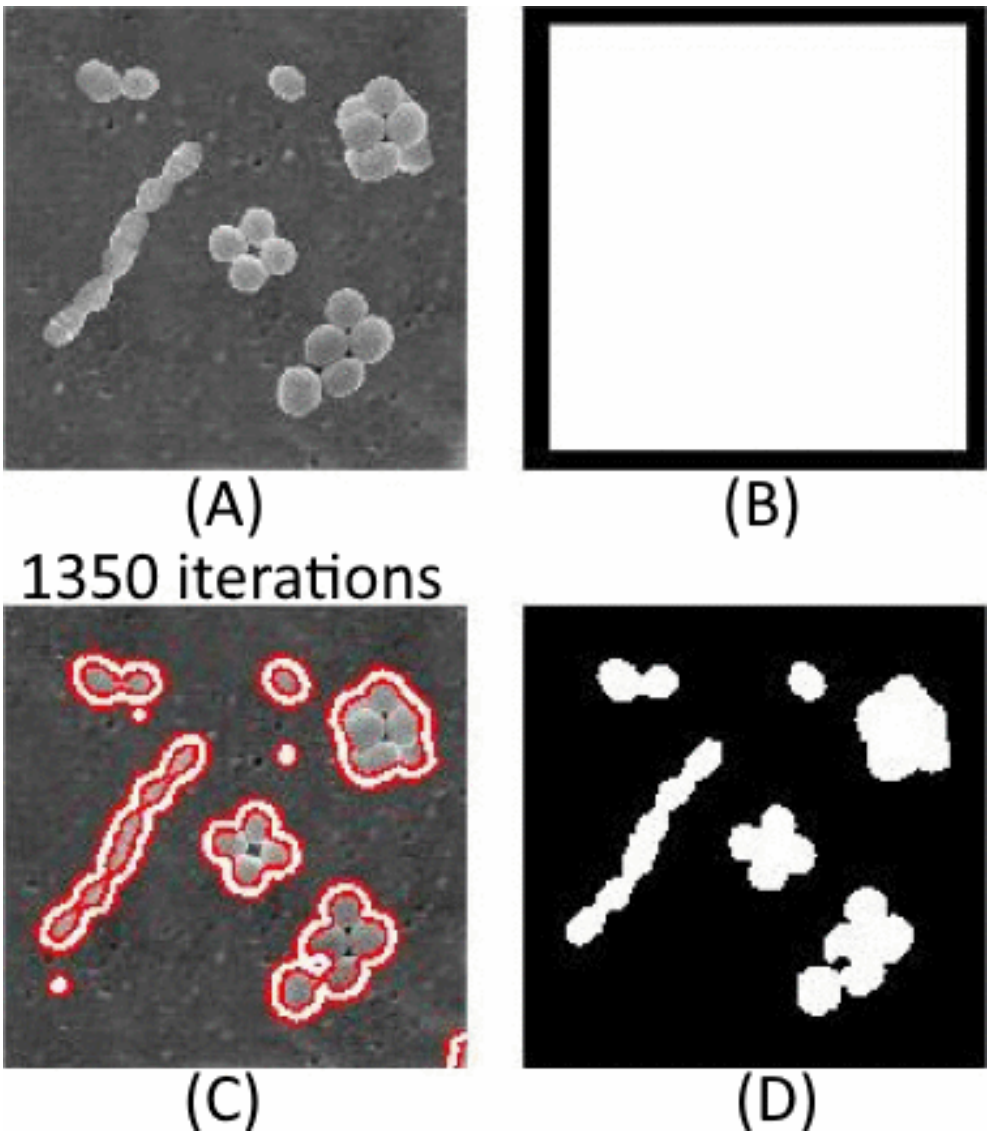


Figure 3.9 - Segmentation of cocci bacterial cells using the Active Contour algorithm. (A) Original colour image of cocci cells (B) Region of interest based on a rectangles selection (C) image after 1350 iterations of the Active Contour algorithm (d) Final segmented image using the Active Contour algorithm. Adapted with permission from [313], © 2010 IEEE.

Finally, the last cell segmentation approach that was reported in (see Figure 3.5) is the use of deformable models, which have mainly been used in medical image segmentation [9], [314], but have also been reported to be used in image registration techniques [315] and image registration techniques [9]. Deformable models are defined as parametric curves (in 2-D) or surfaces (in 3-D) or as the zero-level of a function with one dimension higher than the segmented image. These models can be modified due to the influence of both internal and external forces (as defined by the shape of the curve and by image-based terms, respectively) [9], [314]. The use of deformable models is completed by iteratively moving these curves to minimize a predefined energy functional, with the internal forces, adjusted to keep the model smooth during each iteration and the external forces adjusted to actually moving the curves towards the final segmented shape [9], [314].

The efficiency of these techniques normally is limited to high contrast images with well-defined cell wall limits and uniform illumination [307]. Nowadays (as observed in Figure 3.5) most of the newly developed algorithms use a mixture of several approaches and started to be available in open-source or commercial platforms (a review on these software's is provided in Sub-Section 3.1.4) , in order to increase testability of all those methods described in the literature [9].

3.1.3. Cell Tracking

Cell tracking is the process of characterizing the movement of cells within their surrounding environment, using the segmentation provided by the previously described algorithms along a time-series, providing useful knowledge about the mechanobiology of cell growth, cell division and motility (see Chapter 2) [232], [253].

In live-cell imaging studies, the use of cell tracking techniques and the development of cell tracking software has been associated both with the tracking of entire cell populations and sub-cellular structures during a timeseries, which can provide a meaningful integration between spatial and temporal cellular features of cell organization (see Chapter 2) [232], [253]. Tracking software's and techniques have been the focus of numerous scientific works in the past decades [232], [253], as shown in Figure 3.10.

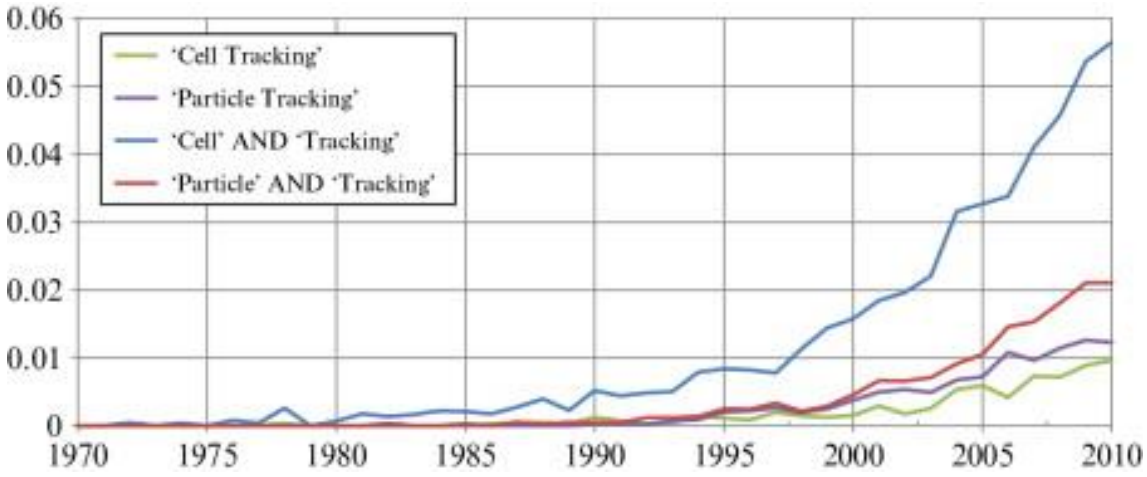


Figure 3.10 – Temporal analysis of publications in the PubMed database (National Library of Medicine, National Institutes of Health, Bethesda, MD, USA) for the indicated combinations of words in the title and/or abstract in the area of Image Tracking. The plot shows a continuous increase in the percentage of published papers in image registration in the biomedical (and related) literature. This analysis was done on the 20th of December using https://www.ncbi.nlm.nih.gov/pubmed/advanced.

Several object tracking methods have been proposed, differing on the number of tracked objects and the consideration of the object's features and type, since the tracked objects can be represented through points, geometric shapes, silhouette and contour, articulated shape model or skeletal model, leading to different developmental approaches [220]. All of these characteristics have an impact on the decision of which method should be used [220] and due to this, tracking methods have been divided into three main categories: Point Tracking, Kernel Tracking and Silhouette Tracking [220].

In the Point Tracking category, the objects are represented by points and tracked based on their position and motion. The main issue with the application of this methodology is the presence of occlusions and the entries and exits of objects from the field of view. This Point Tracking category has been sub-divided into Deterministic and Statistical methods [220]. Deterministic methods associate each object with the implementation of motion constraints, while statistical methods use estimations of random perturbations and noise during the tracking process [220].

The Nearest-Neighbour (NN) association algorithm is the main source of the deterministic approaches. The NN algorithm is based on the calculation of the spatial distances between all objects

in one frame with all objects in the previous frame, and matching the pairs with the smallest distances [316]. This spatial distance can be based on position, shape, colour and size [316].

The combination of the NN algorithm with descriptors based on the scale-invariant feature transform, efficient sub-window search and an updating and pruning method to achieve balance between stability and plasticity was proposed as an efficient visual object tracking algorithm [317], being able to handle occlusions, clutter, and changes in scale and appearance.

The joint probabilistic data association filter (JPDAF) together with the probabilistic data association filter (PDAF) are the main methods behind the statistical approaches. PDAF uses a weighted average of the measurements as input, modelling only one target and considering linear dynamics and measurement models while JPDAF can be seen as an extension of PDAF because it allows multiple targets to be tracked, while both have the same assumptions during the calculation of the target's association probabilities jointly. In both methods, if the model is linear, then the Kalman Filter has a relevant influence. One of the problems of these methods is the incapacity to recover from errors, because only the last measurement is used [316]. The Kalman filter is an optimal estimator, which means that it assumes parameters from indirect, inaccurate and uncertain observations and if all noise is Gaussian, the linear Kalman filter minimizes the mean square error of the estimated parameter. This filter is widely used to obtain the optimal state estimate [316].

A different method [318] combining the JDPAF and a particle filtering [319] was proposed and was named 'Monte Carlo JPDAF'. This method uses three models: the first with near constant velocity, the second with near constant acceleration and a third with both models, which achieved the best performance.

Another statistical method is the multiple hypothesis tracking (MHT), which is one of the most used with point features, but has computational limitations both in time and memory [320]. This method postpones data association until enough information is available. The MHT starts by formulating all possible hypotheses, which develop into a set of new hypotheses each time new data arrives, generating a tree of hypothesis [316]. For each hypothesis, the position of the object in the next frame is predicted and then compared with the measurements, calculating their distance. The associations are made for each hypothesis, generating new hypotheses for the next iteration [220]. The tree of hypotheses should be cut, because it grows exponentially with the measured data. This can be done by clustering, i.e., measurements are subdivided into independent clusters. If a measurement cannot be associated with an existent cluster, a new one is created. Another way of cutting the tree is pruning, meaning that as new iterations are added, a part of the tree is deleted [316].

When tracking objects, one usually obtains multiple measurements of probability and the object with the highest probability is then selected as the next target in the trajectory. If the algorithm selects the wrong measurement or if the correct measurement is not detected, a poor state is estimated. To solve this issue (reducing the computational cost), a validation region (measurement gate) is selected. The measurement gate is a region in which the next measurement has a higher emergence probability [316].

These probabilistic methods have mainly been used to track not only intra-cellular structures (like the ones described in Section 2.3 or homolog structures) [321] but also entire cells [322], [323]. Some works going forward on this approach started to combine multiple methods, including Nearest-Neighbours, Kalman filters and the Multiple Hypothesis Tracking allowing the observations from both the previous and the upcoming images to be used to smooth the object trajectory [324], as can be

seen in Figure 3.11. Unlike PDAF and JPDAF, the MHT method can deal with objects entering, exiting and being occluded from the field of view.

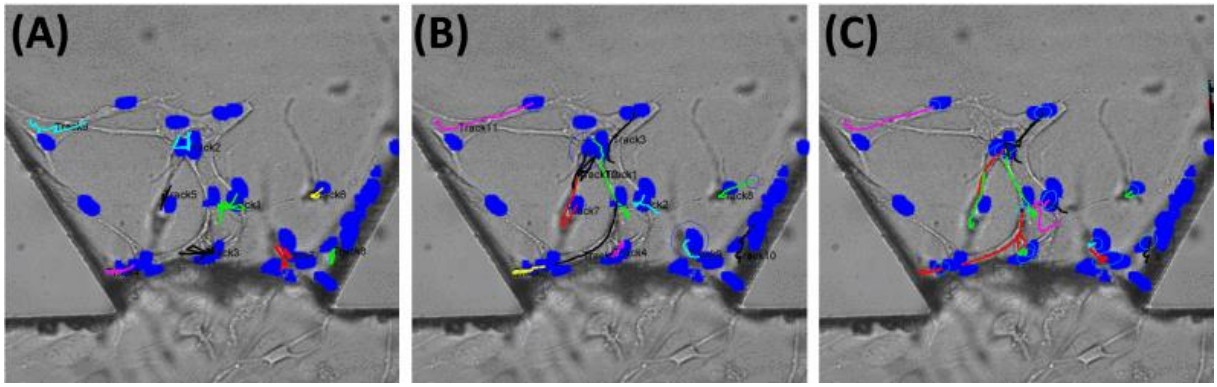


Figure 3.11 – Results for different smoothing methods. (A) nearest neighbour showed the worst results, with several incorrect associations and trajectories. (B) the Kalman filtering showed intermediate results, while (C) the multi-hypothesis tracking showed the best results in tracking cell proliferation. Modified with permission from [324].

Kernel Tracking can be applied with the use of templates and density-based appearance models or multi-view appearance models. Templates are based on basic geometric shapes, while multi-view models encode different views of the object. Mean shift and KLT (Kenade-Lucas-Tomasi) are examples of template and density-based appearance models [220], respectively.

In the Mean shift algorithm, the appearance of tracked objects is defined by histograms while similarities are measured using the Bhattacharyya coefficient [325] and the Kullback-Leibler divergence [326] and then converging towards an ideal object tracking by increasing the similarity between histograms at each iteration [327]. The KLT is an optical-flow method, which uses vectors to show the changes in the image (i.e. translation). A version of this method was proposed in which the translation of a region centred on an interest point is iteratively computed. Then, the tracker evaluates the tracked patch, computing a transformation in consecutive frames [328].

Both methods (Mean shift and KLT) are effective while tracking single objects, but have problems dealing with multiple objects. Silhouette Tracking consists in using precise information about the shape of the objects, using Shape Matching and searching for an object silhouette and its model in each frame. Each translation from frame to frame is handled separately by finding corresponding silhouettes detected in two consecutive frames. Another approach is based on the evolution of the object contour, connecting the correspondent objects by state space models or by minimizing the contour energy [220].

3.1.4. Image Processing Toolboxes

With the development of novel image processing techniques, various computational toolboxes have been published online and publicly available as open-source platforms [329], but most of these developed solutions have been applied into isolated applications, where dedicated and automatic solutions are developed for a specific problem. One of the biggest challenges is to design methods sufficiently generic (automatic or semi-automatic) in order to attain a high specificity and sensitivity for an extensive range of cases [9]. This section provides a small review of various microscopy image processing toolboxes, starting with toolboxes that were applied mainly in eukaryotic cells, followed by toolboxes that were mainly developed towards the segmentation of prokaryotic cells.

❖ ‘CellProfiler’ – This is one of the open source platform that has generated the largest scientific impact (has been cited in more than 6000 publications) [330]. Cell segmentation in ‘CellProfiler’ is normally performed in two steps. First, it uses a block-wise Otsu threshold [258], followed by using a bilinear interpolation, which is applied to separate each cell colony from the background [330]. The second steps used the intensity or shape as a feature for discrimination and segmentation of clumped objects [330]. A revised version of the ‘CellProfiler’ platform (more robust and user friendly) was developed in 2011 (2.0), with the implementation of new segmentation algorithms and new features to facilitated high-throughput works [331]. The user interface of ‘CellProfiler’ 2.0 is presented in has been used to automatically identify and measure various eukaryotic constituents in images, including studies in human cells, mouse cells, yeast colonies, *C. elegans* colonies and other eukaryotic species [332]–[334]. A newer version of the ‘CellProfiler’ platform (3.0) was presented in 2018 [335], with newly developed image processing algorithms for image filtering and noise reduction, cell segmentation, mathematical morphology operations, detection and extraction of cellular features. This version now supports the analysis of entire image volumes (volumetric analysis) and also allows the possibility of a separate “plane-wise” analysis of two-dimensional slices stemmed from the three-dimensional (3D) volume of the object, with an allocated cloud-based framework that further improves the analysis of high-throughput works and a newly developed plugin enable running pretrained deep learning models.

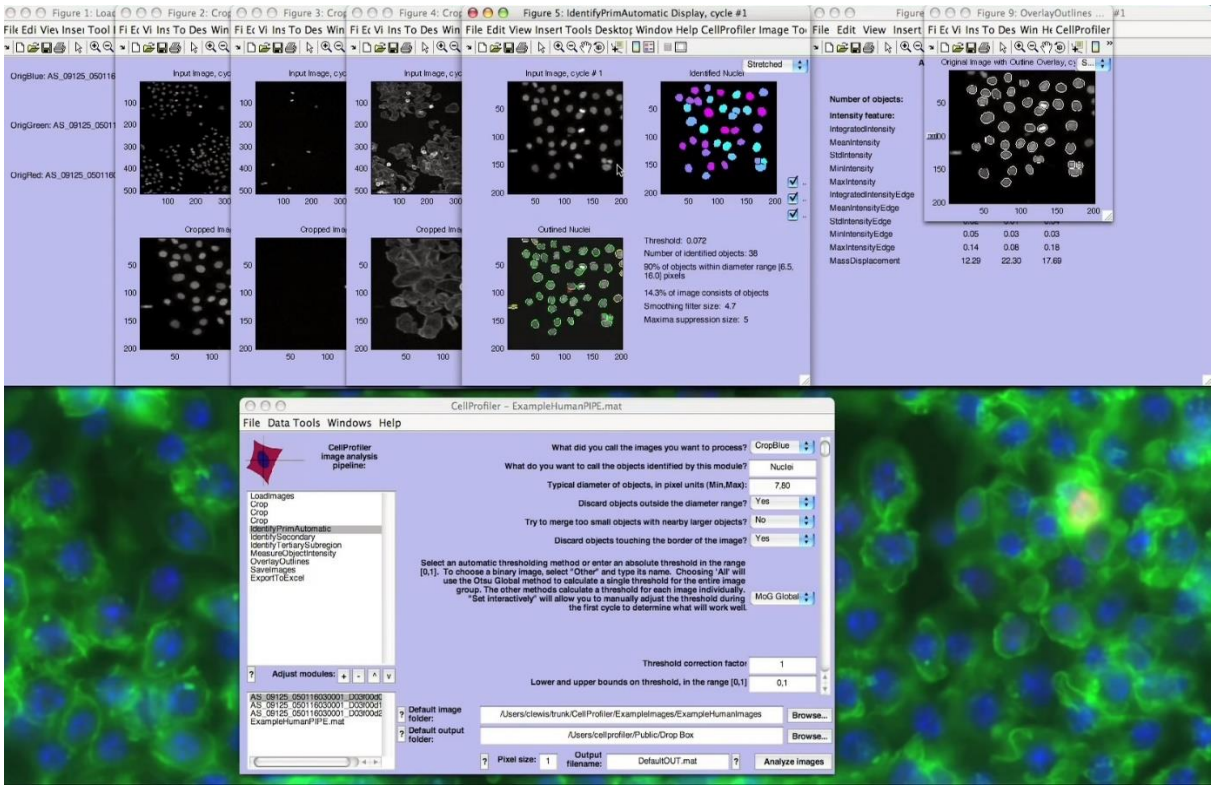


Figure 3.12 – Graphic User Interface of ‘CellProfiler’ 2.0. An example of the results from each module is presented (e.g. ‘Load Images’, ‘Identify Primary, Secondary and Tertiary objects’, ‘Measure objects intensity’, ‘Overlay images’, ‘Save and Export Image’s). Taken from a publicly available website [336].

❖ ‘Cell-ID’ – This tool was originally optimized for segmentation of bright-field images of yeast (mainly Saccharomyces cerevisiae) and other cell types [337]. In bright-field microscopy, the images are taken beneath the focal plan and the cell border pixels are both darker than the image background and darker the cell internal pixels. This is useful for cell segmentation, as it allows that the application of a threshold cut-off value (reported to be 3σ above the background of the fluorescence image in the

[337] case), can be used to independently and automatically for each image separate cells from the background [337]. Cell borders were then aligned with fluorescence images to calculate various cellular parameters (volume, total, total and subcellular fluorescence localization). An updated version of the 'Cell-ID' platform (V1.4) was used in conjunction with the statistical programming framework R to provide a tailored data analysis of both yeast and mammalian cells [338]. Cell segmentation was still done on brightfield images, to avoid the cell wall labelling with fluorescent images, allowing a more efficient scoring of the analysed fluorophores inside the cell wall and avoided the photobleaching of the mentioned fluorophores [338]. Figure 3.12 shows the Graphic User Interface (GUI) of the 'Cell-ID' platform (V1.4) analysing mammalian live cells. This cell segmentation technique can result in an incorrect join of cells that are touching [337]. This was corrected by a cell splitting algorithm, which checks if the minor distance between any two points of the cell boundary divided by the minor axis of both newly divided cells is lower than a pre-defined value (0.5 in Figure 3.12) [338].

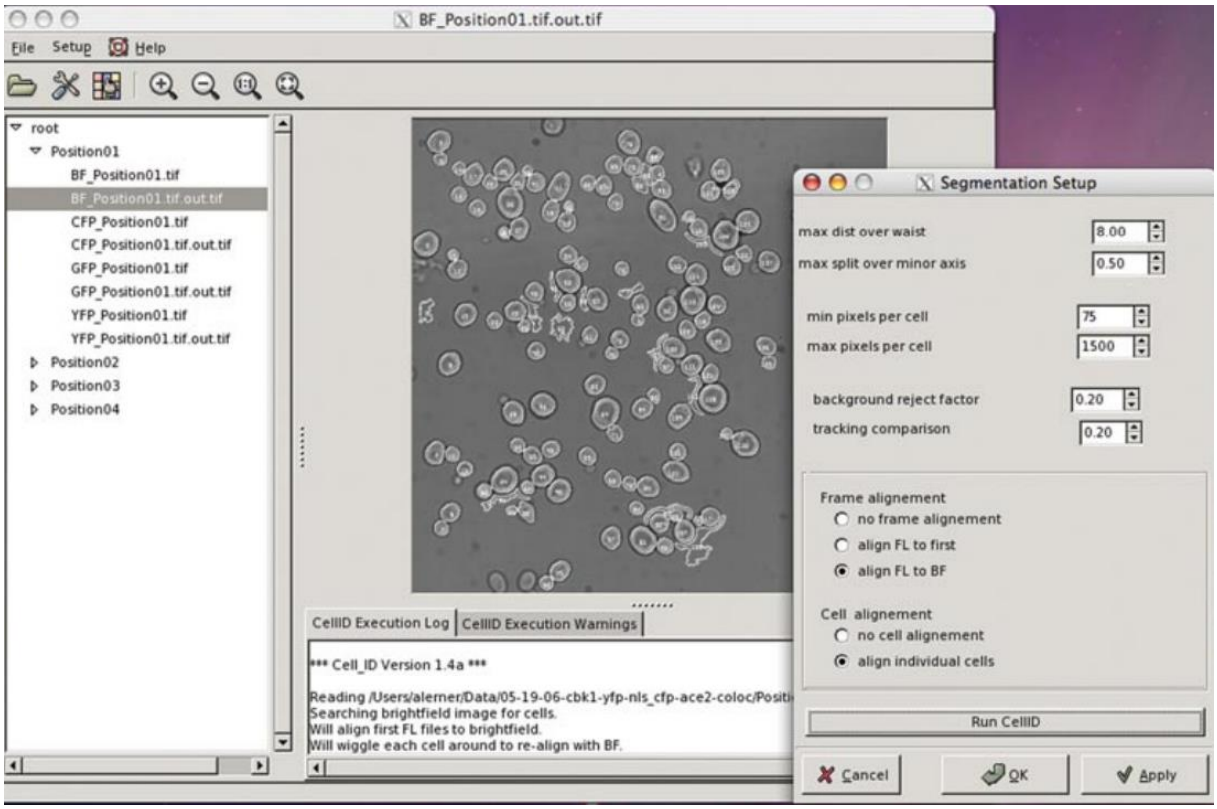


Figure 3.13 – Graphic User Interface of the 'Cell-ID' 1.4 toolbox. Analysis of HEK293 cells (mammalian lymphoid), showing the cell segmentation options: cell splitting algorithm, background correction, fluorescent image alignment and cell alignment/cell tracking. Taken with permission from [338].

❖ 'CellTracker' – This toolbox was especially developed to track the movement of living cells and also to automatically segment cell boundaries and tracking of nuclear and cytoplasmic activity by quantifying the intensity of fluorescently tagged proteins [339]. Cell borders were detected via thresholding and level setting and refined by detecting the cell edges based on an active contour algorithm [339], [340]. An updated version of the 'CellTracker' implemented an improved cell segmentation algorithm that can separate clustered cells, based on the geodesic commute distance (fusion of a geodesic graph-based methods with random walk) to classify pixels [341]. This toolbox is still available online, but its support has been discontinued.

❖ Another toolbox with the same name ('CellTracker') was recently published and developed in MATLAB® [342]. Automated and semi-automated segmentation algorithms were developed and

mainly optimized for Phase-Contrast and DIC images of human cells (e.g. U87 glioblastoma cell lines, as observed in Figure 3.14). While the fully automated is based on the combination of template matching and a tracking algorithm [343]. The semi-automated method is based on the manual selection of the cells and a posterior automatic matching of a specific template for each cell with the best matching template of another cell in the next consecutive frame, which is adaptive process that can able to handle slight cellular deformations [342]. This 'CellTracker' platform also allows a fully manual tracking option, based on a point-and-click resource, allowing the user to define the position of each cell over the studied time frames [342].

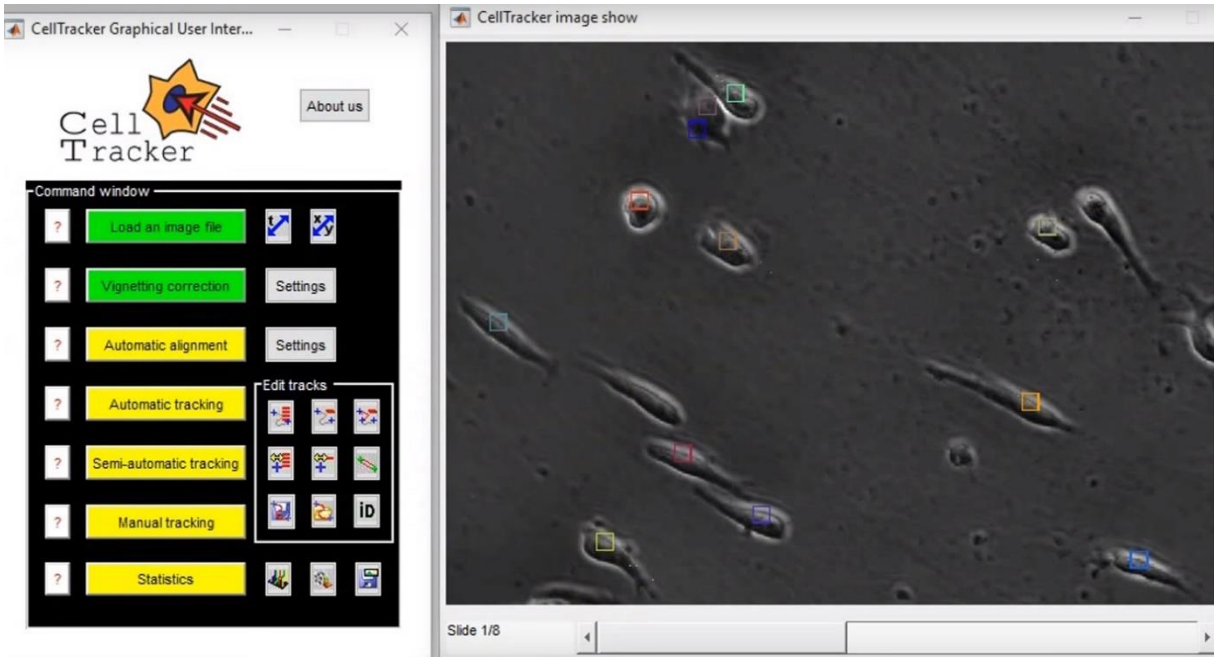


Figure 3.14 - Graphic User Interface of the second 'CellTracker' toolbox. Analysis of U87 cells migrating on fibronectin-coated 6-well plastic cell culture plateEK293, showing the interface options: image loading, vignetting correction, automatic alignment, automatic tracking, semi-automatic tracking, manual tracking and statistics. Adapted from [344].

❖ 'Farsight' toolkit – this toolbox has a segmentation algorithm that was developed to study detailed biological microenvironments, such as histopathology images of the brain and other tissues [345], [346]. The methods present in this toolkit were based on graph-cuts algorithms that can segment foreground signals from the image background. Then, the nuclear seed points are detected by a multiresolution edge detection method, based on Laplacian-of-Gaussian filters limited by an adaptive scale selection of distance-maps and refined by a second graph-cuts algorithm [345].

The algorithms implemented in the above-mentioned tools use a variety of cell segmentation approaches and have been used mainly in eukaryotic cells. The major drawback of using the same segmentation algorithms in prokaryotic cells is that these cells are organized in large and dense clusters. The main consequence of segmenting such clusters, is that accuracy of the algorithms will decrease as the clusters density increases, as its success depends in the initial marking and identification of cell boundaries, which can be a difficult task when cells are tightly clumped together, reducing the possibilities of portability of using such methods in bacteria segmentation [347]. Due to this problematic, some platforms and methods were specifically developed for segmentation and tracking of prokaryotic cells in different microscopy modalities (see Section 2.4.2):

❖ 'CellC' – This tool has mainly been used for the segmentation of bacterial cells (and then counting cell numbers and analysing cell characteristics) in microscopy imaging and initially tested in

fluorescence in situ hybridization (FISH) and 4',6-diamino-2-phenylindole (DAPI) images [348]. The source-code of this software (developed in MATLAB®) was publicly released and the graphic user interface is shown in Figure 3.15. The 'CellC' software segmentation routine starts with the background correction of both uneven illumination problems and the removal of background autofluorescence [348]. Segmentation is done based on automatic global thresholding techniques and the watershed technique [9] is used to separate clusters [348]. Post processing methods in 'CellC' include the removal of holes in cells, removal of small objects (that can result from the over-segmentation of watershed techniques and removal of cells with large areas (this could be the result of dense groups of cells that were not separated [348]. The segmentation results from 'CellC' were positively compared with image processing tools developed in ImageJ, based on a manual cell counting validation [348]. The 'CellC' platform was initially used in several studies of different prokaryotic cell types [349]–[351], but it was later used to analyse eukaryotic cells (e.g. counting fibroblast cells in wound healing studies [352], [353] and counting axons and myelin sheaths from nerve cells to study inflammatory demyelination in Multiple sclerosis [354]).

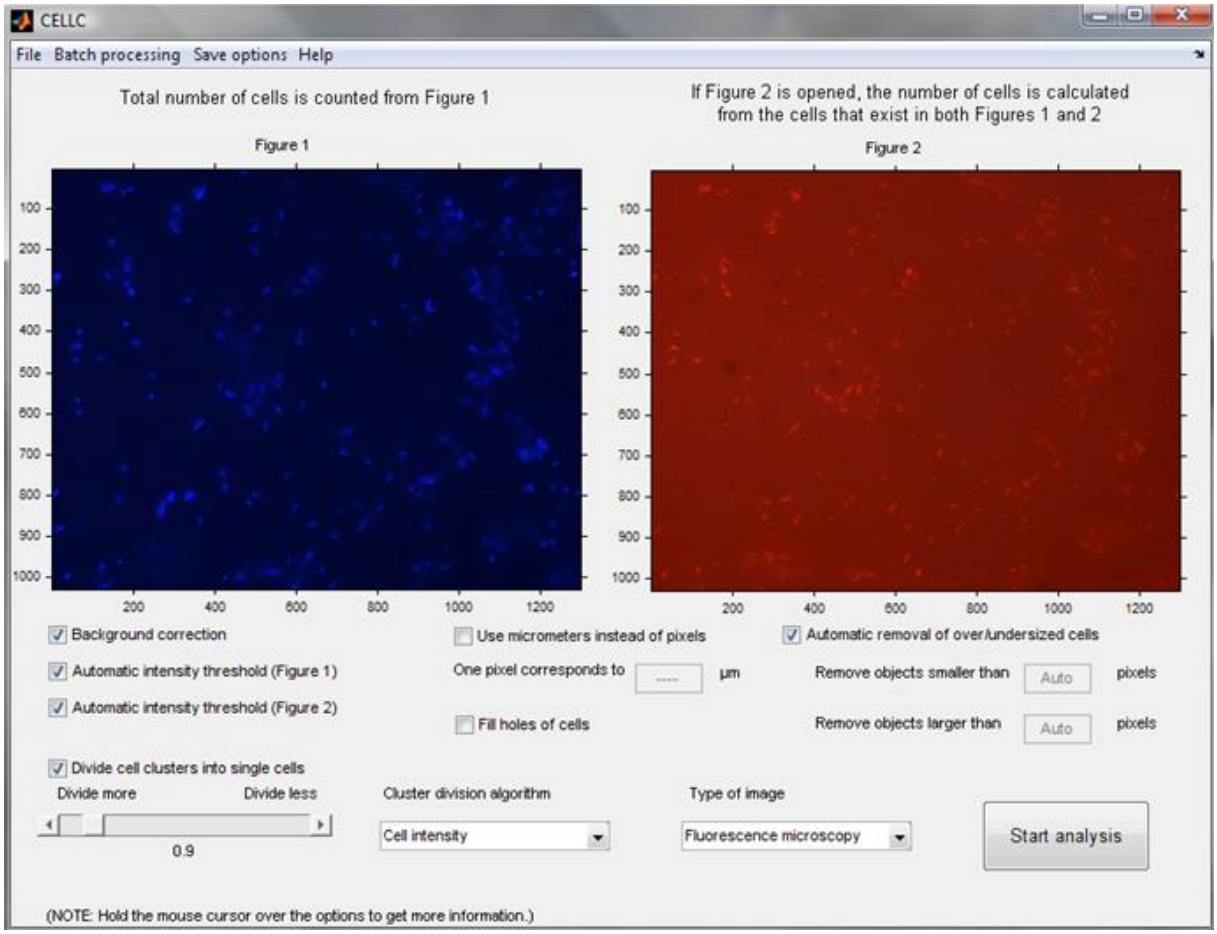


Figure 3.15 - Graphic User Interface (GUI) of the 'CellC' software. Analysis of DAPI-stained bacterial cells (right) and FISH images of bacterial cells (left). Image processing options are shown in the GUI, e.g. (background correction, intensity threshold, removal of oversized and undersized cells and division of segmented clusters in individual cells.

❖ 'CellTracer' – This tool was developed in the MATLAB® environment and using the built-in Image processing tool box (with its source-code of this software publicly released and the graphic user interface is shown in Figure 3.16) [355]. For *E. coli cells* (example shown in Figure 3.16), the proposed segmentation strategy starts with a pre-processing cropping, allowing the user to focus on the visible cell clusters, followed by the employment of a screening algorithm to identify the background and progressing into the application of a Thresholding & Smoothing algorithm that is capable of cell border

identification (in this step several parameters need to be inputted by the user, such as maximum half-cell width, maximum ranking, minimum ranking and global threshold values and the structure element radius). The previous step is followed by a Robust Voting algorithm to further identify the cell border regions (in this step more parameters need to be inputted (minimum ranking threshold, minimum border volume, minimum and maximum half-cell width [355]. The next step is to apply a Convex Model algorithm to identify cells (the parameters chosen for this algorithm assume the rod-shape of the *E. coli* cells, such as the minimum cell score and volume, the structure element radius and a smoothing parameter) [355]. Afterwards, a Global Alignment step is executed, followed by a cell tracking Neighbourhood-based algorithm (in this step, the user also selects several parameters such as the maximum cell displacement, minimum overlapping score, Neighbourhood Size and Scale Factor [355]. Further developments exhibited a satisfactory portability to various types of cells (e.g. Budding yeast and nerve human cells) by just changing parameters in the above-mentioned steps, by skipping some steps or by changing the algorithm in other steps. The new developments also integrated various types of microscopy imaging, namely Phase-Contrast, bright-field and fluorescent [347].

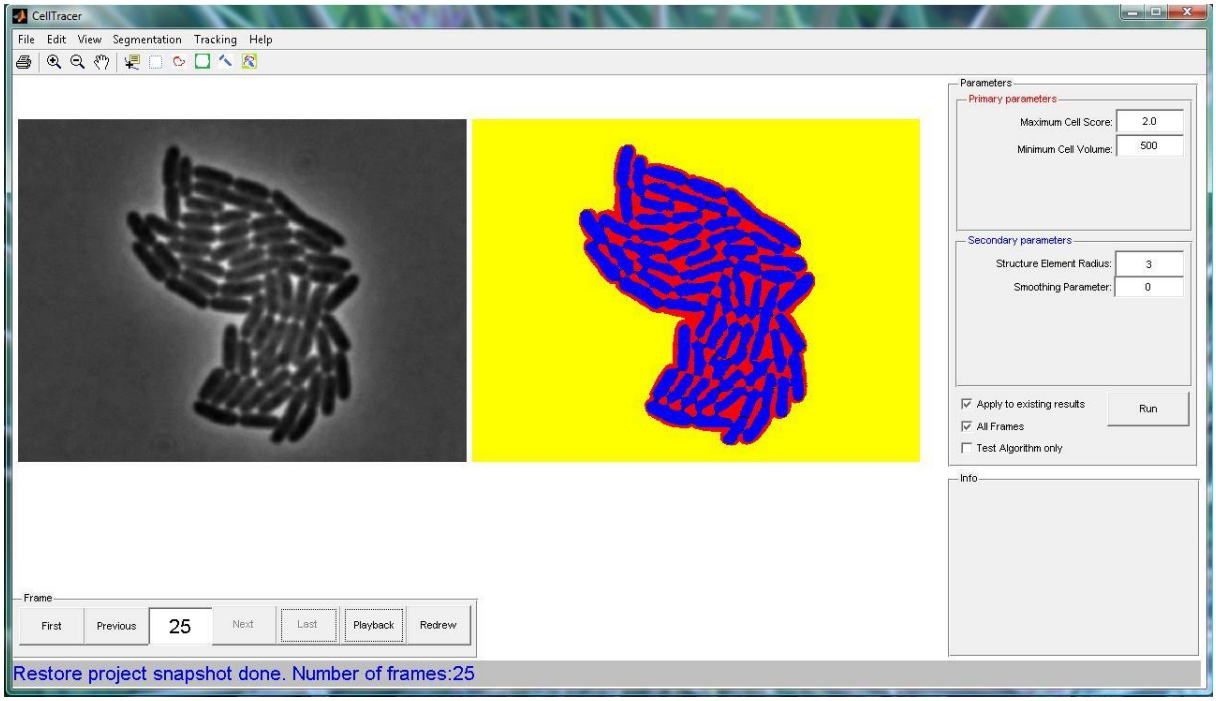


Figure 3.16 - Graphic User Interface (GUI) of the 'CellTracer' software. Analysis of *E. coli* cells during the 'Convex Model algorithm' step.

❖ 'MicrobeTracker' – This toolbox was implemented in MATLAB® and integrated together with an accessory tool, 'SpotFinder', to study the spatial and temporal organization of bacterial cells (initially *E. coli* and *Caulobacter crescentus* [356] and later adapted to other species: e.g. *Pseudomonas aeruginosa* [357]*, Vibrio cholerae* [358] *and Bacillus subtilis* [359]. This tool (see the graphic user interface shown in Figure 3.17) which was developed based on the discontinued 'CellTracker' algorithm [339], has been applied in more than 50 works published in important biology-related journals. The segmentation methodology from 'MicrobeTracker' starts with a step of image inversion to prepare the image for the thresholding step, which makes an initial isolation of the cells from the background. The thresholding step is followed by an edge detection algorithm, that can split individual cells inside segmented clusters, especially when cells are touching (where thresholding techniques normally fail). After this step, if any cell's size surpass their maximum size limit (a value that can be changed by the user), that cell passes through a watershed technique, that also splits cells along one

of the axis, until all of the existing cells don't exceed that value [356]. The next step is to refine the segmentation outlines based on an active contour algorithm that converges towards the actual shape of the cell, which is done by calculating several image forces by calculating 3 gradients of energy at the nodes (attraction to the areas of high local intensity, attraction to the areas of intensity close to the threshold value and attraction to the detected edge lines) and two calculations of image intensity in the vicinity of the nodes (attraction and repulsion to the areas with intensity above and below (respectively) the threshold value [356].

The 'MicrobeTracker' toolbox has two user selectable alternative algorithms to do these calculations (changing their parameters and constraints), with the first one based on the Point Distribution Model [360], which uses descriptors of cell shape to impose constraints on the segmentation (this method does not allow the manual adjustment of the constraints is only able to work with cell shapes that can be linearly described, e.g. does not allow the converging of unusual cell shapes like filamentous or curved bacteria cells) [356]. An alternative method is the Manually Constrained Contour, which has an initial assumption of the segmentation with rod-shape outlines and then converges into a new set of points along the contour using manually pre-set constraints (although this method is slower than the previous one, this was reported to be method chosen to make all of the segmentation, due to its adaptability to all kinds of bacterial shapes) [356]. After the cell outline (from Phase-Contrast and DIC images) is corrected, the accessory tool 'SpotFinder' is used to analyse the fluorescence signal inside each cell (by overlapping the cell outlines with fluorescent images) and detect diffraction-limited spots [356]. This tool uses spatial 2D filtering and a 'ridge removal' algorithm, that removes elongated objects and finds round structures, by calculating local maximum values of the filtered image to use as a seed for a gaussian fitting (with user defined parameters) [356]. In time-lapse images, both the cells and their fluorescent probes are tracked based on the calculation of a "cost", using the perpendicular and parallel distances to the mean main axis, the log ratio of the areas and the angle calculated for both linked cells between two frames. The assigned cell is the one that has the lowest 'cost' (lowest mean distance to the nearest neighbour of all 4 components) [356].

❖ 'Oufti' - The 'MicrobeTracker' [356] toolbox was later discontinued and the authors decided to create a new tool in 2016, called 'Oufti' [361]. This toolbox was developed to address other problems such as the detection of cells in microcolonies, microfluidic chambers or any confluent dense cell samples of touching cells, resulting in large datasets that the 'MicrobeTracker' toolbox was not able to handle [361]. Most of the added features to this new tool are outside of the scope of this research work, since the image processing of bacterial cells in such environments requires the development of different image processing techniques. The main advantage of the 'Oufti' toolbox is that it has all the algorithms previously developed in the 'MicrobeTracker', since this toolbox has been compiled into a standalone program, while 'MicrobeTracker' required the use of the MATLAB® interface. The Graphic User Interface (GUI) of the 'Oufti' was built over the 'MicrobeTracker' GUI (see Figure 3.17) with added buttons added for the new features. New features were added such as plots (e.g. demographs and kymographs cell growth curves) and statistics (e.g. cell intensity and cell dimensions statistics) independent of the MATLAB® interface [361].

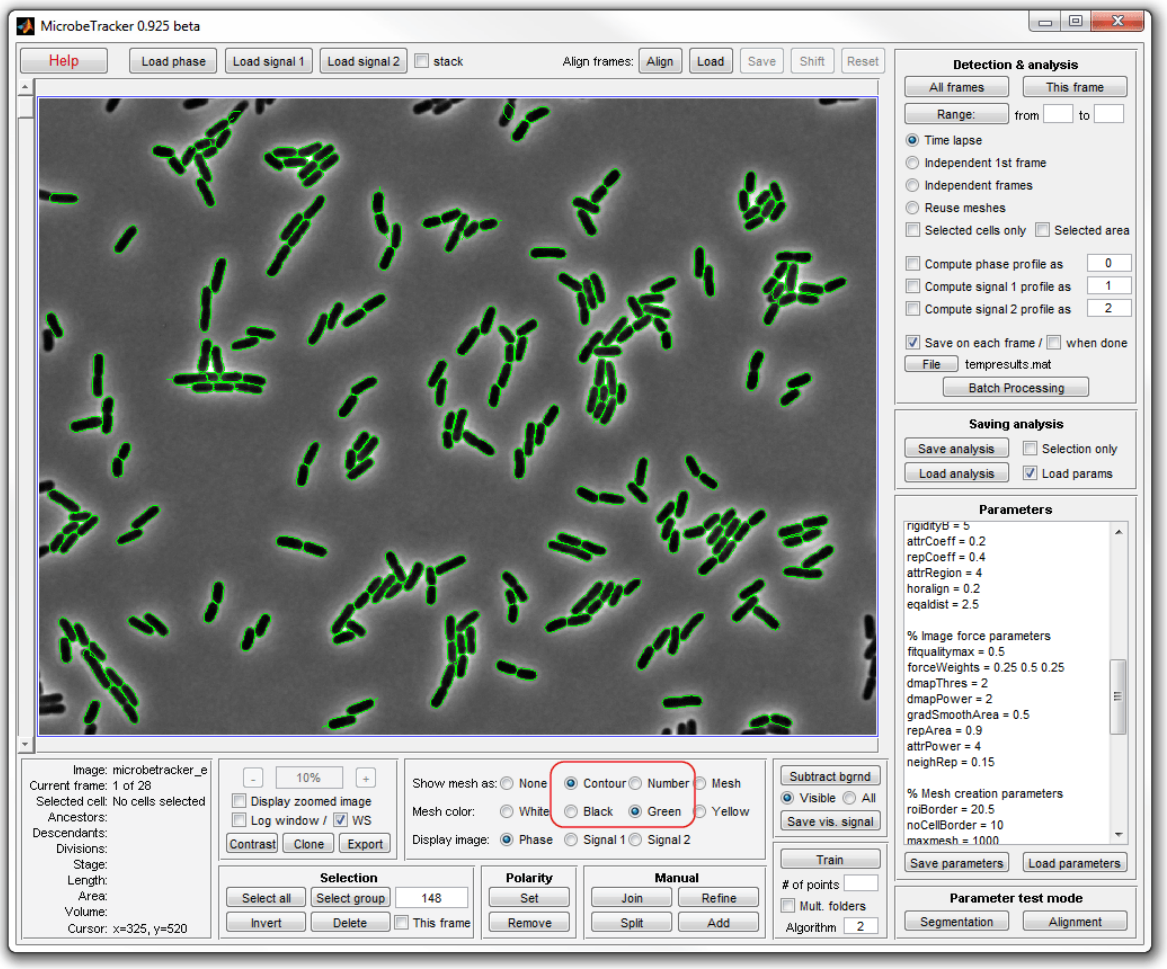


Figure 3.17 - Graphic User Interface (GUI) of the 'MicrobeTracker' toolbox, version 0.925. Analysis of segmented *E. coli* cells (shown in green outline) using the Manually Constrained Contour segmentation algorithm, showing some of the applied constraints and parameters.

❖ 'Schnitzcells' – This toolbox (see the graphic user interface shown in Figure 3.18) was implemented in the MATLAB® environment and provides solutions for segmentation and tracking of *Escherichia coli* cells from confocal and Phase-Contrast images in order to make gene expression studies [362]. The cell segmentation algorithm in 'Schnitzcells' is composed of several steps, starting with a Laplacian of Gaussian filter in order to generate an initial edge segmentation [362] (see an example of this step in Figure 3.18). The next step is to split long or clustered cells and eliminate small cells, based on user-selectable parameters, such as cell dimensions (e.g. minimum cell area, minimum cell length and maximum cell width) and different thresholds (a maximum threshold to determine the maximum number of splits on large clusters and two maximum thresholds to split touching cells, respectively cutting by the major or the minor axis) [362]. If the user is analysis a timeseries, 'Schnitzcells' allows the tracking of each cell over time by connecting cells in different frames by minimizing the distance between the centroid position in each cell, while during divisions, it minimizes the distance between two child cells and the distance between the child's and the old parent cell [362]. In 'Schnitzcells' the user is also allowed to manually correct both the segmentation and the tracking results [362]. One of the identified problems with the 'Schnitzcells' toolbox was the large number of parameters (just for segmentation at least 14 parameters were user-adjustable) that, without proper tuning, can cause the accuracy of the segmentation to decrease notably, presenting a significant number of false positives [363].

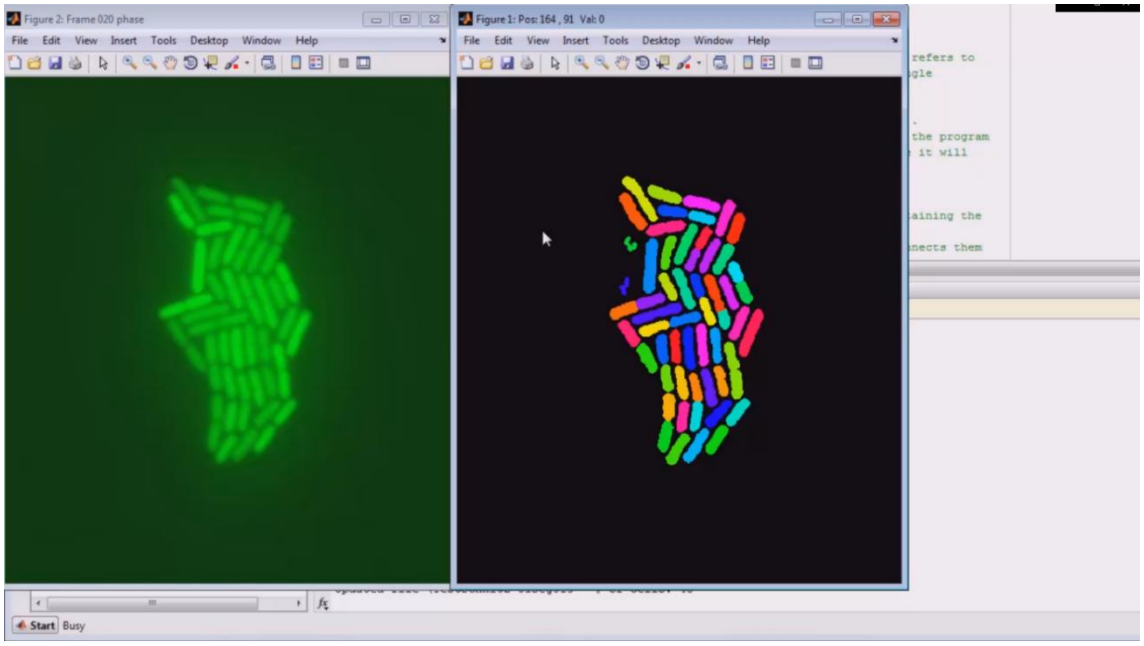


Figure 3.18 - Graphic User Interface (GUI) of the 'Schnitzcells' software. Analysis of *E. coli* cells after the initial edge segmentation step. Small cells that were detected in this step (see small blue and green cell near the white cursor) need to be removed in the next step, while some incorrectly split cells need to be joined back during the manual correction step.

❖ 'MAMLE' – Stemming from the high-throughput production of time-lapsed microscopy images of *E. coli* cells, the Laboratory of Biosystem Dynamics (see Section 1.1 for a detailed introduction to the group) started developing their own image processing toolbox, resulting in the 'MAMLE' (Multi-Resolution Analysis and Maximum Likelihood Estimation) tool, which was proposed for the detection of *E. coli* cells within dense clusters [364]. 'MAMLE' executes cell segmentation in several stages. The first relies on state-of-the-art filtering technique to denoise the image (Block-Matching and 3D filtering), which searches the fixed size blocks of 8x8 that match a reference block, followed by a 3D arrangement of the matching blocks, which is then transformed, thresholded, inverted to augment the basic estimate and finalized by using a collaborative Wiener filter to remove the noise [364]. The previous step is followed by a foreground and background separation algorithm, with the chosen method depends on the processed image: block-wise Otsu threshold, accompanied by a bilinear interpolation (in confocal images) or an iterative range filtering (in Phase-Contrast Images).

The following step is based on the creation of a fuzzy image based on a multi-resolution edge detection method in with a morphological operator followed by a threshold decomposition (using an adaptive method for threshold selection) to create an initial segmentation mask [364]. A Classification algorithm is used to categorize the segmentation masks into classes, namely 'correct', 'under' and 'over' segmentation based on the morphological features (an ideal cell shape is assumed to have a multivariate Gaussian distribution). A correction procedure is then applied by maximizing the likelihood estimate as the objective function to split under segmented images. Over segmented cells are then merged by using the acquired morphological features from the initial segmentation to estimate the maximum likelihood parameter, and exploiting a linear programming-based branch-and-bound technique to obtain the final segmentation mask [364]. This toolbox (see the graphic user interface shown in Figure 3.19) was implemented in the MATLAB® and was initially developed for the segmentation of *E. coli* cells in Phase-Contrast microscopy, but it's segmentation algorithms were also tested for other cell species (e.g. *Staphylococcus* species and Human HT29 Colon cells) and different microscopy modalities, being cited in over 30 research works.

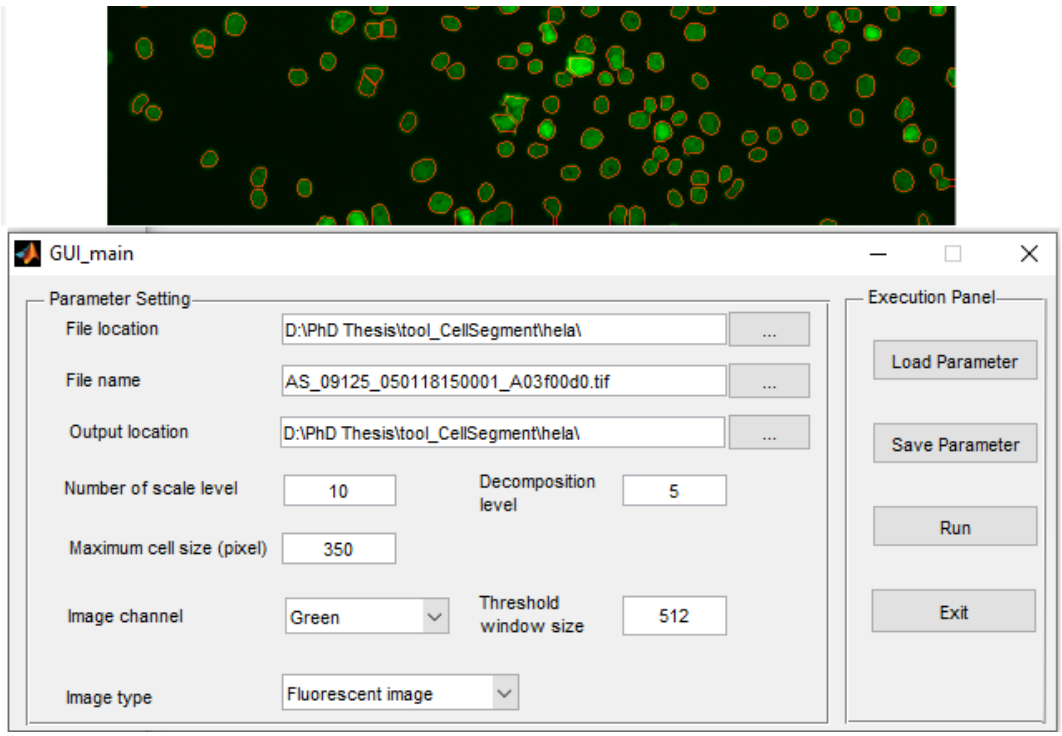


Figure 3.19 - GUI of the 'MAMLE' software. The example shown is based on a Phase-Contrast image of *E. coli* cells. The segmentated lines were produced using the automatic segmentation algorithm followed by a post process manual correction.

❖ 'CellAging' – The 'MAMLE' software was later discontinued when the LBD started a collaboration with the CA3 Group from FCT-UNL, to provide more tailored solutions for segmentation of prokaryotic cells and their internal cell structures, initiating the SADAC project (this research work is inserted in this project, as introduced in Section 1.1). This project started with the development of the 'CellAging' toolbox [365], which adapted a segmentation algorithm called Gradient Path Labelling GPL algorithm, that was specifically developed for the segmentation of Drusen's in Retinal images [304] (see Section 3.1.2 for an description of this algorithm) and also introduced the use of Machine Learning techniques (Classification and Regression Trees algorithm [366]), which were trained (initially for brightfield images) to merge and discard incorrectly segmented objects (over and under segmented objects).

This toolbox added more features from the previously developed ('MAMLE'), such as the possibility of manual corrections after the automatic segmentation algorithm, automatic inter-modal image registration, based on 2-D affine geometric transformations, e.g. translation, rotation, scale, and shear transformations (see Section for a detailed description of these image registration techniques) [229], [234], the possibility of establishing relationships between cells of consecutives frames (creating cell lineages based on the overlapped position of each analysed cell (it is noted that the algorithm checked if the number of cells were augmented in a future frame, which was recorded as a cell division process, and two cells are assigned to the same parent from the previous frame) and finally the possibility of studying cellular processes, such as cell growth, cell division times and studying features such as the detection of fluorescent spots and the distribution of fluorescence along the major axis. This toolbox (see the graphic user interface shown in Figure 3.20) was implemented in the MATLAB® interface and was used in many research works (cited over 35 times) to study transcription events and asymmetries found in several cellular processes found in *E. coli* cells [6], [7], [367], [368].

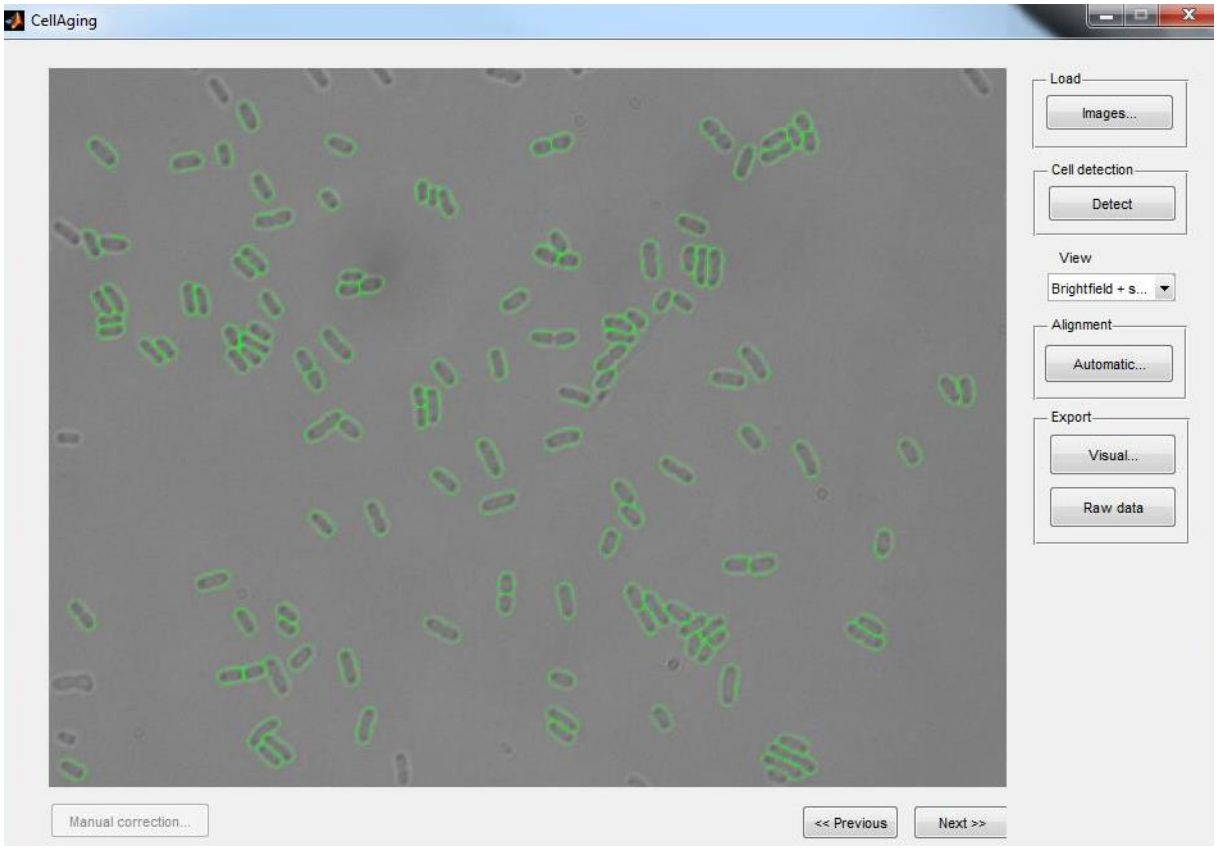


Figure 3.20 - Graphic User Interface (GUI) of the 'CellAging' software. The example shown is based on a brightfield image of *E. coli* cells. The green segmentation lines were produced by their developed automatic segmentation algorithm (followed by a post process manual correction).

❖ 'AutoCellSeg' – This toolbox [369] was recently published and implemented on MATLAB® an automatic supervised segmentation method. The user needs to manually select the bacterial colonies, to extract automatically with the colony's features (area and mean intensity) using a fast marching level set method [370] (alternative *a priori* selection methods are based on creating manual labels using a freehand sketching tool or drawing ellipsoid shapes). This step is followed by an adaptive threshold segmentation step which creates the initial segmentation mask (the thresholds are selected by using a plausibility criterion that is used remove small objects). This mask provides the search space for the numerical computation of a regional maxima which is used as the initial seed for a tailored feedback-based watershed segmentation step (if the initial seed is does not produce adequate results), the user can change the parameters of the H-maxima transform, by tuning the fuzzy trapezoidal membership functions and their variables [371]. This toolbox (see the graphic user interface shown in Figure 3.21) was implemented in MATLAB® and its segmentation algorithm were tested successfully for several colonies of bacterial species (e.g. *E. coli, Klebsiella pneumoniae, Pseudomonas aeruginosa* and *Staphylococcus aureus*) [369]. Finally, the authors also implemented a post-correction step in their workflow that is based on freehand drawing and manual labelling of seeds points and circles using a user-friendly graphical interface (a feature that has only been implemented in recently developed toolboxes) [369].

Most of the above-mentioned image processing toolboxes, didn't have available one or more core components of the SADAC project (image registration, cell segmentation, segmentation of cellular components, cell tracking) or some of their methods did not converge to the images produced by the LBD, so it was decided that the best course of action, would be to produce an in-house toolbox, leading to the development of the 'iCellFusion' and the 'SCIP' platforms, as detailed in Section 4.1.

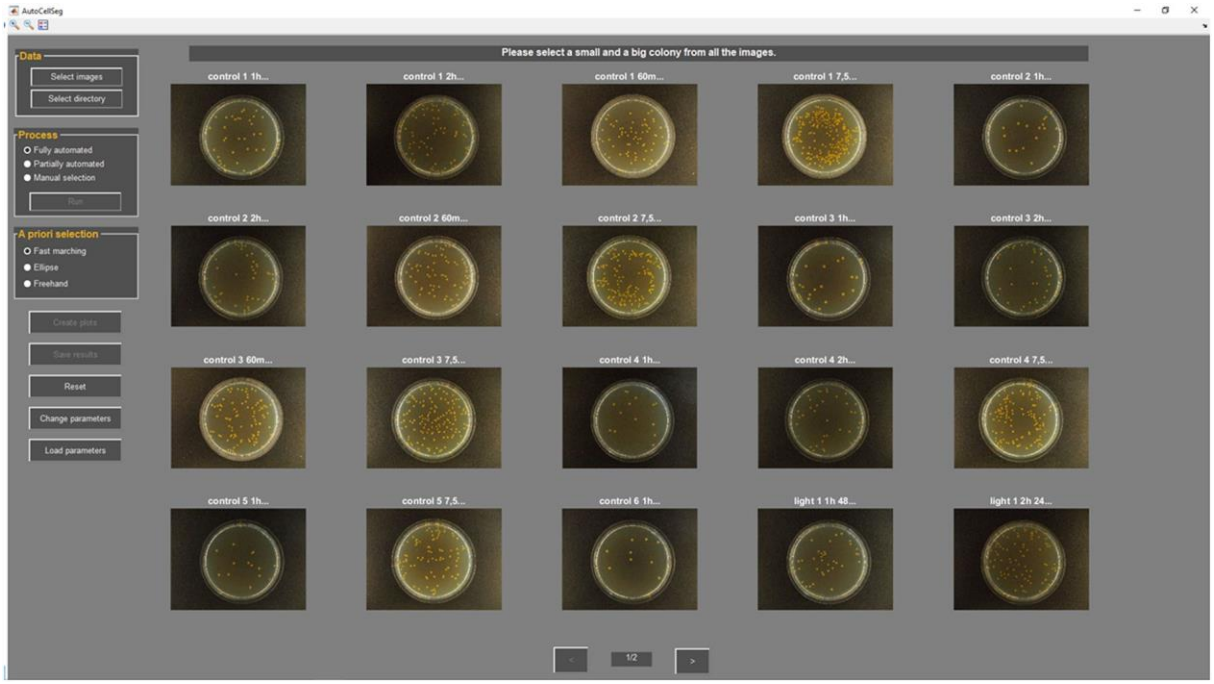


Figure 3.21 - Interactive GUI of the 'AutoCellSeg' software. The example shown shows microbiological assays producing colony forming units. Image taken with permission from [369].

Table 3.1 summarizes the public availability of the above-mentioned toolboxes.

Table 3.1 – Availability of microscopy image processing toolboxes. (Download location was last updated on 20/12/2019).

Name of the Toolbox	Download Location	Main Reference
'CellProfiler'	http://www.cellprofiler.org/	[330]
'Cell-ID'	http://lbms.df.uba.ar/	[337]
'CellTracker' (1)	http://www2.warwick.ac.uk/fac/sci/systemsbiology/staff/bretschneider/celltracker/	[339]
'CellTracker' (2)	http://celltracker.website/index.html	[342]
'Farsight'	http://farsight-toolkit.org/wiki/Main_Page	[345]
'Cell-C'	https://sites.google.com/site/cellcsoftware/Home	[348]
'CellTracer'	http://www2.stat.duke.edu/~mw/mwsoftware/CELLTRACER/	[355]
'MicrobeTracker'	http://microbetracker.org/	[356]
'Oufti'	https://oufti.org/	[361]
'Schnitzcells'	http://easerver.caltech.edu/wordpress/schnitzcells/	[362]
'MAMLE'	https://sites.google.com/view/andreribeirolab/home/software	[364]
'CellAging'	https://sites.google.com/view/andreribeirolab/home/software	[365]
'AutoCellSeg'	https://github.com/AngeloTorelli/AutoCellSeg	[369]

Similarly to the validation of image registration techniques, one of the growing trends in microscopy imaging is the simulation of biological processes using computational modelling is also a viable alternative to create "ground truths" by producing artificial deformable images that can be used for quantitative evaluation of the cell segmentation algorithms [10]. Different image simulation toolboxes and algorithms will be discussed in the next Section.

3.2. Simulation Methods

The validation of the above-mentioned image processing tools has prompted the development of image simulation tools, that can generate synthetic images. Simulated images have already been

used as benchmark with precisely known "Ground-truth" in biological studies by improving of the quantitative measures (e.g. such as Sensitivity, Precision and F-Score) of cell segmentation, tracking and image registration [250].

These image simulation tools can also be used to create "null-models" [372], that can be used to study the statistical patterns in the absence of a particular mechanism (e.g. it could be used to study how the nucleoid affects the location of RNA molecules by removing the nucleoid from the cell).

Section 3.2.1 summarizes the development and implementation of the Stochastic Simulation Algorithm (SSA), which is one of the most used simulation algorithm, while Section 3.2.2 outlines the development of image simulation toolboxes, mainly focusing on simulation of microscopy images of both prokaryotic and eukaryotic cells.

3.2.1. Stochastic Simulation Algorithm

To create such simulation tools, realistic biological models need to be developed using data coming from theoretical and experimental knowledge that arise out of the statistical distributions of cellular geometry [373] and spatial and temporal information [11]. Those models include the cell shape and size, the location of subcellular structures, kinetic and spatial models of cell growth, cell division cell migration and internal cell functions (such as gene expression) [11].

To characterize the state of the elements inside the simulated system, these realistic biological models need to be explicitly written into a system of chemical reactions, which their evolution can be predicted using two different approaches (which have been extensively reviewed and compared in [374]: a deterministic approach, where the most popular method answers this problem by solving a system of coupled ordinary differential equations (ODEs), and where stochasticity within the system is neglected [374] and a stochastic approach, where two popular methods have answered this problem by solving analytically the chemical master equation (CME) (see equation 3.2), which is also known as the Kolmogorov forward equation for a stochastic kinetic process (a solution that is unfeasible when a large number of reactants is present) or by providing exact simulations of trajectories of the CME [374], with Gillespie's algorithm being the most prominent algorithm with several of its formulations [375]–[378]. This research works focuses on the Stochastic Simulation Algorithm (SSA) [378], which is a Monte Carlo method that simulates numerically the time evolution of well stirred reaction systems, where time goes forward in discrete steps and in each step a reaction is explicitly executed and the effect on the number of each molecule is calculated.

This algorithm assumes that all the intervening molecules reacting in a homogeneous and thermally equilibrated mixture, and as previously said provides exact simulations of the trajectories of the CME (see equation 3.2), by estimating the exact temporal moment that an event will occur using the probability $P(x, t|x_0, t_0)$ of having a given concentration x in the simulated volume at next time step 't', knowing the initial time step (t_0) and the initial concentrations of all the molecules (x_0) and using the propensity function, $a_\mu(x)$, counts the probability that the molecules x in the volume at time 't' react in the next infinitesimal time interval '$t + dt$', changing absolutely the molecules in the volume by v_μ, via reaction the R_μ. This propensity function can be written as $a_\mu(x) = h_\mu(x)c_\mu$, where $h_\mu(x)$ is the number of possible reactant combinations in the simulated volume (see for the possible values of $h_\mu(x)$) and $c_\mu$is a kinetic constant such that $c_\mu dt$ gives the probability that in the next infinitesimal time 'dt' a determined molecule will spontaneously react via R_μ [376], [377], [379].

$$\frac{\partial P(x,t|x_0,t_0)}{\partial t} = \sum_{\mu=1}^{R}\left[a_\mu(x-v_\mu)P(x-v_\mu,t|x_0,t_0) - a_\mu(x)P(x,t|x_0,t_0)\right] \quad \textbf{(3.2)}$$

Table 3.2 - Possible values of $h_\mu(x)$ for each reaction type. Taken from [380] (adapted from the formulations found in [378]).

$h_\mu(x)$	Type of reactions
1	$\rightarrow$ ***products***
X_i	$S_i \rightarrow$ ***products***
X_iX_j	$S_i + S_j \rightarrow$ ***products***
$\frac{X_i(X_i-1)}{2}$	$2S_i \rightarrow$ ***products***
$\prod_{i\in S_\mu}\prod_{q=1}^{N_{(i,\mu)}}\frac{X_i-q+1}{q}$	$\sum_{i\in S_\mu} N_{(i,\mu)}S_i \rightarrow$ ***products***

The stochastic approach has been extensively reviewed and divided into two classes: network-based and network-free [381]. The first one requires that all the simulated reactions must be established during the initial phase, while the second can be implemented with reaction rules that can encapsulate classes of reactions to generate the needed reactions during the simulations. The SSA algorithm, which will be the main algorithm used in this research work, is based on the network-based approach. The earliest formulations of the SSA algorithm, the Direct Method and the First Reaction Method [378], were considered to be computationally heavy, especially in large biochemical systems prompting the appearance of new methodologies, which improved the computational efficiency, without affecting its exactness, namely the Next Reaction Method [382], the Logarithmic Direct Method [383] and the First Family Method [376].

The main differences of each method have been reviewed in are the on generation of random numbers, and how each they calculate the total propensities of the reactions (which are grouped into "families"), which is represented in step 3 in the following example of the Direct Method formulation:

1. Define R reactions rates $\{k_1, \dots, k_R\}$ and the initial molecule number $x = \{x_1, \dots, x_N\}$ and define the stopping time of the simulation (t_{stop}). Set the initial simulation time to t = 0 and the reaction counter n=0.

2. Calculate the propensity for each Reaction Rate (R), {p1=k1•h1,…, pR=kR•hR} and calculate $p_0 = \sum_1^R p_i$ and store all the propensity values. This is done by using the current population of molecules, while h is the number of all possible distinct molecular interactions in the current state (see Table 3.2 for different types of h).

3. Calculate the pair (τ, μ) using two random numbers r1 and r2 (using a uniform distribution from [0,1], with $\tau = \ln(1/r_1)\cdot(1/p_0)$ while μ has to satisfy the following rule: $\sum_1^{\mu-1} p_i < r_2 \cdot p_0 < \sum_1^{\mu} p_i$

4. Calculate and store the value of the new time interval ($t_{new} = t + \tau$), pair (τ, μ) and increment by one the reaction counter.

5. If $t_{new} \geq t_{stop}$, end the simulation.

6. If $t_{new} < t_{stop}$ then set $t = t_{new}$ and update the molecular numbers according to the type of reaction that occurred using $x = x + v_{\mu}$ (see Table 3.2).

7. Go back to step 2 and continue the simulation.

During these simulations the average time between these reactions can be so small that is not computationally feasible to implement all of them (independently of the chosen method), which led to adjustments of the SSA algorithm in order to boost the simulation performance, such as the introduction of a mechanism that allows the system to skip forward in time by a pre-selected time interval (this is called 'tau-leaping) and a slow-scale simulation mechanism which allows the user to simulate based on the timescale of the slower reactions (which means that it has to skip most of the faster reactions, which are on a different timescale) [376], [384]. These improvements allowed the SSA algorithm to provide a response to both a wide range of molecular populations and of reaction timescales, which provided the necessary tools to simulate complex biological system, and allowed the implemented of a "wait list", by modifying the steps in the SSA algorithm to account for the release of molecules, leading to a so called, delayed SSA algorithm [51]. As mentioned in Section 2.2.3, this delayed SSA algorithm was able to simulate the dynamics of gene expression and gene regulatory models at the single RNA and protein level, even coupled with cell division [53], [68]–[73]

The available simulators based on the SSA algorithm have available the implementation of one or multiple of the above mentioned number generator method [381] (see a comprehensive list of most of the available simulators, their available methods and the implementation languages in Table 1 of [381]). The SGNS2 simulator [67] only allows the use of the Next Reaction Method [382], which can be faster than the Direct Method, since it only samples one random number per iteration and also stores the more efficiently all of the generated time intervals [67], [381], [382].

3.2.2. Image Simulation Toolboxes

A recent review on simulation methodologies of microscopy images and cellular objects has characterized 61 tools and methods based on the simulated objects (spots and particles, subcellular components, nuclei, multiple target, entire cell populations and tissues) [385], the microscopy technique that is simulated (Fluorescence, TIRF, 3D-SMLM, Brightfield, etc), the type of image (2D, 3D, and temporal series), if a sufficient description of the method/tool is provided, if it is publicly available for download and if the generated benchmark dataset is also publicly available [385]. Simulated objects (also called digital phantoms) into two different categories [385]: parametric phantoms, which are created and controlled by a set of previously designed parameters (e.g. controlling size and shape of the cell, time of division) and learning-based phantoms, which are created based on the training of previously acquired image datasets [385]. A discussion on the advantages and disadvantage of both categories is also presented in [385], with the main disadvantage of the first being that one needs to have previous knowledge of a large number of user-defined parameters, while the second one has a large dependency on the choice of the training algorithm and the availability of a large number of representative datasets [385].

Another category separation that above mentioned review presents is the difference between moving and non-moving phantoms [385]. Most of the initial simulations toolboxes only focused on the spatial information of the cell (non-moving), producing just a single frame of the desired synthetic image model, producing point-like objects of Fluorescence in situ hybridization (FISH) spots [386],

which were randomly placed in a 3D space, as can be seen in Figure 3.22, while some of the newer simulation tools started to simulating moving phantoms to study the dynamics of cell motility (e.g. the simulation of Phase-Contrast image of fish epidermal keratocytes [387]). The step towards the simulation of moving phantoms allowed the generation of artificial time-lapsed microscopy images, which marked a very important step for the validation of automatic tools used in live cell imaging, as a time series extends the observation from a unique time-point (just 1 frame) to the observation various frames containing cellular dynamics, such as measuring protein or RNA levels or even observing cell migration, cell division and cell growth [1], [192].

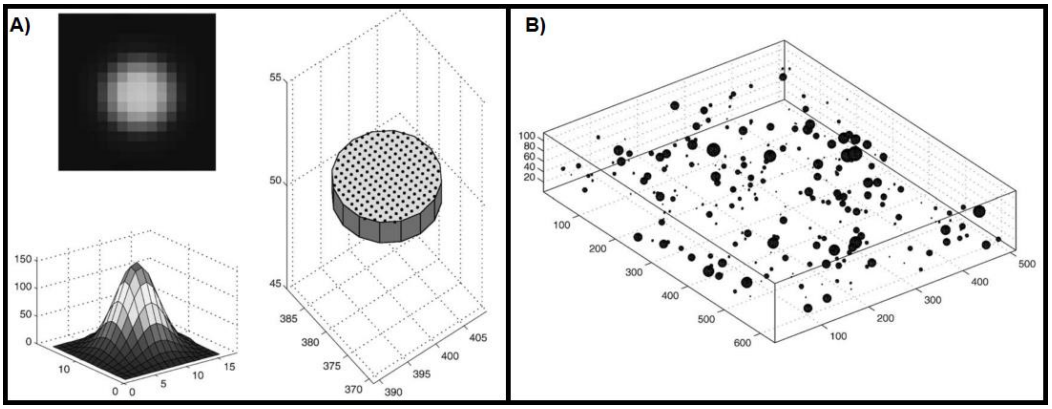


Figure 3.22 – Generation of 2D and 3D non-moving phantoms. (A) Slice of a simulated point-like object (based on the Fluorescence in situ hybridization spots) (B) Random 3d spots in a Rectangular Prism. Adapted with permission from [386].

Finally, the above mentioned review [385] has also focused on the simulation of the different image detection techniques and how to emulate how a real image acquisition system behaves (e.g. adding noise to the images or adding uneven lighting or optical aberrations [385].

In this section, five image simulation toolboxes and their implemented methods are extensively reviewed, based on their importance to this research work (see Table 3.3 for a summary of the public availability of the chosen toolboxes).

A complex simulator was designed to produce a simulated image of large eukaryotic cell populations by creating a parametric model for each individual cell geometric contour (including size, shape and texture) [388]. This simulator also included common errors obtained in the image acquisition systems, such as uneven lighting and optical aberrations [388], as can be seen in the workflow presented in Figure 3.23-1.

The simulation of the nuclei and the cytoplasm can be observed in Figure 3.23-2-A and Figure 3.23-2-B, respectively, while the overlap of both structures is shown Figure 3.23-2-C. This simulator [388] evolved towards the development of a standalone and publicly available toolbox called 'SIMCEP' [389], which provided a framework to validate and test various image processing toolboxes, such as the previously mentioned in Section 3.1.4: 'CellProfiler' [330] and 'CellC' [348] and other image processing software's such an open-source and Java-based image processor (ImageJ v1.36b) and a commercially available software called MCID Analysis (from Imaging Research Inc., Catharines, ON, Canada; Evaluation ver. 7.0).

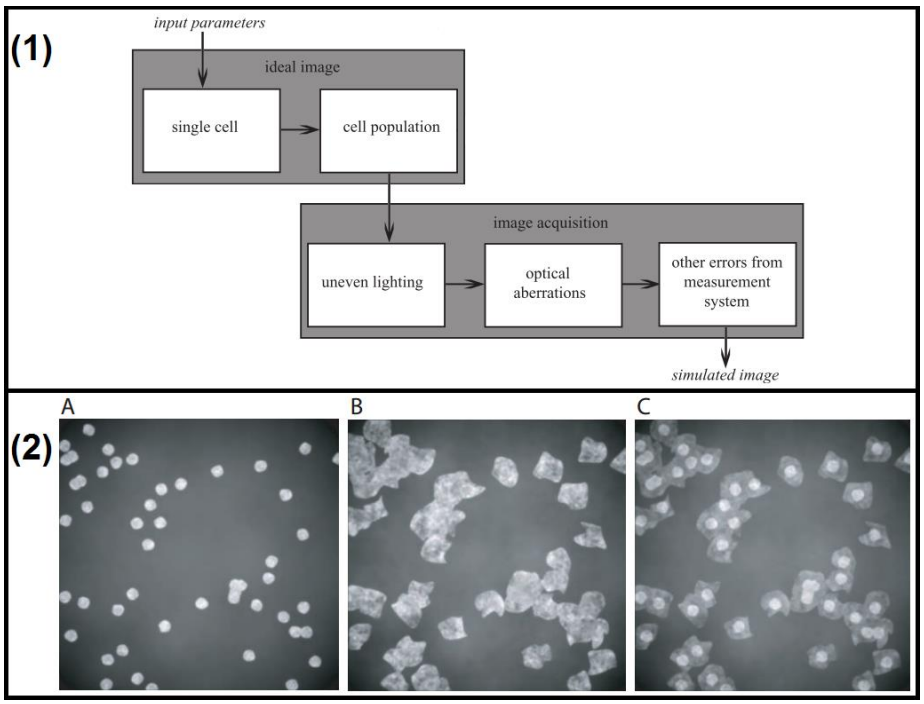


Figure 3.23 – Predecessor of the 'SIMCEP' simulator. (1) Workflow of the image generation; (2-A) nuclei images; (2-B) cytoplasm images; (2-C) Overlapped image of the simulated nuclei and cytoplasm. Adapted with permission from [388].

The 'SIMCEP' simulator produced a benchmark dataset of synthetic images along with manually labelled images was generated in order to be used as the ground-truth for the validation of image processing tools (e.g. k-means clustering, Expectation Maximization, Otsu's threshold and the Global Minimization of the Active Contour Model) [390], [391]. These images were produced with different cell parameters, such as probability of clustering, cell radius, and cell shape and image parameters such as background noise and illumination disturbance [391].

In a subsequent study, eleven methods for segmentation of subcellular constituents (which have a spot-like structure) were validated on simulated microscopy images from 'SIMCEP' and their testing was done both on microscopy images from real experiments coming from human and yeast cells but also used [392]. Their results showed that no algorithm outperformed the others for all cases, as the selection of the detection algorithm should consider each situation (type of cells, quantity and quality of images, quantity of spots, etc) and then apply the method according to that situation [392].

The same group that developed the 'SIMCEP' simulator, followed their work on image simulation by developing parameterized models of different bacterial populations [393]. The five proposed models can be observed in Figure 3.24-A (Additional information on bacteria shapes has been provided in Section 2.2). A simulation of a population set containing both *E. coli* and *Micrococcus luteus* cells is shown Figure 3.24-B, using their respective parametrized models [393].

This simulator ('SIMCEP') is capable of creating a population with similar characteristics (but each cell parameters are drawn from a random variable), but also sub-populations with specific characteristics, for example, stress response to drugs and gene knock-downs [393]. The first version of the 'SIMCEP' toolbox was designed only to simulate 2D images, which led to the development of a new toolbox to extend that model to a higher dimension, although it limited the maximum number of generated cells, since its extension to 3D was not straightforward and computationally efficient [394].

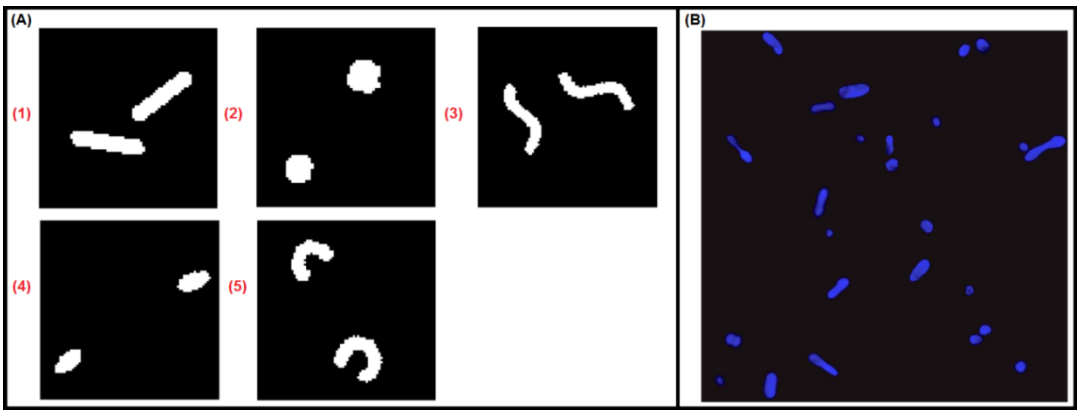


Figure 3.24 – Parameterization of bacterial shape models based on the 'SIMCEP' image generation toolbox. (A-1) bacilli-type bacteria (rod-shaped), like the *Escherichia* and *Salmonella* genera; (A-2) cocci-type bacteria (spherical-shaped), like the *Streptococcus* and *Micrococcus* genera; spirochetes-type or spirilla-type bacteria (corkscrew-shaped), like the *Treponema* and *Brachyspira* genera; (A-3) coccobacilli-type bacteria (intermediate shape between spheres and rods), like *Haemophilus* and *Chlamydia* genera; vibrio-type bacteria (curved rods or comma-shaped) like the *Caulobacter* and *Vibrio* genera. (B) Population of 40 cells sampled from models learned for *E. coli* and *M. luteus* bacteria. Both synthetic cell types show variation in cell sizes and shapes. Adapted with permission from [393].

The 'CytoPacq' artificial phantom simulation toolbox was recently published with the main objective of generating benchmark datasets that can be used to test and validate the accuracy of image processing algorithms (registration, segmentation, tracking), with the capability of simulating multi-dimensional representations of cells (e.g. microspheres, granulocytes, HL-60 Nucleus, colon tissue cells and images of lung cancer cells) and also allows the simulation of the image acquisition process, starting from the light transmission process to the retrieval of the digital object [394], [395]. The final version of the 'CytoPacq' toolbox incorporated several of the methods that the research group developed over the years and was equipped with three different modules [394]–[399] (see the complete workflow in Figure 3.25-1):

1. The first module, 'Digital Phantom', is divided into three smaller frameworks, that can generate digital artificial objects that mimic the cell structure and behaviour (see Phase I in Figure 3.25-1). The first framework, 'CytoGen', can generate realistic distributions of objects by creating an ellipsoid in black and white (see Figure 3.25-2-A), then that ellipsoid is deformed using partial differential equation-based methods (see Figure 3.25-2-B) followed by texture creation, which is done by defining the internal structures (see Figure 3.25-2-C) [394], [395]. The second framework, 'MitoGen', can generate 3D-time-lapsed images of fluorescence-stained dividing cells (e.g. HL60 population) by mimicking the observed temporal and spatial organization of these cells (shape, size, texture, cell growth, cell division due to mitosis and cell motility) [396]. The third framework, 'FiloGen', can generate 3D-time-lapsed images of moving cells (e.g. lung cancer cells) with growing and branching filopodial protrusions, with spatial and temporal attributes that can be tuned by the user on a molecular level (length, thickness, number, level of branching, and lifetime of the filopodia) [397].

2. The second module, 'OptiGen', is the optic system simulator (see Phase II in Figure 3.25-1) and simulates the transmission of the signal through the lenses, the objective, the excitation filter and the emission filter (various sets of equipment can be simulated for each part), capable of generating optical aberrations such as uneven illumination and image blurring, based on the real point spread function [398].

3. The third module, 'AcquiGen', is the digital CCD camera simulator of the phenomenon's that occur during image capture (noise, resampling, digitization) by changing the camera selection, the

acquisition time, the dynamic range usage and the stage movement in the z axis (see Phase III in Figure 3.25-1). This module also allows the simulation of photobleaching. An example of the cell passing through the second and the third module is shown in Figure 3.25-2-D [395].

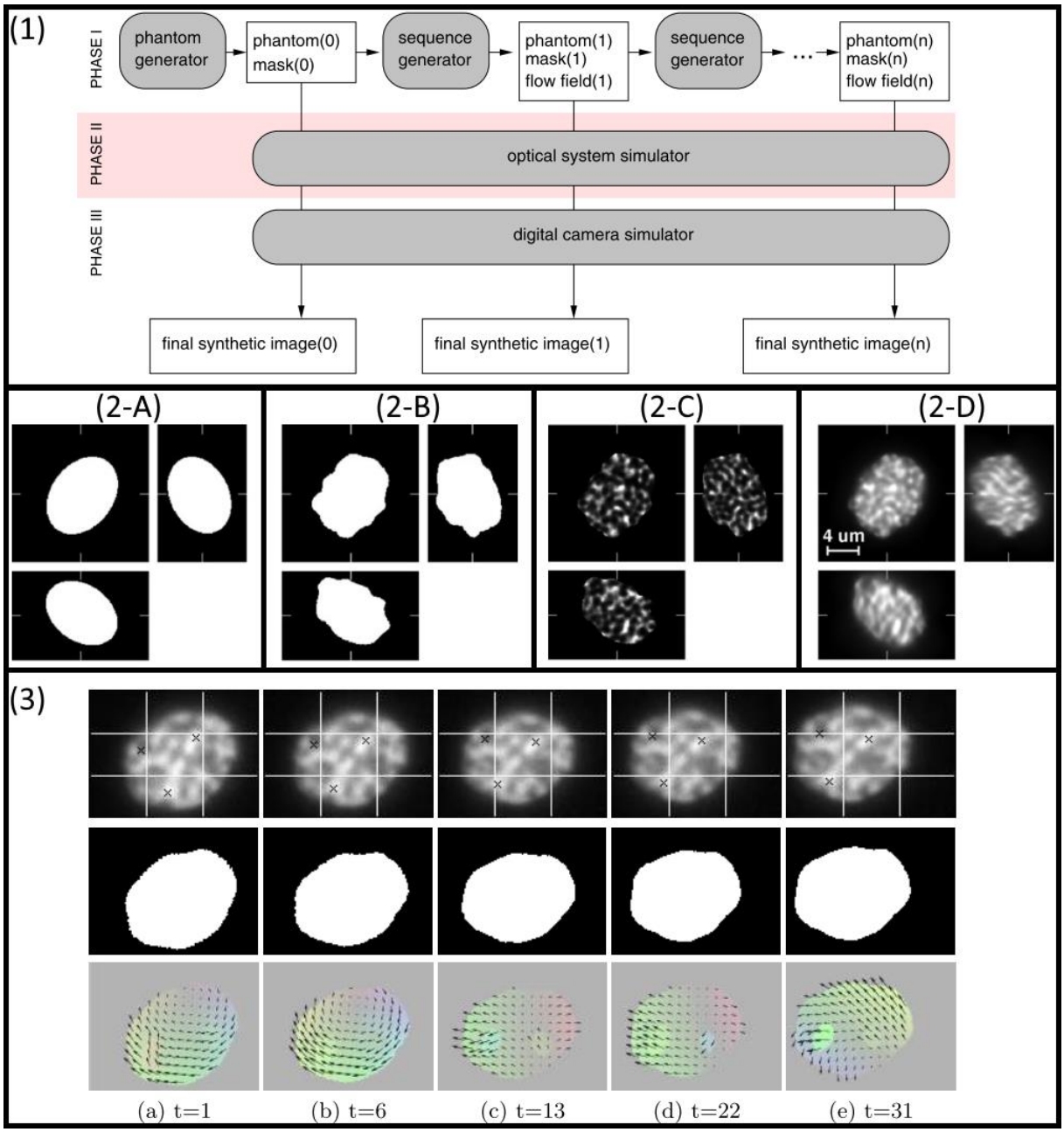


Figure 3.25 – 'CytoPacq' workflow and its artificial object generation. (1) Workflow of the CytoPacq toolbox showing its three functional modules [398]; (2) Steps for the artificial image generation of a HL-60 Nucleus [395]; (3) Artificial time-lapse observation of a generated HL-60 Nucleus [398]. All images were adapted with permission from the respective reference.

The generated artificial images were validated using four different methodologies: by visual comparison of real images acquired in a laboratory, by comparison of the log intensity histograms of the artificial and real image, by comparison of the computed descriptors, namely the entropy and the second to sixth central moments using Quantile—Quantile plots from real and synthetic data and by computing the 3D Haralick texture features, such as angular second moment, contrast, correlation and variance [395]. The 'CytoPack' toolbox is able to reproduce not only the spatial information, but also the temporal information by simulating motion of selected biological objects and generating an artificial time-lapse observation, as can be observed in Figure 3.25-3 [398], [399].

The third reviewed toolbox in this section is the 'SimuCell' platform, which was developed to generate synthetic microscopy images with a heterogeneous cellular population and diverse cell phenotypes (their simulated objects can be the nucleus, cytoplasm, lipid droplets and nuclear bodies) [400]. 'SimuCell' allows, the development of novel phenotypes by creating new Plug-ins and that the algorithm development cycle is dependent of the comparison between the full ground truth and the result analysis [400] (see the workflow presented in 'SimuCell' in Figure 3.26-A). Each cell can be modelled with different shapes own distinct distribution of biomarkers (e.g. the distribution can be constant, linear or angular, can also depend on the distance to the edge of the object, can depend on the distance to other objects inside the same cell or the distance to nearby cells) [400].

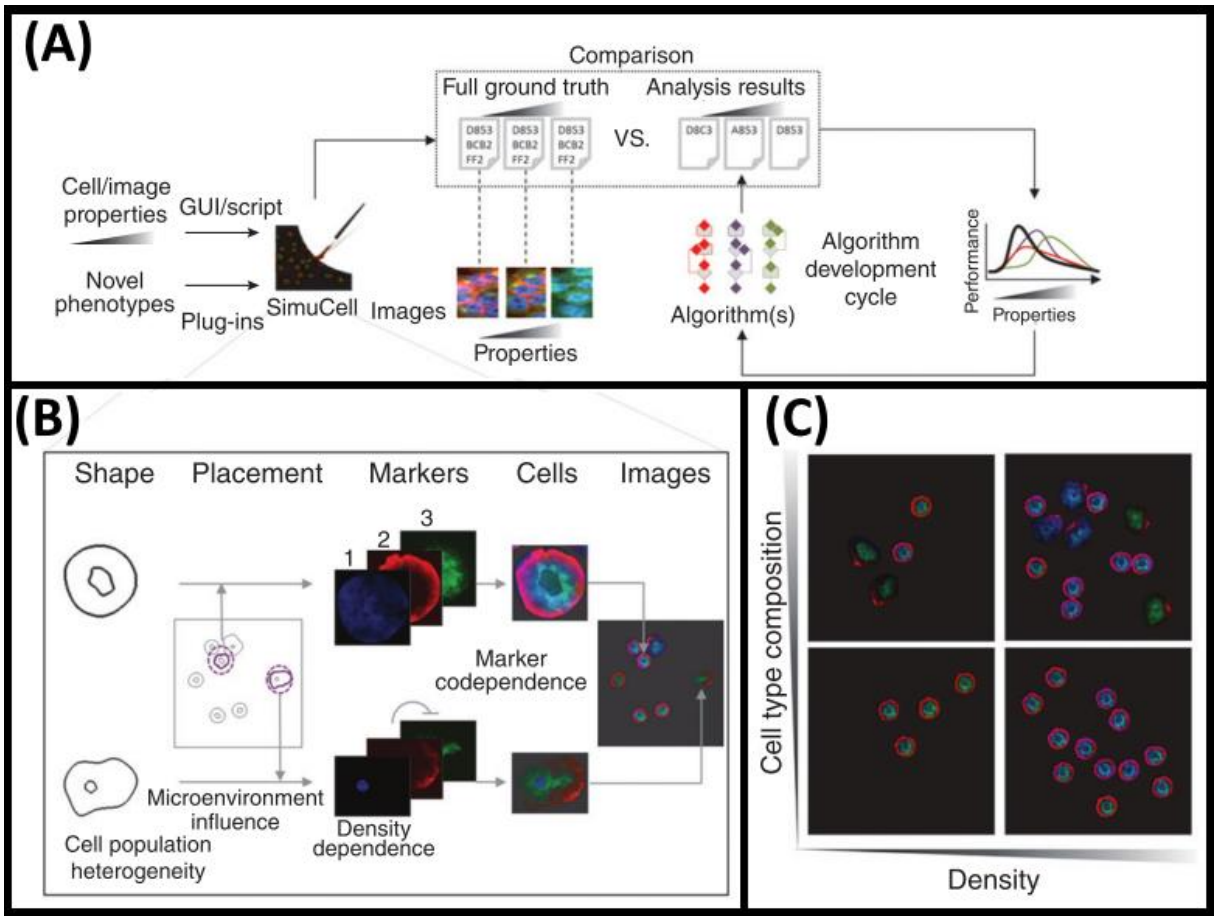


Figure 3.26 – 'SimuCell' artificial object generation toolbox. (A) Workflow of the 'SimuCell' toolbox; (B) Observation of the cell population heterogeneity and creation of different phenotypes in the same image; (C) Examples of images with different densities and different cell type composition. Adapted with permission from [400]

When considering other simulations platforms, the main innovation of the 'SimuCell' toolbox is that it allows that the distribution of biomarkers inside the cell to be affected by the cell's microenvironment (see an example in Figure 3.26-B), making the placement of each cell an important task in 'SimuCell', which can be in clusters, near existing cells, randomly placed and with the possibility of allowing cell overlapping. Figure 3.26-C shows four examples of different images with different phenotypes and with different densities. Examples of simulated cellular organelles include the nucleus; nuclear body; cytoplasm and lipid droplet. Each object can be rendered using its own specific Plug-in. The 'SimuCell' toolbox can also simulate image artefacts that occur during the process of image acquisition, such as adding basal brightness, or adding linear or radial image gradient. It can also simulate cell artefacts, such as adding cell staining or misfocusing some cells [400].

The ‘CellOrganizer’ toolbox uses a different approach to generate synthetic images of cells, based on a machine learning-based approach, another group developed a several methods generate the whole cell, including structures like the nucleus, proteins, cell membrane and cytoplasm components such as microtubules [401], which was later followed by the development of a publicly available toolbox called [402]. Although the model was capable of extracting a very precise shape model from real image data, using Bayesian networks as their modelling strategy (see Figure 3.27-A), the model could not be described in precise mathematical terms [401].

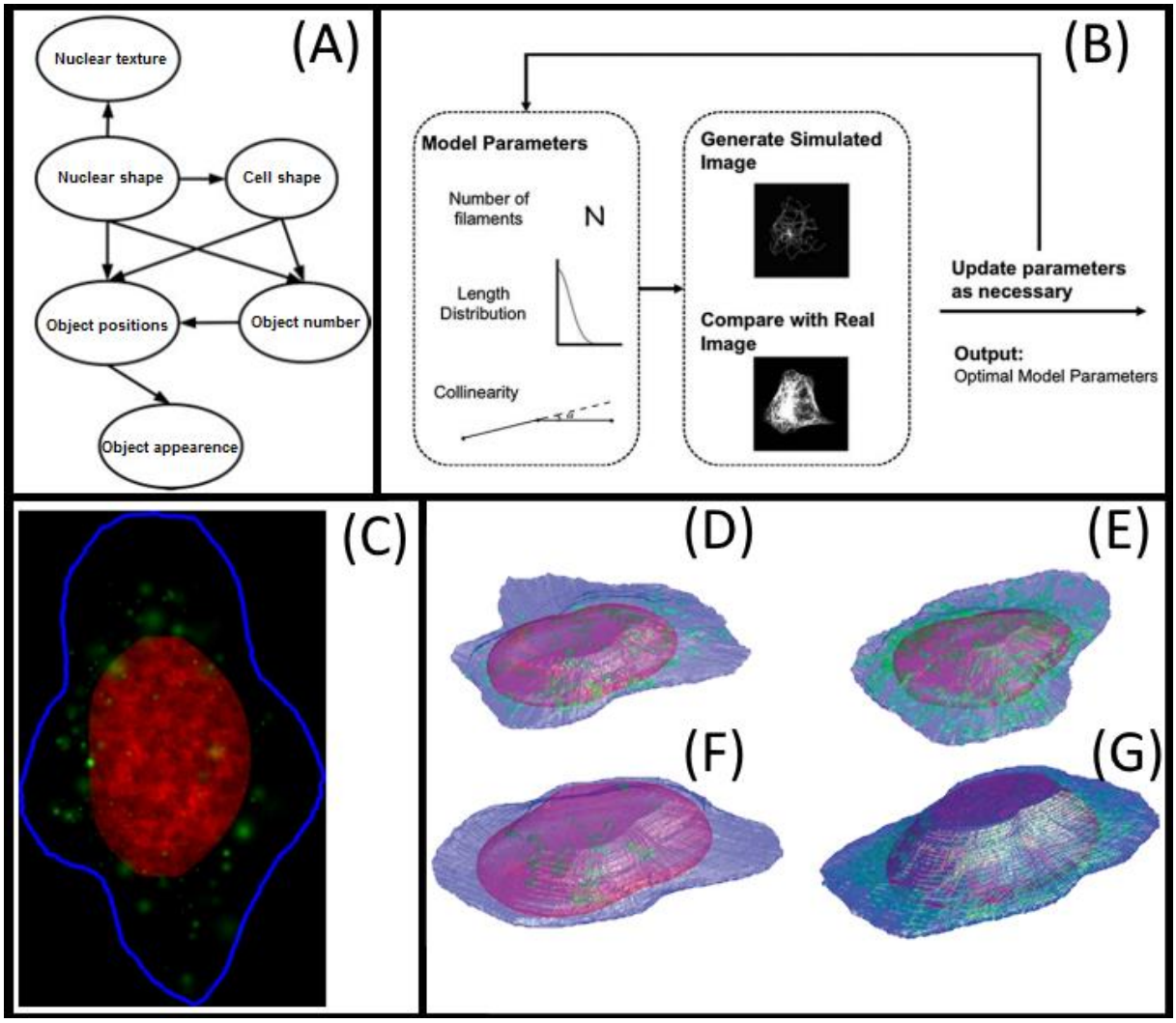


Figure 3.27 - ‘CellOrganizer’ artificial object generation toolbox. (A) Description of the models as Bayesian networks [401]; (B) Overview of inverse modelling approach for estimating parameters of the microtubule generative model [402]; (C) Example of a synthetic image generated by a 2D model of the lysosomal protein LAMP2 (DNA distribution is shown in red, cell outline in blue, and lysosomal objects in green) [402]; Synthesized 3D images of a (D) Lysosome, (E) Mitochondria, (F) Nucleolus, and (G) Endosome displayed in pseudo color surfaces for different protein location patterns (green), with nuclear (red) and cell shapes (blue) [403]. All images adapted with permission from the respective reference.

An overview of the modelling approach for the generation of synthetic microtubules is shown in Figure 3.27-B. Examples of a simulated 2D image of a lysosome (see Figure 3.27-3) and 3D images of a Lysosome (Figure 3.27-D), a Mitochondria (Figure 3.27-E), a Nucleolus (Figure 3.27-F) and an Endosome (Figure 3.27-G).

Another toolbox (‘CompuCell3D’) capable of simulating tissue development, homeostasis or even diseases over a timeframe was develop to aid the experimental studies in this area [404]. The graphic user interface of ‘CompuCell3D’ is presented in Figure 3.28-A, showing the cell drawing tool.

The proposed cell modelling in 'CompuCell3D' is based on a multi-cell and open source Monte Carlo algorithm [405] and is capable of simulating a cell-sorting model (shown in Figure 3.28-B), where the less-cohesive cells (lighter grey) envelop the more cohesive (dark grey) and condensing cells (forming a central cluster domain), simulating vascular tumour growth (as shown in Figure 3.28-C) or simulating angiogenesis models [405]. Instead of focusing on individual cell modelling, 'CompuCell3D' generates large cell populations by adopting the statistical large-Q Potts model to simulate the reorganization of uniformly distributed cell-like objects to assure that a natural cell shape is achieved [404].

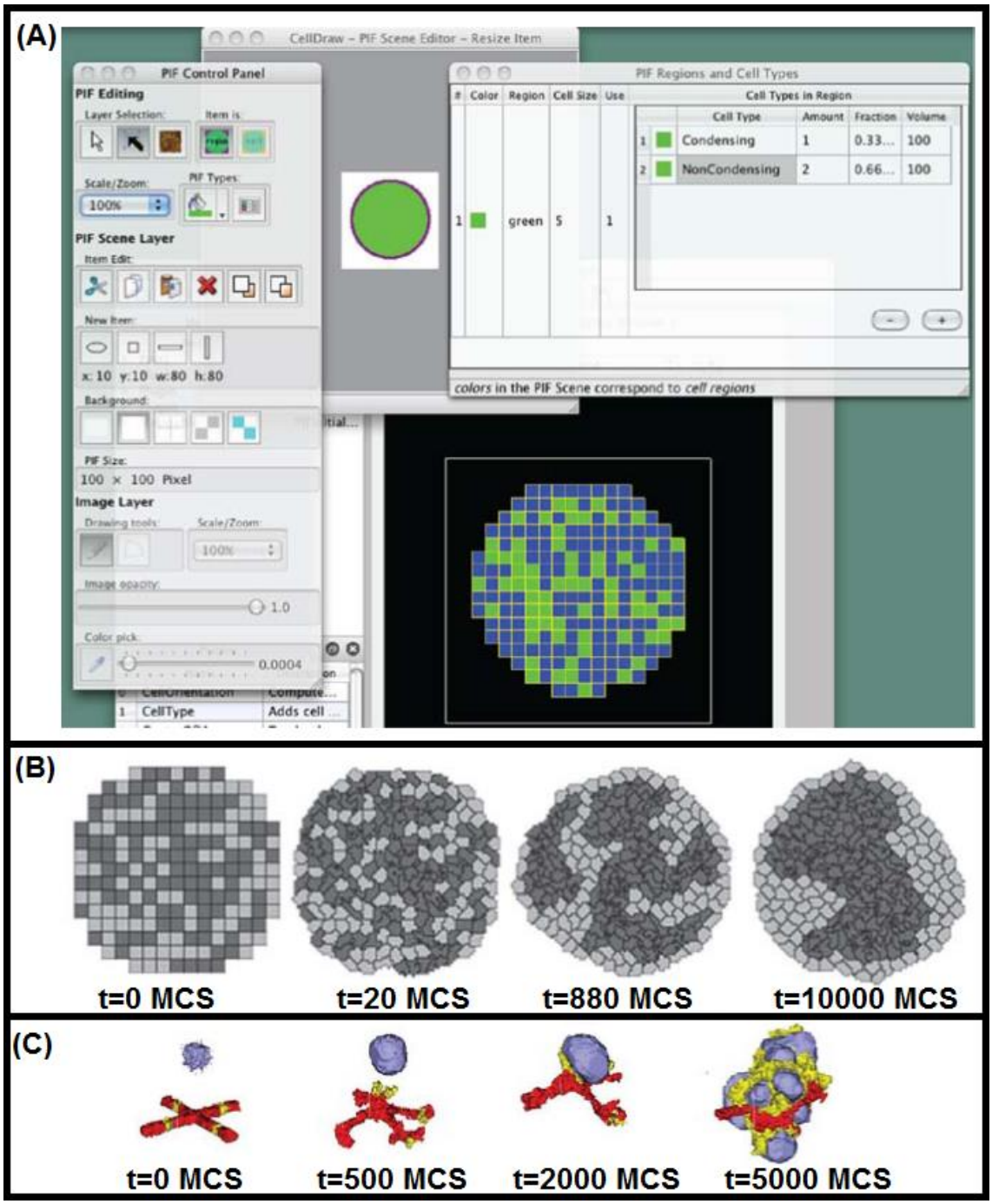


Figure 3.28 - Graphic User Interface of 'CompuCell3D' and snapshot of temporal simulations. (A) 'CompuCell3D' GUI and its graph drawing tool (B) Temporal snapshots of the cell-sorting simulation from 'CompuCell3D'. MCS is one Monte Carlo Step (C) Snapshot of vascular tumor simulation taken at different steps. Adapted with permission from [404].

Most of the reviewed image simulation toolboxes (see summary in Table 3.3) based their artificial object generation on mathematical models of cell shape, while one of the examples used data mining techniques to acquire the models and its parameters directly from experimental images, providing realistic simulations of biological processes, working towards a 'Virtual Cell' model [406]. The future work of the image generators will be focused on cell functionality, as cell morphology has been already studied extensively [406].

Table 3.3 - Availability of microscopy image processing toolboxes. Download location was last updated on 20/12/2019.

Toolbox	Download location	Main Reference
'SIMCEP'	http://www.cs.tut.fi/sgn/csb/simcep/tool.html	[391]
'SimuCell'	http://www4.utsouthwestern.edu/altschulerwulab/simucell/	[407]
'CytoPacq'	http://cbia.fi.muni.cz/simulator/index.php	[398]
'CellOrganizer'	http://cellorganizer.org/Downloads/	[402]
'CompuCell3D'	http://www.compucell3d.org/	[404]

The study of cell functional models (spatial and temporal cellular organization) can be used to implement mathematical models of cell migration, which is still being envisioned by scientists due to the complex nature of cell migration [41] or implement mathematical models of cell growth and division [408], allied with the usage of time-lapsed microscopy. To tackle this problem, bacteria species such as *E. coli* are the perfect organism to be used as a model, as several biological mechanisms and their spatial and temporal organization (see Section at the molecular level are fundamentally conserved along various species.

3.3. Machine Learning

Machine learning started as a field of study around 1960 from studies in pattern recognition and the development of statistical techniques in classification tasks, being regarded as one of the fields in computer science and engineering with highest growth in recent years [409]. This field of study is divided in several approaches, methods and algorithms, such as the use of Decision Trees, Support Vector Machines, Artificial Neural Networks, Logistic Regression Modelling, Instance-based learning, Clustering, Deep Learning Algorithms, Bayesian networks, Genetic Algorithms, Instance-based Learning, Clustering, Fuzzy Logic and several other algorithms [410].

This section will focus on the five algorithms that were implemented in the cell segmentation, cell tracking, and classification tasks performed in this in this research work: Decision Trees, Support Vector Machines, Logistic Regression Modelling, Instance-based Learning (namely the k-nearest neighbour algorithm) and Clustering (namely the DBSCAN algorithm), and how they can be used in each specific task.

3.3.1. Overview and Approaches

The design of classification tasks starts with two steps that are prior to the decision of the method: data collection and feature selection [410].

Data collection is normally considered to be the step with the highest cost during the classification tasks (in terms of time and monetary resources), depending if the user is using previously

acquired data or if he needs to acquire new data [410]. Usage of previously acquired data is normally associated with data mining techniques, specifically with the use of large databases, both stored in offline data storage units or on online servers [411]. As stated in Section 1.1, all the data used for this work was collected in the Laboratory of Biosystem Dynamics, so no data mining techniques were required during data collection, although it should be noted that the collected data has been stored online, and can be used in future works using the aforementioned data mining techniques.

The feature selection step allows the transformation of the initial set of collected data into sets of informative and non-redundant inputs. Taking into account machine learning tasks related to image processing, different types of features can be extracted: low-level, fixed shape-matching, flexible shape features and object descriptors [412]. Low-level feature extraction is related with features that can be automatically extracted from the collected image, without any additional data about spatial information (e.g. shape) [412]. These type of features are associated with edge detection operators, such as first-order (Roberts, Sobel, Canny, Prewitt, etc) and higher order operators (e.g. Laplacian, Zero-crossing, Laplacian of Gaussian, etc) [412], curvature and motion estimation operators (e.g. Curve fitting, Optical Flow, etc) [412].

Fixed shape-matching is related to the use techniques such as thresholding and subtraction to extract simple shapes, the use of techniques such as the direct implementation of the Fourier transform for template matching or the extraction of shapes such as lines, circles, ellipses and other arbitrary shapes using the Hough transform [412]. Sections 3.1.1 and 3.1.2 has also provided a review on these features.

Flexible shape features (also known as deformable) are related to the use of techniques such as active contours, snake algorithms, deformable templates, skeletonization and distance transforms [412]. Finally, the object descriptors are related to the use of chain codes, Fourier descriptors (e.g. shift invariance, Fourier expansion, elliptic Fourier) and region descriptors [412].

Machine learning tasks are typically classified into several categories, depending on whether there is data available to processed by the learning system:

- Supervised learning: This is a category where all available inputs were labelled by an expert in the field of study to their respective output class. Training [413];

- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal (discovering hidden patterns in data) or a means towards an end (feature learning)

- Semi-supervised learning: This is a category halfway between supervised and unsupervised learning, where the available inputs and outputs are separated into an incomplete training data, with labelled and unlabelled data sets [414]

- Active learning: the computer can only obtain training labels for a limited set of instances (based on a budget) and has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labelling.

- Reinforcement learning: training data (in form of rewards and punishments) is given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

The data acquired for this work was all labelled by the bioinformatics experts from the Laboratory of Biosystem Dynamics (see Section 5.1)

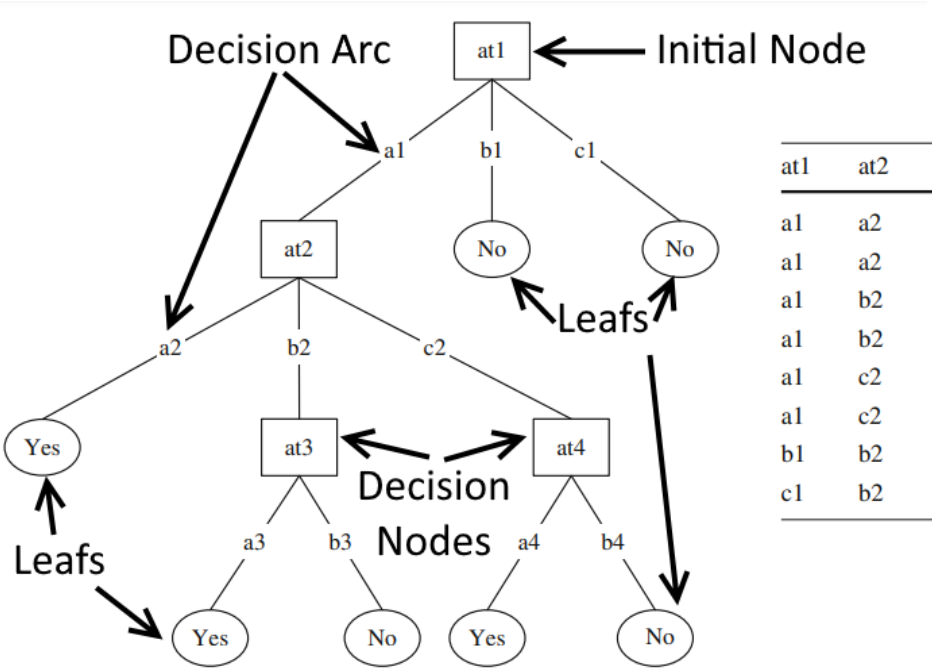
3.3.2. Applied Models

As mentioned, this sub-section will focus on the six algorithms that were applied in this work, namely: Decision Trees, Support Vector Machines, Logistic Regression Modelling, Instance-based Learning, Clustering Algorithms.

3.3.2.1. Decision Trees

Decision trees (DTs) are one the most widely used machine learning algorithms, especially due to its simplicity and the possibility of the explicit visualization of the entire decision-making process, and can be divided into two distinct categories [308]. The first category is named Classification Trees, which can predict the outcome of a class, based on the initial inputs and the structure of the tree [308]. The second is named Regression Trees and they are used to make a real number prediction such as price of goods and the minimum time to cure a specific disease (this type of Decision Trees will not be discussed in detail, since it is not used in this research work) [308].

The Classification Trees algorithm implementation starts with an initial decision node, which is labelled with the input class that can separate more efficiently the classification problem connected to other internal nodes with a decision arc, except the last node (which is called the leaf node) which is labelled with the output class (see the example in Figure 3.29) or with a probability distribution of all the output classes. There exist several measuring metrics to select the best split (Gini impurity, information gain, Shannon entropy, Tsallis entropy, variation reduction, etc) that led to the development of several decision trees algorithms (from 1970 to 2000) such as ID3, C4.5, CART, CHAID, MARS [308], [415]–[418], or more recently the Unified Criterion Decision Tree algorithm [419].



at1	at2	at3	at4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

Figure 3.29 – Example of a Decision Tree and its training set. The squares represent the Decision nodes (at1 was chosen as the initial node), the lines represent the attributes chosen for each decision arc and the circles represent the leaf, which is labelled with the output class. Adapted with permission from [416]

The CART algorithm was introduced in 1984 [308] and is still one of the most relevant and used algorithms. This algorithm is based on simple binary decisions or split into linear combinations, where each node asks a yes or no question (e.g. “value >=100 then category=hot”, value <100 then category=cold, etc), allowing the data to be fragmented more slowly and into several partitions [308]. This algorithm allows for the use of both categorical and continuous input and outputs and mainly uses the ‘Gini index’ and the ‘Twoing’ criteria to select the best nodes [420]. The algorithm is mainly based on creating the biggest possible tree, and then prune that tree at the lower levels to create numerous smaller trees. These trees are then evaluated based on different parameters such as accuracy, tree complexity and cost [420].

The relevant algorithm is the C4.5, which was, which was introduced by Ross Quinlan in 1993, as an extension of the ID3 algorithm [418], with several improvements, such as the handling of both continuous and discrete attributes, the handling of missing attributes and with differing costs, and the ability to prune the trees after the final tree is created. The C4.5 algorithm has two main differences from the CART algorithm: the first one is that it uses normalized information gain (calculated as the difference of entropy) as the selection measurement criteria [418]. The second main difference is that it allows nodes to be split into several decisions, and not just a binary decision [418].

Decision trees have been studied and applied in areas such as Bioinformatics [421], Healthcare [415] and in Computer Vision applications [422]. In this research work, only the CART algorithm and its several variations have been applied to discard or join over segmented objects (see Section 4.3.1) and in classification of different development phases of FtsZ rings (see Section 4.3.2).

The main advantage of using DTs over other machine learning algorithms is that they are very easy to interpret (so they are not used as classification ‘black boxes’), they can easily handle missing and skewed values (as they are robust to outliers) and are a non-parametric approach that does not require any previous assumptions of the data distribution [423].

3.3.2.2. Support Vector Machines

Support Vector Machines (SVM) are machine learning techniques that were first presented to separate binary class problems based on linear or nonlinear separation methodologies [424].

The first method (linear) assumes the existence of an optimal hyperplane in the input feature space, separating each class, while maximizing the distance between the two classes and the hyperplane (an example of this linear separation is shown in Figure 3.30). The second method uses a ‘kernel trick’ that implicitly maps the inputs in a higher dimension feature space that are nonlinear in the original feature space [424]. The choice between of different separation methodologies is one of the most important steps in creating a SVM algorithm, since each problem can be optimized (in term of accuracy, training speed, memory constraints) based on that training choice [425].

The adaptation of SVM algorithm to multiclass has been normally branched down into a series of binary classifications, following either the One-Against-One (OAO) or One-Against-All (OAA) strategies [426], where for n classes, there needs to exist n^2 and n separation hyperplanes for the OAO and OAA methodologies, respectively [426]. However, the multiclass SVM can be extended to a one-shot multiclass classification which only needs a single optimization operation and the classification of all the classes is performed in a single step, requiring fewer support vectors than the previously mentioned multiclass approaches [427].

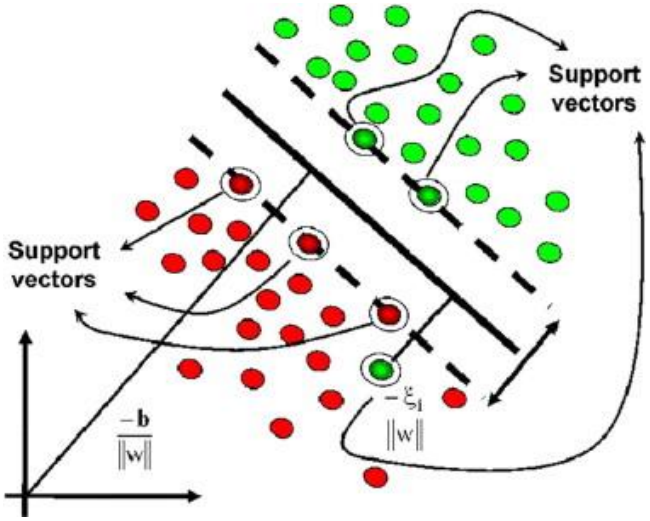


Figure 3.30 – Example a support vector machine application. Visualization of two support vector (separating the two classes (red and green) and a 2D representation of the hyperplane (black line) that divides the feature space. Taken with permission from [428].

SVM algorithms have been applied to Biomedical and Biotechnology applications, such as face recognition [429], using gene expression to classify different cancers [430] and classifying objects such mass spectra [429], [431], proteins [428], [431], DNA sequences [431] and recognize splice sites [432]. In this research work several methods have been applied in the classification of different development phases of FtsZ rings (see Section 4.3.2).

3.3.2.3. Logistic Regression Modelling

Logistic regression is a classification method that is extremely efficient in binary response problems. This method is based on the logit function, which is the natural logarithm of the ratio between the probability that an event will occur (*p)* and the probability that the same event will not occur. The logistic model (see equation 3.3) includes all the predictor variables (called covariates, represented by β_i) and the input variables (represented by x_i) [433], [434]. The logistic regression does not assume that the dependent and independent variables have a linear relationship, but between the logit of the probabilities and the predictor values [433], [434].

$$\log\left(\frac{p}{p-1}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \quad \textbf{(3.3)}$$

Equation 3.3 can then be solved for *p*, as demonstrated in equation 3.4, where *p* is the probability that the classified

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}} \quad \textbf{(3.4)}$$

The relations between the input variables and the covariates can then be evaluated in order to compute the importance of each selected predictor to the classification problem, which can lead to studies of collinearity and methods for predictor selection and removal [435]. The dependent variable is required to be categorical while the independent variables should not be normally distributed, nor linearly related between each other [433], [434].

The use of Logistic regression-based algorithms in Biomedical and Bioinformatics applications have been extensively studied [436], [437], with case studies such as modelling gene regulation [438],

identifying predefined sets of biologically related genes [439] and computer vision applications such as face recognition [440]. In this research work several methods have been applied in the classification of different development phases of FtsZ rings (see Section 4.3.2).

3.3.2.4. Instance-based Learning

Nearest-Neighbour Algorithms and all their associated methods are the most commonly used instance-based learning algorithms in classification problems. K-Nearest-Neighbour Algorithms are a non-parametric method that calculate the '*k*' (where '*k*' is a closest training examples in the feature space to the input example [316].

Specifically considering object tracking applications, the easiest way to classify an object in one frame to as the same object in the next frame is to calculate the distance between both the position of objects (this distance is normally calculated from the centroid of the objects). The most common method is the Euclidian Distance [441] between points to find matching objects between frame n and n+1 (see equation 3.5, where $\mathrm{d_p}$ is the Euclidean distance between two objects [316].

$$d_p = \sqrt{(a_n - a_{n+1})^2 + (b_n - b_{n+1})^2} \quad \textbf{(3.5)}$$

In cell tracking studies, the variables (a_n, a_n) and (b_{n+1}, b_{n+1}) usually represent the centroid coordinates of each object in frame *n* and *n+1*, respectively, but other variables can be introduced in the distance calculation, such as area, overlap percentage, perimeter (in conjunction with the centroid). Having the distance between each object in frame *n* and all objects in frame *n+1*, correspondences are made based on the minimum distance. The object in frame *n+1* closer to each object in frame n is assigned to it. If two objects in *n+1* are assigned to the same object in *n*, the closer object is assigned, until all correspondences between frames are unique [316].

The performance of simple Nearest-Neighbour algorithms is dampened when the objective is the identification of cells inside clusters in cell tracking studies [442], especially since bacteria often organize spatially in this way (see Section 2.1). One of the main problems of clustered objects that can move inside the cluster, or rotate as a whole cluster is illustrated in Figure 3.31 (examples A and B), as the use simple application NN algorithms to track these frames, will fail the identification of at least two of the objects of frame *n*+1. The example shown in Figure 3.31 is an extreme case where all objects would be mis-classified, as all object centroid's shifted positions in frame *n*+1, overlapping with the centroids of different objects from the previous frame *n*.

The use of Nearest-Neighbour Algorithms-based algorithms in Biomedical applications have been reviewed in [443], with case studies such as nuclei tracking [444] and the identification of lung cancer in computed tomography images, in combination with a Genetic Algorithm [445]. In this research work, different methodologies based on nearest-neighbour algorithms have been applied in the development of tracking algorithms (see Section 4.3.3).

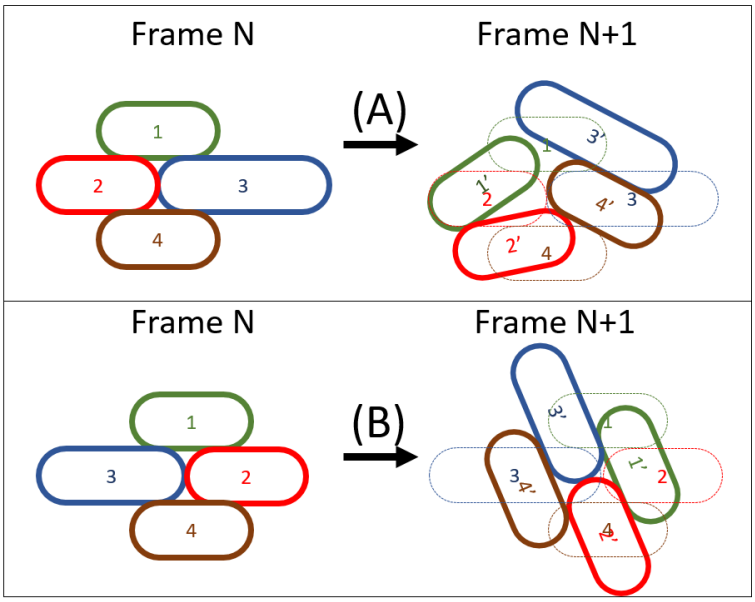


Figure 3.31 – Examples of possible misidentifications using a simple NN Algorithms. Example (A) shows how cells can move and push other cells. Example (B) shows a simple rotation of the entire cluster. In both examples, most cells would be tracked incorrectly in frame N+1, using a simple Nearest Neighbour Algorithm, as the centroids of each cell in frame N+1 (the numbers are a close representation of the centroid) are closer to other cell centroid's from the previous frame (N).

3.3.2.5. Clustering

The above-mentioned problem of object tracking inside clusters (see Figure 3.32), requires different algorithms that accounts for cluster features and singularities. One of the most used methods that allows the tracking clustered objects and the correctly identification of objects inside the clusters is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [446].

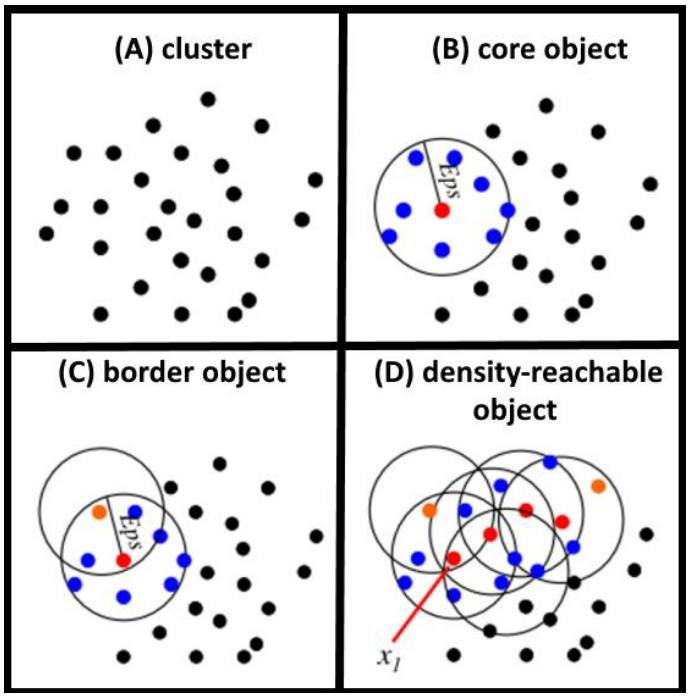


Figure 3.32 - Application of clustering algorithms to the tracking of cells inside clusters. (A) Shows an example of a cluster. (B) Shows the definition of a core object (Red dot), which is when its local density is higher than 'MinPts' (defined as the minimal number of neighbourhood objects), 'Eps' is the neighbourhood radius (C) a border object (Orange dot) is defined when its local density is less than 'MinPts'. (D) Two density-reachable objects are defined if a chain of core objects exists with distances between them smaller than 'Eps'. Adapted from [447].

The revised form of this method [447] formalizes the notion of "cluster" and "noise", using the definition of density to characterize clusters, where '*MinPts*' is the minimal number of objects in the neighbourhood, and '*Eps*' is the neighbourhood radius (see Figure 3.32).

Objects can be divided into three categories: core, border, noise and density-reachable objects (see Figure 3.32). An object is a core object if its local density is higher than '*MinPts*'. It is considered a border object if its local density is less than '*MinPts*' and it belongs to the neighbourhood of a core object. An object is classified as noise if in its Eps radius there are less than '*MinPts*' objects and none of these are a core object. Finally, two density-reachable objects are identified if there exists a chain of core objects between them (see Figure 3.32), with distances between them smaller than '*Eps*' [447].

This approach improves clustering identification when the data has dense adjacent clusters [447]. The DBSCAN algorithm also uses the concept of core-density-reachable objects, which is similar to the chain of density-reachable objects, but it eliminates border objects from the chain's ends and these objects remain unclassified until all core objects are identified [447].

The DBSCAN algorithm has two main steps: '*dbscan*' and '*ExpandCluster*'. The first step lies in finding a core object and returns all objects that are core-density-reachable from that one [447]. If it is a core object, a cluster is produced. If it is a border object, then it has no core-density-reachable objects. After all chains from the initial core object are known, a cluster is finally assigned to its best density-reachable chain and all border objects. If the object is unclassified, the algorithm runs the '*ExpandCluster*' step [447].

The use of clustering techniques in Bioinformatics applications have been extensively reviewed in [448], with the DBSCAN algorithm being used in case studies such as the mining of biomedical images [449] or the grouping of genes with similar gene expression patterns [450]. One of the recent advances in clustering algorithms have been based on the integration of nearest-neighbour algorithms into existing clustering algorithms (e.g. DBSCAN) to provide even better results, when dealing with clusters, by achieving parameter-free algorithms [451], [452]. This approach is used in this research work, where the DBSCAN algorithm is combined with a nearest-neighbour algorithm for the development of a tracking algorithm (see Section 4.3.3).

Chapter 4. Conceptual Contribution

This Chapter presents the conceptual contributions to answer the proposed main research question. Chapter 4 also includes a sub-section with the formulation of the Image Processing framework, the implementation of existing segmentation methods and the development of new segmentation methods. In another sub-section, the formulation of the Image Simulation framework and the implementation of the cell modelling features is also included. Finally, this Chapter also depicts the implementation of several Machine Learning algorithms into Bioinformatics studies. The resulting publications from this research work are cited along the Chapter.

4.1. Contribution for the Image Processing Framework

In the initial steps of this research work, a preliminary image processing toolbox started to be developed, which was published to include cell segmentation, cell tracking and an image registration algorithm, and a spot detection algorithm. This toolbox, 'iCellFusion' was published in [453], and continued the work of the 'CellAging toolbox' [454], but focusing on the fusion of different microscopy methods to provide a better integration of functional and morphological information by fusing Phase-Contrast and fluorescence microscopy images [453].

After the completion of the 'iCellFusion' [453], it was required to developed new segmentation methods for the detection of cellular structures (described in Section 2.3) such as the Nucleoids, the FtsZ ring, Min System Proteins, Protein Aggregates, Inclusion Bodies and other fluorescently labelled structures. From this necessity, a new image processing framework started to be developed, called "SCIP – Single Cell Image Processing Toolbox" [455] with its main objective of being capable of analysing multi-modal, multi-process, time-lapse microscopy images, while improving the image alignment, cell segmentation and tracking algorithms that were previously developed, namely 'CellAging' [454] and 'iCellFusion' [453]. The source code of the tool, an executable file, the Toolbox Manual, the raw images and the segmentation files that are used to validate the tool in Section 6.1 are all publicly available at: http://www.ca3-uninova.org/project_scip [455].

It is noted that in this Dissertation, the images where segmentation was performed are named as 'morphological' images (e.g. Phase-Contrast images), as they are normally used just for their morphological features (cell shape and size) and to detect cell growth and division (as described in Section 2.3). Contrarily, 'functional' images are the images that provide data on internal cell processes (e.g. fluorescence images) but lack the ability of providing clear morphological cell features.

4.1.1. Graphic User Interface and Workflow

The Graphic User Interface (GUI) of the SCIP toolbox is shown in Figure 4.1 (this platform was developed using MATLAB® API version 2016a but was also tested correctly for version 2017a and 2018a). All Buttons and User Controls of interest are shown in the Annexes Section A.1 - SCIP's User Interface - Buttons and Controls. Unless otherwise stated, all functions are native from MATLAB® and are carried out based on their default parameters and implementation [224], [225].

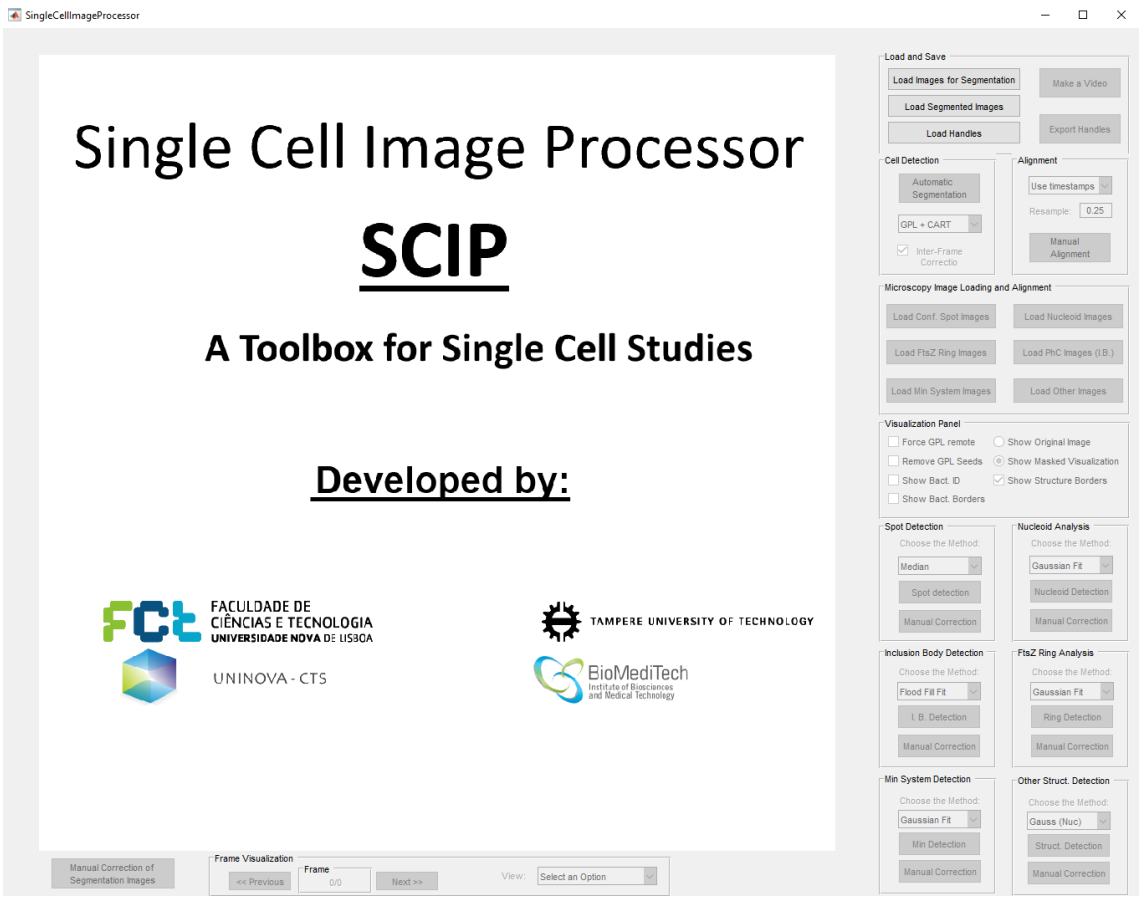


Figure 4.1 - Graphic User Interface of the Single Cell Image Processor toolbox

The toolbox workflow is divided in three major steps (the colours of the boxes in Figure 4.2 represent each step). Initially, the options available to the user in the Graphic User Interface (GUI) are (see buttons in Figure A.1): load morphological images, load segmented masks, or loading a previously saved Handles file (black boxes in Figure 4.2). After completed one of these steps, the user will be able to segment the morphological images, load the functional images, intra-align the functional images, inter-align the segmentation masks, detect the internal cellular structures (dark grey boxes in Figure 4.2) and finally extract the desired results (light grey box in Figure 4.2).

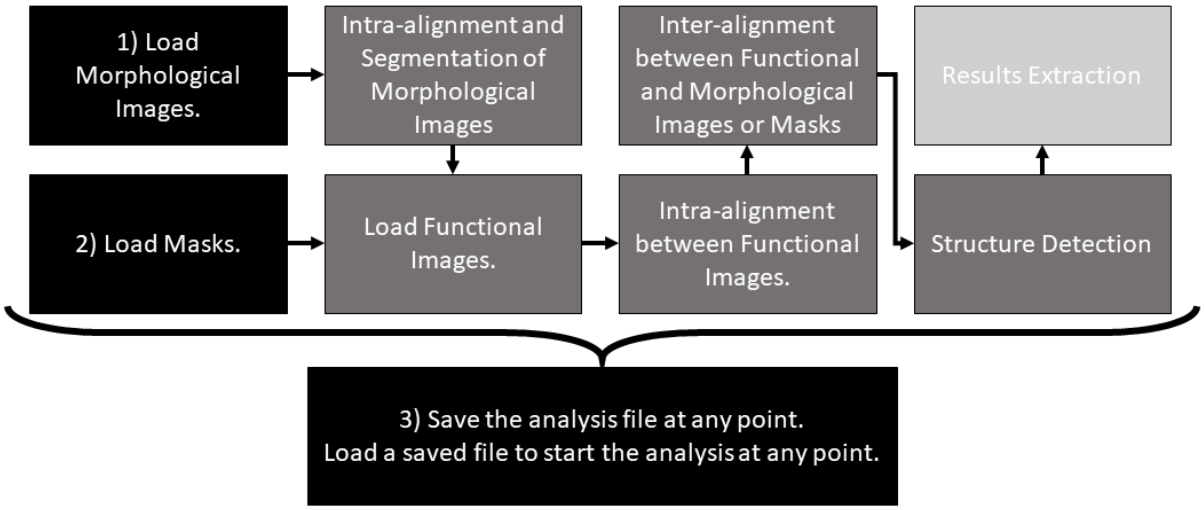


Figure 4.2 – Workflow of the Single Cell Image Processor toolbox

At any time during the workflow process, the user can save a Handles file in a ".mat" file using the "Export Handles option". Note that all steps in this workflow are automatic but can be manually adjusted.

It is noted that only three functional images (Channels) can be loaded at the same time, as the toolbox uses the red, green and blue colours to display the distributions of up to three fluorescent probes). The loaded images are required to be in formats supported by MATLAB® (preferably TIFF - Tagged Image File Format), while the masks should be using the RGB system with a unique colour code for each cell. Loaded images are required to be in formats supported by MATLAB® (e.g. TIFF). Images from a given microscopy modality need to be of the same size, while the size of images of different modalities can differ.

If the user only loads one image, and aligns one functional image, then the user should choose the option 'No Timestamps' (Figure A.7-A). If the user loads a set of images (e.g. from a time series), the filenames must follow the format: 'Setname't001.tiff, 'Setname't002.tiff, etc. This is done by choosing the option 'Use t(\d+) pattern (Figure A.7-A)'.

It is possible to have multiple functional images associated to one morphological image (e.g. 5 multiple functional images taken every minute, for 1 morphological image taken every five minutes). Given multiple allocations to one morphological image (Figure 4.3), SCIP compares the minimums between the absolute difference of the Morphological Indices and the Functional Indices. In case of a tie, SCIP selects the first allocated image.

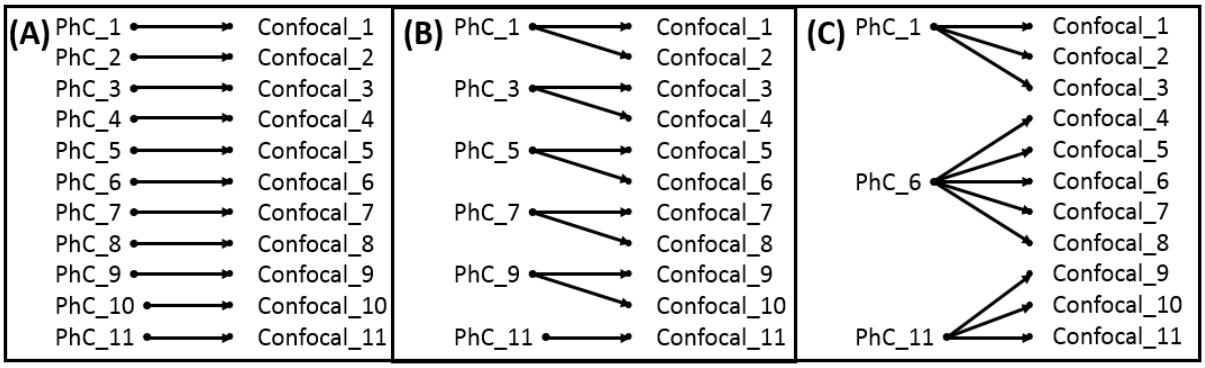


Figure 4.3– Allocation of Morphological Images. (A) 1-to-1 allocation. (B) 1-to-2 allocation. (C) 1-to-5 allocation.

The 'Use Timestamps' option (Figure A.7-A) requires a metadata file 'meta.txt' in the same folder as the images, consisting of lines with the following format (UTC timestamps have a 1 second precision):

- <filename><tab>modified<tab><YYYY>-<mm>-<dd>T<HH>:<MM>:<SS>Z, where:
- <filename>: base name of the file (for example 1.tif)
- <tab>: a tabulation character (ASCII character 9, also known as HT or ^I)
- <YYYY>, <mm>, <dd>: year, month, day
- <HH>, <MM>, <SS>: hour, minute, second

If the images lack timestamps or the required pattern, a warning message is displayed, and the images are not loaded.

A description of the implemented and developed methods is provided, namely the Cell Segmentation methods, the Image Alignment algorithms, and the Structure Segmentation methods is provided in the next sub-sections.

4.1.2. Image Registration Methods

The merging of information between morphological (e.g. segmentation borders) and functional images (e.g. intensity inside the cells) requires the registration of both images. This can be done by the simple overlay of both images, if the images are acquired by the same camera sensors and at the same time point, otherwise (e.g. when images are taken from different viewpoints and/or by different sensors) this process requires the use of image registration algorithms [219], as detailed in Section 3.1.1. In this research work, three different image registration methods (also known as image alignment) were implemented (separated in different Sections)

4.1.2.1. First Registration Method

Based on the survey done in [234], the first method can be classified as an automatic intensity-based process (no features are extracted), intra-modal (as it registers Phase-Contrast images taken at different time-points that with possible spatial drifts that can occur during the acquisition process) and based only on translations transformations, using raw and intrinsic data.

This method is applied on a global domain and is based on an exhaustive search. In the case where the acquired images are taken by the same camera sensor, but timepoints, resulting in a small drift these images cannot be simply overlaid, as usually there are always misalignments between consecutive frames. An example of the implementation is presented in Figure 4.4.

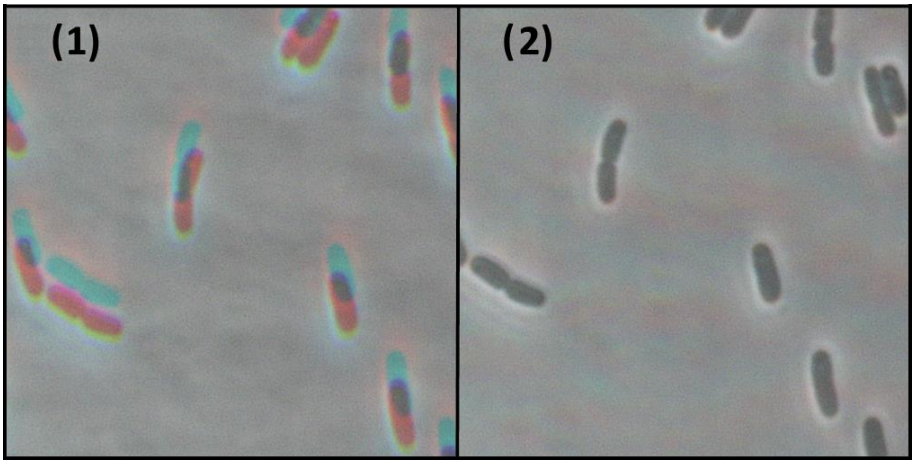


Figure 4.4 – Example of Intra-Modal Registration. (1) Example of a drift in time-series of Phase-Contrast images (3 images acquired every 5 minutes). No intra-modal registration was done; (2) The same time-series of (B-1) but now with the application of an intra-modal registration technique based on a Phase-Correlation method (using the two-dimensional Faster Fourier transform method). The examples shown (1 and 2) represent the top 500x500 square of all the superimposed images, as by applying this transformation, the resolution of the images was changed from 2560x1920 to 2527x1878.

The first step in this process is to remove any uneven background illumination, as this problem can influence the effectiveness of the intra-alignment process and can be done by applying an inverted Gaussian filter (which functions as a high pass filter). The filter specifications are: 51x51 pixel window, a standard deviation (σ) of 8 and an impulse response given by equation 4.1, where x and y are the distance to the origin in the horizontal and vertical axis, respectively.

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.1)$$

Following this filtering process, the intra-modal registration methodology is applied, based on an exhaustive search of the translation matrix that maximizes the cross-correlation function. This is done with the Fast Fourier Transform, as the cross-correlation function between two images i_1 and i_2 can be computed as:

$$i_1 \star i_2 = i_1 * i_2^- = \mathcal{F}^{-1}\{\mathcal{F}\{i_1\} \circ \mathcal{F}\{i_2^-\}\} \quad (4.2)$$

The intra-modal translation matrix, which is described in equation (4.3), is applied to the next consecutive frame is an average of the values of t_x and t_y, that maximize the cross-correlation function, between that frame and up to 10 of the previous frames (so in a timeseries of lower than 10 frames, all frames are used for averaging).

$$T_{intra-modal} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

This first registration method is applied before segmentation for Morphological images, and during the loading of the Functional Images (see Figure A.6), preceding the process that has been defined as inter-modal registration, which is the alignment between images acquired by different microscopy modalities, or in this case between Morphological and Functional Images (as seen in the workflow in Figure 4.2).

4.1.2.2. Second Registration Method

The second method is an automatic intensity-based process (no features are extracted), intra-modal (as it registers the Phase-Contrast images with other microscopy modalities, such as confocal) and based on affine transformations, using raw and intrinsic data. This method is applied on a global domain (optional local adjustments are allowed) and is based on search methods. An example of the implementation is presented in Figure 4.5.

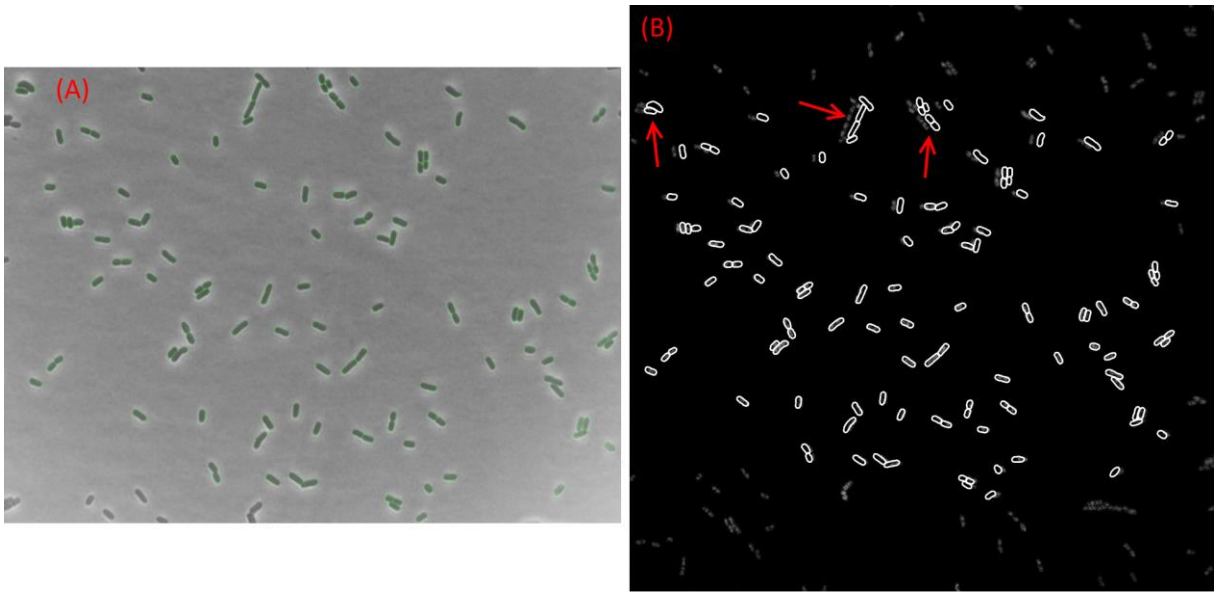


Figure 4.5 – Automatic Alignment Errors. (A) Segmentation done on a Phase-Contrast Image (green borders represent the segmentation); (B) incorrect automatic alignment, showing local errors (see red arrows)

During the alignment process, a progress bar appears, which is interrupted with the Local Adjustments popup (Figure A.7-B). This popup is required because Functional Morphological images may not be recorded exactly at the same time, producing local image distortions. Local adjustments are calculated by a local translation using the cross-correlation computation for each cell cluster. Two cells will belong to the same cluster if the smallest distance between them is less than half of the mean cell width (obtained from all the segmented cells).

The inter-modal registration assumes that since the images are taken by different camera sensors, they are normally taken with different viewpoints, different resolutions and even at different timeframes (e.g. one Morphological image can be used to align different Functional images, as seen in Figure 4.3). This makes the alignment process more complex than in the inter-alignment modality (which just uses translation transformations), which requires the usage of a 2-D affine geometric transformation [456], involving the use of several iterations of translation, rotation, scale, and shear transformations, similarly to what was implemented in CellAging [365].

The main pitfalls in the automatic alignment process are that some images can have cells with zero fluorescence, cells with much higher intensity than the average and some images can have image artifacts or the object blurring due to the inherent image formation process (multiplication of the convolution of the real light sources with the point spread function) that can interfere with the alignment process. The presence of large cell clusters can also influence the correlation processes that are used to automatically find the best transformation matrix. Therefore, if image registration problems persist, as shown in Figure 4.5, the user can use a manual align strategy analogous to what was developed in 'iCellFusion' [453].

4.1.2.3. Third Registration Method

The third method is based on the manual placement of control-points, corresponding to a semi-automatic feature-based process (is based on extrinsic control-point features), intra-modal (as it registers the Phase-Contrast images with other microscopy modalities, such as confocal) and based on affine transformations. By adding more control-points than the minimum for each transformation, the method allows for adjustments when more control-points, using direct methods to calculate the transformations. An example of the implementation is presented in Figure 4.6.

In iCellFusion [453], the manual registration was performed by a feature-based registration process that used a control-point mapping interface. It was found that this decreased the errors mentioned by allowing the user to define additional points to guide the image transformation required to align the different images and is based on the MATLAB® 'Control Point Selection Tool', and offered simultaneous overview of both images to be registered (on the bottom of the GUI) with zoomed user defined areas (on the top). The control-points identify landmarks common to the fluorescence and Phase-Contrast images.

The main problem with the strategy proposed in iCellFusion [453] was that the final results were only observed at the end of the registration process (after the insertion of all the control-points). This problem was solved in the SCIP toolbox by overlaying both images, placing the control-points and moving them until both images are completely aligned. This strategy (as shown in Figure 4.6) also has the advantage of solving local alignment errors, which could not be easily solved with the previous strategy.

After all control-points are placed, and both images are correctly aligned, the user can finish the alignment or move to the next frame. In this Alignment process (as shown in Figure 4.6), if no control-points are placed, then the images are simply scaled, by up-sampling the lower resolution image with a bi-cubic interpolation, as it preserves the image contrast [457].

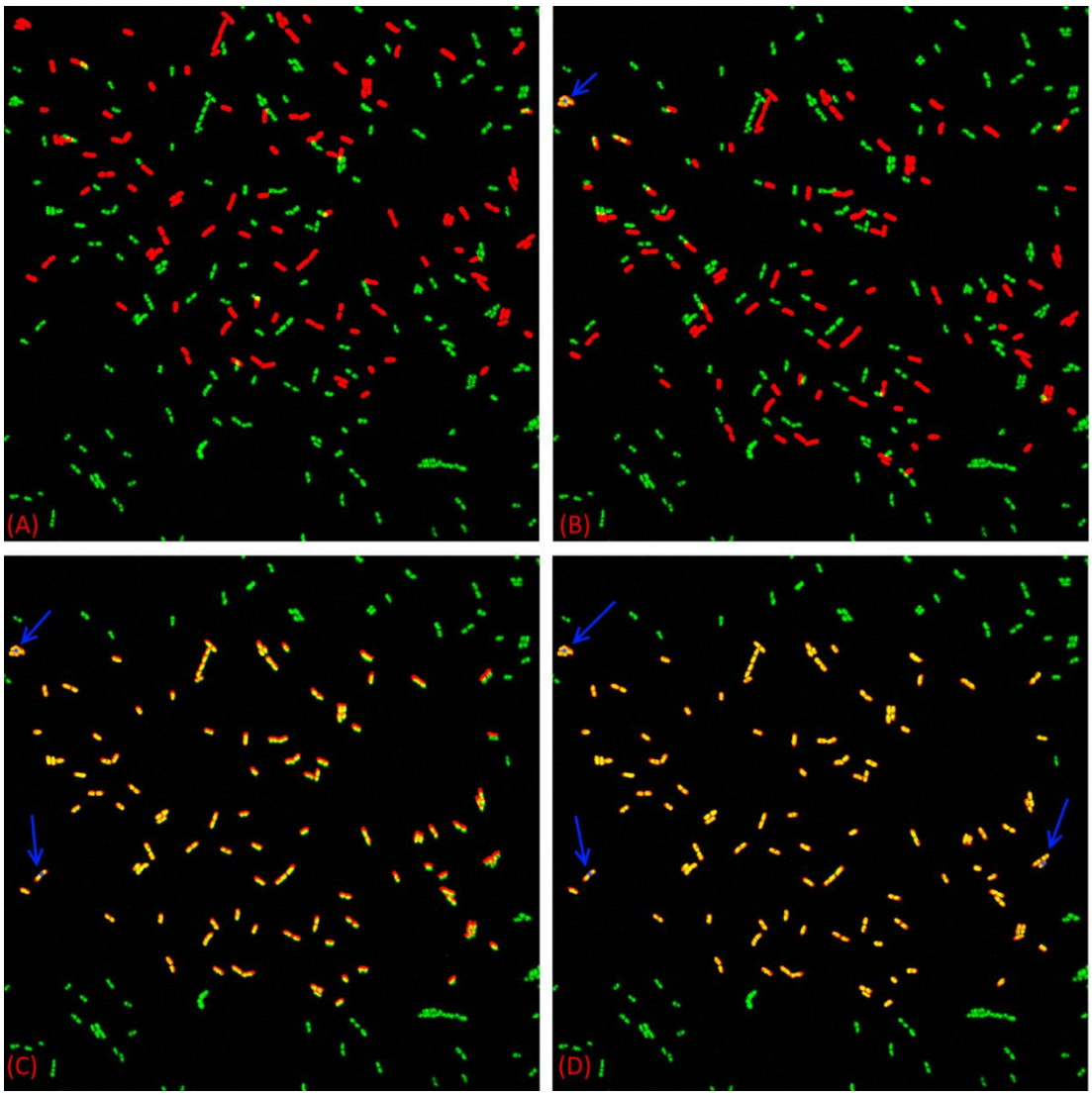


Figure 4.6 – Manual Alignment Strategy with Control Point (blue dots) Mapping. Dots are highlighted with arrows. (A) No control Points; (B) one control point; (C) two control points (D) three control points

If one point is placed, then only translation and scaling can be applied, if two points are placed then translation, rotation and scaling can be applied, and finally if three or more points are placed, then all transformations (scaling - SC, rotation - R, translation - T and shear - SH) can be applied, by multiplying the images with the Transformation Matrix ($T_{inter\text{-}modal}$), as seen in equation (4.4), where $T_{inter\text{-}modal}$ Matrix can be obtained by multiplying each transformation Matrix (see equation (4.5)), obtaining a full affine transformation.

$$T_{inter-modal} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} = [SC][R][T][SH] \quad \textbf{(4.4)}$$

where, each transformation matrix is defined as:

$$SC = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}, SH = \begin{bmatrix} 1 & h_y & 0 \\ h_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(4.5)}$$

4.1.2.4. Registration of Multiple Images

If the user is working with multiple channels, then each image will have to be registered independently, e.g. two different functional images aligned to the same segmentation done on a single morphological image. Figure 4.7 shows an example of an image containing Nucleoids that require the third registration process (based on the manual alignment correction), after the automatic registration process (the second registration method) failed to provide a good result, while the one containing the FtsZ rings was correctly aligned based on the automatic process (the second registration method).

In Figure 4.7-A, the segmentation done on Phase-Contrast images is shown (white borders) automatically registered (with several local incorrections) with a confocal fluorescence image showing nucleoids. In Figure 4.7-B, a second confocal fluorescence image was added, showing FtsZ rings highlighted in green, which was taken from a different viewpoint and that can be automatically overlapped with the segmentation (this can be observed, as the green parts of the image are all inside the white borders, while the red parts of the image aren't). Due to this situation, it is only necessary to manually align the image containing the nucleoids. Figure 4.7-D shows the example where both morphological images are correctly aligned to the segmentation (white borders).

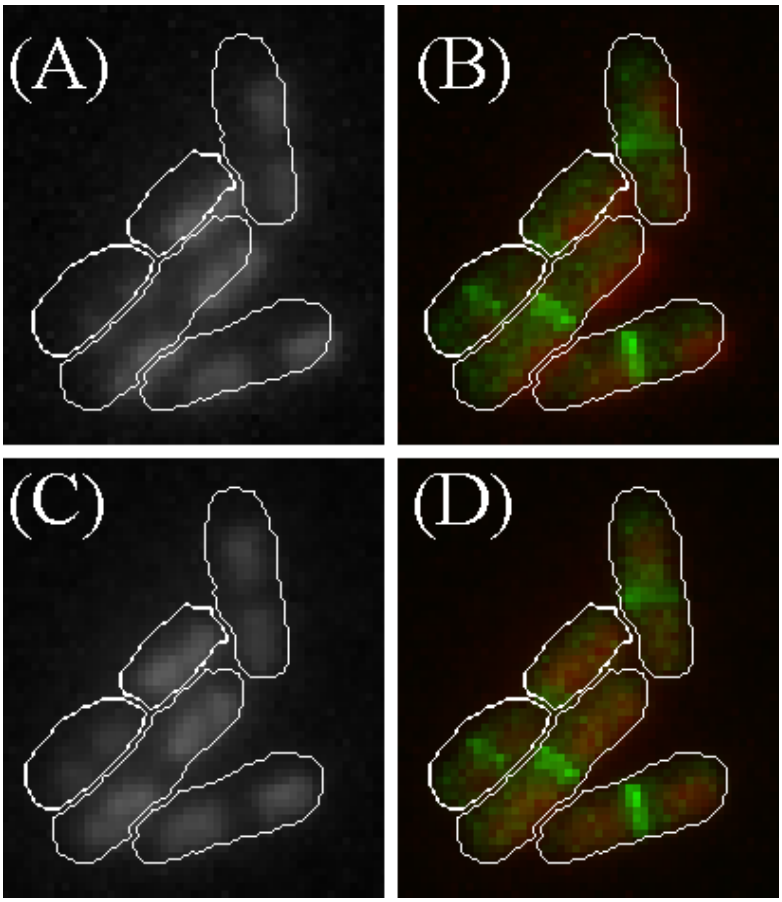


Figure 4.7 – Example of (A) erroneous and (C) correct alignment between the morphological segmentation and the functional images (with Nucleoids). (B) shows how this affects the overlay of this image with the other functional image (with FtsZ Rings) (D) shows the correct overlay of both images.

It is noted that single-channel images in the SCIP tool are shown in grayscale (like in Figure 4.7-A and Figure 4.7-C), while images with information from two or three simultaneous channels are shown using the RGB colour system (see Figure 4.7-B and Figure 4.7-D). Since each image is acquired separately and all image processing algorithms work on grayscale images, it is important to note that the RGB colouring of images with multiple channels is used just for visualization purposes.

4.1.3. Cell Segmentation Algorithms

The selection of the first option of the SCIP toolbox (see button 'Load Images for Segmentation' in Figure A.1), prompts the activation of the Cell Segmentation Interface options (see Figure A.2), where two methods for automatic segmentation can be selected. Afterwards, the automatic segmentation can be performed by pressing the Button 'Automatic Segmentation', similarly to the methods developed in 'CellAging' [458] and 'iCellFusion' [453].

The first method (Path 1 in the Segmentation Workflow, see Figure 4.8) can be selected by choosing the 'GPL+CART' option in the dropdown box of the GUI. The second method (Path 2 in the Segmentation Workflow, see Figure 4.8 and by selection the 'Otsu + Median').

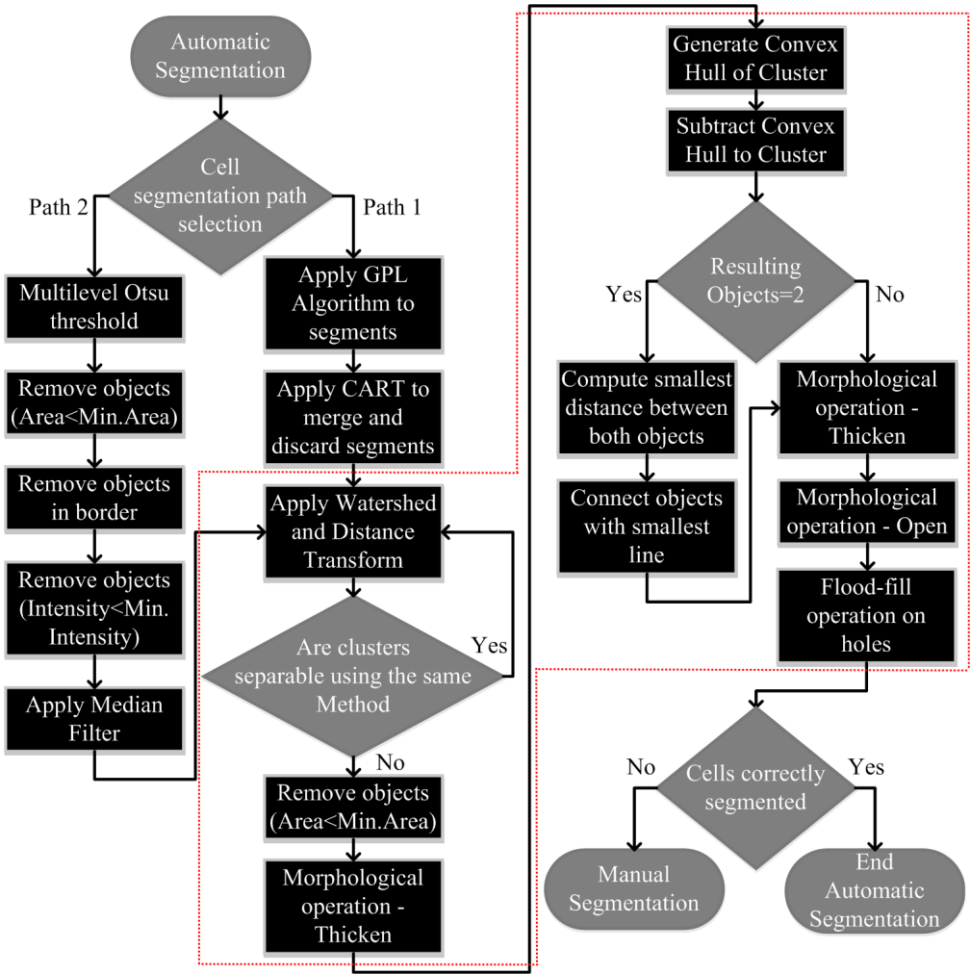


Figure 4.8 - Segmentation workflow of the two cell segmentation algorithms, respectively Paths 1 and 2. The red dash line represents the newly developed steps in SCIP, that were not present in the previous toolboxes.

The first method uses the Gradient Path Labelling Algorithm [458] to create the segmentation seeds. creates over-segmented sections, and then uses the Classification and Regression Trees Algorithm [308] to merge and discard inappropriate sections. This Algorithm was previously trained for differential interference contrast microscopy images [306]. For the SCIP toolbox, a new set of images acquired with Phase-Contrast microscopy has been used to train the merge and discard classification algorithms. The specifications for the implementation of the CART algorithm is presented in Section 4.3.1. Figure 4.9 shows the usage of the 'GPL + CART' segmentation workflow using an example image acquired with Phase-Contrast microscopy showing the initial GPL segmentation (Figure 4.9-A), proceeded by the application of the Discard Algorithm (Figure 4.9-B) and followed by the application of the Merge Algorithm. An example with a large cluster of cells is shown in (Figure 4.9-D). The Inter-Frame Correction option (checkbox in the GUI interface) further enhances the segmentation's quality using information from subsequent frames to merge or discard sections.

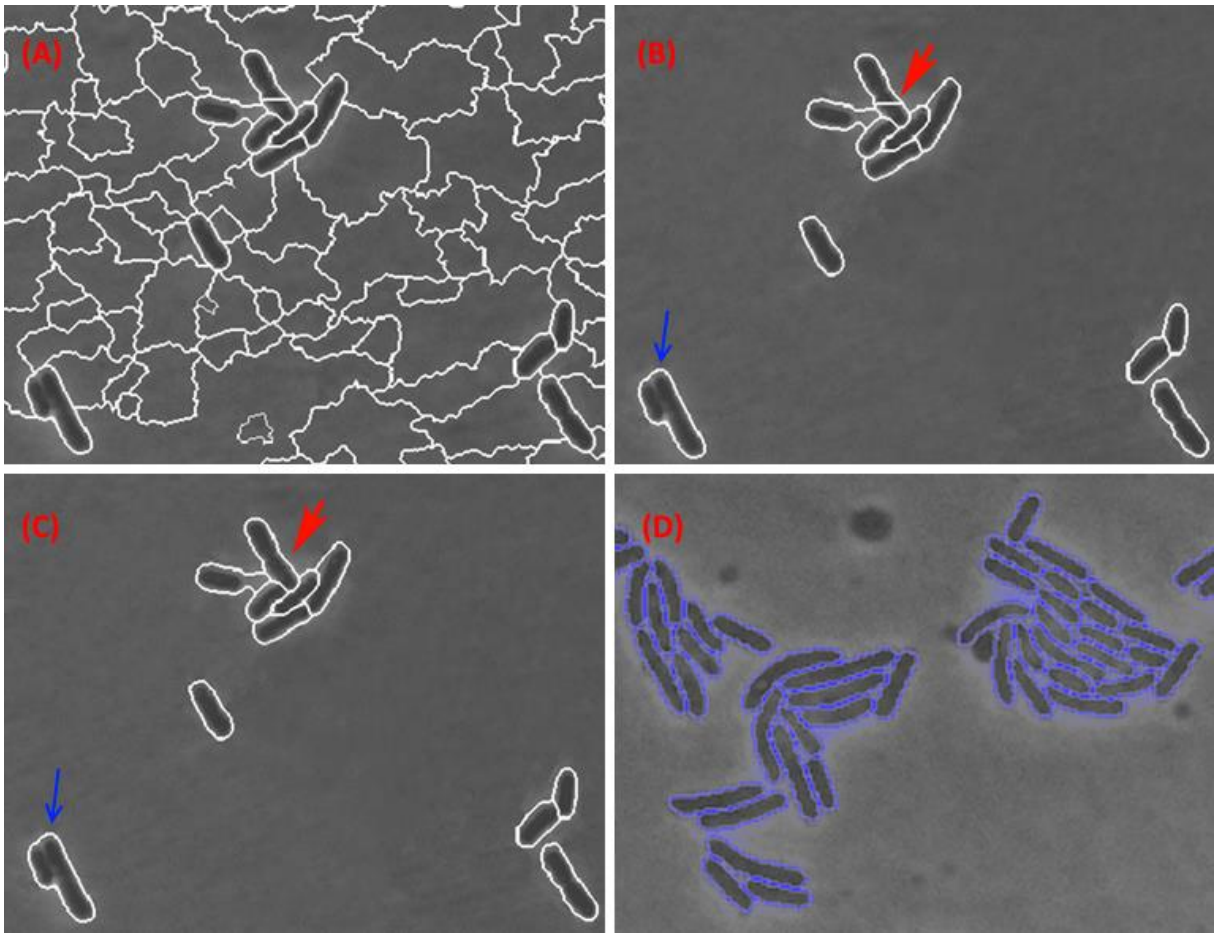


Figure 4.9 – Example of the 'GPL + CART' usage of a Phase-Contrast image. White and Blue Borders represent the segmentation contours. (A) Initial GPL segmentation; (B) Application of the Discard Algorithm; (C) Application of the Merge Algorithm. Red arrow indicates where the merge algorithm was applied) (D) Example of the final application of both algorithms in a clustered environment.

The second method is based on the segmentation method developed in 'CellAging' [458], based on Multilevel Otsu's thresholding [258] and the implementation of a median filter to reduce noise while preserving cell edges. The resulting objects are removed according to their minimum area and minimum pixel intensity

Both methods ('Otsu + Median' and 'GPL + CART') still had segmentation problems, especially for dense clusters, so in the SCIP toolbox, additional segmentation steps were implemented (see red

dashed line in Figure 4.8). The additional steps gathered resulting masks from either segmentation methods ('Otsu + Median' and 'GPL + CART') and started a cluster separation by combining the Watershed Algorithm and the Distance Transform (using the '*watershed*' and '*bwdist*' functions), which is an iterative method, that can check if no more clusters can be divided. Finally, the Convex Hull of each cluster is generated (using the '*convhull*' function) and is subtracted from original cluster. This subtraction is connected by the smallest possible line to make a final separation of the clusters. These steps are followed by the morphological operations of thickening, opening and flood-filling to provide filter the shape of small incorrections and holes.

During the automatic segmentation process, a progress bar appears, which disappears when the segmentation is complete. After this process is complete, the 'Manual Corrections of Segmentation Images' button and the 'Alignment' panel are enabled (see section 4.1.2).

Manual corrections and optimizations have been implemented using the same strategy that was developed in the 'iCellFusion' software [453]. This process starts by pressing the button "Manual Corrections of Segmentation images" (bottom left button in GUI, as seen in Figure 4.1) opens a new window (see Figure A.3), which allows the interaction with the segmentation results by selecting cells using the primary mouse button (which become highlighted in green), by adding an interactive polygon segment (with the '*impoly*' function). Clicking on the edge points of the polygon allows the tuning of the shape (see Figure 4.10-A). Finally, the user can also draw or modify the segmentation results using the '*imfreehand*' function (see Figure 4.10-B). Cells to which a segmentation line is added or modified change their highlight to a distinct colour (white borders in Figure 4.10-C). A 'Help' Menu, with all the options and specific keys can be accessed by pressing 'F1' (see Figure A.4).

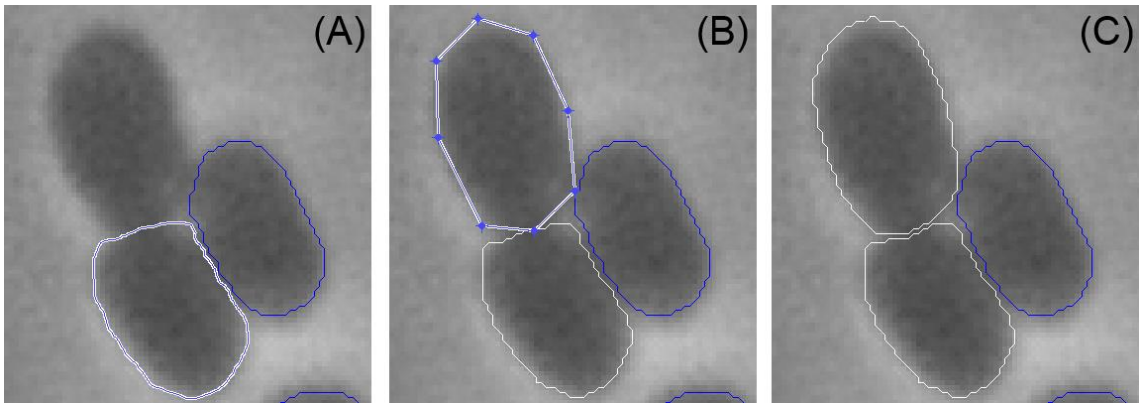


Figure 4.10 - Example of an on-going process of manual segmentation correctio. (A) using a free hand drawing function (B) using a predefined polygonal shape, as it is visualized by the user. (C) final results of manual segmentation (in white) and automatic segmentation (in blue).

If the new polygon does not intersect any existing segment, a new segment is then created (see Figure 4.10C). Otherwise, the action is queried (popup menu in Figure A.5-A), with the possible actions being listed. Before closing the manual correction window, the user must save the changes by pressing 'u' and confirm by selecting 'Yes' from a new popup menu (Figure A.5-B).

The result of clicking each button of the queried action in Figure A.5-A is shown in Figure 4.11 (here red borders represent the resulting segments).

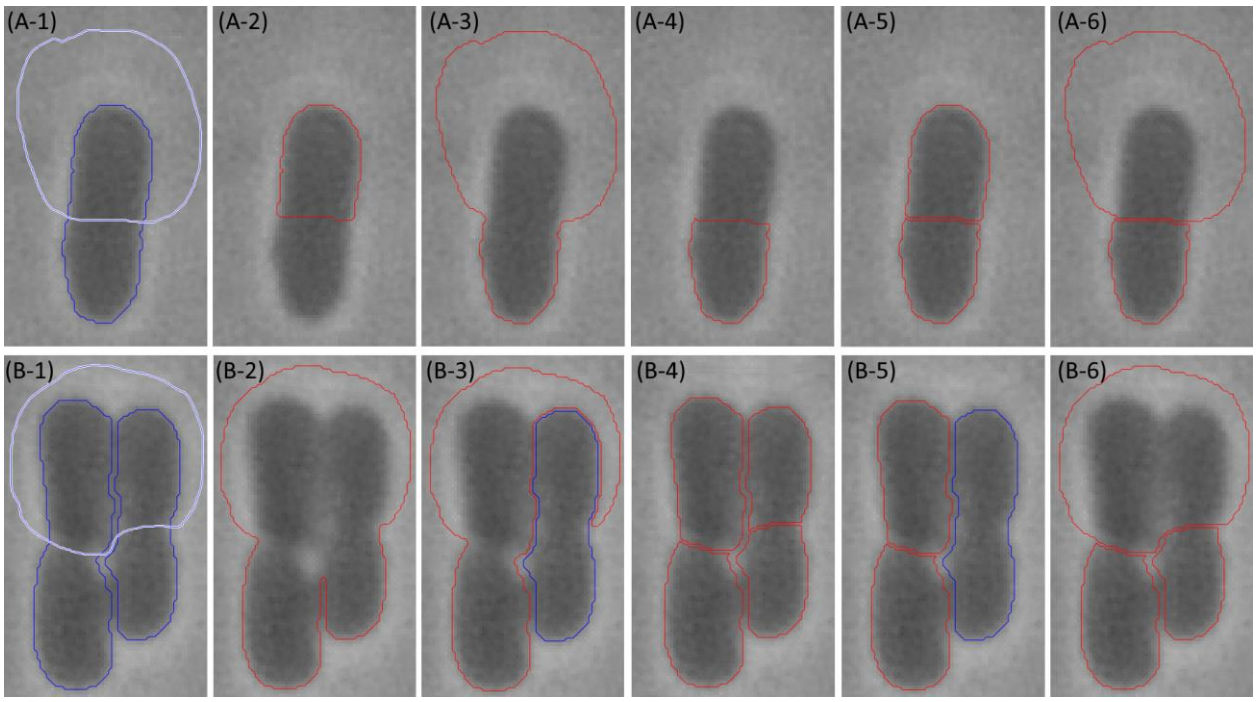


Figure 4.11 – Manual corrections when the segmentation overlaps with existing objects. (A-1) overlap with a single cell. Results of pressing on: (A-2) 'i' button; (A-3) 'u' or 'e' buttons; (A-4) 'u' button; (A-5) 's' or 't' buttons; (A-6) 'a' or 'r' buttons. (B-1) overlap with two cells. Results of clicking on: (B-2) 'u' button; (C-3) 'e' button; (B-4) 's' button; (B-5) 't' button; (B-6) 'r' button.

In some cases, clicking on different buttons result in the same effect, depending if only one or more cells are intersected. The following effects occur if the user presses on the button:

- 'i', the resulting segment will be the intersection between the drawing segment and the existing segment (Figure 4.11-A-2).
- 'u', the resulting object is the union of the drawing segment and the existing segment (Figure 4.11A-3) or the union of all existing segments that touch the drawing segment (Figure 4.11B-2).
- 'x' or 'd', the resulting segment is the subtraction of the drawing segment with the existing segment (Figure 4.11-A-4), even when two objects exist.
- 'e', the resulting object is the extension of the largest segment (Figure 4.11-A-3). For two existing segments, only the largest one is extended (see difference between red and blue borders in Figure 4.11-B-3).
- 's', the result is the split between all touching segments. With 1 cell, the resulting object is two separated objects (Figure 4.11-A-5). With two objects, the result is four new objects (Figure 4.11-B-4).
- 't' or 's', both split the object by the intersection line (Figure 4.11-A-5). When two objects exist, if the user clicks on 't' only the largest segment is split (Figure 4.11-B-5).
- 'a' or 'r', which splits and joins the drawing segment with the existing segment (Figure 4.11-A-6). If multiple segments touch the drawing segment, only the 'r' button will create a new segment based on the drawing and split all those touching the drawn segment (Figure 4.11-B-6).

After the segmentation process (automatic or manual) is completed, the loading panel of other types of microscopy images is activated by the Microscopy Image Loading Interface option, by loading images with e.g. Nucleoids and FtsZ Rings. After the alignment of the images and the segmentation process is also completed and if the user loaded a timeseries, the SCIP toolbox will continue its image processing workflow by providing a cell tracking step, including the possibility of tracking divisions and the lineages along the divisions.

4.1.4. Cell Tracking Algorithm

The cell tracking step starts by assigning an ID to every cell in the first frame and assign a parent ID to all the cells in the following frame (this process is continued to all the following frames). This parent ID assignment is based on the cell that has a biggest overlap percentage from the previous frame based on a Nearest-Neighbour approach. The proposed methodology works similarly to Hand et al. [253] where the nearest cell is used instead of the most overlapping cell, as in both methodologies, the image registration (based on the intra-modal alignment process) allows the cells to be tracked more efficiently (this is verified in the results validation in Sub-Section 6.1.2., where the lineages were manually inspected.

It should be noted, that this methodology works on *E. coli* cells that are observed by the LBD group, because these cells are normally fixed or are placed in an agarose gel which reduces the cells motility. It is also important to note, that *E. coli* cells in normal conditions divide in two almost identical cells and always along the minor axis by the mid-cell point, which facilitates the identification of the cell division process. An example of this division and tracking step is shown in Figure 4.12.

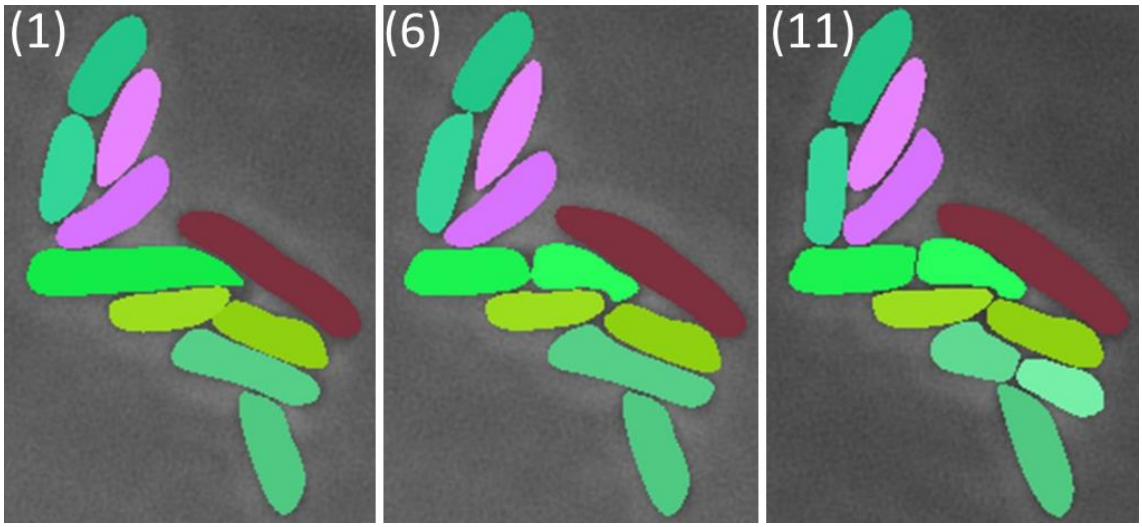


Figure 4.12 - Example of cell tracking and division detection results. Assorted colours represent the segmentation of different cells. A similar hue indicates shared ancestry. Numbers represent the time (in minutes) of the timeseries acquisition.

An example of the lineage tracking procedure is presented in Figure 4.13. If a new cell appears near another, it is assumed that both descend from the same cell from the previous frame and the algorithm tracks it as a division forcing both cells to have their parent id to be assigned as the same.

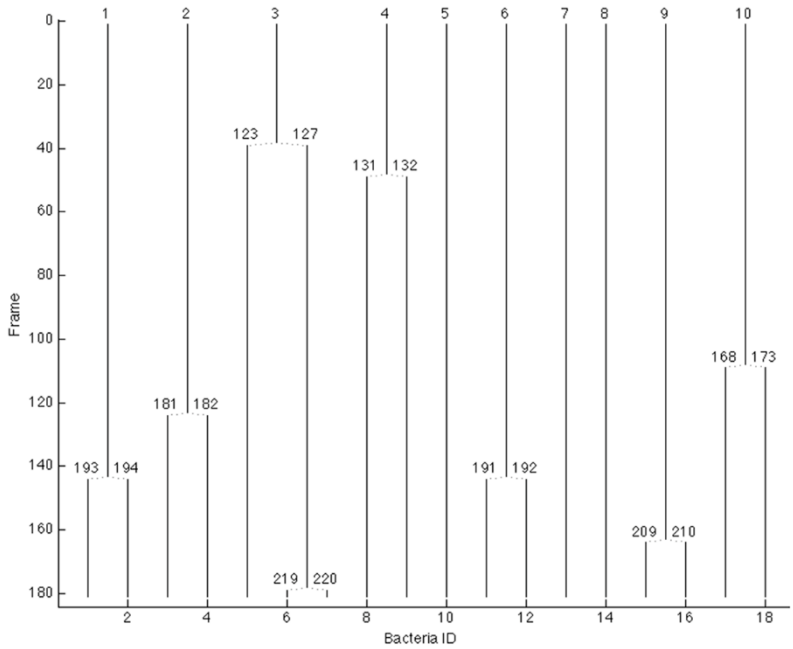


Figure 4.13 - Example of a cell lineage plot of a timeseries with a duration of 180 minutes. The numbers at the top represent the ID of the cells at the start of the measurement. When a division occurs, the new daughter cells have a new id, which is incremented from the total numbers of cells at the time.

Two main errors can occur during the lineage tracking. The first one is based on the detection of three candidate cells for the same parent (rather than 2). An example of this error is highlighted by a red ellipse (A) in the lineage plot of Figure 4.14. A second error can occur when a new id is assigned to a cell that is already identified, which breaks that lineage and creates a new cell lineage. This type of error is shown in the lineage plot in Figure 4.14, highlighted with the red arrows (B and C).

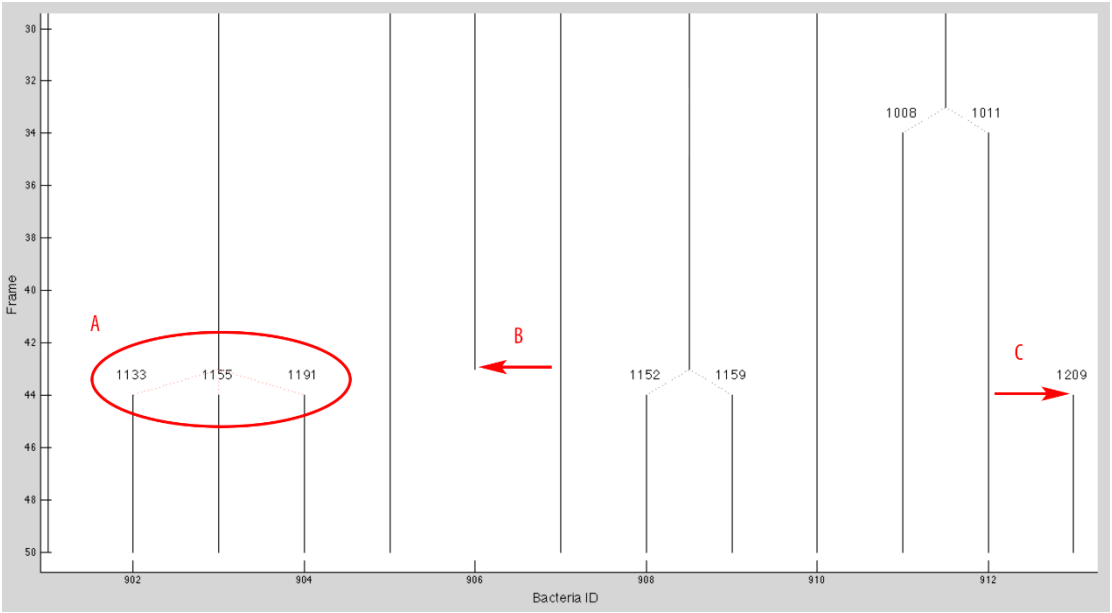


Figure 4.14 - Example of lineage construction errors of the tracking algorithm. (A) one cell dividing into three cells; (B) a cell disappears in frame 43; (C) the same cell reappears in frame 44 with a new id and no parents.

4.1.5. Segmentation of Cellular Components

After the cell segmentation, and the image registration process between morphological and functional images, it is possible to observe several cellular components. As discussed in Section 2.2., several cellular structures are distributed in organized clusters or in localized sections of the cell (e.g. poles) [81]. For these types of structures, the segmentation of 2-D borders can extract important cellular features, necessary for the characterization of the dynamics of bacterial processes. However, some structures are not spatially organized along the cell, can be sparsely localized in the cell's cytoplasm or in some cases its spatial organization is affected by intracellular compartments or by a secondary molecular complex [81]. The main structures of interest used in this research work are the Nucleoids, the FtsZ Rings, inclusion bodies and protein aggregates.

4.1.5.1. Gaussian Segmentation Algorithm

The Gaussian Segmentation Algorithm was first developed specifically for a study, where it was necessary to study Nucleoid properties over different temperatures and correlate those properties with the exclusion of exclusion of protein aggregates from the center of the cell [117]. To study the physical properties of the Nucleoid (size, position from the center and number of Nucleoids), an algorithm was developed that was able to detect and segment the Nucleoids based on applying the GPL algorithm [304] to label each pixel, based on its gradient azimuth, creating a gradient path.

The resulting labels are reduced by tagging them as equivalents, which happens when two labels belong to the same maximum. After this step, the position and number of seeds can be obtained. The seeds are used for a Segmentation Algorithm based on a two-dimensional Gaussian profile of the resulting objects, which was initially specifically tailored for DAPI-stained nucleoids segmentation [117] and is described by equation 4.6.

$$G(x,y) = A \times e^{\left(-\left(a(x-x_0)^2+2b(x-x_0)(y-y_0)+(y-y_0)^2\right)^d\right)} + z_0 \quad (4.6)$$

where

$$a = \frac{\cos^2\theta}{2{\sigma_x}^2} + \frac{\sin^2\theta}{2{\sigma_x}^2} \quad (4.7)$$

$$b = -\frac{\sin 2\theta}{2{\sigma_x}^2} + \frac{\sin 2\theta}{2{\sigma_y}^2} \quad (4.8)$$

$$c = \frac{\sin^2\theta}{2{\sigma_y}^2} + \frac{\cos^2\theta}{2{\sigma_y}^2} \quad (4.9)$$

The nucleoid modelling function allows translation in the three axes (x_0, y_0, z_0), amplitude scaling (A), rotation (θ), width adjustment in x and y planes (σ_x and σ_y) and amplitude profile adjustment (*d*) between a square shape, a bell shape and a thin shape.

Initially for the detection of the DAPI stained cells [117], a value of d a value of (*d*)=10 was empirically defined and the Levenberg-Marquardt Least-Squares [459](Moré, 1978) optimization algorithm was used for the calculation of the other parameters, allowing the use of the z0 value as threshold to obtain the segmented masks of the nucleoids.

The algorithm was not implemented in any toolbox, and there was no possibility of manual corrections of the GPL seeds, and the values of (*d*) had to be changed directly in the MATLAB® code. Due to this situation, it was decided to implement the Gaussian Segmentation Algorithm in SCIP, allowing the user to manually correct the placement of the seeds and the change of several parameters. To apply the Gaussian Segmentation Algorithm to any structure of interest (e.g. Nucleoids, FtsZ Rings, Min System proteins), the user needs to select the option 'Gaussian Fit' on the dropdown menu of the specific Structure (see Figure A.8 for an example in the Nucleoid Detection Box) and click on the button "Structure Detection" (which is specific for each structure).

When the Gaussian Segmentation Method is chosen for the segmentation of a specific structure, a new window appears on the screen (see Figure 4.15). One of the parameters that can be changed in this window is the shape parameter (*d*). The other parameters that can be changed are based on the rejection of Nucleoids, when two seeds are used but the fitting of the Nucleoids overlaps (see radio box in Figure 4.15). The first option, totally rejects overlapped Nucleoids while the second option rejects nucleoids if the overlap is bigger than X% of the total area of that Nucleoids (X can be changed in the edit box in Figure 4.15, inside the Gaussian Fitting Parameters window). The third option doesn't reject any overlapping Nucleoids, while the last option is to consider overlapping Nucleoids as one single Nucleoid.

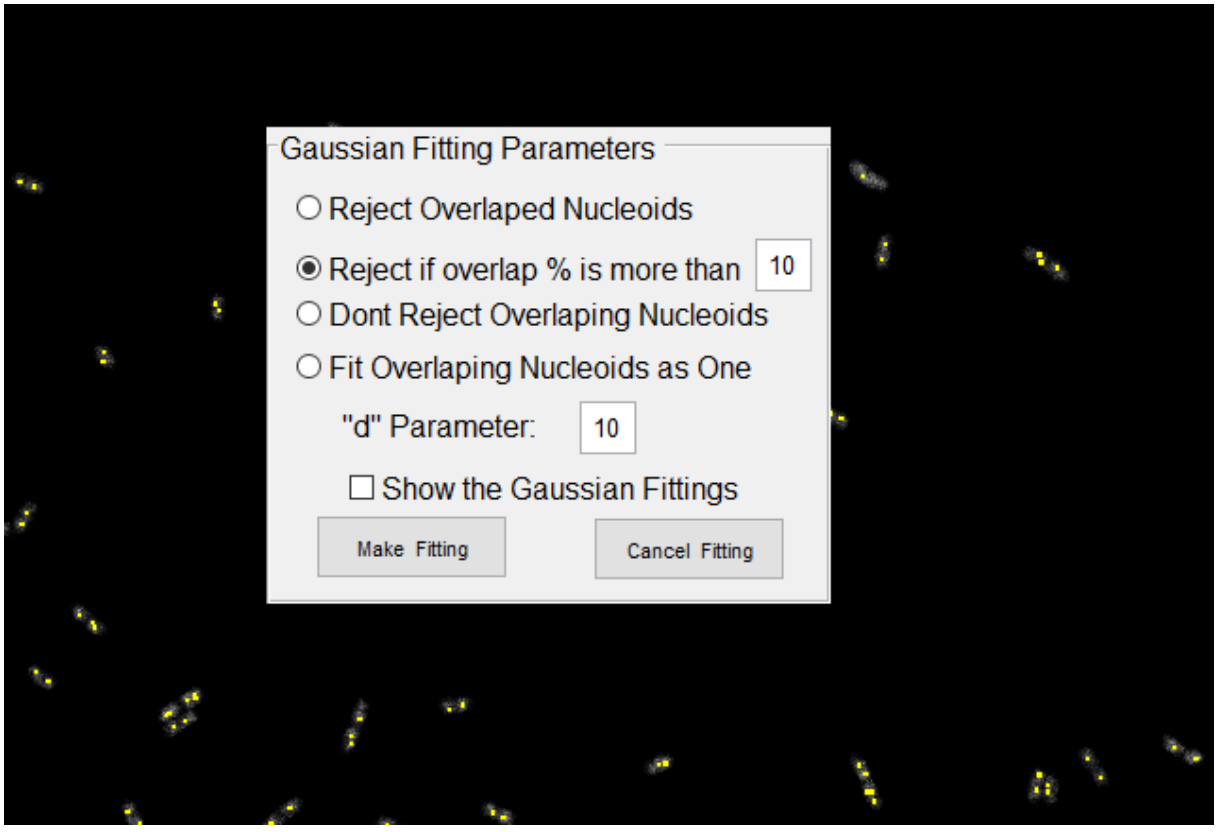


Figure 4.15 – Gaussian Fitting parameters window. This example shows the seed position for the Nucleoid Detection in the yellow dots.

The user can also see the Gaussian Fitting of each cell, by clicking on the checkbox 'Show Gaussian Fittings'. Figure 4.16 shows examples of one and two Nucleoids fitting using the Gaussian Segmentation Algorithm.

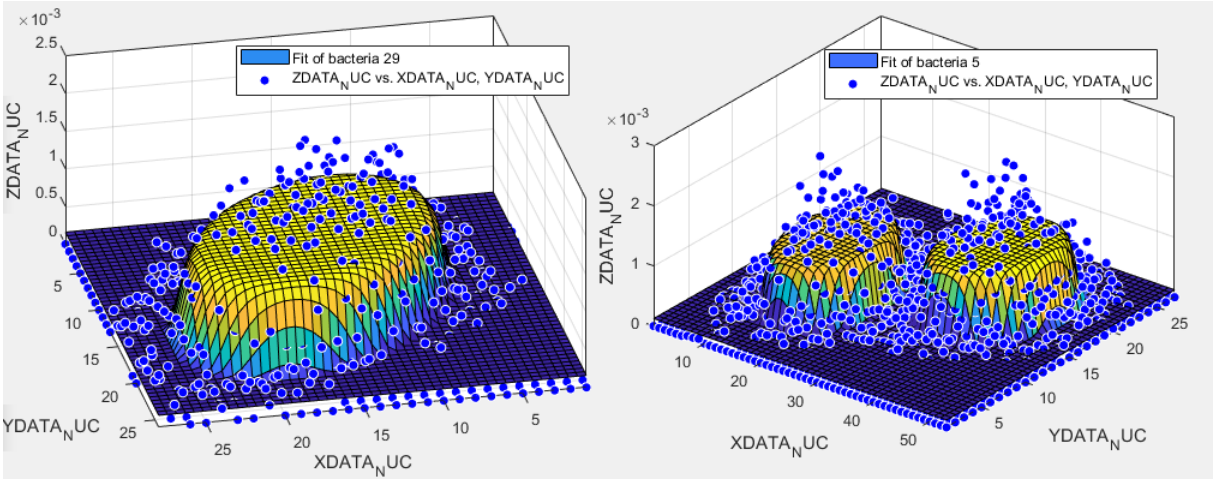


Figure 4.16 – Visualization of the Gaussian Fitting of one (left) and two (right) nucleoids.

The main issues with the implemented Gaussian Segmentation Algorithm are based on the segmentation of structures that can have large morphological changes during the cell's lifetime, such as the FtsZ ring and the Min System protein and the dependency of the algorithm on the correct placement of seeds, which might require manual selection/correction of seeds. Finally, objects like the inclusion bodies, which can also be difficult to segment, due to intensity and contrast of the objects. The segmentation done (with the Gaussian Algorithm) on structures of interest, is shown in Figure 4.17, that shows correct and incorrect segmentation examples.

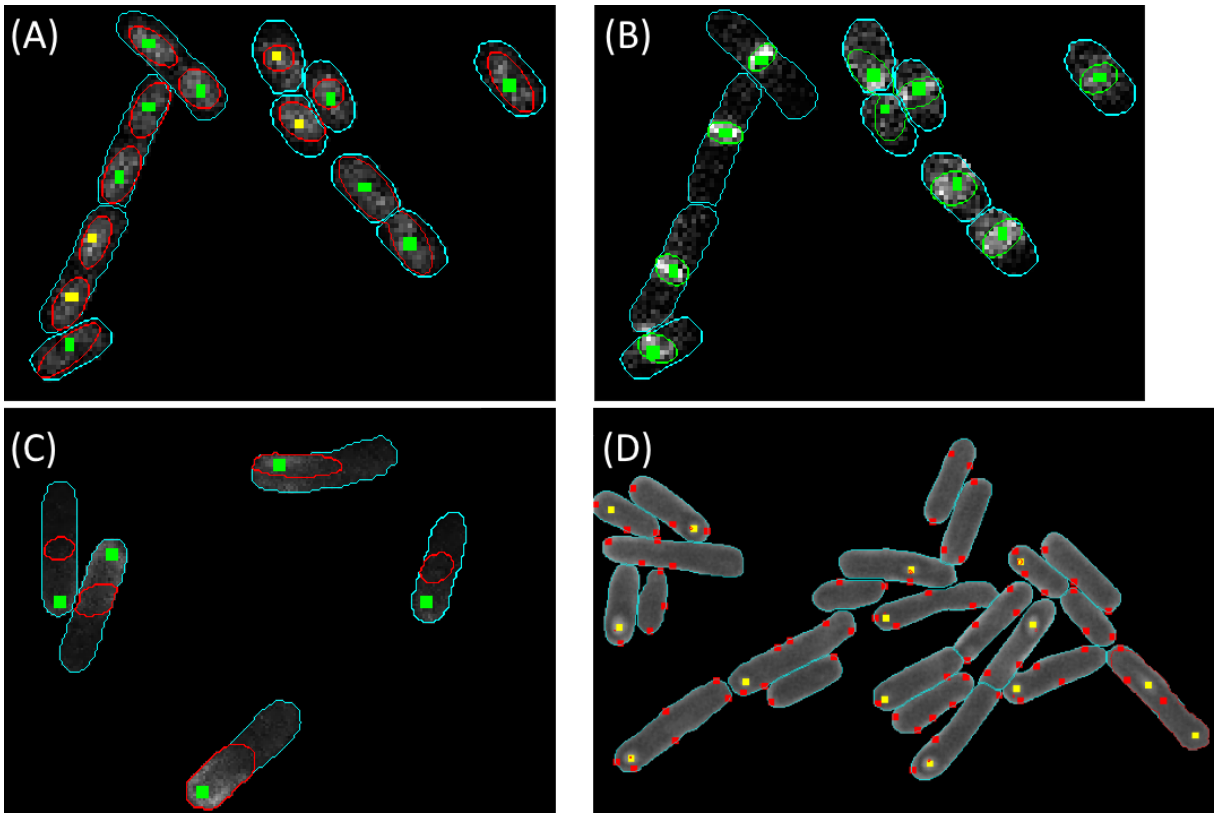


Figure 4.17 – Example of usage of the Gaussian Algorithm. (A) In Nucleoids; (B) in FtsZ Rings; (C) In Min D proteins, (D) in Inclusion Bodies. Yellow squares show automatically placed seeds with the GPL algorithm, green squares are manually corrected seeds and red squares are seeds removed by a seed removal algorithm.

To make a completely automatic analysis of cellular structures (after the parameter selection), a second algorithm was developed with the intention that it doesn't depend on seed selection like the Gaussian Algorithm, although the user will still need to decide the best thresholding method. This

algorithm should be able to adapt better to the morphological changes that occur during the lifetime of a cell and to different intensities and contrast inside the cell.

4.1.5.2. 'TreshMorph' Segmentation Algorithm

The developed Algorithm is based on Thresholding techniques and morphological operations), so it has been named as 'TreshMorph'. The first step in the 'TreshMorph' Segmentation Algorithm (the full workflow is presented in Figure 4.18 and the parameter selection window is presented in Figure 4.19) is to select a threshold level to separate the structures of interests from the background. Three different threshold methods can be selected. The first is based on the Global Otsu's Global image threshold method [258], minimizing the intraclass variance of the black and white pixels (using the '*graythresh*' function). The second is based on the Multilevel image threshold (using the '*multithresh*' function) with several levels. If the Multilevel image threshold is selected, the number of levels can also be defined by the user (second edit box in Figure 4.19) and which level is used to threshold the image (first edit box in Figure 4.19). A threshold based on the mean and standard deviation of the intensity inside the cell can also be selected. In this case different amounts (positive of negative) of the standard deviation to be added to the mean intensity can be chosen.

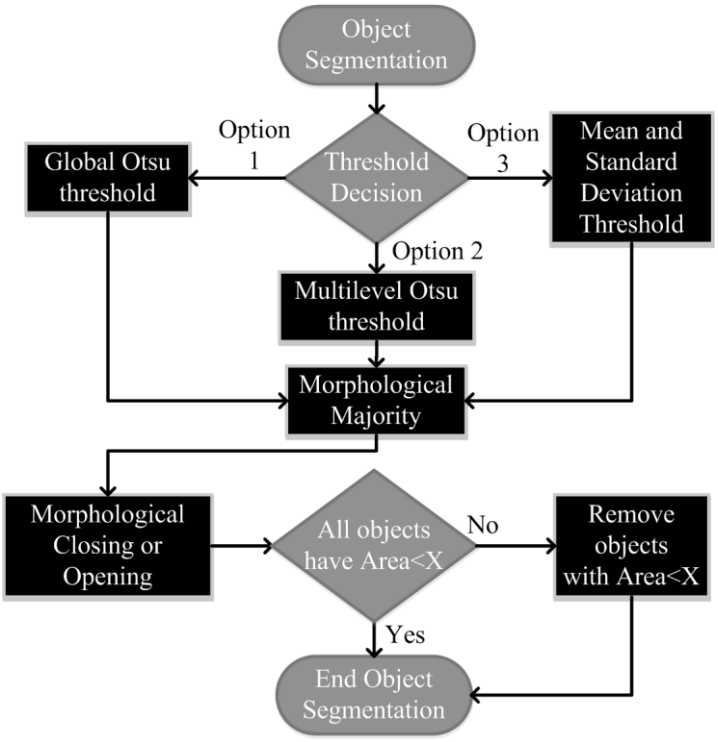


Figure 4.18 – Segmentation workflow of the 'TreshMorph' Segmentation Algorithm.

After the threshold level definition, the '*im2bw*' function is used to obtain the binary images (masks) of the structures based on the specific threshold. This is then followed by the application of the '*bwmorph*' function with the 'majority' operation, which sets a pixel to 1 if five or more pixels in its 3-by-3 neighborhood are 1s and to 0 otherwise.

The next step is a morphological 'closing' operation (by selecting the radio box 'Morphological Close' in Figure 4.19) which performs a dilation followed by an erosion, which tends to enlarge and smooth the boundaries of the structures, while also removing small holes in the mask (but doesn't remove small objects like the 'opening' operation). The user might also use the 'opening' operation,

which can obtain better results for different structures, by selecting the radio box 'Morphological Open' in Figure 4.19.

Finally, all objects with a size smaller than X pixels can be removed from the analysis (X can be changed in the edit box 'Delete objects smaller than X pixels" in Figure 4.19), using the '*bwareaopen*' function, unless this step removes all the objects inside the cell (if it does, then this step is skipped), which then finalizes the automatic object segmentation.

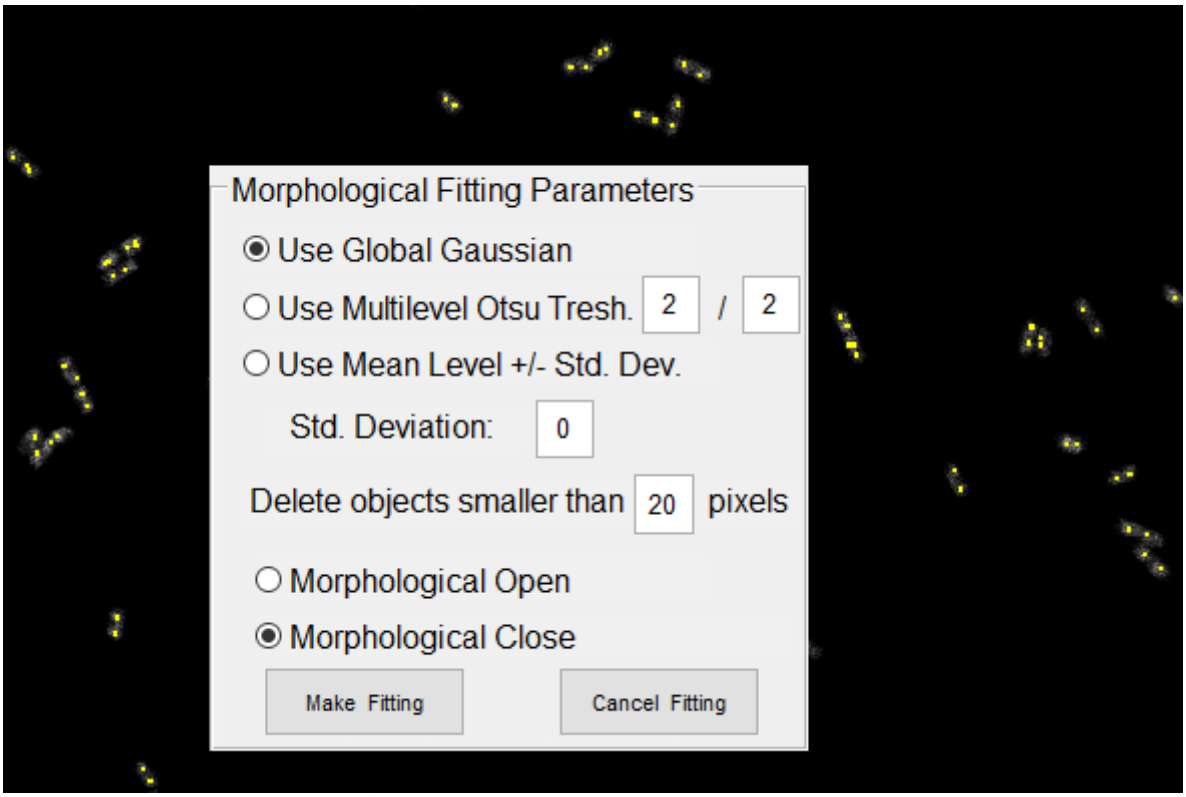


Figure 4.19 - Activation of the Morphological Fitting parameters window for a Nucleoid Detection Example.

The segmentation done (with the Gaussian Algorithm) on structures of interest, is shown in, that shows correct and incorrect segmentation examples with different thresholds. For Nucleoids (see Figure 4.20-A), the Otsu's Global threshold is chosen. For FtsZ rings (see Figure 4.20-B), the multilevel (two levels) Otsu's threshold is chosen.

For MinD proteins (see Figure 4.20-C), a global threshold value based on the mean intensity of each cell is chosen. For the inclusion bodies. For the inclusion bodies (see Figure 4.20-D), the multilevel (three levels) Otsu's threshold is chosen.

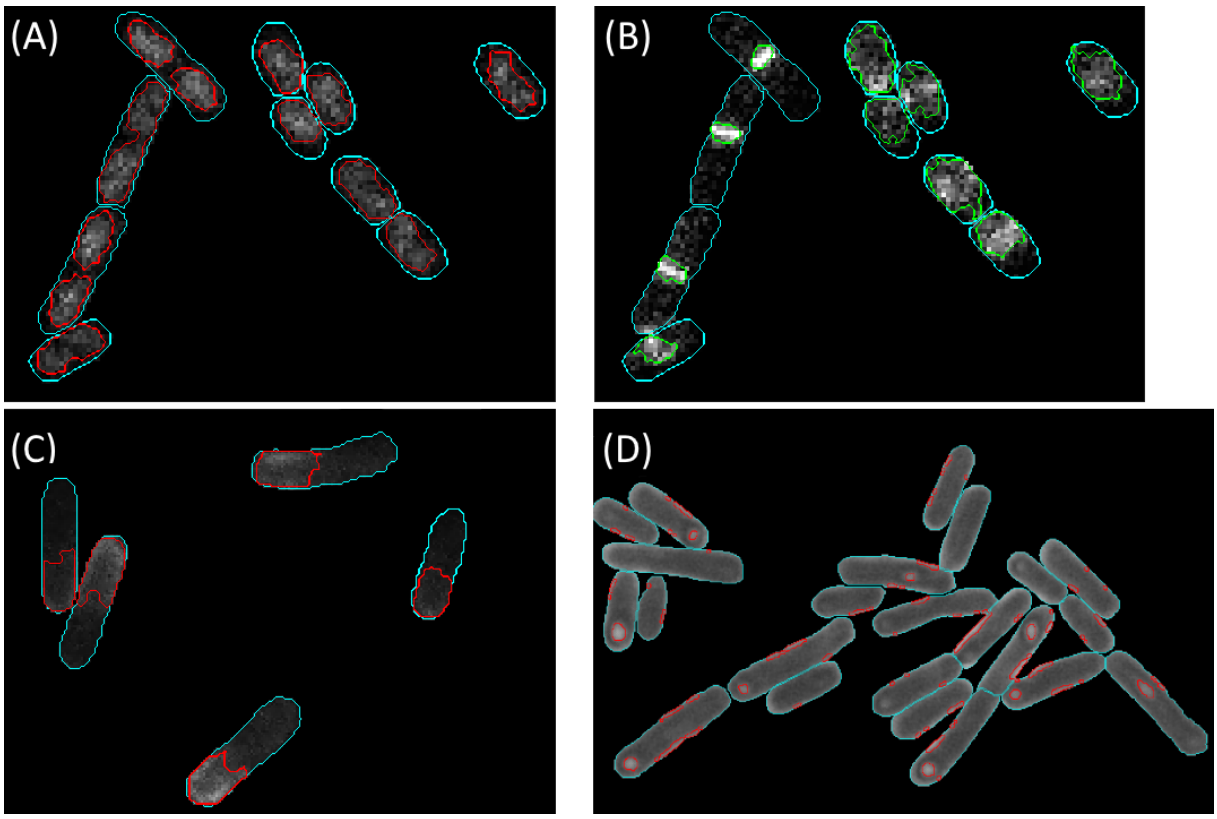


Figure 4.20 – Example of usage of the 'TreshMorph' Algorithm. Segmentation of (A) Nucleoids; (B) FtsZ Rings; (C) Min D proteins and (D) Inclusion Bodies.

4.1.6. Detection of Inclusion Bodies and Manual Seed Correction

For the segmentation of Inclusion Bodies, which can be directly detected in segmented Phase-Contrast images, either the Gaussian Segmentation, using the seeds to start the fitting (Figure 4.17-D) or the 'TreshMorph' Segmentation, with multilevel Otsu's threshold (Figure 4.20-D) didn't provide a satisfactory segmentation, as in the first one the segmentation is not able to be expanded from the seeds, while in the second there is a large presence of false positive segments.

To make a correct detection of the inclusion bodies (which are normally associated with bright spherical objects that can even be observed in morphological images), the first thing that needs to be fixed is the seed placement. As seen in Figure 4.17-D, there are numerous red squares, which correspond to GPL seeds that are false positives (do not correspond to inclusion bodies) and need to be removed. Most of these false positive detections are caused by the high intensity of the aura outside cells in Phase-Contrast images.

To remove false positive seed detections three methods are used. The first is based on calculating the Euclidean distance between the seed center and the cell border. If the closest pixel from the cell border is distanced less than 5 pixels from the seed center, that seed is removed (transforming yellow squares into red squares, as seen with the blue arrow in Figure 4.21-A). The second method removes seeds with the GPL variable 'path minimum amplitude' at zero. This value indicates that the seed is linked to the background independently of its position in the cell (as the background has an intensity value of 0, since it is cut from the image based on the segmentation, as seen with the green arrow in Figure 4.21-A). The final method is based on subtracting the GPL 'path

maximum amplitude' with the GPL 'path minimum amplitude' of the seed. If the subtraction is lower than a pre-defined threshold (for our examples, a value of 20 was found to be adequate), the seed is removed, as seen with the white arrow in Figure 4.21-A.

Based on the seed rejection algorithm, the next step to improve the inclusion body segmentation was to combine the 'TreshMorph' algorithm (which didn't use the GPL seeds) and the seed removal methods to create the cells that also have an accepted seed inside the segment. An example of this combination, using a multilevel threshold with two levels is shown in Figure 4.21-B. This combination can miss the segmentation of inclusion bodies, as the segmentation is not expanded from the seeds (see white arrow Figure 4.21-B).

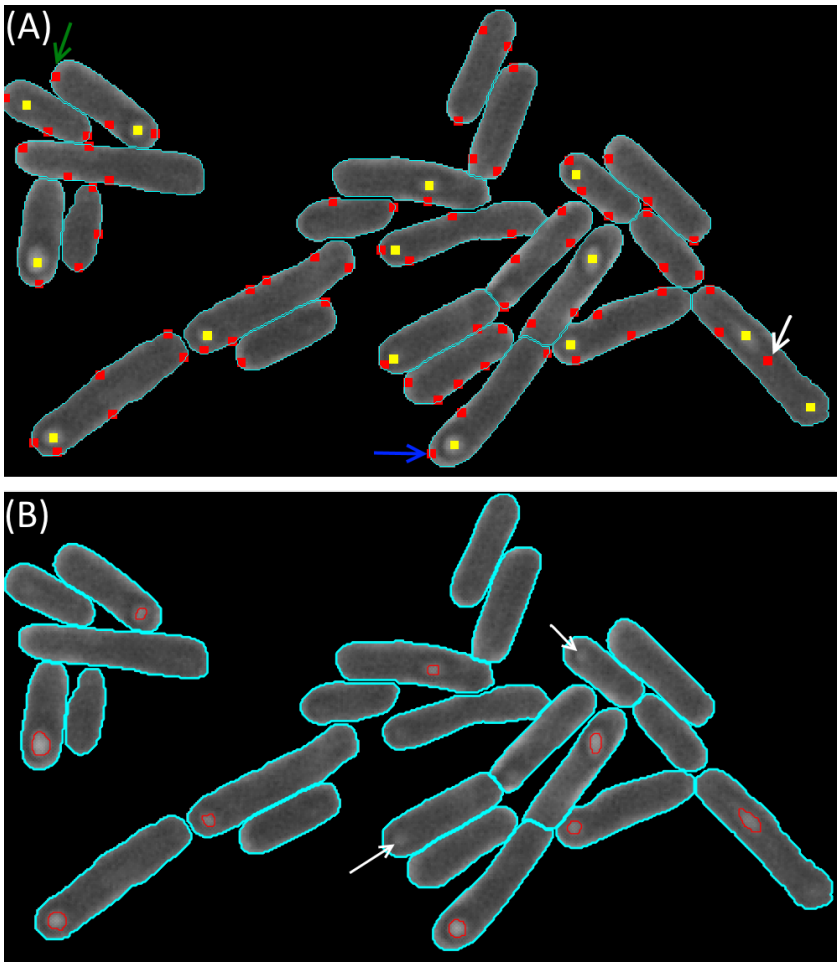


Figure 4.21 – Automatic seed correction and inclusion bodies segmentation. (A) shows in red squares, examples of rejected seeds. Blue arrow is an example of the first rejection method, green arrow an example of the second rejection method and white arrow an example of the third rejection. (B) Segmentation of inclusion bodies based on the combining the 'TreshMorph' Algorithm with a multilevel threshold and the seed rejection methods. White arrows represent inclusion bodies not segmented by the 'TreshMorph' Algorithm but accepted as a correct seed.

One solution to this issue can be the usage of different levels of threshold until an area is fitted. However, this solution might require too many iterations. Another solution is to try other algorithms which can use the seeds as the initial fitting (which the 'TreshMorph' doesn't) but are able to expand the fitting towards the borders of the inclusion bodies (which the Gaussian doesn't). Good candidates to test the fitting are based on Region-Growing algorithms [309], [310] or the Region-based Active Contour model [311], which should be able to adjust to local and global intensity features of the inclusion bodies.

As aforementioned, the removed seeds are shown with a red colour, while non- removed seeds are shown in yellow (based on the seed removal methods). New seeds can then be manually added, and yellow seeds can be removed from the analysis (in all image modalities) by just clicking on the pixel where the seed needs to be removed or added. When a pixel is selected, the program checks all its 8-connected neighbour pixels (pixels that touch one of its edges or corners). If no neighbouring pixel is already a seed (yellow or red) a new seed is created on that pixel. If any of the neighbours is a seed, that seed is deleted. Examples of red, yellow and green seeds are shown in Figure 4.22.

Seed removal can also be done by drawing an interactive a freehand region of interest (ROI), by clicking the 'd' key. After the ROI is finished, any seed (yellow or green) inside it is deleted. All seeds are deleted if 'alt'+'d' is selected.

In addition to the existing seeds (red, yellow and green), the user can add markers inside a cell, which can be used to mark interesting features inside the cell. Two distinct types of markers (see Figure 4.22) can be added by clicking on 'o' (to add orange marks) and 'p' (to add pink marks) and clicking inside the cell (similarly to the seed correction mechanism). The program saves the coordinates of each marker inside the structure of each cell. This marking mechanism was created to allow the user to signal and track over time any structures of interest inside any cell.

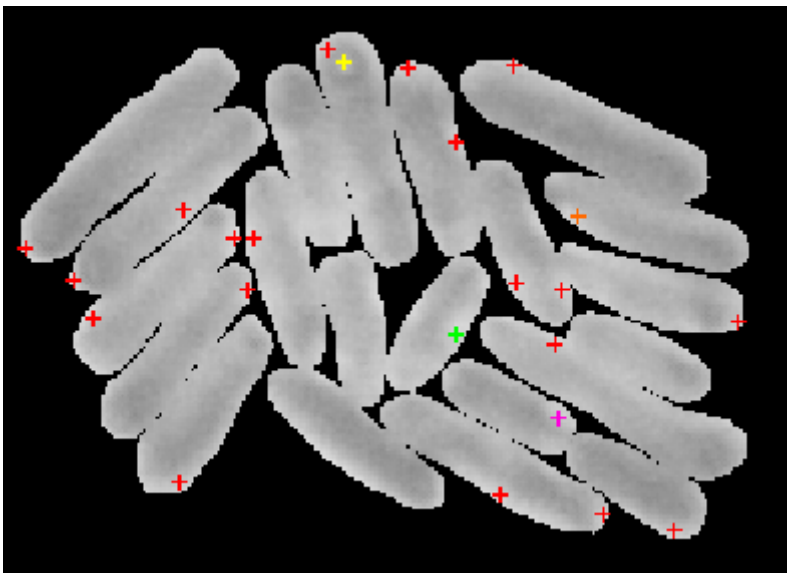


Figure 4.22 - Manual Seed Correction. Seeds automatically selected for deletion are shown in red. Yellow seeds were not automatically selected for deletion. Green seeds were manually added. Orange and pink marks are also shown.

4.1.7. Protein Aggregates (Spot) Detection

The spot detection methods implemented in the SCIP tool, were based on the methods developed in 'CellAging' [458] and 'iCellFusion' [453]. The SCIP tool integrates all existing methods, which differ in the filters that can be applied (Median, Kernel and Gaussian). An example of MS2-RNA-GFP spot detection using the SCIP tool with a median filter is presented in Figure 4.23.

The parameters for each filter, as described in 'CellAging' [458] and 'iCellFusion' [453] are based on the intensity values inside the cell, to select the threshold values to detect fluorescent spots, as seen in options panels in Figure A.9. For each method, the spots that exceed the maximum accepted area and spots smaller than the minimum accepted area are removed.

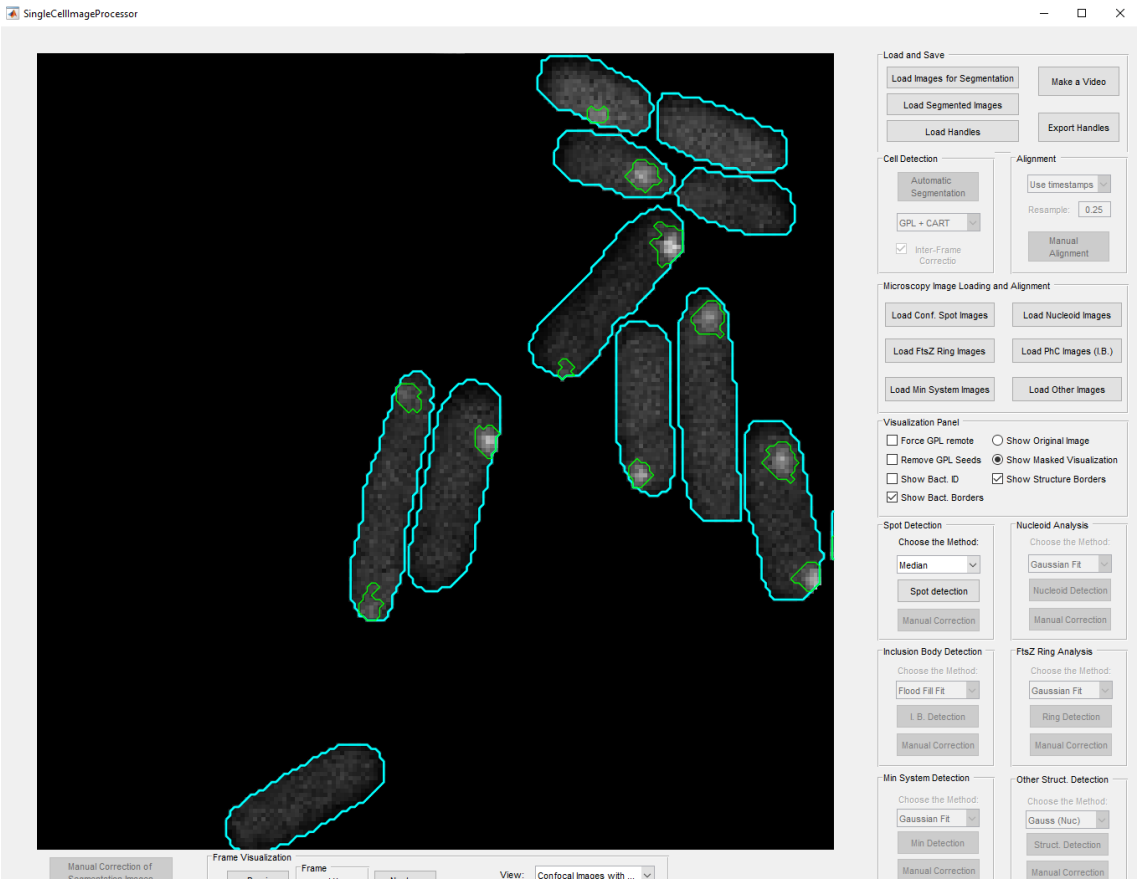


Figure 4.23 - Examples of segmentation of mRNA spots using the median Algorithm.

The correct detection of protein aggregates (or spots) has been an important step in the study of the study of the spatial organization in bacteria or even studies of gene expression. These protein aggregates are formed with fluorescent proteins (e.g. green fluorescent protein, yellow fluorescent protein) merged with the structures of interest (e.g. RNA molecules) to form structures such as MS2-RNA-GFP complexes and IbpA-YFP complexes [6], [117] or fluorescent Tsr-Venus clusters [184], which can all be detected with the implemented algorithms.

4.1.8. Singe-cell and population-level colocalization

The segmentation of different internal cellular components might not provide relevant information as some structures are not spatially organized along the cell can be sparsely localized in the cell's cytoplasm or in some cases it's spatial organization is affected by intracellular compartments or by a secondary molecular complex [81]. In such studies that are interested in how the localization of one structures affects the localization of the another (so called "co-localization" studies), it is necessary to provide quantitative tools, such as correlation coefficients [460]. These correlation coefficients are used to study cellular functions of proteins and other molecules. In many cases, the function of a molecule can be inferred from its association with specific intracellular compartments or other molecules.

In the SCIP toolbox, the calculation of two of the most popular and useful correlation coefficients are provided: the single-cell Pearson Correlation Coefficient (PCC), which can be calculated for the entire cell and also along the Major and the Minor Axis and the Manders Coefficients (M1 and M2) of the first versus the second channel [197].

The PCC can be calculated using the '*corrcoef*' function, which is defined as, in this case for N pixels in image A and B:

$$\rho(A,B) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{A_i - \mu_A}{\sigma_A}\right)\left(\frac{B_i - \mu_B}{\sigma_B}\right) \quad \textbf{(4.10)}$$

where, μx and σx are the mean and standard deviation of the x images (A and B).

To calculate the PCC along the Major and Minor Axis, it is required to normalize the lengths and the center coordinates of each cell. This problem was approach by using the Principal Component Analysis (PCA) Algorithm [461], using the '*pca*' function to calculate the principal component coefficients and scores. The initial step of the PCA algorithm is the calculation of the cell centre ($P_{centroid}$), calculating the mean values of the x and y coordinates, $\overline{x}$ and $\overline{y}$ respectively. It is possible to obtain a zero-mean pixel list matrix by subtracting the $\overline{x}$ and $\overline{y}$ values. With this matrix, it possible to compute the covariance matrix (C) along each dimension, as determined by equation (4.11):

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{pmatrix} \quad \textbf{(4.11)}$$

where x and y are the coordinates of $P_{centroid}$, and each covariance value is determined in (4.12):

$$cov(X,Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \mu_X)\,(Y_i - \mu_Y) \quad \textbf{(4.12)}$$

The covariance matrix is used to determine the two eigenvectors ($v_{\lambda_1}$and v_{λ_2}) and the corresponding eigenvalues ($\lambda_1$and λ_2, with $\lambda_1 > \lambda_2$), using singular value decomposition and QR decomposition [462]. Since $\lambda_1 > \lambda_2$, the direction of the major axis will be represented by the eigenvector v_{λ_1}, while the minor axis will be represented by the eigenvector v_{λ_2}(which is perpendicular to v_{λ_1}). The score matrix (S), can be defined in the new system of coordinates using the equation (4.13):

$$S = P_{centroid}[v_{\lambda_1} \quad v_{\lambda_2}] = [S_{\lambda_1} \quad S_{\lambda_2}] \quad \textbf{(4.13)}$$

This score Matrix S is used to calculate the major and minor axis length (MaxL and MinL respectively) and respective midpoint of each direction (D_{λ_i}), which is used to calculate the re-centered score matrix ($S_{re\text{-}centered}$), with the new coordinates for each segmented cell:

$$\begin{cases} MaxL = \max(S_{\lambda_1}) - \min(S_{\lambda_1}) \\ MinL = \max(S_{\lambda_2}) - \min(S_{\lambda_2}) \\ D_{\lambda_i} = \left(\min(S_{\lambda_i}) + \max(S_{\lambda_i})\right)/2,\ with\ i = 1{,}2 \\ S_{re-centered} = \left[(D_{\lambda_1} - S_{\lambda_1})\right](D_{\lambda_2} - S_{\lambda_2}) \end{cases} \quad \textbf{(4.14)}$$

Using this system, the pixel coordinates are normalized along each direction (Major and Minor axis), and it is possible to obtain an intensity profile by summing all the pixels that have the same coordinates along that direction.

RNAp-GFP molecules are shown in the green channel and HupA-mCherry-tagged nucleoids in the Red channel an example. This normalization is done by doing the PCA normalization for each cell and then dividing the score matrix into 10 equal bins and summing. The intensity profile is also normalized by summing the intensity along each bin and divided by the total intensity inside each cell.

In the SCIP toolbox, the calculation of Manders overlap coefficient (r) and Coefficients (M1 and M2) of the first versus the second channel are based on the following equations (4.15, 4.16 and 4.17) [197]:

$$r = \frac{\sum_{i=1}^{N} S1_i S2_i}{\sqrt{\sum_{i=1}^{N} [\![(S1)]\!]_i)^2 \sum_{i=1}^{N} [\![(S2)]\!]_i)^2}} \quad \textbf{(4.15)}$$

$$M_1 = \frac{\sum_{i=1}^{N} S1_{i,coloc}}{\sum_{i=1}^{N} S1_i}, with\ S1_{i,coloc} = S1_i\ if\ S2_i > 0 \quad \textbf{(4.16)}$$

$$M_2 = \frac{\sum_{i=1}^{N} S2_{i,coloc}}{\sum_{i=1}^{N} S2_i}, with\ S2_{i,coloc} = S2_i\ if\ S1_i > 0 \quad \textbf{(4.17)}$$

For the Manders Coefficients calculation, pixel intensities are normalized by subtracting the mean intensity inside the cell. Based on the fluorescent intensities of each channel, the pixel values inside one channel versus the corresponding values inside the other channel can also be provided. Both the PCC and the Manders coefficients are independent of the image brightness, but might be sensitive to noise in the images, so sometimes they might require a pre-filtering process or a background correction process [463].

If three channels are used to study three differently labelled probes (for example in a RGB configuration: Red, Blue and Green), the approach is to make a one-vs-one comparison and to make three different studies (Red vs Green, Green vs Blue and Blue vs Red) calculating the co-localization coefficients for each study.

4.1.9. Structure detection of three simultaneous channels

As mentioned in the previous Sub-Section, the SCIP tool allows the visualization of three simultaneous channels. The visualization of MS2-GFP-RNA spots in the green channel, Nucleoid in the Blue channel and FtsZ Rings in the red channel, with the bacterial strain and growth conditions specified in Section 5.1 and is shown in Figure 4.24 in a single channel configuration, containing each structure of interest and the corresponding segmentations. Figure 4.25 presents the three possible combinations of simultaneous visualization of two channels of each structure of interest, with (A) and without (B) segmentation.

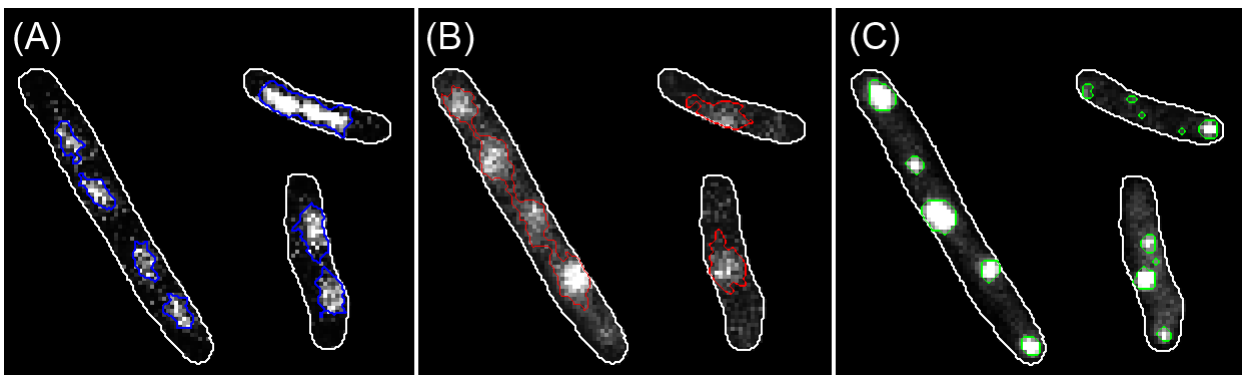


Figure 4.24 - Examples of visualization of a single channel of (A) Nucleoids (segmented in blue colour), (B) FtsZ Rings (segmented in red colour) and (C) MS2-GFP spots (segmented in green colour).

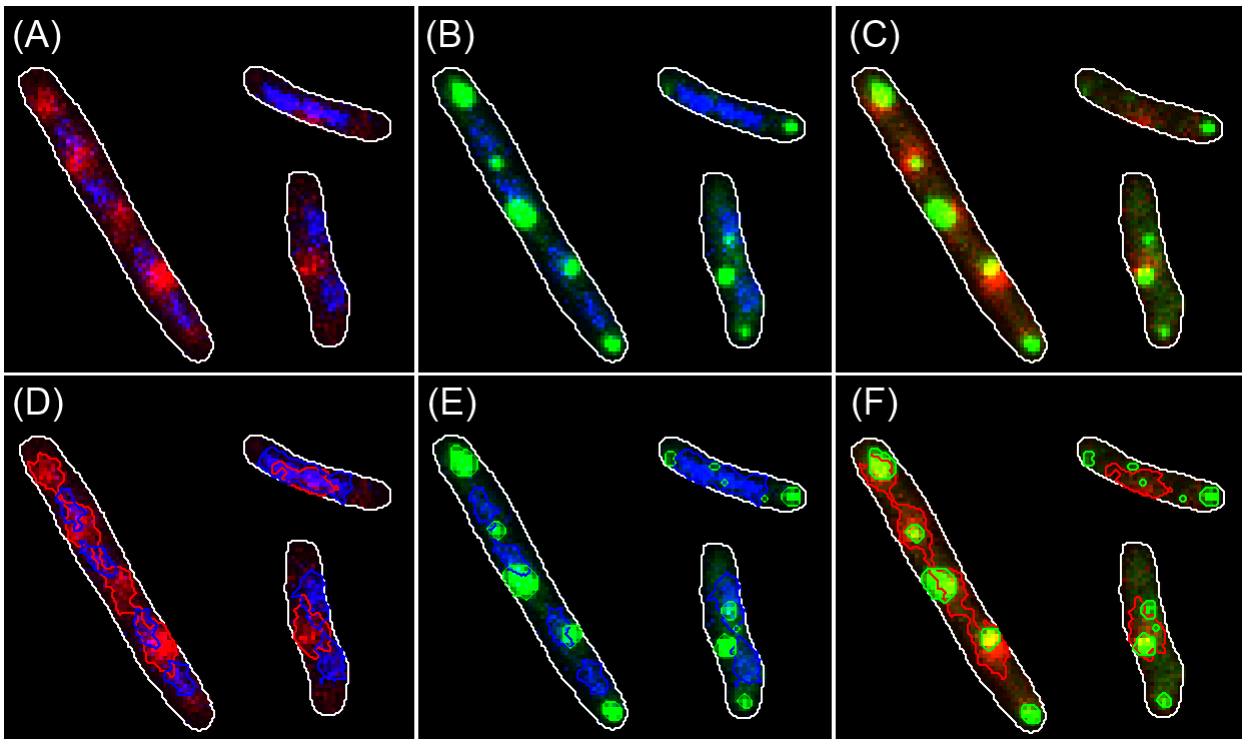


Figure 4.25 - Examples of visualization simultaneous visualization of two channels with (top) and without (bottom) segmentation: (A) both Nucleoids (in blue) and FtsZ Rings (in red), (B) Both Nucleoid (blue) and MS2-GFP spots (in green). (C) both Nucleoids (in blue) and FtsZ Rings (in red), (C) both Nucleoids (segmented in blue) and FtsZ Rings (segmented in red), (E) Both Nucleoid (segmented in blue) and MS2-GFP spots (segmented in green colour) (F) FtsZ Rings (segmented in red colour) and MS2-GFP spots (segmented in green colour).

The visualization of all structures of interest is presented in Figure 4.26 with (A) and without (B) segmentation. This example is presented to show the software's ability to handle 3 different fluorescent proteins taken at the same time in different channels. In this example, the segmentation uses the 'TreshMorph' Algorithm, with a Threshold selection based on the mean fluorescence intensity and the Multilevel Otsu, respectively for the detection of Nucleoids and FtsZ Rings. The segmentation of MS2-GFP-RNA spots, uses the median filter

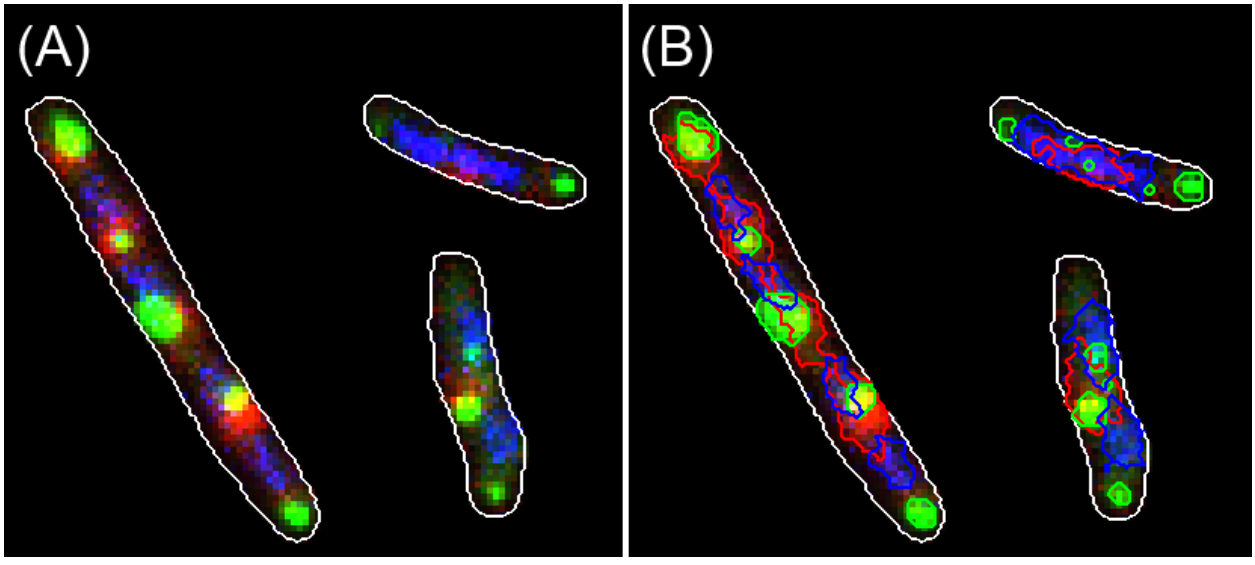


Figure 4.26 - Example of visualization simultaneous visualization of three channels: (A) Nucleoids (in blue), FtsZ Rings (in red) and MS2-GFP spots (in green) with no segmentation and (B) Nucleoids (segmented in blue colour), FtsZ Rings (segmented in red colour), and MS2-GFP spots (segmented in green colour).

4.2. Contribution for the Simulation Framework

The contribution of this research work for the development of an image simulation framework is divided into several steps, with two toolboxes published during this research work. The first one, named 'miSimBa' (Microscopy Image Simulator of Bacterial Cells), simulated images that reproduced the spatial and temporal organization of *E. coli* cells [464] by modelling realistically cell morphology (shape, size and spatial arrangement), cell growth and division, cell motility. The second developed platform allowed a generic representation of bacterial cells and was used to validate cell tracking algorithms. The second platform, which will be named 'Image Tracking Generator' was mainly developed by Pedro Canelas during his Master Thesis [465], [466], while for this research work, the toolbox was then tested extensively [467]. This next sub-section focuses on the implementation of the image generator and its basic features.

4.2.1. Graphic User Interfaces

The first image simulator ('miSimBa') interface were implemented using MATLAB, integrating its Object-Oriented Programming capabilities [464]. This toolbox allows the user to select several inputs such as the number of objects (randomly generated between the chosen minimum and maximum number), the desired width and height of images, temperature of the simulation (this changes the statistical distribution of the Major and Minor Axis of the cell, the movement and the cell division rate, as reported in Section 2.1), the simulation time and the frame rate (resulting in a fixed number of simulated frames). The tool graphical user interface is presented in Figure 4.27.

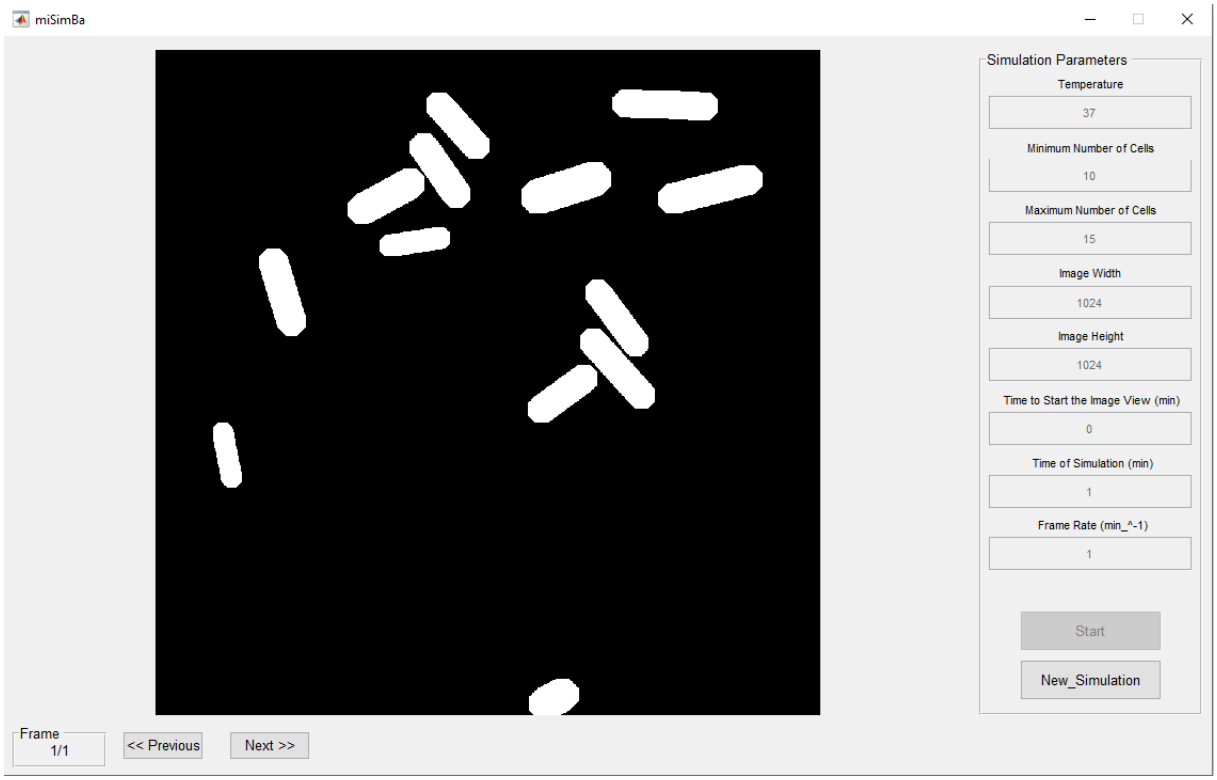


Figure 4.27 – Graphical interface of the 'miSimBa' Toolbox and a simulation example.

The ‘miSimBa’ image generator automatically creates a .mat file containing the images of each frame, and the properties of each cell object, similarly to main output of the image processing toolbox described in the previous section. The properties of each cell object are the following: ‘Cell ID’, ‘Division Flag’, ‘Parent ID’, ‘Centre’, ‘Orientation’, ‘Major Axis Size’, ‘Minor Axis Size’, ‘Contour’, ‘Pixel List’. The ‘Cell ID’ is a unique number that identifies each cell. When a division occurs, the ‘Division Flag’ of that object is turned from 0 to 1 and each Daughter cells receive a new ‘Cell ID’, and the ‘Cell ID’ of the parent is recorded into both Daughter cells. The ‘Centre’ position is recorded in x and y pixel coordinates, along with the ‘Contour’ and the ‘Pixel List’, which are the map of all the pixels that the cell border and the rest of the cell occupies respectively, the Major and Minor Axis are recorded in pixel units, the ‘Orientation’ is defined as the angle between Major Axis of the object and the X Axis (in radians).

The second image simulator (‘Image Tracking Generator’) interface and the tracking methods were implemented using the C# language from Visual Studio 2015. The time-series generator allows the user to change several settings such as the number of objects, frames, clusters, and their features. The tool interface is shown in Figure 4.28.

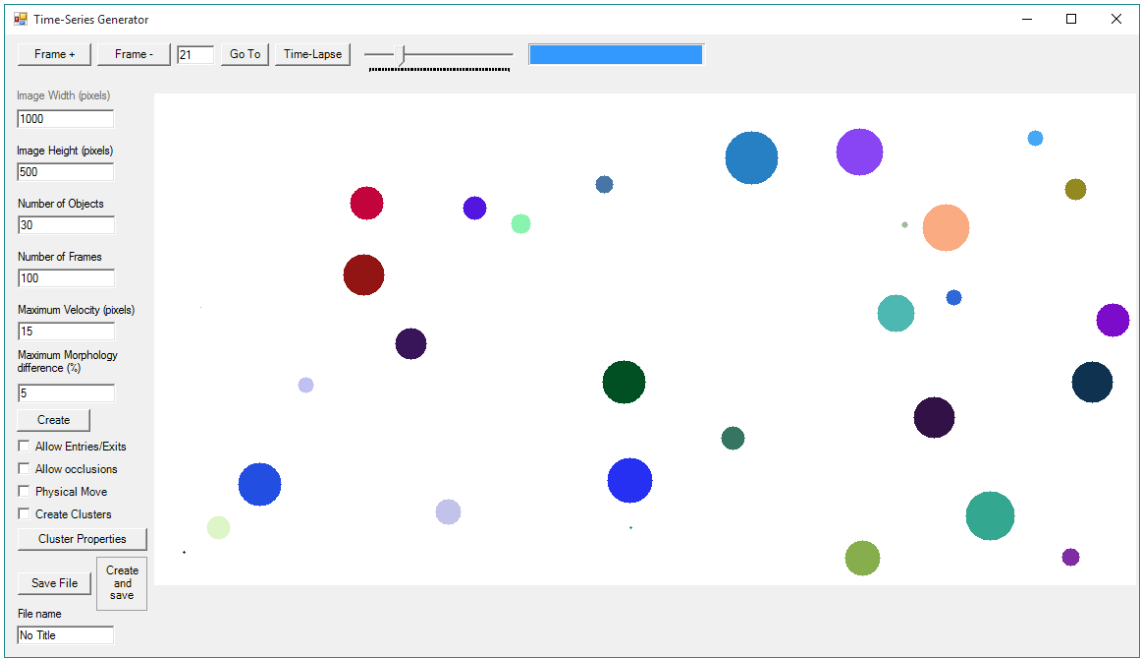


Figure 4.28 - Graphical interface of the ‘Image Tracking Generator’ toolbox and a simulation example.

The generator automatically creates a .csv file containing the object’s the shape-related factor called ‘Morphology’, which is a rational number between 0 and 1 (as defined in the sub section 4.2.2.1). Other properties were added similarly to the ‘miSimBa’ toolbox: ‘Cell ID’, ‘Division Flag’, ‘Parent ID’, ‘Centre’, ‘Orientation’, ‘Major Axis Size’, ‘Minor Axis Size’. The objects lack the ‘Contour’ and ‘Pixel List’ properties (the pixel maps can be efficiently saved in the .mat files, but no in .csv files).

Object shape will be detailed in sub section 4.2.2.1. At the top row of the window (see Figure 4.28) there are frame handlers, to advance forward and backward in the time-series, or to go directly to a specific frame. The ‘Time-Lapse’ button reproduces the full time-series with a framerate of 25 frames/second.

The left bar (see Figure 4.28) contains the boxes to write the desired inputs, such as the width and height of images, in pixels. The user can also choose the number of objects in each frame, and the

total number of frames. The 'Maximum Velocity' is the maximum distance, in pixels, that an object can travel between frames, while the 'Maximum Morphology Difference' is the maximum difference of the 'Morphology' factor that an object can have between frames, in percentage.

The 'Physical Move' button controls the option of giving objects physical limitations to their kinetics. If it is selected, each object has a velocity and orientation assigned to it, meaning that its position dynamics will depend on these two variables. If it is not selected, objects will move arbitrarily between frames. It is also possible to select 'Allow Entries/Exits', which allows the objects to enter and exit the image limits. If unselected, objects collide and are reflected by the edges of the image when reaching them. When the option 'Allow Occlusions' is selected, objects move without restrictions and can overlap. If it is not selected, objects collide between them similarly as when colliding with the edges. Object growth and movement will be described in detail in sub section 4.2.2.2 and 4.2.2.3, respectively.

The 'Create Clusters' option can be used to create forced object clustering with all objects having have the same physical features. In this setting, 'Physical Move' is automatically selected and 'Allow Occlusions' is deselected, blocking the correspondent checkboxes. Object Clustering will be described in detail in sub section 4.2.2.4.

4.2.2. Object Modelling

This sub section focuses on the modelled features, namely object shape, movement, growth, division and clustering, which were improved from the previous toolbox towards a realistic simulation of the bacterial cell spatial and temporal organization.

4.2.2.1. Object Shape

To create a realistic simulation of bacterial cells, it is necessary to study how they are classified by their shape. Bacterial cells can have a spherical shape (coccus) a rod-shape (bacillus), while other bacteria have shown a vast diversity of shapes, such intermediate shapes (coccobacillus) or curved/corkscrew shapes (spirochete, spirillum and vibrio), or even square and star shapes, each of them with its specific purpose [16], [18] Bacteria can also have a wide range of cell sizes (volumes that range from 0.02 to 400 µm3), where even a vast variability can be observed within the same species [19], [20]. These variations can be explained due to cell adaptation to external factors, such as lack of nutrients leading to starvation, situations of extreme temperatures (low and high) or of extreme dryness [20] (see Section 2.2 for an overview of Bacterial morphology). To create such shapes, it is required to create mathematical representations of these shapes, as observed in Figure 4.29.

Figure 4.29 - Examples of models of bacterial cell shapes. Spherical shape (coccus) in dark grey, a rod-shape (bacillus) in orange, intermediate shape (coccobacillus) in green and curved shapes (spirochete, spirillum and vibrio) in blue.

The first approach, in the 'miSimBa' toolbox [464], to create such bacterial shapes was by defining the mathematical model of the rod shape of *E. coli* cells, which was done by creating a rectangle (black line in Figure 4.30) with the length of the major axis (horizontal green line in Figure

4.30) and the height of the minor axis (vertical green line in Figure 4.30) and taking the convex hull of two equal semi-circles with the radius of half of the minor axis and placing their centres at the major axis line, by a distance of half of the minor axis from the border (see Figure 4.30).

Figure 4.30 – Mathematical modelling of the rod shape of *E. coli* cells (red colour). Minor and Major Axis in Green. The semi circles have a radius defined as half of the minor axis.

The initial approach of the second image simulator ('Image Tracking Generator') [465] was to create simpler round-shaped objects, similarly to spherical bacteria (coccus) using just a cell radius, which was converted into a morphology factor, which determined the maximum radius of the objects (corresponding to morphology value 1, which by default represented 30 pixels). All the results reported in this research work of cell tracking studies (see section 6.2) were done with using round-shape.

A new approach was implemented to use the parameters from both toolboxes and the mathematical model shown in Figure 4.30, to change the shape of the cell towards more realistically bacterial shapes. Cells with equal '*MajorAxisSize'* and '*MinorAxisSize'* will have coccus shape [467]. The gradual increase of the '*MajorAxisSize'*, will lead to the modelling of intermediate shapes (coccobacillus) and the large increase of the '*MajorAxisSize*' parameter will return bacillus shapes [467].

A theorical parameter 'Curvature' was proposed to recreate curved shaped cells, to curve the green line of the mathematical shape (see Figure 4.30) [467]. A theorical value of 0 would simply recreate the straight rod shape cells, while a value of 1 would join both end of the Major Axis, and an intermediate value could recreate the curved shapes of some bacterial cells. The 'Curvature' parameter was not implemented in the published version of the toolbox.

4.2.2.2. Object Growth and Division

Bacterial cell cycle is normally divided in three stages, specifically a period between its "birth" and the initiation of DNA replication, a replication period when the cell increases its mass and size (Cell Growth) and, finally, a binary fission process into two new daughter cells (Cell Division), which is repeated over the next generations [30], as detailed in Section 2.2.1.

When dealing with an image, the spatial modelling of cell growth (the increase in mass and size) must be done by adding new pixel to the existing border pixel (see Figure 4.31-A), which are mapped with the 'Contour' property. For *E. coli* cells the added pixels are create along the major axis and this has to be done by forming new pixels in the middle of the cell, and pushing the existing pixel outwards of the centre of the cell, recreating the creation of new murein polymer (as described in Section 2.2.1). When bacteria are organized in clusters, the cell growth can be halted due to the cells not having space to grow, or they require the pushing of other touching cell or even the bending and growing in different directions. The temporal modelling of cell growth can be defined as a stochastic temporal process until it reaches a division event (when it normally reaches the doubling of its initial size). The kinetic constant of *E. coli* cell doubling time has been reported to be around 3600s in favourable conditions [72], [203].

When a cell division event is flagged, the parent object needs to be 'cut' in two daughter cells, as observed in Figure 4.31-B, by gradually deleting the pixels of the middle section, using the semi circles with a size of half of the minor axis, until the two daughter cell are only touched by their tips, and the cell movement will finally separate both touching cells.

The distribution of bacterial cell sizes can be obtained from the literature review of experimental studies. It should be noted that these distributions can even be dependable on the applied external conditions, as observed in [117].

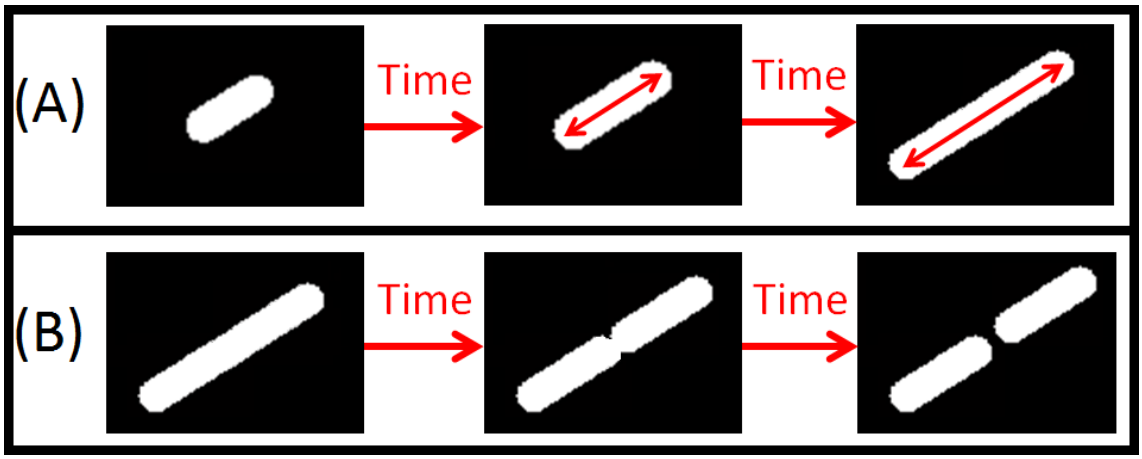


Figure 4.31 – **Modelling of cell growth and cell division.** (A) Spatial simulation of cellular growth along the major axis of the cell. (B) Spatial simulation of cell division at the centre of the major axis.

Although this process emulates how other cell shapes (bacillus, coccobacillus, vibrio) change their cell size, this actually needs to be changed in truly spherical shaped cells (cocci) as they do not have an elongation process [468], but create a division septum at mid-cell, which allows them to create two daughter cells roughly of the same size of the parent cell due to entropic forces [16]. Due to this, in the first version 'Image Tracking Generator' [465], no division process was implemented.

The new version has implemented object division as the 'miSimBa' toolbox, but also added a new division process of spherical shaped cells, where the parent cell "splits" in half, originating two daughter cells, by splitting the object with a morphology factor m into two objects with a factor m/2, and quickly growing both objects to a morphology factor of m and inheriting from the parent all the physical parameters of the parent cell and sharing the same cluster force (if inside a cluster). An example of the second implementation of cell division modelling is shown in Figure 4.32. Modelling of the temporal organization of the cell division process is based on the Stochastic Simulation Algorithm (SSA) approach, as detailed in section 3.2.1.

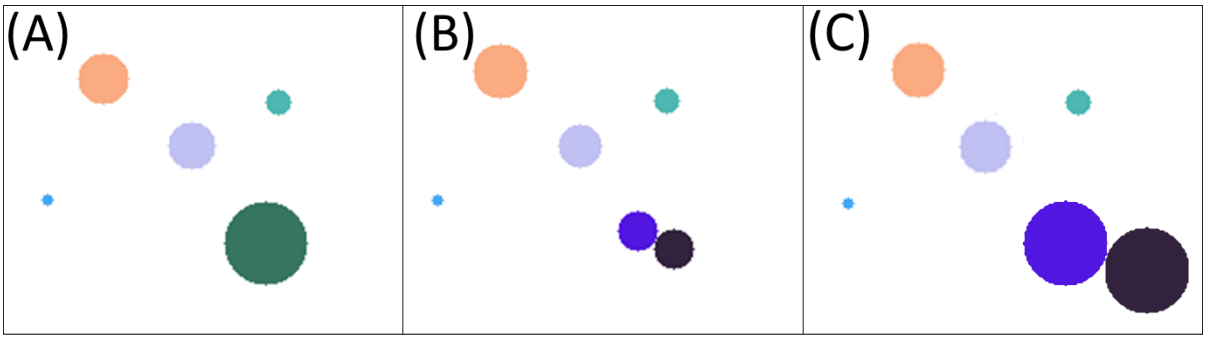


Figure 4.32 - Example of object division from frame (A) to frame (B), and the rapid growth towards the same size of the parent cell in frame (C).

4.2.2.3. Object Motility

Bacterial growth as a colony can also be dependent on the capability to move in the direction of more favourable conditions, which at its basic form is normally associated with Brownian random movement or active movement towards a specific gradient, e.g. chemicals (chemotaxis) and temperature (thermotaxis) [35].

In the 'miSimBa' toolbox, cell motility is modelled by random movements, done by the rotation of the axis angle of the bacteria (changing the orientation angle as seen in Figure 4.33-B) or by a geometric translation of the centre of the bacteria (moving all the other pixels in that direction, as seen in Figure 4.33-C) or by mixing both types of movement. To simulate this stochastic process, a random number is sampled using the maximum velocity and the maximum rotation that a cell can have (this numbers can also be obtained from experimental studies of the condition that have to be simulated.

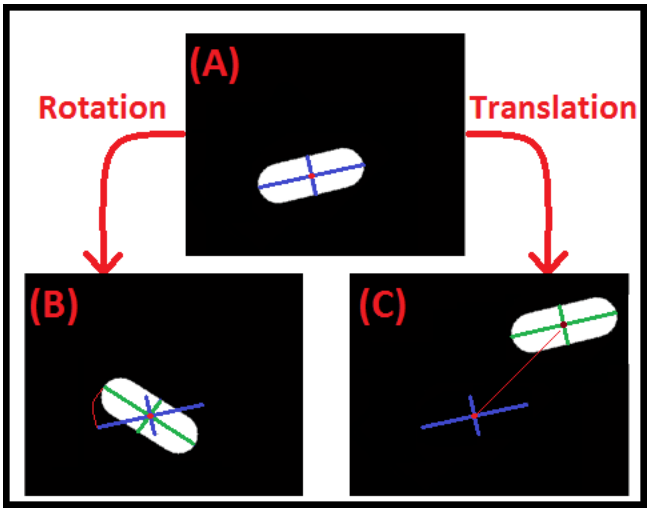


Figure 4.33 – **Modelling of cell motility.** (A) Initial state of the bacteria before movement. (B) Rotation movement, the centre remains in the same place and the axis move their orientation angle. (C) Translation movement, the centre moves, but the axis remains with the same angle. Note: In both (B) and (C) Blue line represents the initial state of the Axis and the green line the new state.

In the 'Image Tracking Generator' toolbox, additional options were added to recreate cellular movement, which can be changed according to the user selection. The user can select if the objects move arbitrarily through the image (consistent with the Brownian random movement) where in each frame, each object can move to a new x and y coordinates by a randomly sampled distance and orientation that cannot be higher than the 'Maximum Velocity' value in pixels.

The user can also select an option to give objects a 'Physical Move', where the velocity is randomly sampled, but the first given orientation is fixed, and it is assigned to the object, until it collides with image boundary (which happens if the 'entries and exits' option is deactivated), changing its orientation using is then reflected respecting Snell's Law or it collides with another object (which happens if the 'occlusions' option is deactivated) and both objects change their orientations in an approximation to the reflection laws, but ignoring differences in their morphologies, as seen in Figure 4.34. This is not a totally correct approximation, as cell with higher mass or larger cells will be able to push smaller cell if they are moving, or smaller cells can even stop their movement when they collide with larger cells. This type of examples needs to be studied case by case, when creating realistic simulations of cell movement.

When the ‘occlusions’ option is activated, the cell do not interact and will simply continue their movement (replicating images where the cells are in different z-planes, but still are observed in the same x-y plane. When the ‘exits and entries’ option is not activated, the objects go out the image boundaries, but are still simulated, allowing them to be re-enter the image (the only difference is that the objects are not rendered in the simulation). Modelling of the temporal organization of the cell motility process is based on the Stochastic Simulation Algorithm (SSA) approach, as detailed in section 3.2.1.

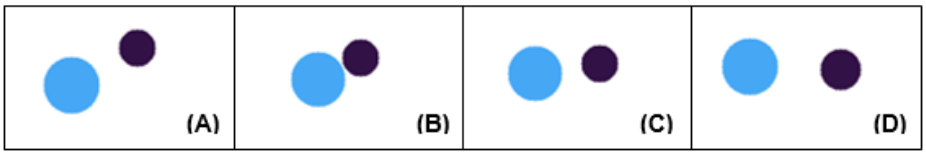


Figure 4.34 - Collision between objects with "Physical Move". Objects in: (A) Frame 10; (B) Frame 16; (C) Frame 19; (D) Frame 23.

4.2.2.4. Cluster Creation

In terms of spatial arrangement, bacteria can be organized in single forms or be grouped in pairs (diplo prefix), in chains (‘strepto’ prefix). Cocci bacteria can also organize in groups of 4 (tetrad), 8, 16 or 32 (‘sarcinae’) or in grape-like clusters (‘staphylo’ prefix). Bacilli bacteria can organize in palisade structures (side by side) or can be in unstructured spatial clusters [18], as detailed in section 2.2.

While the ‘miSimBa’ tool only allowed random creation of clusters, the ‘Image Tracking Generator’ toolbox allowed a forced creation of clusters with similar properties, using the button ‘Cluster Properties’ (see Figure 4.28), which leads to a new window with the options for clusters’ creation, as seen in Figure 4.35. The properties that can be changed by the user are desired number of clusters, objects per cluster, and size of the clusters in pixels. It is also possible for the user to choose between two types of objects’ kinetics: ‘Follow the Leader’ and ‘Alternative Movement’, and the user of a ‘Cluster Centre Force’ property and its strength value, as detailed below.

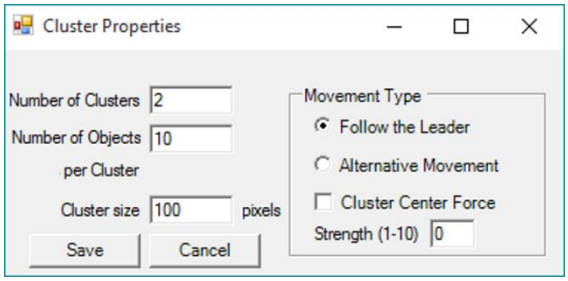


Figure 4.35 - Interface options for cluster properties.

If the user selects the ‘Follow the Leader’ option (as shown in Figure 4.36-B), each cluster has a leading object, mimicking the ‘strepto’ spatial organization. The characteristics of the other objects of the same cluster are dependent on the leader’s behaviour. The leader “receives” the physical parameters at first frame (velocity and orientation) and at each frame the other objects of its cluster will move in the leader’s direction, minimizing the distance to it, but respecting the “non-collision” rule. If two objects from different clusters collide, one of them will start belonging to the other cluster. This may cause the “merging” of clusters.

When the user selects the 'Alternative Movement' option (as shown in Figure 4.36-A) all objects of each cluster have the same physical parameters, which means that they move in the same direction with the same speed (with a small independent arbitrary component), which mimics alternative organizations, depending on the number of cells and the cluster center force. The 'Cluster Centre Force' option, is exclusively applied for 'Alternative Movement' that creates an attraction force at the cluster's centre, with a selectable strength selected by the user. This force keeps cluster's objects together, even when colliding with the image borders or other objects. Increasing the strength, the objects will move faster to the cluster's centre. In this mode of motility, when objects from different clusters collide, they will be "left behind" by their cluster until they can join it again.

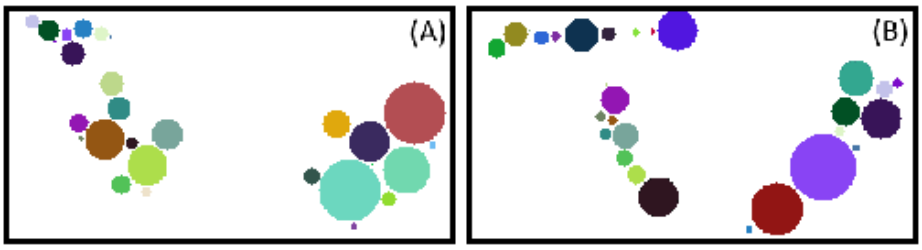


Figure 4.36 - Exemplificative frames of cell movement. (A) 'Alternative' Movement (B) 'Follow the leader' Movement.

4.3. Contribution to the development of new Machine Learning Techniques

4.3.1. Merge and Discard Classifiers of segmented objects

As mentioned in Section 4.1.3, one of the implemented segmentation algorithms ('GPL+CART') was implemented by using the Gradient Path Labelling Algorithm [304] to create the segmentation seeds. This creates an over-segmented labelled imaged image, as can be seen in Figure 4.9-A in section 4.1.3. The proposed 'GPL+CART' algorithm then uses the Classification and Regression Trees Algorithm (CART) [308] to merge and discard inappropriate sections [306].

The CART® software (for Windows) [308], version 4.0 was the chosen program to train the discard and the merge classifiers. The chosen options were the standard ones: Gini as the splitting method, the same probability for both classes, 10-fold cross validation as method for testing tree, the best suggested tree is the minimum cost one, regardless of the tree size. The minimum node size was changed, so the parent nodes must have at least 2 cases and the terminal nodes must have at least one case. Similarly to the dataset obtained for the training in Brightfield images [306], the Area (A), Perimeter (P) Shape factor given by Perimeter2/(4 ∗ π ∗ Area) (S), Major Axis Length (LH); Intensity Variance of the entire segment (VAR) and the Ratio of the contour intensity over the inside intensity (R) were all calculated for each segment. Figure 4.37 shows an example of the dataset (the full dataset has 568 examples, divided into 392 cases of discarded segments and 176 segments that were kept).

An example of the implementation of the discard classification algorithm on the example shown in Figure 4.9-A in section 4.1.3 is presented in Figure 4.9-B. The example shows two segments that can be corrected using the merging classifier (in red), but also shows a segment that cannot be corrected (in blue), since it must split into two different segments.

C:\sim\discardClassTrainingSet.xls7

	A	P	S	VAR	L	R	actio
1	36092	1251.786	3.454931	4.707133	38	0.9795918	1
2	36092	1251.786	3.454931	4.707133	38	0.9795918	0
3	57158	1775.883	4.390784	3.579046	39	0.9659864	0
4	1716	231.1371	2.477491	1.448665	28	0.9861111	0
5	189	59.79899	1.505622	-2.394402	15	0.993007	0
6	19275	874.3057	3.155894	-2.65276	25	0.9862069	0
7	2255	255.7645	2.308469	3.039326	29	0.9724138	0
8	8117	636.4407	3.971097	-1.517732	29	0.9862069	0
9	14973	968.2885	4.982999	-1.72183	29	0.9793103	0
10	9334	596.6417	3.034936	-1.674024	25	0.9724138	0
11	366	75.59798	1.242595	77.63738	46	0.9470199	0
12	205	56.87006	1.255462	10.28272	21	0.9527027	0
13	26313	1078.874	3.520152	2.656438	31	0.9861111	0
14	14188	807.7788	3.65977	4.69132	33	0.9861111	0
15	4150	458.6589	4.033869	5.558774	27	0.9931034	0
16	31739	1325.124	4.402607	0.6595434	34	0.9863014	0
17	19481	910.2885	3.384831	23.58575	37	0.9407895	0
18	14199	764.0803	3.271978	4.042409	35	0.9794521	0

Figure 4.37 – Example of the discard dataset. 'actio' represents the classes (1 is to discard and 0 is to keep the segment) and with inputs: Area (A), Perimeter (P) Shape factor (S), Length (L), Variance (V) and the contour and intensity ratios (R).

After rejecting the background segments, there are many cases where cells are split into two or more segments (see Figure 4.9-C). To solve this, another classifier was trained to merge adjacent segments belonging to the same cell. To create this dataset the chosen inputs were the Shape factor of two segments touching segments (F), Variance of the image in the contact area between both segments (VAR), Length of the contact zone between both segments (C) and the Ratio between intensity of the image in the contact zone and the pixel intensity in the contour of the segments (R) similarly to what was calculated in [306]. An example of the dataset is shown in Figure 4.38. (full dataset has 668 examples, divided into 355 cases of merged segments and 313 kept segments).

C:\sim\MergeSet.xls7

	R	VAR	C	F	actio
1	0.9446226	2201.798	27	2.708637	0
2	1.135992	13.78983	60	1.766307	0
3	0.5159003	8715.455	34	1.517564	1
4	1.04938	1113.034	26	1.93442	0
5	1.075929	16.81621	54	1.954501	0
6	0.5034951	8321.131	33	1.702305	1
7	0.5073143	10703.88	38	1.394948	1
8	0.596743	10160.09	31	1.511775	1
9	1.144173	11.06719	23	1.821455	0
10	1.124036	3.058824	52	2.045564	0
11	0.9523705	1261.07	48	2.023871	0
12	0.7034463	7028.258	32	1.487738	1
13	0.2756073	7799.887	40	1.709674	1
14	0.4941045	8965.845	45	1.509065	1
15	0.3878751	7645.879	44	1.694615	1
16	1.171488	6.435484	32	1.500291	0
17	0.6045402	8672.06	32	1.536121	1
18	1.254215	29.66154	26	1.729013	0

Figure 4.38 – Example of the merge dataset. 'actio' represents the classes (1 is to merge and 0 is to keep the segment) and with inputs: intensity ratios (R), Variance (V), Contact Zone (L) and the contour and Shape factor (F),

The merge classifier, also based on the CART algorithm, identifies the bacteria blocks that are connected and over-segmented. After identification, those blocks are merged in order to reduce the

over-segmentation. However, blocks that do not belong to the same bacterium must not be joined, even if physically connected. The merge classifier distinguishes these blocks from the image intensity shifts at the borders between blocks, since pixels within a bacterium exhibit lower intensity than those located on cell membranes.

After the removal and merging of segmented areas, it is common that a small number of bacteria segments still need to be corrected due to errors of the Discard and Merge classifiers. This correction was automatically implemented, as detailed in section 4.1.3, using splitting algorithms (Watershed and Distance transform, and other morphological operations) and can finally be manually corrected using the manual correction procedure that was also implemented.

Examples of final manually corrected segmentation results can be seen in Figure 4.9-D.

4.3.2. FtsZ Ring Classification

During this research work, the LBD group started a new collaboration to understand, how sub-optimal temperatures influence Z-ring placement, fluorescent FtsZ-rings and nucleoids were observed at the single cell level, to study the existence of uncertainties in the Z-ring placement along the major cell axis [167], [469]. During this study, it was required to classify the three stages apparent of the FtsZ ring [165], [166], as mentioned in section 2.3.5.

The initial step in the classification process normalized each cells by the major axis, using Principal Component Analysis [470] to normalize the major and minor axes lengths and the center coordinates of each cell, in order to obtain the intensity distribution of the fluorescence levels along the major cell axis [167]. After this step, the normalized cell area was divided into three regions, the two poles and midcell, by dividing each cell in three normalized bins along the major axis located between position: [0, 0.25],]0.25, 0.75[and [0.75, 1] [167]. From each cell, the mean and standard deviation coming from each of the mentioned regions were extracted, resulting in 6 different inputs for each of the classification Algorithms.

The main difference between the initial classification process (which had 300 examples) [167] and the final classification procedure (which had 500 examples) was initially, three different stages of the FtsZ ring were classified, while the final procedure joined the two initial steps of the FtsZ Ring formation, since it was only required to analyse the cells that had FtsZ Rings in the last stage of development. Since the classification output had just two variables, the first corresponding to all examples in the initial and intermediate stage (see Figure 4.39 A1, A2, and A3) and the second to all examples in the last stages of development (see Figure 4.39 B1, B2, and B3), the classification procure was reduced from a multiclass problem to a binary problem, which generally increases the classification performance. The comparison between both classification procedures comparing to the previous FtsZ Ring classification process is presented in section 6.1.10.

The classification procedures are done using the Machine Learning packages present in MATLAB™. Three different Machine Learning Algorithms: Decision Trees (DT), Support Vector Machine (SVM), and Regularized Multinomial Logistic Regression (RMLR), which were reviewed in section 3.3.

The evaluation of the performance of each ML Algorithm was done by calculating the accuracy (the ratio of correctly classified samples to the total number of samples averaged over the folds) of each method with a 10-fold cross-validation , which randomly partitions the data into 10 subsets, and trains the Algorithm with 9 subsets and evaluates the performance on the last subset (a process

that is then repeated 10 times). Our accuracy results are based on repeating this validation process 100 times and averaging the result.

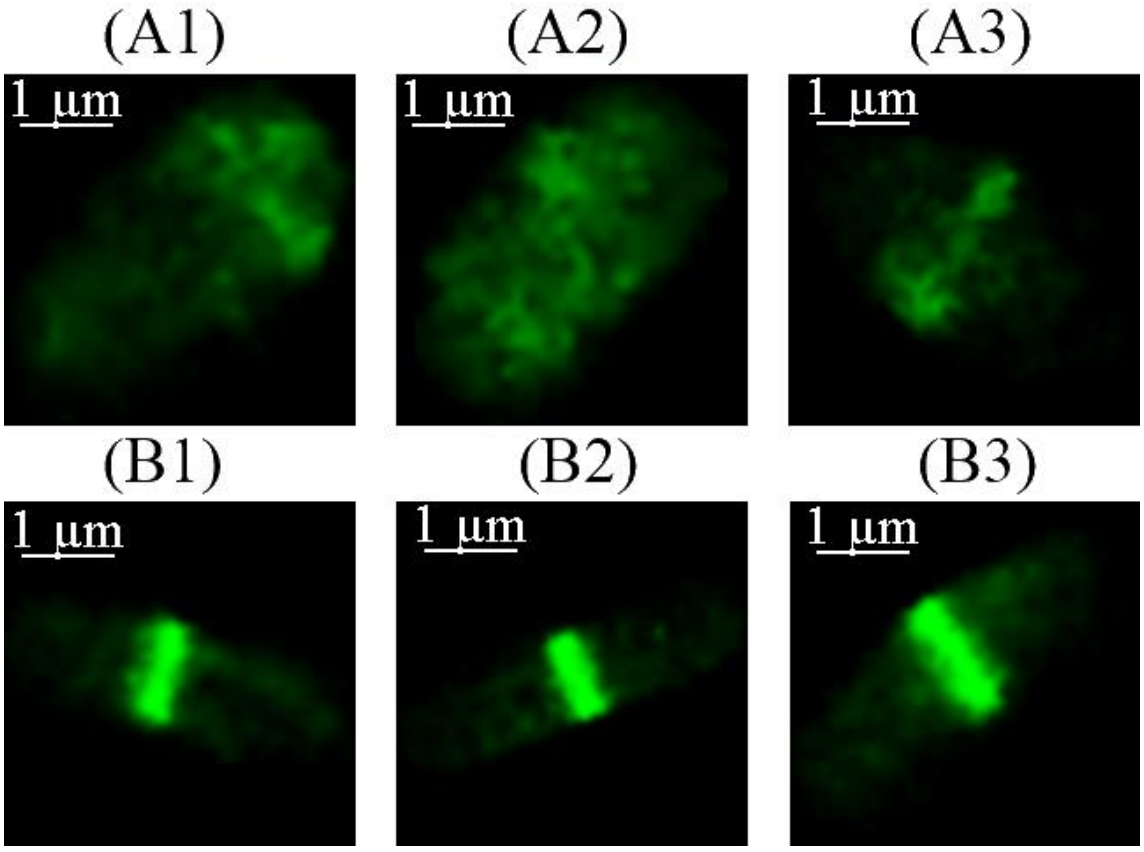


Figure 4.39 – Classification of the three FtsZ formation stages. In this example, the initial phases (A1 and A2) were joined with the intermediate phase (A3), while the last stage (where the ring is formed) is represented by B1, B2 and B3.

4.3.3. Tracking Algorithm

As mentioned in section 3.3.2.4, inn this research work, the implemented tracking algorithms to that were tested using the artificial data generated by the developed Simulation Framework is based on the implementation of nearest-neighbour algorithms and the DBSCAN method, and the combination of both algorithms.

The first tracking algorithm that was tested was a simple nearest-neighbour, which only took into consideration the position of the centroid of each object in each frame of the time-series and uses the Euclidian Distance between points to find matching objects between frame n and n+1. Being $\mathrm{d_p}$ the distance between two objects:

$$d_p = \sqrt{(x_n - x_{n+1})^2 + (y_n - y_{n+1})^2} \quad \textbf{(4.18)}$$

Equation 4.18 is similar to equation 3.5 (see section 3.3.2.4), where (x_n, x) and (y_{n+1}, y_{n+1}) represent the centroid coordinates of each object in frame *n* and *n+1*, respectively in the x-plane and the y-plane.

A second algorithm was implemented, which not only considers the differences between the positions of each object in each frame, but also considers a shape-related factor, called morphology. This algorithm calculates the taxi-cab distance [441] percolated by each object between frames n and

n+1 using equation 4.19, where m_n is the morphology factor of each object in frame n, and m_{n+1} the morphology factor in n+1, the difference, d_m, between these variables is calculated by:

$$d_m = |m_n - m_{n+1}| \quad \textbf{(4.19)}$$

The total difference, d_t, between each object in each frame pair is given by equation 4.20 with α and β being the weights given to each partial distance [441]. Here different weights are used (as presented in the Results section), in order to study the best way to combine them:

$$d_t = \alpha \cdot d_p + \beta \cdot d_m \quad \textbf{(4.20)}$$

As mentioned in section 3.3.2.5, the in this research work, the DBSCAN algorithm is also used to track clusters of artificial objects [446]. In this work, the revised version of the DBSCAN algorithm is used [447], which as mentioned formalizes the notion of "cluster" and "noise", using the definition of density to characterize clusters, meaning that to define a cluster, the density of the neighborhood of each point has to be higher than a given threshold.

After identifying the clusters in all frames with DBSCAN, a novel algorithm for object tracking was developed. This algorithm assumes that objects are grouped and move in clusters, treating each cluster as a separate object while tracking. The first step (with all clusters identified) is to isolate the clusters and calculate their centroid, in coordinates x and y:

$$Cluster_{centroid} = {\sum_{i=1}^{N} x_i}/{N} \quad \textbf{(4.21)}$$

After all centroids are calculated, they are processed as objects, since they have their own coordinates. A nearest-neighbour algorithm (similarly to equation 4.18, but with the position of each cluster) is then applied to these coordinates, which provided an improvement in the tracking of the clusters and resulting. The results of the implementation of the nearest-neighbour, the DBSCAN and their combination are presented in section 6.2.

Chapter 5. Experimental Developments

This section presents the experimental setups of the description of the research experiments. The microscopy settings, bacterial strains, growth conditions and induction protocols are detailed for each experiment presented in Chapter 6. It is noted that for the image processing framework, all of the work documented here, mainly the preparation and manipulation of the bacterial cells was done by the colleagues at Laboratory of Biosystem Dynamics under the supervision of Professor André Sanches Ribeiro, namely, Nádia Gonçalves, Ramakanth Neeli-Venkata and Samuel Oliveira. For the image simulation framework, the details of the produced benchmark data are also presented.

5.1. Experimental setup

5.1.1. Bacterial strain

The experiments reported in this research work used different strain of *E. coli* cells. The CM735-derived strain NK9386 was used to study the Nucleoid. This strain expresses the endogenous gene hupA fused with fluorescent protein mCherry, under the control of the native promoter, incorporated into the chromosome [96] (a kind gift from Nancy Kleckner, Harvard University, U.S.A). This strain was transformed with pEG12-ftsz::gfp [164] (kind gift from Kenn Gerdes, Copenhagen University, Denmark) under the control of a lac promoter.

To collect empirical microscopy data on the Min system, the *E. coli* W3110 derived strain FW1561 was used. This gene expresses the endogenous gene minD minE fused with fluorescent protein superfolder GFP (sfGFP), under the control of the native promoter, incorporated into the chromosome [94] (a kind gift from Cees Dekker, Delft University of Technology, Netherlands).

The simultaneous study of the nucleoid, Z-ring and protein aggregates (MS2-GFP tagged RNA) on individual cells, was only possible with the use of the strain FW1551 expressing the endogenous gene hupA fused with fluorescent protein TagBFP, under the control of the native promoter, incorporated into the chromosome [94] (a kind gift from Cees Dekker, Delft University of Technology, Netherlands). This strain was transformed with medium copy plasmid expressing MS2-GFP reporter system, a single-copy BAC vector plasmid expressing the target gene PtetA-mRFP1-96BS with a 96 MS2-GFP binding site array and pEG12-ftsz-mCherry. All these plasmids have different origin of replications and different antibiotic resistances.

To validate the spot detection algorithm, the target gene PtetA-mRFP1-96BS with a 96 MS2-GFP binding site array was constructed in a single-copy BAC vector by restricting out the Plar promoter with BamH1 restriction endonuclease from a BAC clone carrying a target gene Plar-mRFP1-96BS [5] (a kind gift from Ido Golding, University of Illinois, IL), and replacing it with PtetA amplified from the pTetLux1 plasmid [471].

To measure intracellular concentrations of RNA polymerases (RNAP), the *E. coli* RL1314 strain was used (kind gift from Robert Landick, University of Wisconsin-Madison, USA), carrying GFP tagged

RNAPs (RNAP-GFP) . Changes in fluorescence levels with, e.g., media richness, are consistent with RT-PCR (rpoC transcript levels) and plate reading measurements .

5.1.2. Growth Conditions and induction

All overnight cultures were grown in LB (Lysogeny broth) supplemented with appropriate antibiotics, when required, for 15 hours at 37 °C with shaking (250 rpm). Subcultures were subsequently made by diluting overnight cultures into fresh LB and the subcultures were left in the incubator at 37 °C with shaking (250 rpm) until the cells attain mid-logarithmic phase (OD600 ~ 0.3). Cells at this stage were either induced with appropriate inducers or taken to the microscopy chamber for visualizing under the microscope.

For the visualization of the FtsZ-GFP expression, cells were induced by adding 40µM IPTG to the culture and left in the incubator for 30 minutes prior to microscopy. For the visualization of the Min system, mid-logarithmic cultures were placed on agarose gel pad. There is no induction required in this case.

For the visualization of inclusion bodies, the cells were exposed to osmotic stress during time-lapse microscopy. Sodium Chloride (125 and 300 mM of NaCl) was added to the growth media and pumped into the thermal chamber (set to 37°C) for 1 hour. For population microscopy imaging, the cells were kept under osmotic stress for 60 minutes (osmotic stress-inducing media with 125 and 300 mM of NaCl).

To recreate the visualization of the 3 fluorescent proteins simultaneously, first, the reporter plasmid activation was performed by adding 0.4 % of L-arabinose to the culture and incubated at 37°C for 60 minutes. Following the reporter activation, the target plasmid was activated by adding 50ng of Anhydro tetracycline to these cultures and left in the incubator for 30 minutes. Cells were pelleted and proceeded to microscopy.

5.1.3. Microscopy preparation and Image Acquisition

In single time point microscopy, induction was performed as described in methods. Cells were then left in the shaker incubator at 37 °C, prior to image acquisition. From this, 8 µL of cells were placed on 1% agarose gel pad prepared in LB media. Images were taken after placing the cells under observation.

In time-lapse microscopy measurements, cells (NK9386: FtsZ-GFP and FW1561) were placed on a microscope slide between a coverslip and LB agarose gel pad containing with or without IPTG. Images were captured for 1 hour every 1 min by confocal microscopy or HILO microscopy.

Imaging was performed by a Nikon Eclipse (Ti-E, Nikon) inverted microscope with a C2+ point scanning confocal system and a 100x Apochromat TIRF (Total internal reflection fluorescence) objective (1.49 NA, oil). For population imaging, three channels fluorescence was measured using a 461 nm laser (Melles-Griot) HQ514/30 filter for TagBFP, a 488 nm argon laser (Melles-Griot) and HQ514/30 filter for green fluorescent spots. HupA-mCherry fluorescence was measured using a 543 nm He-Ne laser (Melles-Griot) and HQ585/65 filter (Nikon, Tokyo, Japan). For time-lapse microscopy measurements of FtsZ-GFP and mCherry-tagged nucleoid(s) (NK9386: FtsZ-GFP strain), similar

microscopy setup was used excluding the blue laser. Images were acquired using a medium pinhole, gain 90 and 3.36 µs pixel dwell. The acquisition rate of confocal images was of one image per minute and of the phase contrast had an acquisition rate of 1 image per 5 minutes. The duration of each time-series is reported for each specific study. A photo of the microscopy setting is shown in Figure 5.1.

Highly Inclined and Laminated Optical sheet (HILO) microscopy [175] was used to visualize the fluorescent labelled Min system. The fluorescence signal was recorded using an EMCCD camera (iXon3 897, Andor Technology) with a 488nm laser, along with the HQ515/30 filter and the Texas Red filter (Nikon, Tokyo, Japan). Phase contrast images were captured simultaneously by a CCD colour camera (DS-Fi2, Nikon).

The software for image acquisition was NIS-Elements (Nikon, Tokyo, Japan). Slides were kept in a temperature-controlled chamber (Bioptechs, FCS2) at stable temperature (37 °C, unless stated otherwise).

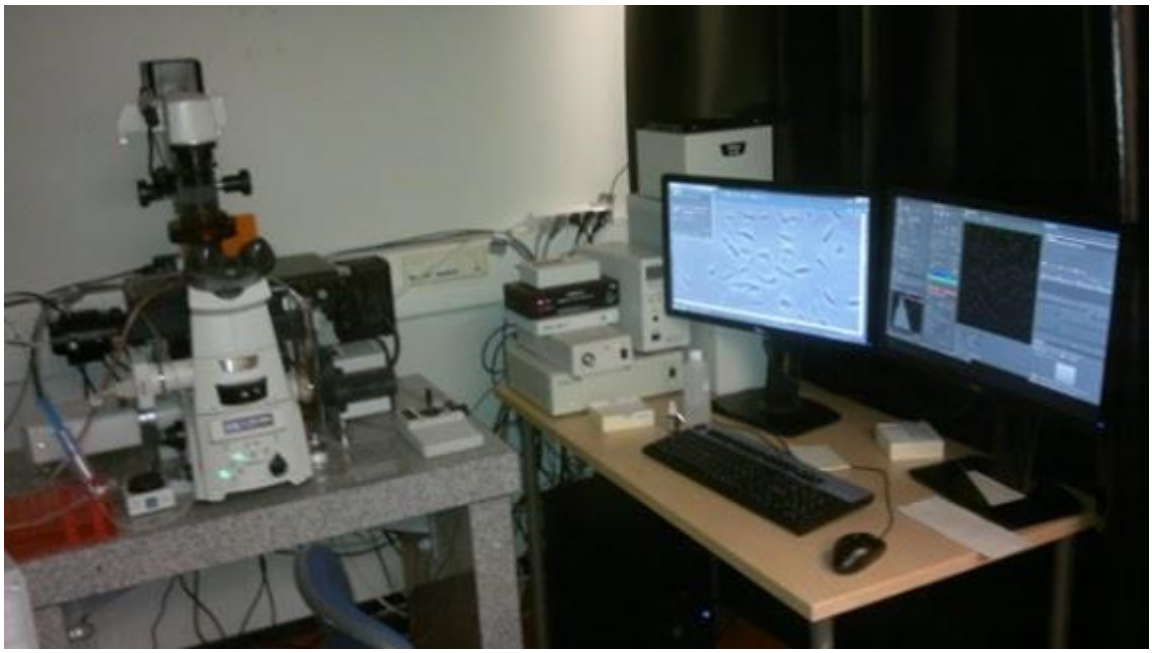


Figure 5.1 – A photo of the microscopy setting. Taken from: sites.google.com/view/andreribeirolab/home/laboratory

Chapter 6. Validation and Discussion

This section presents the answers to the Research Questions, the confirmation of the research hypothesis and the validation of the methods described in Chapter 4. The validation of the image processing is based on benchmark data acquired by manual inspection and manual segmentation of the data. The validation of the simulation toolbox is based on a qualitative evaluation, and it is used to create new benchmark data that has been used to evaluate cell tracking methods. Finally, the dissemination results are also presented, including the publication of articles in journals and book chapters and participation in conferences and courses.

The integration of this research work on the SADAC project (Study of the kinetics of asymmetric disposal of aggregates in cell division and its correlation to functional aging from in vivo measurements, one event at a time), allowed the collaboration with several students and researchers from both the CA3 and the LBD groups. These colleagues have been important in this research work in the implementation, testing and validation process.

The validation process of the image processing techniques was done by manual inspection and manual segmentation of the data. The toolboxes and all the data were submitted to peer-reviewed journals indexed to the Web of Science, to renowned international conferences (with peer-reviewing process and with relevant technical programmes) and selected for publication in Book Chapters. Qualitative validation was done by surveying groups of microscopy experts. These experts were acquainted and contacted via the participation the EMBO Practical Course “Microscopy, Modelling and Biophysical Methods”.

Both the image processing and simulation toolboxes are publicly available, with the source codes and data were published. The websites also serve as a contact point for microscopy experts to test the developed toolboxes.

6.1. Image Processing Validation

Using the experimental setups described in Section 5.1, the validation of the implemented methods in the SCIP toolbox is provided in this Sub-Section.

Here, the Cell Segmentation Algorithms, the Tracking Algorithm, and the Structure Segmentation Algorithms (Gaussian and ‘Treshmorph’) are evaluated based on manually segmented/inspected images. The Gaussian and the ‘Treshmorph’ Algorithms were tested with various sets of parameters and for several structures of interest, namely, the Nucleoid, FtsZ Ring, MinD protein and other protein aggregates. For these structures, the supervised evaluation scores are provided (namely Accuracy, Sensitivity, Specificity, Precision and F1-score and calculated using the same equations as [472]). For the inclusion bodies, the same evaluation scores for the Seed Removal Algorithm and the same scores for the segmentation of inclusion bodies are presented. When applicable, the detection times that are presented are based on an Intel Core i5-3470 CPU @ 3.20 GHz with 16 GB RAM memory running a 64-bit Windows 7 operating system.

A global performance analysis of all the implemented algorithms is also provided, discriminating the algorithms and parameters that had the best scores for each of the studied cases. A temporal analysis based on signal-to-noise ratio of the images is also provided.

It should be noted that this detection performance can be case-dependent, and may change on different conditions as: using different dyes for structure staining, using different acquisition techniques (e.g. epifluorescence versus confocal), and different environmental conditions, which affect the visualization and organization of the internal structures (for example by lowering the binding affinity of the stain to the structure of interest). Therefore, the specific algorithms and parameters that provided the best results in these examples might not be the best for other examples. This is the reason why the SCIP image processing toolbox allows the user to test all the mentioned algorithms.

6.1.1. Image Registration

6.1.1.1. First Registration Method

The validation of the proposed intra-modal registration algorithm was done on two time-series of 121 minutes, that is also used for the detection of the MinD proteins in Section 6.1.5. This was a long time-series and has a good representation of the normal drift that occurs during the acquisition of Phase-Contrast Images (which have a resolution of2560x1920 pixels, with each pixel equal to 0.049 µm), which are acquired every 5 minutes (see section 2.4 for a detailed explanation challenges and limitation of the microscopy image acquisition).

To validate the proposed model, the 2-D Pearson Correlation value is calculated on the registration of consecutive images, when no image registration is done, when different image registration types are done (translation, rigid, affine) and the proposed registration method. As mentioned in section 4.1.2.1. The automatic registrations methods (translation, rigid, affine) are based on using the 'imregconfig' MATLAB function, using the 'Monomodal' Input Argument and the 'imregister' function. As mentioned in section 4.1.2.1, the proposed method first method is based on finding the best translation transformation, using an exhaustive search of the translation matrix that maximizes the cross-correlation function, while the MATLAB functions use by default a Regular Step Gradient Descent optimization method to find the best transformation.

As can be observed in Table 6.1, based on the Pearson correlation value, the drift between each consecutive frame is random, although in in the later stages of the time-series, there exists a higher correlation value than in the earlier stages, which means that the system was able to stabilize during the image acquisition process. This is also confirmed by the gradual increase in correlation value of the proposed registration method, and as can be seen in the example of Figure 6.1 (e.g. the less colour difference that is observed, the more correlated the images are). From Table 6.1 it is possible to observe that our proposed intra-model registration method can align all the consecutive with better results than the default MATLAB implementations, even when applying more complex transformations (e.g. affine), and that the transformation that had the closest results to our proposed method was based on finding the translation transformation. It is noted that it was not possible to achieve a Pearson Correlation value closer to 1 due to the differences between each consecutive frame (e.g. the growing of cells and the cell division process add more 'signal' to each frame), as can be observed in the examples shown in Figure 6.1.

Table 6.1 - Quantitative evaluation (Pearson Correlation value) of several image Intra-Modal registration algorithms

Registered Images (time in minutes)	No registration	Automatic Registration (Translation)	Automatic Registration (Rigid)	Automatic Registration (Affine)	Proposed Intra-Model Registration
1 to 6	0.0642	0.0651	0.0245	0.0158	0.8317
6 to 11	0.1384	0.5794	0.0130	0.0368	0.8366
11 to 16	0.1407	0.5817	0.0006	0.0069	0.8442
16 to 21	0.0389	0.5971	0.5941	0.0162	0.8440
21 to 26	0.4704	0.7019	0.0135	0.0021	0.8506
26 to 31	0.3648	0.6575	0.0130	0.0021	0.8506
31 to 36	0.1575	0.6392	0.5680	0.0149	0.8552
36 to 41	0.6032	0.7578	0.0122	0.0130	0.8536
41 to 46	0.1838	0.6516	0.0239	0.0623	0.8545
46 to 51	0.3595	0.6873	0.0038	0.0071	0.8540
51 to 56	0.0395	0.0402	0.0534	0.0006	0.8618
56 to 61	0.3590	0.6973	0.0125	0.0051	0.8668
61 to 66	0.0510	0.0565	0.6075	0.0348	0.8688
66 to 71	0.0336	0.0338	0.0333	0.0104	0.8674
71 to 76	0.1432	0.6685	0.0231	0.0011	0.8717
76 to 81	0.4138	0.7836	0.0208	0.0077	0.8714
81 to 86	0.3302	0.7134	0.0196	0.0271	0.8764
86 to 91	0.2850	0.7072	0.0028	0.0059	0.8769
91 to 96	0.2976	0.7460	0.0144	0.0805	0.8801
96 to 101	0.4666	0.7700	0.0165	0.0028	0.8802
101 to 106	0.5095	0.7670	0.0014	0.0306	0.8867
106 to 111	0.7239	0.8168	0.0020	0.0080	0.8903
111 to 116	0.5173	0.7870	0.7570	0.0169	0.8864
116 to 121	0.6904	0.8230	0.0135	0.0271	0.8888

The implementation of the proposed image registration technique is shown to improve the cell tracking results in Section 6.1.3.

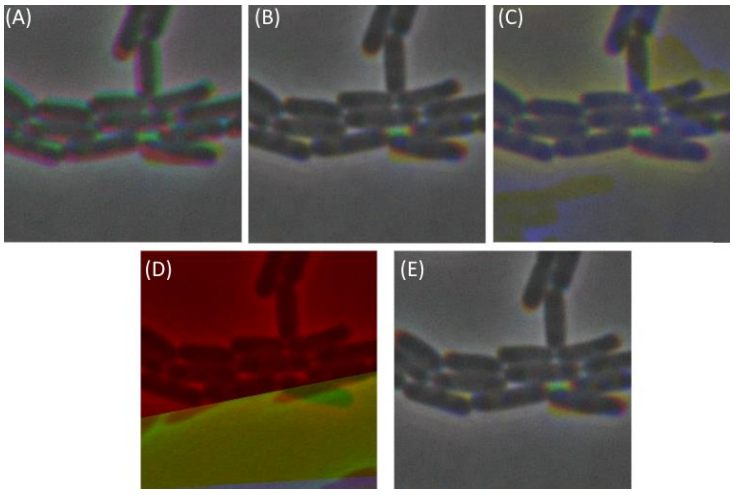


Figure 6.1 – Example of Intra-Modal Registration of drift in time-series of Phase-Contrast images (3 images acquired every 5 minutes). (A) No intra-modal registration was performed; (B) Automatic Registration using only Translation transformations; (C) Automatic Registration using Rigid transformations; (D) Automatic Registration using Affine transformations; (E) Automatic Registration using the proposed Intra-model methodology. The examples shown, represent a 250x250 square of all the superimposed images (with a 2560x1920 resolution.

6.1.1.2. Second and Third Registration Methods

The validation of the second and third proposed registration methods are based on the registration of one time-series of 31 minutes, with the multimodal registration of images containing the segmentation borders done on Phase-Contrast images and Confocal microscopy images of Nucleoids stained by HupA-mCherry, which is also used to validate the Nucleoid detection algorithm (see section 6.1.4). The Phase-contrast images have a resolution of 2560x1920 pixels, with each pixel equal to 0.049 µm and have to be aligned with Confocal image with a resolution of 2048x2048 pixels, with each pixel equal to 0.062 µm (see Section 2.3.5) with a different field of view. Due to this problem, for all examples, the transformed image is the one containing the segmentation masks, due to their smaller field of view (e.g. the segmentation images are transformed from 2560x1920 to 2048x2048). The initial validation of the second algorithm is based on the comparison of several multimodal automatic image registration methods starting with a simple overlay of both images, which is presented in Figure 6.2-A.

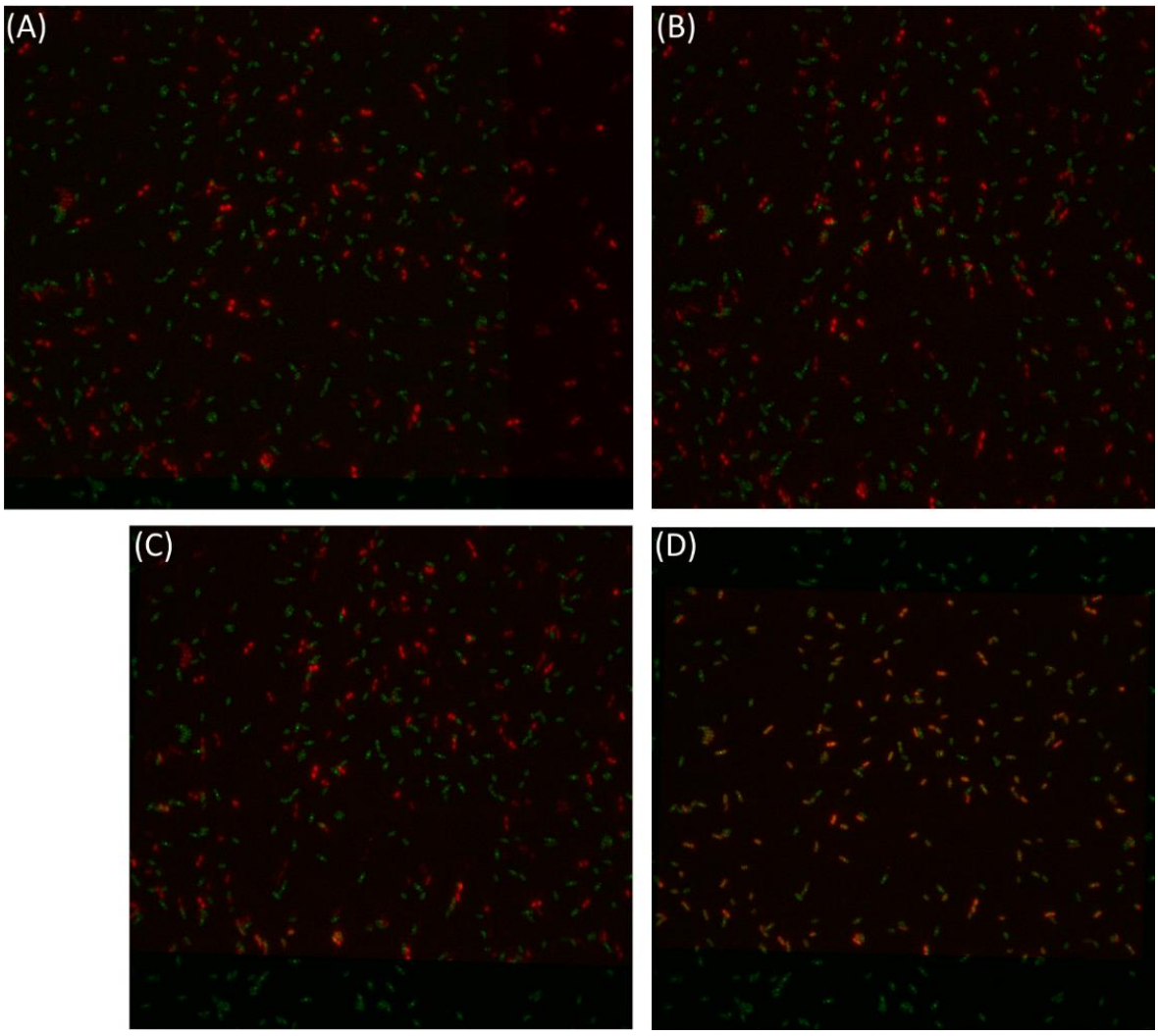


Figure 6.2 – Example of the application of different image registration transformations. (A) Simple Overlay; (B) Simple Resizing; (C) Rigid Transformation; (D) Affine Transformation. Note: In (A) the images have a different resolution, so the overlay is done by aligning the right corner of both images.

The simple resizing process (by just using a scaling transformation) [224], [225], is shown in Figure 6.2-B. An evolutionary algorithm [473] is used to calculate a rigid transformation (see Figure 6.2-C) and an affine transformation (see Figure 6.2-D). The evolutionary registration process [473] is done by changing slightly the parameters from the last iteration (the parent), and checking if that perturbation yields a better result to the new iteration (the child). If the child has better results, it is transformed into the new parent and this iterations continue until the maximum number of iterations is completed or a stoppage factor is achieved, if the parent has better results, it remains the parent and the perturbation is done in a different way [473]. All of these methods were based on the default parameters of the '*imregister*' and '*imregconfig*' functions, with the Multimodal input argument.

The second proposed method, as mentioned in section 4.1.2.2, is mainly based on the search of the best affine transformation, along with local adjustments are calculated by a local translation using the cross-correlation computation.

Finally, other intensity-based algorithm that was also tested, called phase-correlation, is based on registering images in the frequency domain in order to detect different geometric transformations [474]. This algorithm [474], which is invariant to image brightness, was not able to converge (see Figure 6.3) and find the location of a strong peak when trying to find the best rigid or similarity transformation (which includes all rigid transformations and adds the scaling process).

Several feature-based algorithms were also tested (see Figure 6.3). Three out of the six tested algorithms (Speeded Up Robust Features – SURF [475], Features from Accelerated Segment Test – FAST [476] and Binary Robust Invariant Scalable Keypoints – BRISK [477], failed in the feature detection step [219], particularly in the detection of features in the microscopy image containing DAPI-stained Nucleoids (see the text "Detected: 0" in Figure 6.3).

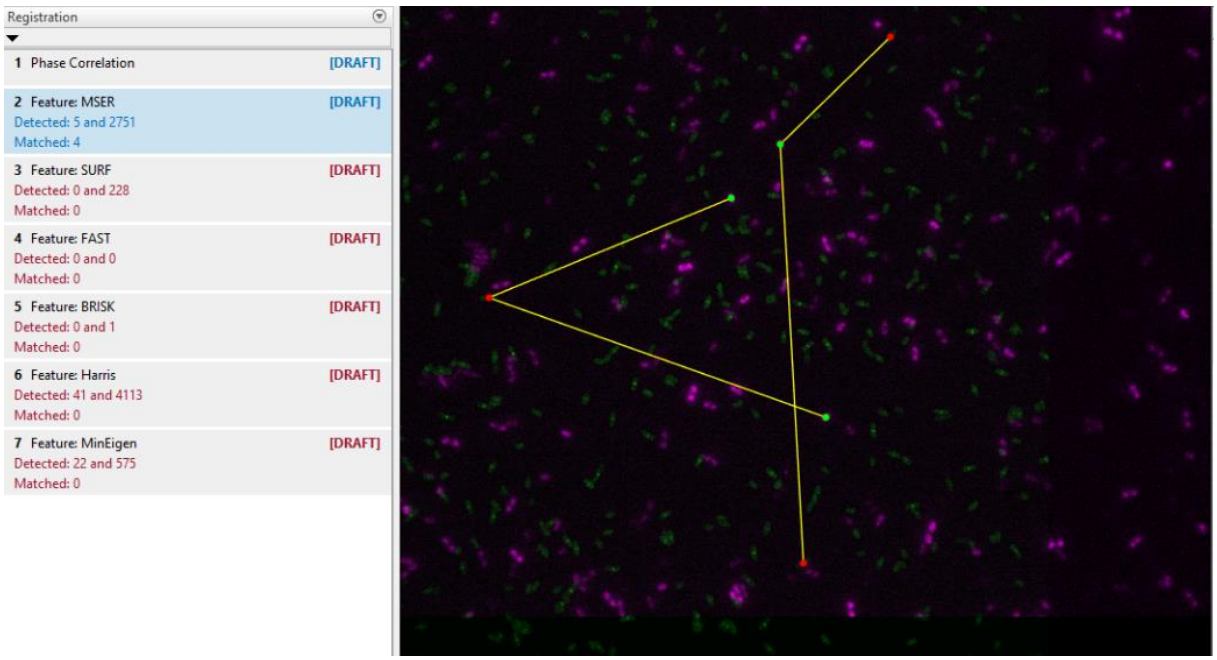


Figure 6.3 - Example of the unsuccessful application of intensity-based and feature-based registration methods. Phase-Correlation (intensity-based) and the MSER, SURF, FAST, BRISK, Harris, MinEigen (feature-based) methods were tested. Visualization of the feature matching step using the MSER algorithm is shown on the right.

Both the FAST [476] and BRISK [477] algorithms have been used to detect corner features in images, making their descriptors not suitable for this type of images, as there are no visible corners to

be detected. The SURF [475] algorithm, uses an Hessian Matrix for the detection step and a scale- and rotation-invariant descriptor, but was still not able to detect any features on the moving images.

The Harris–Stephens algorithm [478], which combines the detection of corners with the detection of edges, was the algorithm, out of the six tested algorithms, that was able to detect more features (41 and 4113 respectively on the moving and fixed images). The Harris–Stephens algorithm [478] was also not able to successfully pass the feature matching step. The MinEigen algorithm [328] was also tested, as it calculates the minimum eigenvalues, which normally represent corners and salt-and-pepper textures, also detected several features (22 and 575 respectively on the moving and fixed images), but was also not able to pass the successfully pass the feature matching step.

The Maximally Stable Extremal Regions (MSER) algorithm [479], which has been reported to be more robust to affine transformations than the previous feature-based algorithms [479], was the only feature-based method to pass the feature matching step (4 features were matched when using an Exhaustive Search Method to calculate the distance between the matched features [234], as observed on the right side of Figure 6.3, while only 2 features were matched using an approximate nearest neighbour search method [480], so this method was also chosen to test the image registration process.

As mentioned in Section 4.1.1 and observed in Figure 4.3, the chosen example have 5 functional images (each Confocal is acquired for every minute) for each morphological image (each Phase-Contrast is acquired every 5 minutes. Due to this situation, the previously mentioned allocation method was used, since it was implemented in the SCIP toolbox. Table 6.2 presents the quantitative evaluation of several automatic image registration algorithms.

Table 6.2 - Quantitative evaluation of several automatic image registration algorithms, based on the Pearson Correlation method. In the first column, the first number correspond to the time-point of the Phase-Contrast image and the second number correspond to the aligned confocal image.

Registered Images (time in minutes)	Automatic Registration (Resize)	Automatic Registration (Rigid)	Automatic Registration (Affine)	MSER	Second Proposed Method
1 to 1	0.0028	0.0812	0.4942	0.0044	0.5340
6 to 6	0.0044	0.0799	0.4939	0.0042	0.5310
11 to 11	0.0035	0.0785	0.4713	0.0042	0.5250
16 to 16	0.0048	0.0772	0.4501	0.0040	0.5012
21 to 21	0.0032	0.0771	0.4284	0.0041	0.4805
26 to 26	0.0025	0.0712	0.4138	0.0035	0.4595
31 to 31	0.0022	0.0696	0.3987	0.0032	0.4277

As can be observed in Table 6.2, for this example, both the Affine Transformation and our proposed methods produced the best quantitative performances. Both algorithms showed a reliable performance on the centre of the image, while failing some registration problems near the image borders, due lack intrinsic features, even when the feature matching step was successful and the difference in field of view. In Figure 6.4, an example of our proposed algorithm is presented.

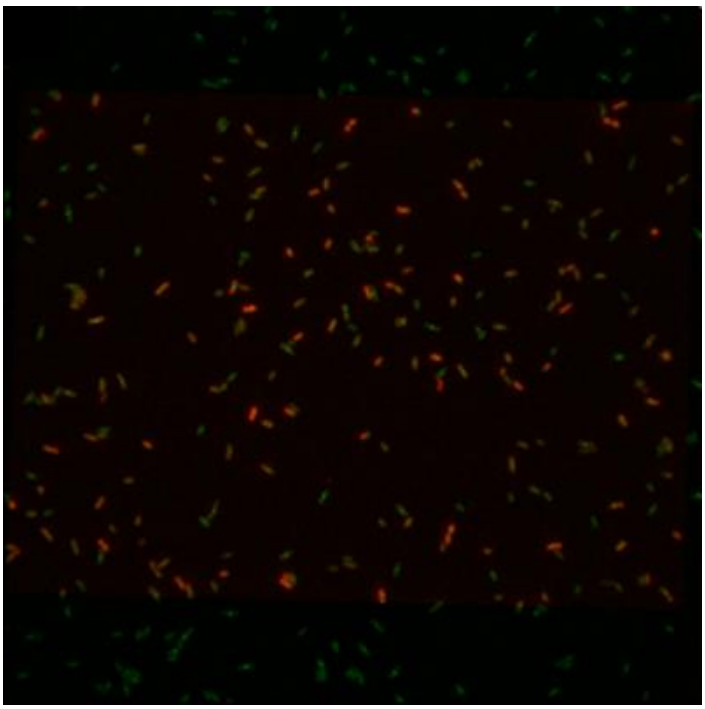


Figure 6.4 - Example of the application of our second proposed registration method.

As mentioned in Section 4.1.2.2, since this type of images lack intrinsic features, even when the feature matching step was successful, the quality of the matched features was not sufficient to make the registration process able to converge to a final solution. Due to this situation, and since no extrinsic features were previously placed in the Microscope, further improvement of the registration process was done by manual implementation of extrinsic features, by placing corresponding control points in the fixed and the moving images [242]. An example of the placement of twelve control-points is shown in Figure 6.5.

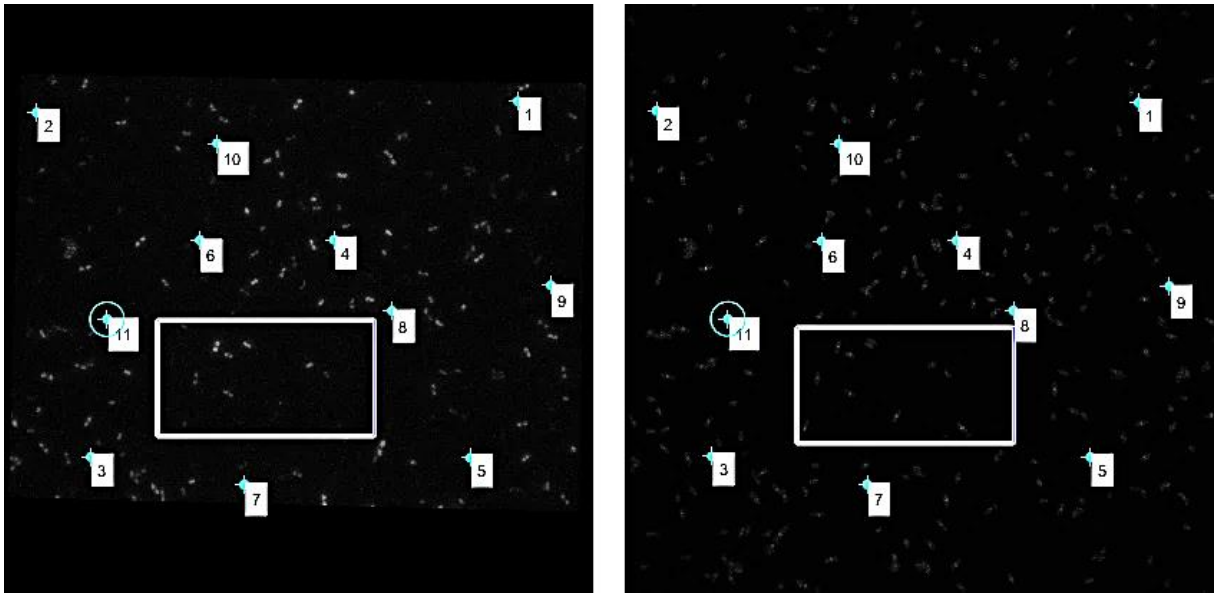


Figure 6.5 - Example of the application of a manually-based control-point image registration method. (Left) Moving Image containing the cell segmentation borders done on Phase-Contrast images (Right) the Confocal microscopy images of Nucleoids stained by HupA-mCherry.

Using the twelve manually placed control-points (see Figure 6.5), other geometric transformation were estimated [229] (see Figure 6.6) using different interpolating functions for the

image resampling [243] (linear, near-neighbour and cubic), with the near-neighbour giving the image sampling results. The proposed manual optimization based on the affine transformation [229], [234] was estimated (see Figure 6.6A), showed better quantitative results (comparing Table 6.2 with Table 6.3) registration process than the previous affine registration process. The affine transformation allows the preservation of collinearity and incidence and parallelism, so straight and parallel lines remain straight and parallel, but it does not preserve the length and angle of the lines [229], [234]. It is noted that at least 3 non-collinear control-points are necessary to directly estimate a global affine transformation [219]. With additional control-points (when placed correctly), the parameters of the mapping functions are estimated by a least-square function [219].

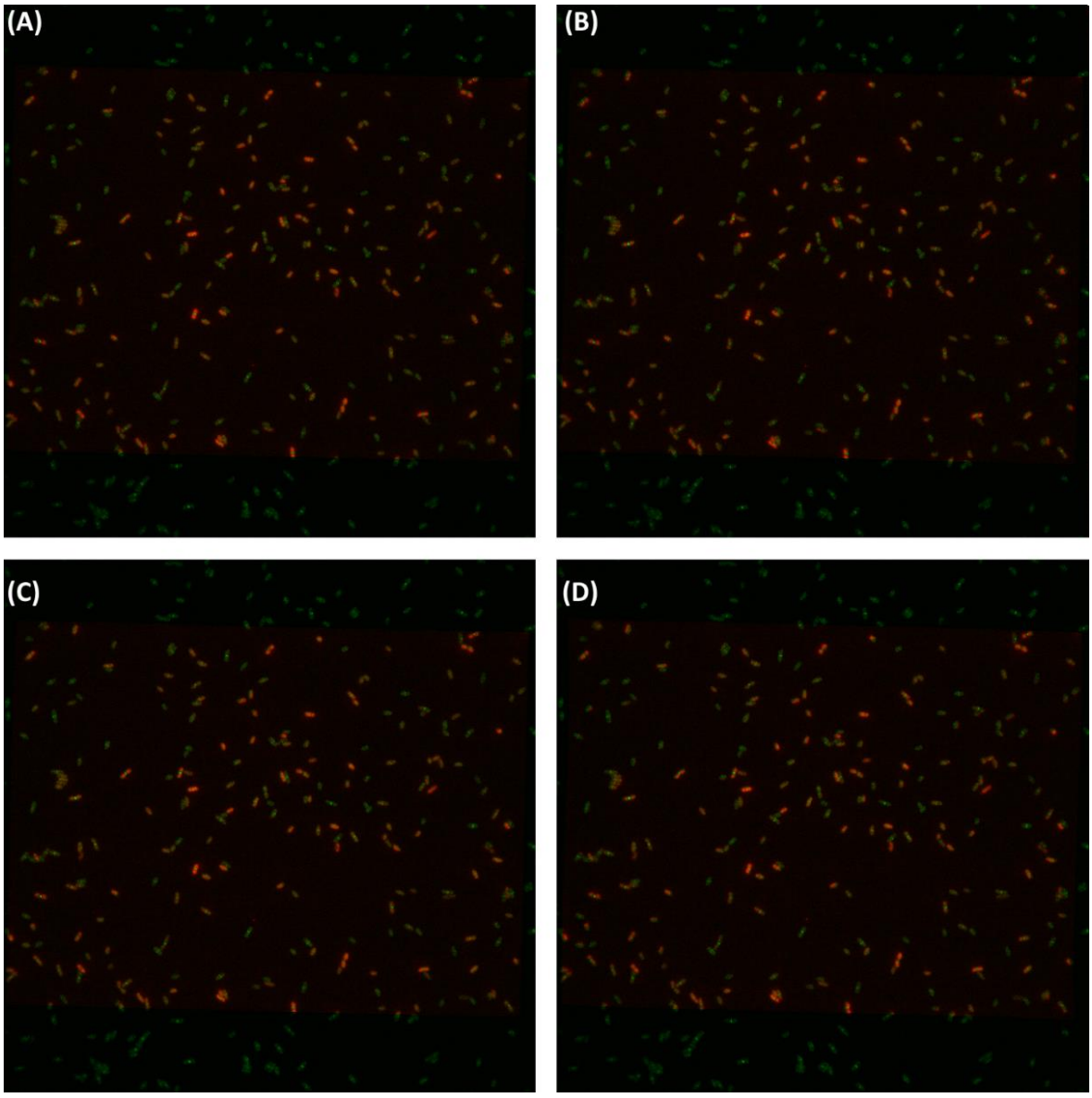


Figure 6.6 - Example of the application of different manual image registration transformations after the manually-based control-point image registration processing. (A) Third registration method with the Affine transformation; (B) Projective transformation; (C) Curved transformation – 2nd-degree Polynomial; (D) Curved transformation – 3rd-degree Polynomial.

Using the same twelve manually placed control-points (see Figure 6.5) it is also possible to provide estimations of projective transformations [229], [234] (see an example Figure 6.6-B).The projective transformation, contrary to the affine transformation, does not preserve parallelism, but still preserve collinearity and incidence, so straight lines still remain straight, but parallel lines are converged towards a vanishing point [229], [234]. Examples of the search of curved transformations is

presented in Figure 6.6-C and Figure 6.6-D, respectively with the search of 2nd and 3rd degree polynomials.

Projective transformations are useful when the camera sensors are placed in a tilted apparatus, correcting the resulting deformations, but can be harder to compute than the affine transformation due to a larger and less constrained search space, with its non-linear variables varying widely in sensitivity [481], so these type of transformations are normally only required when large deformations are found in the registered images [229], [234].

The Pearson Correlation was not calculated in the simple overlay example, since both images have a different resolution, requiring both images to be padded with zeros to calculate the value, but this padding counts toward the calculation of the correlation, which changes completely the comparison with the other examples.

Table 6.3 - Quantitative evaluation of several manual correction image registration algorithms, based on the Pearson Correlation method. In the first column, the first number correspond to the time-point of the Phase-Contrast image and the second number correspond to the aligned confocal image.

Registered Images (time in minutes)	Manual Correction Projective	Manual Correction 2nd-degree Polynomial	Manual Correction 3rd-degree Polynomial	Proposed Manual Correction (Affine)
1 to 1	0.5940	0.5935	0.5895	0.6012
6 to 6	0.5610	0.5612	0.5590	0.5672
11 to 11	0.5540	0.5535	0.5410	0.5602
16 to 16	0.5420	0.5419	0.5400	0.5501
21 to 21	0.5362	0.5290	0.5083	0.5362
26 to 26	0.5150	0.5110	0.4821	0.5154
31 to 31	0.4855	0.4856	0.4381	0.4861

For this example, the Affine Transformation produced slightly better quantitative performances, after the manual placement of the control-points. The automatic Affine registration showed a reliable performance on the centre of the image, while failing some registration problems near the image borders, which were vastly improved by the manual adjustments performed with the addition of 12 Control Points (see Figure 6.6). The use of simpler methods (e.g. like the Affine Transformation) is able solve the type of registration problems appear in this work, since projective and curved deformations are avoided by placing all the microscopy's camera sensors in the plane, as validated in Chapter 6.1.1.

6.1.2. Cell Segmentation

As mentioned in section 4.1.3, there were two segmentation algorithms implemented, the 'GPL+CART' and the 'Otsu + Median'. The 'GPL+CART' algorithm requires the use of Merge and Discard Classifiers, which were created with the CART® software (as detailed in section 4.3.1. This section is then divided into two parts, the first one presents the results obtained from the creation of these machine learning classifiers, while the second one presents the results of a full segmentation pipeline example, obtained with the both implemented algorithms: 'GPL+CART' and the 'Otsu + Median' and the additional steps that were developed for the SCIP toolbox.

6.1.2.1. Merge and Discard Classifiers

Figure 6.7-A shows the best suggested classification tree (with the minimum relative cost) for the merge classifier, with the chosen options (Gini as the splitting method, the same probability for both classes, 10-fold cross validation). This tree has 14 nodes (see rules in Annex A7) and has a classification accuracy of 96.023% for class 0 (keeping the segment) and 93.112% for class 1 (discard the segment), as can be confirmed in the confusion matrix in Figure 6.7-B. Finally, the variable that has the largest importance to the classifier is the Intensity Variance of the entire segment, as denoted by V, with a 100-importance score, followed by the area of the segment denoted by A. (as large and very small segment tend to be discarded), as can be confirmed in Figure 6.7-C.

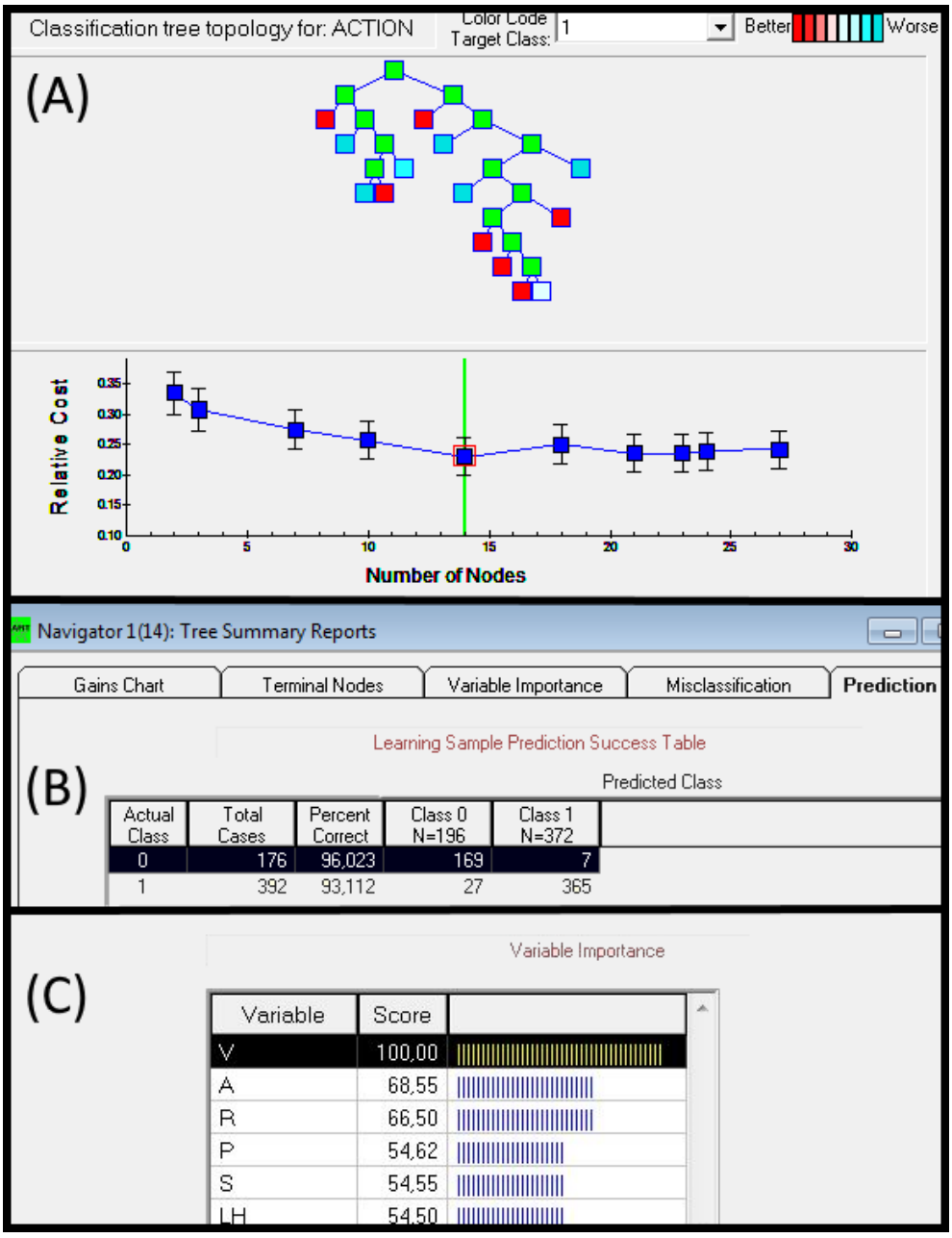


Figure 6.7 –Representation of the discard classifier. (A) The best tree is shown with the lowest relative cost. (B) the prediction accuracy s shown for the best classification tree; (C) the variable importance graph shows that variable V has the highest importance.

Figure 6.8-A shows the best suggested classification tree (with the minimum relative cost) for the merge classifier, with the chosen options (Gini as the splitting method, the same probability for both classes, 10-fold cross validation). This tree has 12 nodes (see rules in Annex A7) and has a classification accuracy of 93.201% for class 0 (keeping the segment) and 96.338% for class 1 (merge the segment), as can be confirmed in the confusion matrix in Figure 6.8--B. Finally, the variable that highest contribution to the classification algorithm is the ratio between intensity of the image in the contact zone and the pixel intensity in the contour of the segments (denoted by R), as It makes sense that segments that need to be merged, will share a similar intensity values in the shared contour zones, as can be confirmed in Figure 6.8-C.

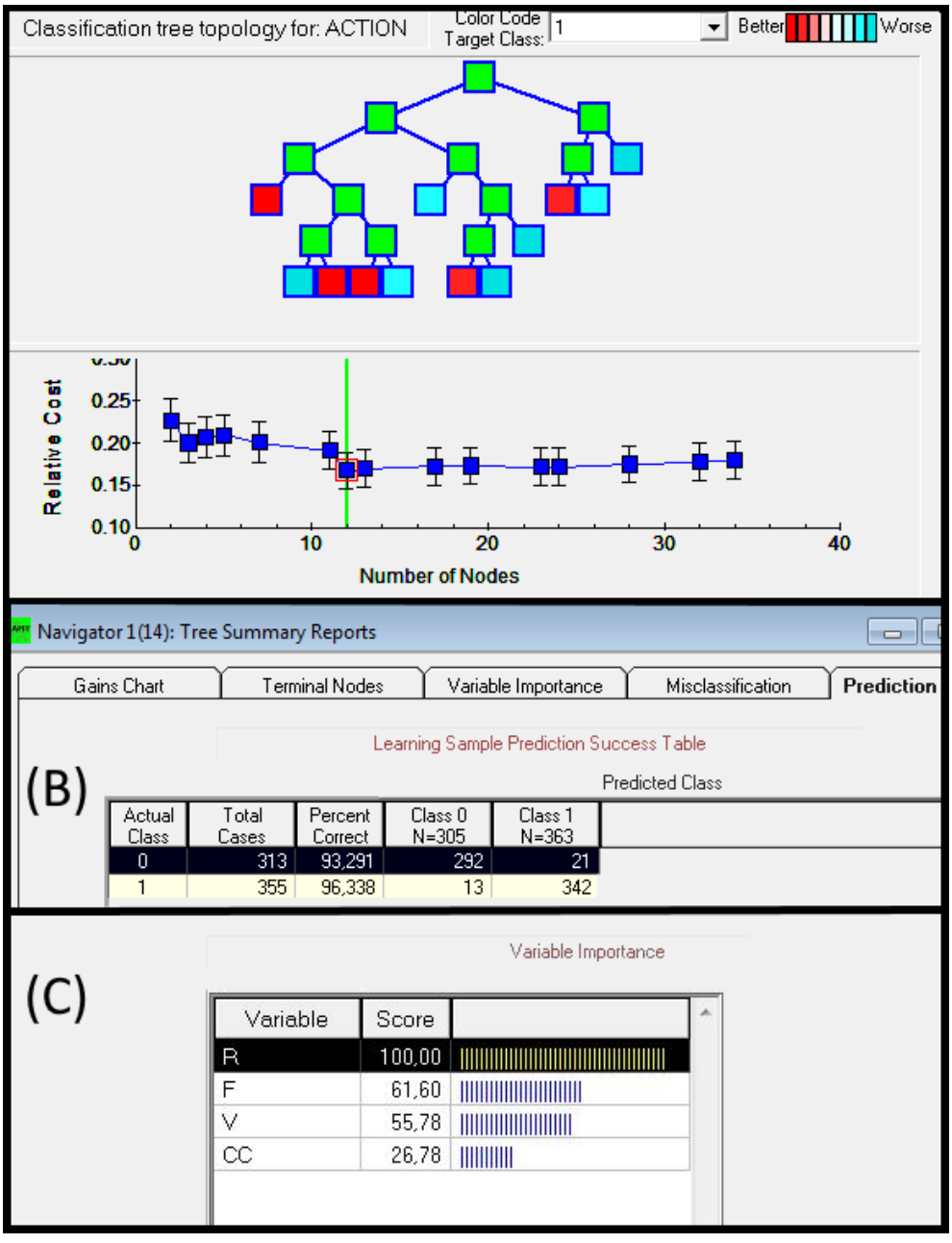


Figure 6.8 –Representation of the merge classifier. (A) The best tree is shown with the lowest relative cost. (B) the prediction accuracy s shown for the best classification tree; (C) the variable importance graph shows that variable R has the highest importance.

6.1.2.2. Evaluation of the cell segmentation algorithms

In this section, the validation of Cell Segmentation algorithm is based on one time-series of 31 minutes, starting with 116 *E. coli* cells and ending with 274 cells, for a total of 5730 analysed cells, (this time-series is also used to study the detection of the Nucleoids and the FtsZ Rings, as seen in section 6.1.4), taken at 37ºC. Since the segmentation is done on Phase-Contrast images, there are only 7 frames of cell segmentation, as each segmentation is aligned to several confocal images and the Phase-Contrast images are acquired every 5 minutes (as detailed in the section 4.1.1). The analysis is actually done after the registration of the Phase-Contrast images to the Confocal data-space (2048*2048 pixel resolution).

The first analysis, is based on the correct identification of the cells (presented in Table 6.4), using the implemented segmentation algorithms: 'Otsu + Median', 'GPL + CART' and the additional splitting and morphological functions added to these algorithms ('New Steps').

Using the manually correction procedure detailed in section 4.1.3, a validation dataset was created corresponding to a total of 1302 cells (116, 153, 128, 186, 213, 232, 274, respectively for each frame). Based on this validation set, it was computed which cells were correctly identified, cells that had to be split to be correctly identified (see the red arrows in Figure 6.9-A), cells that actually required to be merged and background segments that were still not cleaned.

From Table 6.4, it should be noticed that both algorithms (even before the added 'New steps') detected a small number of background segments (false negatives). The images in this time-series were acquired with a clean background (as it is possible to observe in Figure 6.9-C) and the environmental conditions were favourable (lower or higher temperatures, along with the introduction of chemicals, such as NaCl, can lead to artefacts present in the image). The algorithms were also built to discard segments that belong to the background (using thresholds and filters in the case of the 'Otsu + Median' and the discard classifier in the case of the 'GPL + CART').

Table 6.4 - Quantitative evaluation of the implemented segmentation algorithms at the cell detection level: 'Otsu + Median', 'GPL + CART' and the same algorithms with the addition of new steps based on splitting methods.

	'Otsu + Median'	**'GPL + CART'**	**'Otsu + Median'+ New Steps**	**'GPL + CART' + New Steps**	**Manual Correction**
Total Number of detected cells	1287	1287	1293	1305	1302
Correct number of identified cells	1264	1253	1276	1269	-
Cells that require splitting	18	22	12	14	-
Cells that require merging	4	10	4	10	-
Background Segments	1	2	1	2	
Accuracy	97.2%	96.2%	98.0%	97.5%	-

From the results in Table 6.4 it possible to observe that both algorithms had a high accuracy (more than 97%) and that the added steps were able to split 6 cells in the 'Otsu + Median' case and 8

cells in the 'GPL + CART' case. The added steps are not able to merge or remove background cells (but the segmentation algorithms already provide a reliable way to remove those from the analysis.

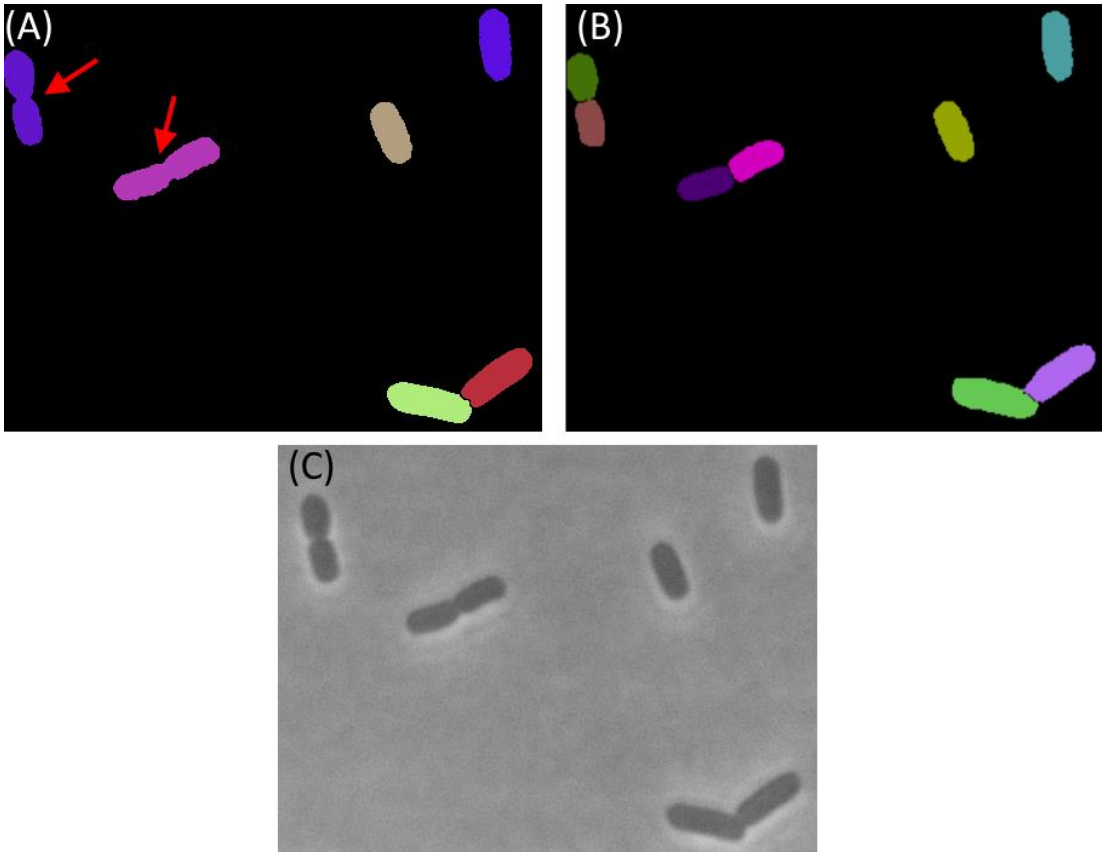


Figure 6.9 – Cell Segmentation results. In (A) the 'Otsu + Median' algorithm was not able to correctly separate the four cells indicated by the red arrow. In (B), the added 'New Steps' based on splitting algorithms allowed a correct splitting of the examples shown in (A). (C) shows the original Phase-Contrast Image, where the segmentation was applied

The analysis presented in Table 6.4 shows that the algorithms are able to identify correctly the cells, but as seen in Figure 6.9, the added steps can add or remove some pixels from the cells, so it was also necessary to do a pixel-based analysis, as shown in Table 6.5, and for this analysis it was decided to compare the best algorithm (namely the 'Otsu + Median', and including the added steps) with the manual corrected dataset. The total number of analysed pixels are 29360128 (2048*2048*7).

Table 6.5 - Quantitative evaluation of the implemented segmentation algorithms at the pixel level: 'Otsu + Median', 'GPL + CART' and the same algorithms with the addition of new steps based on splitting methods.

	'Otsu + Median'	**'Otsu + Median'+ New Steps**	**Manual Correction**
Number of true positive pixels (TP)	1031120	1036120	1057335
Number of true false pixels (TF)	28302793	28296052	28302793
Number of false positive pixels (FP)	52005	36425	-
Number of false negative pixels (FN)	26215	21215	-
Accuracy (%)	99.9%	99.9%	
Sensitivity (%)	97.5%	98.0%	-
Specificity (%)	99.8%	99.9%	-
Precision (%)	95.2%	96.6%	-
F1 score (%)	96.4%	97.3%	-

From the results in Table 6.5, it possible to observe that the 'Otsu + Median' provides reliable scores of sensitivity, specificity and precision (all over 97%). The added steps not only improved the correct identification of cells, by splitting some of the analysed cells (see Table 6.4), but also improved cell segmentation at a pixel level, by removing pixels at the border that were part of the background, by removing pixels at mid-cell during the splitting process, and adding a few pixel that were previously part of the background.

6.1.3. Cell Tracking Algorithms

As mentioned in section 4.3.3, the tracking algorithm is based on the assignment of the cell on one frame, which maximizes the percentage of area overlap, from the previous frame based on a Nearest-Neighbour approach with an Euclidian distance, similarly to the study in [253].

The cell tracking procedure were evaluated by manually inspecting the cell lineages along a timeseries. Table 6.6 shows the evaluation of a timeseries of four hours, with Phase-Contrast images being taken every 5 minutes (total of 49 images), with the numerical error quantification counted whenever a new id was assigned to a previously identified cell. These errors were previously shown in the lineage plot presented in Figure 4.14, highlighted with the red arrows (B and C). It is possible to observe that the proposed intramodal image registration method (see section 4.1.2.1 and 6.1.1.1) provided an improvement of the cell tracking process.

Table 6.6 - Quantification of the error percentages in cell tracking and division detections with and without intramodal registration at 37 ºC.

Image registration Method	Error Percentage (no. cells)
With Intramodal image registration	0% (115)
Without Intramodal image registration	3.478% (115)

Table 6.7 shows, for each temperature, the number of divisions and the error percentage in detecting these events. Such errors are detected when one bacterium has 3 (rather than 2) candidate children. From Table 6.7 it is possible to observe that the cell tracking method based on a simple nearest-neighbour algorithm was able to solve 99% of the lineages, when the Phase-Contrast images were previously aligned with the intra-modal image registration process (see section 6.1.1). It should be noted that times-series with even more cells and higher division times might require the use of other tracking algorithm, especially in highly clustered images, which can reduce drastically the efficiency of the nearest-neighbour algorithm (which led to the implementation of a nearest-neighbour algorithm that takes into account the morphology of the cell, as seen in section 4.3.3.

Table 6.7 - Quantification of the error percentages in cell tracking and division detections in each temperature condition using intramodal image registration.

Error Quantification	22 ºC	37 ºC	43 ºC	Total
Cell Tracking Error %	0%	0%	0.660%	0.506%
(Number of bacteria)	(162)	(115)	(909)	(1186)
Division Error %	0%	0%	0.331%	0.243%
(Number of divisions)	(58)	(52)	(302)	(411)

6.1.4. Nucleoid and FtsZ Ring Segmentation

In this section, the validation of the Structure Detection Algorithms (Gaussian and 'TreshMorph') is presented, based on one time-series of 31 minutes, starting with 116 *E. coli* cells and ending with 274 cells, for a total of 5730 analysed cells. The structures of interest in this example are the Nucleoids and the FtsZ Rings, which can be observed by the fusion of the fluorescent aggregates (HupA-mCherry and GFP, respectively) and these structures can be simultaneously observed (see Figure 6.10-A) by Confocal Microscopy (using the Red and Green channel, respectively).

Table 6.8 presents the statistical metrics for the nucleoid segmentation algorithms, while Table 6.9 presents the statistical metrics for the FtsZ protein, using the Gaussian Segmentation with different 'd' parameter values and 'TreshMorph' (TM) with different threshold (T) values for both tables. Table A.1 and Table A.2 present the Confusion Matrix Tables of the pixel segmentation analysis of the Nucleoids and FtsZ, respectively, for both algorithms and their input parameters.

Table 6.8 - Statistical metrics of the nucleoid segmentation algorithms. Results are shown for the Gaussian Algorithm with different 'd' parameter values and the 'TreshMorph' Algorithm (TM) with different threshold (T) values. Here 'mean' and 'std' represent the Mean and Standard Deviation of the pixel intensities inside each cell.

	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	F1 Score (%)	Detection Time (s)
TM (T = Global Otsu)	87.46	73.27	98.62	97.66	83.72	44.4
TM (T = ML Otsu - 2)	74.06	41.14	99.94	99.83	58.27	58.73
TM (T = mean)	86.81	72.56	98.01	96.63	82.88	51.3
TM (T = mean + 1/3 std)	90.19	97.25	84.64	83.27	89.72	42.50
TM (T = mean + 2/3 std)	92.01	93.49	90.85	88.93	91.15	47.41
TM (T = mean + 1 std)	91.64	87.16	95.16	93.40	90.17	43.13
TM (T = mean + 4/3 std)	89.46	79.02	97.66	96.37	86.84	48.45
TM (T = mean + 5/3 std)	86.22	70.03	98.96	98.14	81.74	66.17
Gaussian with d = 2	84.40	68.33	97.03	94.76	79.40	1461.8
Gaussian with d = 3	85.67	70.74	97.41	95.55	81.30	1777.7
Gaussian with d = 4	85.44	69.76	97.76	96.07	80.83	1737.0
Gaussian with d = 5	84.81	67.98	98.04	96.47	79.76	1759.8
Gaussian with d = 6	83.99	65.81	98.28	96.79	78.35	1860.3
Gaussian with d = 7	83.36	64.15	98.46	97.03	77.24	1858.6
Gaussian with d = 10	81.31	58.92	98.92	97.72	73.52	1831.8
Gaussian with d = 15	79.17	53.67	99.22	98.19	69.40	1749.0
Gaussian with d = 20	78.25	51.58	99.21	98.09	67.61	1735.2

From Table 6.8 it is possible to observe that the 'TreshMorph' (TM) using a threshold (T) based on the mean intensity was able to provide the best overall score. This value is calculated for each cell, so it can adapt to darker and lighter cells. Changing slightly the values based on the standard deviation (from mean + 1/3 std to T = mean + 1 std, it was possible to observe the large decrease on the sensitivity (from 97% to 87%), but a large increase in specificity and precision (from around 84% to 97%), which is probably caused by the drop in the intensity values near the border of the nucleoids, especially during the division process.

The Gaussian Algorithm had a considerably lower sensitivity, which lowered its overall F1 score, maintaining a high specificity and precision. The main advantage of this algorithm is that it provides a mathematical description of the structure of interest, which has been proved to be very

useful in the creation of models and description of the temporal and spatial organization of these structures of interest.

From Table 6.9 it is possible to observe that the 'TreshMorph' (TM) using a threshold (T) based on a multilevel Otsu methodology provided the best overall results (with an F1 score of 81.40%), although using a global Otsu threshold also provided a good overall score (with better precision, but lower sensitivity). The multilevel Otsu methodology calculated two different intensity thresholds and using the lowest threshold value for the intensity cut-off, proved to have the better results (as using the highest value had a huge decrease in the sensitivity, as it underestimated the borders of the distribution of the FtsZ Rings.

Table 6.9 - Statistical metrics of the algorithm of FtsZ Rings detection (Accuracy, Sensitivity, Specificity, Precision, F1 Score for one example time-series. Here 'mean' and 'std' represent the Mean and Standard Deviation of the pixel intensities inside each cell.

	Accuracy	Sensitivity	Specificity	Precision	F1 Score	Detection Time (s)
Gaussian (d=2)	89.82	50.26	96.93	74.61	60.06	1500.8
Gaussian (d=3)	90.11	51.21	97.10	76.03	61.20	1644.8
Gaussian (d=5)	90.34	53.77	96.92	75.82	62.92	1805.3
Gaussian (d=7)	90.33	55.64	96.57	74.45	63.69	1740.4
Gaussian (d=10)	90.29	59.40	95.84	71.99	65.09	1676.3
Gaussian (d=13)	90.15	61.13	95.37	70.35	65.41	1864.4
Gaussian (d=15)	90.09	62.98	94.97	69.22	65.95	1616.6
Gaussian (d=17)	89.91	63.13	94.73	68.30	65.61	1742.1
Gaussian (d=20)	89.73	64.56	94.26	66.92	65.72	1672.3
TM (Global Otsu)	93.80	70.51	97.99	86.30	77.61	73.3
TM (Multilevel Otsu - 2)	88.26	23.99	99.82	95.95	38.39	64.7
TM (Multilevel Otsu – 1)	94.42	80.19	96.97	82.65	81.40	67.2
TM (T = mean)	90.59	89.04	90.87	63.69	74.26	57.7
TM (T = mean - 1/6 std)	88.34	91.89	87.70	57.31	70.59	60.0
TM (T = mean + 1/6 std)	92.02	85.41	93.21	69.35	76.54	53.4
TM (T = mean + 2/6 std)	92.80	80.78	94.96	74.22	77.36	52.6
TM (T = mean + 3/6 std)	93.05	75.59	96.19	78.08	76.82	55.1
TM (T = mean + 4/6 std)	92.96	70.05	97.09	81.17	75.20	57.4

An example of the segmentation results is shown in Figure 6.10-B, the segmentation borders of the structures of interest (red line for nucleoids and green line for FtsZ rings) are done with the and the Multilevel Otsu – first level and the 'TreshMorph' Algorithm 'TreshMorph' Algorithm (T= mean fluorescence intensity of bacteria + 2/3 of standard deviation of fluorescence) respectively for the Nucleoids and FtsZ rings, as they provided the best results in our test case. It is possible to observe that, as detailed in the literature, when the cells are close to divide, the Nucleoid and the FtsZ proteins (in the last stage) aren't colocalized (characterized by large intensity values of the FtsZ proteins in the mid-cell and large intensity values of the nucleoids at the poles), while cells that are still growing have a large colocalization of the FtsZ proteins and the Nucleoids.

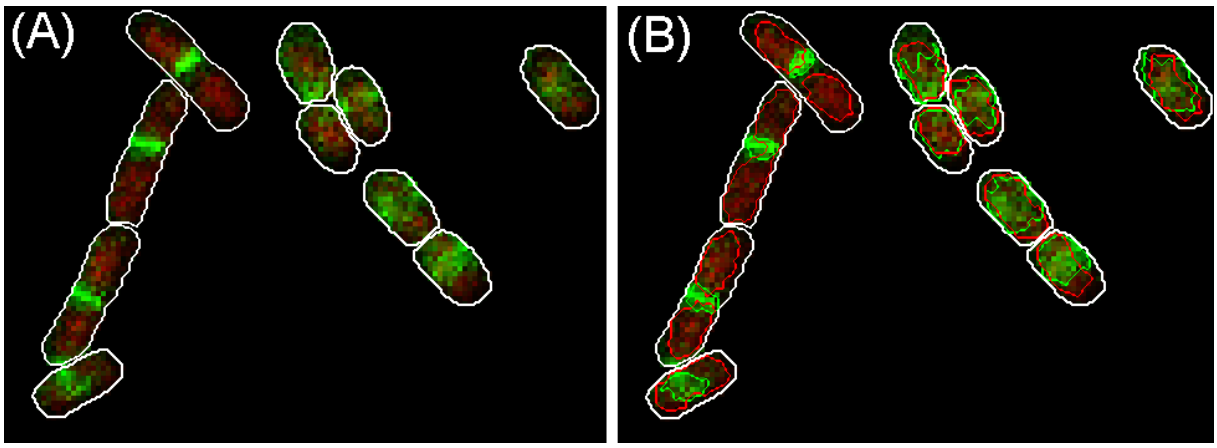


Figure 6.10 - Examples of simultaneous visualization of Nucleoids (in red colour) and FtsZ Rings (in green colour). Visualization (A) with no Segmentation (B) with 'TreshMorph' Segmentation (red lines for nucleoid segmentation and green line for FtsZ Rings).

6.1.5. MinD Protein Segmentation

For this example, one time series of 121 minutes was chosen to test the detection of the Min System, using Confocal Microscopy. This example started with 7 cells and finished with 14, for a total of 1318 cells analysed. Table 6.10 shows the statistical metrics for the MinD-GFP proteins segmentation algorithms, namely the Gaussian Segmentation with different 'd' parameter values and 'TreshMorph' (TM) with different threshold (T) values. Table A.3 shows the Confusion Matrix Tables for each case.

Table 6.10 - Statistical metrics of the algorithm of MinD proteins detection (Accuracy, Sensitivity, Specificity, Precision, F1 Score. Here 'mean' and 'std' represent the Mean and Standard Deviation of the pixel intensities inside each cell.

	Accuracy	Sensitivity	Specificity	Precision	F1 Score	Detection Time (s)
Gaussian (d=5)	75.52	29.98	90.48	50.84	37.72	245.0
Gaussian (d=10)	76.35	41.99	87.62	52.70	46.74	280.2
Gaussian (d=13)	76.45	43.84	87.16	52.85	47.93	279.4
Gaussian (d=14)	76.44	46.21	86.36	52.66	49.23	281.2
Gaussian (d=15)	76.50	46.12	86.47	52.83	49.25	270.6
Gaussian (d=16)	76.49	45.86	86.56	52.83	49.10	276.0
Gaussian (d=17)	76.35	45.21	86.57	52.51	48.59	257.6
Gaussian (d=19)	76.30	45.11	86.54	52.38	48.48	273.9
Gaussian (d=20)	76.17	45.46	86.25	52.06	48.54	278.6
Gaussian (d=25)	75.78	44.50	86.05	51.15	47.59	266.9
TM (Global Otsu)	78.50	96.0	72.75	53.63	68.81	15.05
TM (Multilevel Otsu-1)	89.46	58.35	99.68	98.34	73.24	16.5
TM (T = mean)	95.28	85.13	98.69	95.54	90.01	14.4
TM (T = mean – 1/6 std)	93.64	92.62	93.97	83.46	87.80	14.6
TM (T = mean – 2/6 std)	89.34	95.64	87.27	71.15	81.60	14.5
TM (T = mean + 1/6 std)	92.20	69.84	99.54	98.03	81.57	14.9
TM (T = mean + 2/6 std)	88.96	55.92	99.81	98.99	71.46	14.8

From Table 6.10 it is possible to observe that the 'TreshMorph' (TM) using a threshold (T) based on directly on the mean intensity was able to provide the best overall score. As previously mentioned, this threshold value is calculated for each cell, so it can adapt to darker and lighter cells. Slight lowering

and increasing of this threshold (by 1/6 of the standard deviation of the intensity values inside the cell) reduced the overall results as shown in Table 6.10, although the decreased in the threshold provided a slight increase in the Sensitivity. The results shown in Table 6.8, Table 6.9, Table 6.10 show the importance of allowing the user to choose between several input parameters, as depending on the structure of interests, the image acquisition system and the noise levels, each case can be optimized using different parameters.

6.1.6. Protein Aggregates (Spots) Detection

As mentioned in section 4.1.7, the spot detection methods implemented in the SCIP tool, were based on the methods developed in 'CellAging' [458] and 'iCellFusion' [453]. The SCIP tool integrates all existing methods, which differ in the filters that can be applied (Median, Kernel and Gaussian). Using *E. coli* cells expressing fluorescent protein aggregates composed of an RNA molecule tagged by MS2-GFP, all the implemented methods were tested for several single frame images at 37 °C (see results in Table 6.13). For this evaluation, the default parameters for each method were chosen (see the values in Figure A.10).

Table 6.11 - Quantitative evaluation of the spot detection filters (Median, Kernel, Gaussian) at 37 °C.

	Median	Kernel	Gaussian
Number of true positives (TP)	184	184	181
Number of false positives (FP)	11	20	19
Number of false negatives (FN)	1	1	3
Sensitivity	0,995	0,995	0.984
Precision	0,944	0.902	0.905
F1 score	0,968	0.946	0.943

Using the spot detection filter that gave the better results (Median Filter), we evaluated (see results in Table 6.12) the same RNA molecules tagged by MS2-GFP in different temperatures, which showed that the detection system and the constructed strain are robust to changes in temperature.

Table 6.12 - Quantitative evaluation of the spot detection method using the Median Filter at 22 °C, 37 °C and 43 °C.

	22 °C	37 °C	43 °C	Total
Number of true positives (TP)	410	184	174	768
Number of false positives (FP)	16	11	5	32
Number of false negatives (FN)	0	1	1	2
Sensitivity	1,000	0,995	0,994	0,997
Precision	0,963	0,944	0,972	0,960
F1 score	0,981	0,968	0,983	0,978

6.1.7. Inclusion Bodies Detection

To study the detection of inclusion bodies *E. coli* cells were exposed to osmotic stress as this type of stress leads to an increase in the amount of visible inclusion bodies (Oliveira et al., 2016). 3 conditions were tested: no stress, medium and high stress (0, 125 and 300 mM of NaCl). The phase-contrast images of cells under the osmotic stress were analysed for 60 minutes.

In this example the detection time wasn't tracked, as this is based on the initial seed placement of the seeds, which happens in every example. So instead of the detection time, the time spent on the decision to delete the seed is counted instead (the seed deletion procedure was detailed in section 4.1.6). Table 6.13 presents the calculation of the detection statistics of each example of stress.

Table A.4 (in annex) shows the equivalent confusion matrix of the 3 examples and the result of joining all examples).

Table 6.13 – Statistical metrics of the algorithm of inclusion body detection (Accuracy, Sensitivity, Specificity, Precision, F1 Score for 3 examples of low, medium and high stress and also the results from joining all examples.

	Accuracy	Sensitivity	Specificity	Precision	F1 Score	Removal Time (s)
Example 1	98.75 %	71.43 %	99.88 %	96.15 %	81.97 %	70.8
Example 2	97.91 %	88.94 %	99.73 %	98.53 %	93.49 %	120.3
Example 3	97.81 %	89.71 %	99.39 %	96.65 %	93.05 %	130.9
Total Average	98.01 %	88.51 %	99.59 %	97.25 %	92.68 %	322.0

The results of the seed detection and removal are quite satisfactory (most scores close to or above 90%), particularly in the conditions where cells have many inclusion bodies. The Sensitivity score was mostly affected by False Negatives, not detected by the GPL algorithm, rather than by the rejection of detected seeds. Thus, to improve Sensitivity in the future, one would have to adjust the GPL algorithm [482] or add a new algorithm specialized in the detection of the missing inclusion bodies. It is noted that for this research work and similarly to the analysis of the spot detection algorithm, this analysis was based on the detection of inclusion bodies and not on the segmentation of inclusion bodies, since the until now, the biological studies only required the identification of the number of inclusion bodies

It should be noted that the SCIP toolbox allows to manually add new seeds simply by clicking once on the inclusion body, which allows for quick improvements, even in images with large number of cells. It is also noted that the SCIP toolbox allows the manual drawing of inclusion bodies.

6.1.8. Singe-cell and population-level colocalization

This example used 2 different channels, as specified in the Section "Bacterial strain, growth conditions, and induction". RNAp-GFP molecules are shown in the green channel and HupA-mCherry-tagged nucleoids in the Red channel with a total of 776 analyzed cells 776 cells at 30 ^{o}C.

This example is used to show which features can be extracted based on the fluorescence intensity levels inside each cell (in each channel). The first presented feature is the fluorescence along the normalized Major and Minor axis of each cell (see Figure 6.11), which can be used to calculate the Pearson Correlation Coefficient (PCC) along each axis (see Table 6.14).

The second feature is the plotting of each pixel fluorescence intensity on the first channel versus its intensity on the second channel (e.g. see Figure 6.12), that can be used to calculate the Manders Coefficients (see Table 6.14).

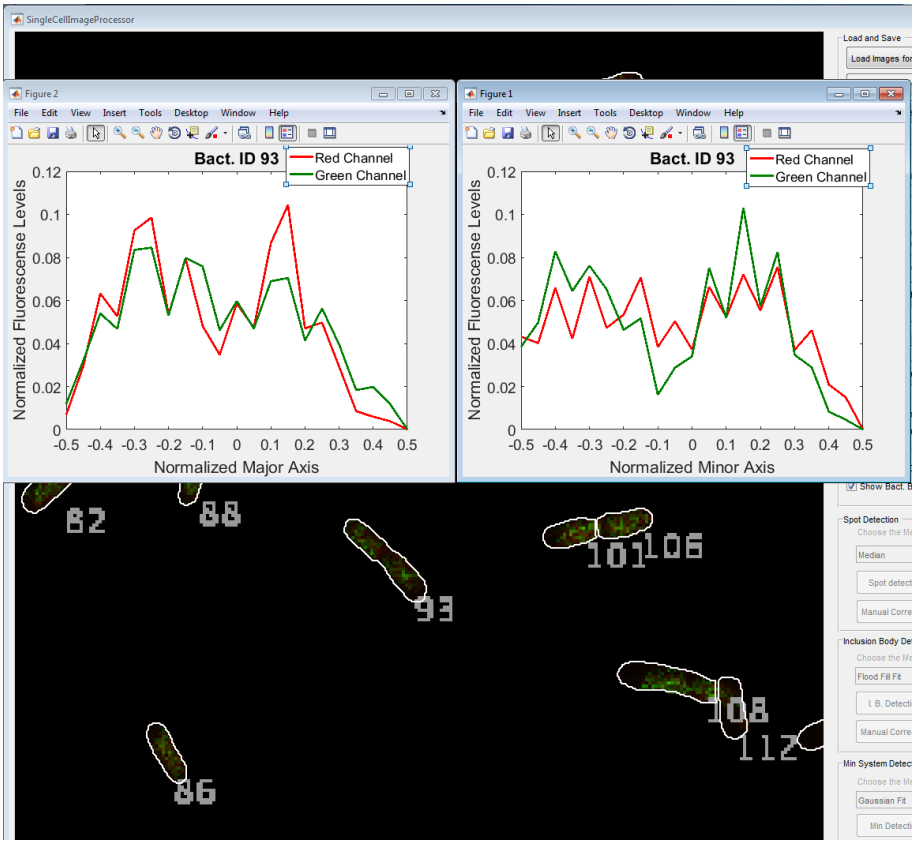


Figure 6.11 - Example of single-cell co-localization of bacteria Nucleoid and RNAp (cell with ID 93).

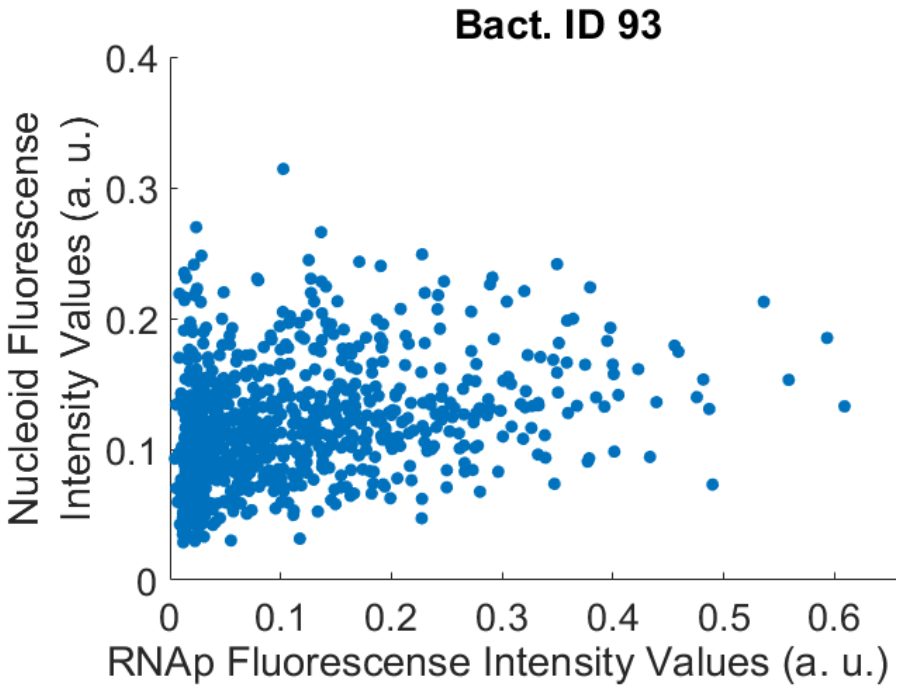


Figure 6.12 - RNAP fluorescence intensity versus nucleoid fluorescence intensity values of Bacteria with ID 93.

The last features to be calculated are the Pearson Correlation Coefficients and the Manders Coefficients (see Table 6.14 for a demonstration of examples based on the cells with ID 86, 93, 101 and 106).

Table 6.14 – Pearson Correlation Coefficient (PCC) between RNAp fluorescence and Nucleoid fluorescence in each cell, along the Major and Minor Axis of the specific cells. The Manders Coefficients were also calculated (M1 and M2 correspond to the Nucleoid and the RNAp, respectively as the reference channel).

	Bacteria ID 86	Bacteria ID 93	Bacteria ID 101	Bacteria ID 106
Global PCC (p-value)	0.4213 (5.5×10-25)	0.3959 (5.1×10-38)	0.5747 (1.5×10-47)	0.4316 (6.6×10-26)
Minor Axis PCC (p-value)	0.951 (3.4×10-11)	0.867 (3.6×10-7)	0.956 (1.4×10-11)	0.982 (2.5×10-13)
Major Axis PCC (p-value)	0.968 (1.3×10-9)	0.928 (1.3×10-9)	0.981 (4.9×10-15)	0.933 (6.5×10-10)
Manders Coefficients (M1)	0.447	0.445	0.5081	0.486
Manders Coefficients (M2)	0.667	0.654	0.7312	0.616

Finally, the cell space (major and minor axes) is normalized by calculating the center coordinates of each cell and applying the Principal Component Analysis algorithm (Abdi and Williams, 2010). The cell space along the major axis is divided in 10 normalized bins.

Note that the normalized major axis bins (see Figure 6.13) are ranged from 0 to 0.5 (divided in 10 bins) because the poles are not known (since this is not a time-series). Each cell is "folded" in half, and the sum is done from the bins starting at the cell center ('0' in the x-axis) to both poles ('0.5' in the x-axis). The fluorescence is also normalized by dividing the intensity inside each bin by the total intensity inside each cell (so that the sum of all bins will be 1).

For this example, comprised of 776 *E. coli* cells at 30 ^{o}C, both the normalized fluorescence levels of RNAp molecules (Figure 6.13-A) and nucleoids (Figure 6.13-B) are plotted over the Normalized Major Axis.

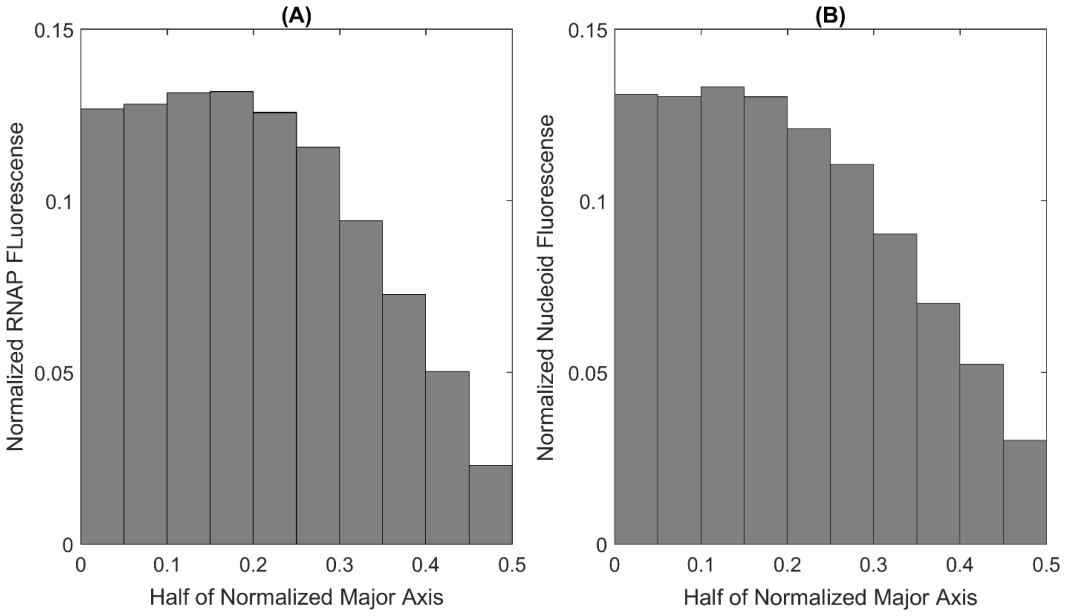


Figure 6.13 - RNAP and nucleoid fluorescence along the major cell axis. (A) RNAP fluorescence along the major cell axis (binned) as normalized by the total mean fluorescence of the cells (RL1314 strain). (B) Normalized average nucleoid fluorescence intensity distribution along the normalized major axis of the cells (RL1314 strain). Measurements are from 776 cells at 30 ^{0}C in both cases. In both figures, in the x axis, '0' corresponds to the cell center, while '0.5' corresponds to both extremities (cells folded in half, with unknown poles).

6.1.9. Global Performance Analysis

In a time-series, it is important to analyse the performance of the detection algorithms over the acquisition time, especially in cases where the signal-to-noise ratio starts to degrade due to loss of fluorescent probes after a division event or due to difficulties with the acquisition setup (like maintaining focus and laser power).

A global temporal analysis of the performance scores was performed on Example 1 (31-minute time-series with Nucleoids and FtsZ proteins) and Example 2 (121-minute time-series with MinD proteins), to check the presence of signal degradation and subsequent performance drop. The analysis of nucleoids, FtsZ proteins and MinD proteins are displayed respectively in Figure 6.14-A, B and C.

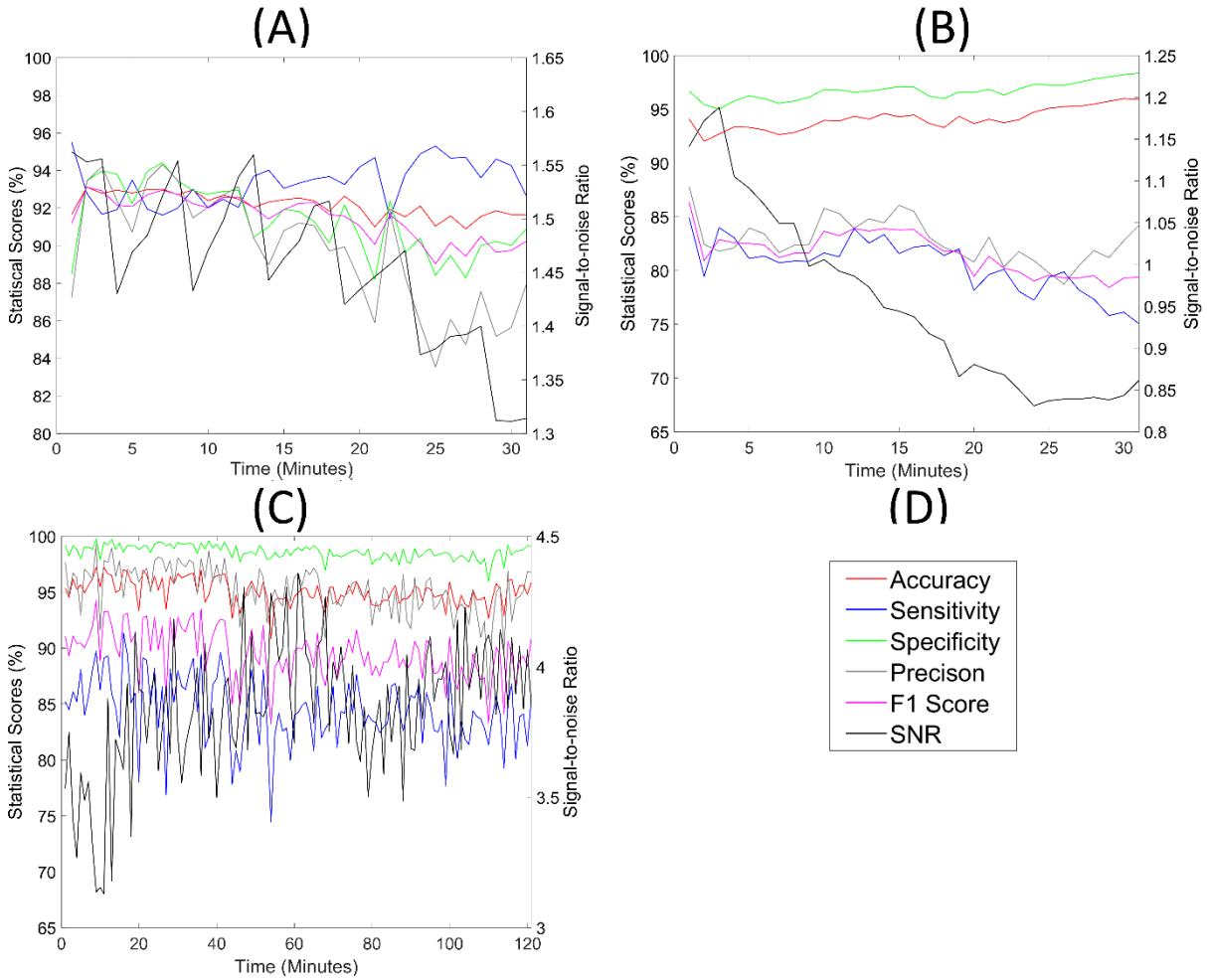


Figure 6.14 – Temporal analysis of the best segmentation algorithm scores. (A) Nucleoid Segmentation - 'TreshMorph' with T = mean + 2/3 standard deviation; (B) for FtsZ ring segmentation - 'TreshMorph' with T= Multilevel Otsu – first level; (C) for MinD proteins segmentation - 'TreshMorph' with T = mean (D) legend of the statistical scores for all figures. Also included is the mean SNR (right axis) for each frame.

The average of the signal-to-noise ratio (SNR) was also included in Figure 6.14-A, B and C for each structure of interest, respectively. The SNR is calculated by averaging (for each frame) the division of the mean fluorescence intensity by its standard deviation (for each cell). Interestingly, the SNR values tend to slightly decrease (especially on the last frames) on both structures of Example 1 (Nucleoids and FtsZ rings), while the SNR of Example 2 shows a tendency to increase. Because of this, the statistical scores are more stable in Example 2, than on Example 1. Example 1 has a less stable curve (especially the Precision Score for the Nucleoids and the Sensitivity for the FtsZ rings). One

interesting effect is the oscillation in the SNR of the Nucleoids, which is caused by cell division, since our segmentation is done every 5 frames, which means that divisions are only detected every 5 frames.

Finally, a global performance analysis of all Algorithms present in this tool is performed based on the supervised evaluation scores [472], namely Accuracy, Sensitivity, Specificity, Precision and F1-score. The algorithms, presented in Table 6.15 include the structure segmentation and detection (Nucleoids, FtsZ Rings, Min System, Cell borders, Inclusion bodies and spots) and cell tracking.

The consolidated benchmark analysis of all algorithms that have better overall scores is presented in Table 6.15. Most of the presented scores in Table 6.15 are between 95% and 85%, which is adequate for most of the biological applications, like studying the influence of different environmental conditions to different structures of interest (e.g. [117], [483], [484]). The lower Sensitivity and Precision values (especially for the detection of FtsZ Rings) are a result of the large morphological changes during the cell lifetime of that cellular structure [165] and are also related to the decrease of the signal-to-noise ratio detected for this structure (see Table 6.15).

Table 6.15 – Average benchmark results of automatic detection algorithms for the different structures present in *E. coli* cells. In Cell Tracking, it is only possible to calculate the Accuracy, because only True Positives and False Positives can be calculated. In Cell Segmentation, the scores were based on the pixel-level analysis and not on the cell-level detection.

Algorithm	Accuracy	Sensitivity	Specificity	Precision	F1 Score	Best Algorithm
Cell Border Segmentation	99.9%	98.0%	99.9%	96.6%	97.3%	'Otsu and Median' + Added Steps
Cell Tracking	99.5%	-	-	-	-	Nearest Neighbour
Spot Detection	98.2%	99.7%	97.2	96.0%	97.8%	Median Filter
Nucleoids Segmentation	92.0%	93.5%	90.9%	88.9%	91.1%	'TresMorph' Algorithm (T= mean + 2/3 of standard deviation of the bacterial fluorescence intensity)
FtsZ Rings Segmentation	94.4%	80.2%	97.0	82.7%	81.4%	'TresMorph' Algorithm (Multilevel Otsu – first level)
Min System Segmentation	95.3%	85.1%	98.7%	95.5%	90.0%	'TresMorph' Algorithm (T= mean bacterial fluorescence intensity)
Inclusion Bodies Detection	98.0%	88.5%	99.6%	97.3%	92.7%	GPL Algorithm + Tailored Seed Selection

It is important to mention that the SCIP tool allows the user to manually correct the segmentation and detection results if a higher sensitivity is required. Due to this, it is believed that an algorithm that can adapt to each of the stages observed during the cell lifetime, which could be done by using Machine Learning Techniques (similarly to the ones implemented in the next section) would be able to improve the statistical scores of the FtsZ Ring Detection Algorithms.

6.1.10. FtsZ Ring Classification

The FtsZ Ring stage classification procedures were all done are done using Machine Learning packages present in MATLAB™. The Decision Trees were created using the '*fitctree*' function, the Support Vector Machines were created using the '*fitcsvm*' and the Regularized Multinomial Logistic

Regression (RMLR) were created using the '*logitMn*' function, which is part of the Pattern Recognition and Machine Learning Toolbox. The dataset had 250 cases of the initial and intermediate FtsZ Ring stages, and 250 cases of the final stage, making it a balanced binary dataset. No cost of misclassification was introduced into the algorithms.

As mentioned in the evaluation of the performance of each ML Algorithm was done by calculating the accuracy (the ratio of correctly classified samples to the total number of samples averaged over the folds) of each method with a 10-fold cross-validation , which randomly partitions the data into 10 subsets, and trains the Algorithm with 9 subsets and evaluates the performance on the last subset (a process that is then repeated 10 times). The accuracy results, presented in Figure 6.15, are based on repeating this validation process 100 times and averaging the result.

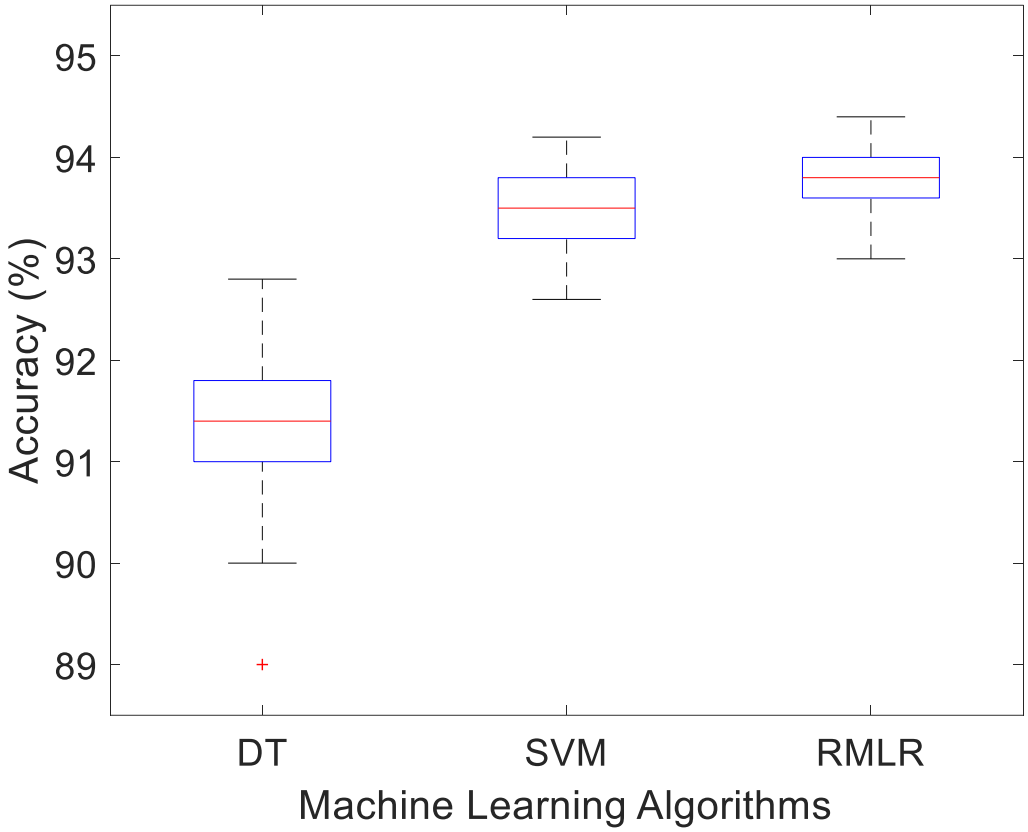


Figure 6.15 – Box Plot with the accuracy percentage of 100 runs, calculated for each Machine Learning Algorithm

In the Decision trees case, different input parameters were used, such as the Split criterion: Gini's diversity index, the twoing rule and the maximum deviance reduction and the algorithms used to select the best split predictor at each node: Standard CART algorithm, Curvature test and the Interaction test. The best results were obtained with the Gini's diversity index and the standard Cart split predictor [485]. The Pruning criterion was not changed during these tests.

In the Support Vector Machines case, different input parameters were used, such as using changing the Kernel scale parameter to 'auto' (changing it from '1'), forcing the software to select an appropriate scale factor using a heuristic procedure. The Kernel function was changed from 'linear' to 'polynomial', with different order numbers, a flag to standardize the predictor data as also used. The best results were obtained with the settings with 'auto', 'linear' and the standardization flag. No other parameters were tested.

In the Regularized Multinomial Logistic Regression (RMLR), there were only one parameter changed from the default values: the regularization parameter (lambda) [486], that was changed from a range of [$1e^{-6}$, $1e^{-2}$], with the default value of $1e^{-4}$, the one that gave the best results.

As can be observed in Figure 6.15, the average accuracy values of the FtsZ stage classification algorithms (when the first two stages were joined together) were 91.4% (DT), 93.5% (SVM) and 93.7% (RMLR), respectively, which improved drastically from the values obtained in [167]: 73.0% (DT), 76.9% (SVM) and 79.4% (RMLR), respectively.

This difference was mainly explained due to lowering of the complexity of the problem (2 classes versus 3 classes) and the increase in the available labelled samples (500 versus 300). The new classification procedure was considered satisfactory and allowed the automatic analysis of 16441 *E. coli* cells and identify the cells with 2 separate nucleoid and FtsZ Rings in the last stage of development [483].

6.2. Image Generator Validation

In this Section, three different machine learning algorithms were tested to show applicability of the Image Generator Toolbox for cell tracking purposes: Instance-based learning (simple Nearest-Neighbour and Nearest-Neighbour with Morphology) and the DBSCAN Clustering algorithm. The generated images have a 1000x500 pixel size (first and second experiment) and 1500x1000 (third experiment).

In this Section, a False Positive (FP) is counted when one object is incorrectly tracked from one frame to another and a True Positive (TP) is accounted when one object is tracked correctly between two consecutive frames. It is important to note that errors that occur in the beginning of a time series are typically propagated through the entire sequence.

All the tables in the next Sub-sections present the estimation of the tracking errors, which are based on false discovery rate, calculated as FP/(FP+TP). The presented values are based on the mean tracking errors from 100 time-series with a total of 100 frames for each time-series, for each of the examples tested.

The results presented here have been expanded from the work done by Pedro Canelas during his Master Thesis [466] and from the results presented in [465], [467].

6.2.1. Simple Nearest-Neighbour

The tracking performance of the Simple NN algorithm is presented in Table 6.16. The tracking performance is based on the calculation of the tracking error (in percentage) based on the ground-truth produced by the image generator and is calculated on every frame and accumulated until the end of the time-series. For the example presented in Table 6.16, the morphology shape-related factor called was set to 0.05 (this value was chosen to emulate biologically inspired objects that slowly change their shape over time, such as bacterial cells).

The results from Table 6.16 (9600 time-series of 100 frames) show that this simple algorithm can handle the increase in the number of objects while keeping a small velocity, and that when raising the velocity to 20 and 30 the tracking performance was significantly reduced.

Table 6.16 - Tracking errors (in percentage) of the Simple Nearest-Neighbour Algorithm.

Obj.	V=5	V=10	V=15	V=20	V=25	V=30
10	0,00	0,51	0,53	2,49	5.25	9,34
20	0,00	0,92	1,06	4,19	10.43	19,20
30	0,20	1,11	2,29	5,41	13.54	21,24
40	0,26	1,27	3,23	5,93	19.63	24,01
50	0,27	1,40	3,90	9,13	22.33	30,45
60	0,06	1,58	5,63	12,38	25.11	39,66
70	0,26	1,64	6,01	14,00	28.37	41,19
80	0,24	1,84	6,62	15,74	30.12	45,06
90	0,29	1,90	7,40	18,19	32.44	48,32
100	0,27	1,20	7,85	19,94	33.78	49,76
110	0,25	1,68	9,40	20,32	35.58	50,26
120	0,22	1,69	10,57	21,16	36.95	51,86
130	0,40	3,02	12,11	24,05	38.12	55,83
140	0,55	3,71	14,16	26,57	40.37	58,07
150	0,44	3,79	14,88	29,73	45.99	61,70
160	0,42	4,12	14,91	33,74	50.28	63,89

6.2.2. Nearest-Neighbour with Morphology

In this second experiment, the morphology of the object is considered along with the centre of the object, as detailed in section 4.3.3. Table 6.17 presents the results of the tracking performance of the NNm Algorithm. Here, 10 time-series of 100 frames were also created for each example with different objects, different maximum velocity and distinct morphology factors.

The Nearest-Neighbour with Morphology Algorithm was tested in two configurations; the first giving a 60% importance to the calculated distance between objects (α factor in equation 4.20) and 40% to the calculated morphology difference (β factor) and the second with 40% for α and 60% for β.

Table 6.17 - Tracking errors (in percentage) of the Nearest-Neighbor with Morphology Algorithm.

	α = 60% and β=40%									
	m factor= 0.05					**m factor= 0.1**				
Obj.	**V=5**	**V=10**	**V=15**	**V=20**	**V=30**	**V=5**	**V=10**	**V=15**	**V=20**	**V=30**
10	0.00	0.24	0.00	1.86	6.07	0.00	0.00	0.41	0.55	8.22
20	0.00	0.92	0.00	2.27	11.28	0.00	0.00	1.43	0.37	14.17
30	0.06	1.00	1.52	2.47	14.36	0.00	0.15	1.68	5.42	19.20
40	0.00	0.46	2.45	3.36	18.10	0.00	0.33	2.00	8.71	21.06
50	0.00	0.98	3.05	6.63	24.44	0.00	0.33	3.33	8.42	25.66
60	0.06	1.08	3.20	8.82	30.61	0.34	0.31	4.10	8.27	27.12
70	0.15	1.26	4.03	9.98	31.71	0.15	0.44	5.11	10.55	30.80
80	0.24	1.43	4.66	11.00	34.27	0.00	0.88	5.40	13.76	34.87
90	0.22	1.51	5.74	12.15	36.83	0.00	1.25	5.77	14.41	35.11
100	0.27	1.47	6,02	14.71	41,60	0.20	1.96	6.03	17.79	40,55
110	0.24	1.55	6.21	14.88	41.84	0.31	1.86	7.11	17.90	42.94
120	0.00	1.10	6,27	14,92	42,05	0.13	1.74	9.24	19,68	44,45
130	0.15	1.80	8.59	17.07	45.97	0.20	1.99	9.31	20.15	48.20
140	0.22	2.19	9,29	18,34	48,96	0.27	2.66	8.34	21.59	48,90
150	0.24	2.60	10.15	22.67	53.02	0.34	2.88	9.59	24.20	55.02
160	0.13	3.32	10,32	25,49	55,35	0.19	3.03	10.26	25.39	55.32

	α = 40% and β=60%									
	m factor= 0.05					m factor= 0.1				
Obj.	**V=5**	**V=10**	**V=15**	**V=20**	**V=30**	**V=5**	**V=10**	**V=15**	**V=20**	**V=30**
10	0.00	0.00	0.00	0.41	4.89	0.00	0.00	0.00	0.50	4.21
20	0.00	0.66	0.00	2.63	6.99	0.00	0.00	1.43	0.02	8.90
30	0.00	0.81	0.29	2.74	8.99	0.00	0.18	1.46	3.73	11.39
40	0.00	0.46	1.50	3.16	14.92	0.00	0.48	0.45	5.52	15.66
50	0.08	0.69	1.75	5.58	19.52	0.14	0.43	1.96	6.41	21.51
60	0.06	0.84	1.62	6.66	24.02	0.34	0.02	2.41	7.13	22.69
70	0.18	1.17	2.73	7.01	25.15	0.00	0.45	3.82	8.11	24.47
80	0.24	1.37	3.10	7.30	26.65	0.00	0.65	4.00	9.90	27.80
90	0.22	1.03	3.99	9.12	28.67	0.30	0.82	4.10	10.52	30.53
100	0.19	0.84	4,26	10,57	33,37	0.20	1.07	3,96	12,07	32,44
110	0.23	1.06	4.33	10.69	34.06	0.21	1.14	5.68	13.16	35.62
120	0.00	0.84	4,53	10,80	33,95	0.13	0.79	6,39	14,07	37,09
130	0.24	0.95	5.02	13.81	35.41	0.31	1.40	6.06	14.33	38.18
140	0.18	0.89	6,36	14,58	39,30	0.25	1.61	5,34	15,18	41,36
150	0.21	1.25	6.62	18.21	42.40	0.29	1.96	7.30	17.29	47.54
160	0.13	1.88	7,27	20,78	46,81	0.19	2.03	8,06	18,93	49,38

The impact of the shape-related factor was also studied using both 0.05 and 0.1. For this section, the previously published results were extended to include lower velocities and less objects when comparing our analysis against the simple NN algorithm [465]. From Table 6.17, it is possible to observe that tracking results are improved by using the NNm Algorithm (e.g. in the worst case scenario the error percentage was reduced from 64% to 47%) for the m factor =0.05 case, but at lower velocities, the Simple NN algorithm achieves similar results (compared with NNm) even with a large number of objects.

It is important to remark that, as most of bacterial cells in live-cells imaging are placed in agarose gel, where they do not move very fast, but they are able to grow and create large clusters of cells, which explains how the Simple Nearest-Neighbour was able to produce scores of correct lineage tracking of over 95% in the example shown in section 6.1.3. When cells have faster movement capabilities, other tracking algorithms need to be used and compared. It should also be noted that the second configuration (40% for α and 60% for β) gave better results than the first one, so giving more importance to the morphology factor, improved the results (comparing the results for the same number of objects and same velocities). Results might still be improved by using different configurations of the α β parameters, so this optimization will be one of the future endeavours of this research work.

6.2.3. Cluster Tracking

The 'Create Clusters' property was used to test the same tracking algorithms (Simple NN and NN with Morphology Algorithms with α=40%). The simulated parameters were number of clusters (1, 5 and 10), number of objects per cluster (10 and 15), maximum velocity (5 and 10), Alternative Movement, Centre Force (4) and morphology factor (0 and 0.05).

The tracking results are presented in Table 6.18 and Table 6.19, respectively for Simple and Morphology NN Algorithms, respectively. For the Cluster creation, we used 10 time-series (and

averaged the results) of 200 frames and calculated the object tracking error on every frame accumulated throughout the time-series.

Table 6.18 - Tracking errors (in percentage), within clusters with different properties, using the Simple Algorithms with different number of clusters (1 to 10), different number of objects per cluster (2 to 15), and different maximum velocities (2, 5, 10) and different morphology factors (0 and 0.005).

Simple NN Algorithm							
Nº of Clusters	Obj. / Clusters	m factor= 0			m factor= 0.05		
		V=2	V=5	V=10	V=2	V=5	V=10
1	2	1.14	0.99	1.78	1.59	1.82	11.76
	5	2.95	0.58	2.62	1.93	2.32	14.23
	8	3.16	5.88	18.39	2.03	6.44	19.71
	10	3.32	7.79	30.42	0.93	9.88	23.33
	13	3.96	10.97	42.00	2.18	10.08	31.52
	15	4.63	11.74	50.91	2.94	10.74	38.06
3	2	0.02	1.23	4.75	0.00	2.70	10.08
	5	0.05	2.40	7.69	0.00	4.57	9.01
	8	0.81	4.35	16.80	1.40	8.12	23.46
	10	1.07	7.11	27.83	2.20	9.07	30.08
	13	2.64	11.51	36.39	2.59	12.94	39.30
	15	3.05	14.74	43.77	2.76	16.77	45.44
5	2	0.00	1.84	4.70	0.06	1.30	4.77
	5	0.23	2.22	6.25	0.72	3.28	9.74
	8	0.45	3.06	10.80	1.17	6.11	24.17
	10	0.70	7.48	34.71	1.57	10.95	31.89
	13	2.39	13.16	35.59	2.82	15.94	34.64
	15	3.06	17.43	45.22	3.53	16.06	44.51
7	2	0.26	1.46	7.77	0.60	1.06	8.72
	5	0.58	2.55	12.78	1.04	1.95	14.52
	8	1.34	6.33	20.59	1.64	7.55	20.72
	10	1.58	11.21	33.76	1.81	11.78	40.35
	13	2.82	16.49	40.20	3.24	16.61	44.50
	15	3.14	19.81	48.55	4.01	17.75	48.96
10	2	0.39	1.85	8.27	0.15	3.12	9.27
	5	0.99	3.39	17.81	0.25	5.40	17.13
	8	1.67	7.46	24.56	1.25	7.81	30.65
	10	1.95	12.20	38.26	1.52	11.64	42.47
	13	3.01	17.67	40.64	3.79	18.69	49.02
	15	3.84	21.14	53.90	4.87	23.52	57.34

If when inside a cluster, there are more objects in ‘t’ then in ‘t-1’, these ‘extra’ objects are labelled as ‘Possible Entry’. If there are fewer objects, they are labelled ‘Possible Exit’. This tagging is temporary and compares the "Possible Exit" features to the features of all other objects of the frame t-1, linking it to a "Possible entry" in another cluster (meaning that it left one cluster to join another), classifying it as noise, or as an object leaving the image.

Table 6.19 - Tracking errors (in percentage), within clusters with different properties, using the Nearest Neighbour Algorithm with Morphology (α = 40% and β=60%) with clusters (1 to 10), different number of objects per cluster (2 to 15), and different maximum velocities (2, 5, 10) and different morphology factors (0 and 0.005).

NN with Morphology (α = 40% and β=60%)							
Nº of Clusters	Obj. / Clusters	m factor= 0			m factor= 0.05		
		V=2	V=5	V=10	V=2	V=5	V=10
1	2	0.00	0.00	0.00	0.08	0.02	1.94
	5	0.00	0.00	0.00	1.93	0.00	7.92
	8	0.05	0.89	2.59	2.90	4.49	11.98
	10	0.01	1.27	4.88	3.04	5.52	13.83
	13	0.14	2.06	13.41	3.07	5.62	17.43
	15	0.18	3.76	21.14	1.75	4.63	20.76
3	2	0.00	0.14	1.47	0.00	0.29	2.31
	5	0.38	1.08	1.66	0.00	1.23	4.08
	8	0.97	1.12	7.30	0.06	1.59	11.64
	10	1.18	1.26	10.33	0.03	2.13	12.52
	13	1.36	4.47	15.07	0.69	5.83	18.29
	15	1.81	5.29	20.24	1.92	8.12	22.44
5	2	0.00	1.09	1.76	0.00	0.56	2.03
	5	0.00	1.78	2.58	0.10	0.68	5.77
	8	0.52	1.94	8.77	0.15	2.33	10.11
	10	0.71	1.80	12.98	0.15	4.69	15.93
	13	1.02	5.43	18.15	0.53	5.59	18.48
	15	1.54	7.16	20.77	0.93	5.95	22.07
7	2	0.00	0.32	1.46	0.24	0.31	1.91
	5	0.20	0.41	2.82	0.41	0.35	5.13
	8	0.54	2.97	8.52	0.47	1.58	9.97
	10	0.78	3.92	15.08	0.48	3.60	17.84
	13	1.19	5.56	21.29	1.49	5.61	24.09
	15	1.22	8.14	25.78	1.99	6.86	27.11
10	2	0.00	0.90	3.01	0.00	0.87	2.92
	5	0.04	0.97	6.93	0.15	2.31	7.20
	8	0.13	2.54	9.99	0.41	3.59	13.15
	10	0.48	3.78	16.15	0.54	4.55	19.71
	13	0.91	6.04	22.73	1.72	8.83	28.52
	15	1.11	8.73	28.36	2.22	10.13	34.12

From Table 6.18, it is possible to observe that the simple NN cannot handle clusters adequately (for V=10, m factor = 0.05 and 160 objects/cluster, there exists a 4,12% error while for V=10, m factor = 0.05, 10 clusters and 15 objects/cluster, for a total of 150 objects we have a 57.34% error rate). Comparing those results with Table 6.19, it is possible to observe that the NNm algorithm handles much better the cluster creation, giving almost one half of the errors (worst case scenario of 34.12% versus 57.34% for the same configuration).

The main difference between DBSCAN 1 and DBSCAN 2 algorithms is that, in the first case, the classification is done after the tracking and in the second it is done before the tracking, equalizing the number of objects between the clusters. Results from both DBSCAN Algorithms are presented in Table 6.20 (m factor =0.00) and Table 6.21 (m factor =0.05) and show no major differences between both DBSCAN algorithms.

From the comparison of Table 6.19 with Table 6.20 and Table 6.21, it is possible to notice that DBSCAN Algorithm does not provide a significant improvement over the NNm algorithm for large cluster numbers, since the DBSCAN algorithm tries to separate each cluster in every frame. Therefore, if the number of clusters is the same between the actual frame and the previous one (t and t-1), then their results are matched with the usage of the NNm algorithm, treating them as isolated objects using their centroids for the calculation. If the number of clusters changes, the first step is skipped and the number of objects inside each cluster is checked.

The biggest advantage of the DBSCAN algorithms was observed with 1 clusters, where the worst case scenario of 15 objects/cluster gives a score of around 21% versus the 13% errors obtained with the DBSCAN 1 and 2, respectively for a m factor of 0, while the values were of 20.76% for the NNm and around 10% for the DBSCAN with a m factor of 0.05. For 3 clusters, the DBSCAN had errors of around 16% and 19% (m factor of 0 and 0.05 respectively), while the NNm reported errors of around 20 and 22%. With more than 3 clusters, the DBSCAN and the NNm reported similar values.

Table 6.20 - DBSCAN1 (DB1) and DBSCAN1 (DB2) tracking errors (in percentage) comparison for different number of clusters, objects per cluster, and maximum velocities, with m factor =0.

mmd = 0.00							
		Vmax=2		Vmax=5		Vmax= 10	
Clusters	Objects/ Cluster	DB1	DB2	DB1	DB2	DB1	DB2
1	2	0.00	0.00	0.00	0.00	0.00	0.05
	5	0.00	0.00	0.00	0.00	0.16	0.16
	8	0.00	0.00	1.93	1.68	1.75	1.67
	10	0.00	0.00	5.55	4.67	2.97	2.97
	13	0.09	0.10	3.99	3.84	5.69	5.54
	15	0.24	0.24	1.92	2.59	12.94	12.96
3	2	1.94	1.95	3.62	3.54	5.70	5.83
	5	3.10	2.47	5.76	6.01	8.81	8.72
	8	2.46	2,24	5.12	5.09	10.60	10.42
	10	0.70	1.02	4.86	4.89	11.28	10.70
	13	0.74	0.90	3.99	3,94	13.46	13,55
	15	0.83	0.83	3.86	3.86	16.44	16.34
5	2	2.46	2.55	1.79	1.74	10.78	10.69
	5	5.01	5.20	2.46	2.58	15.87	16.80
	8	3.17	3.04	4.38	4.46	14.04	14.09
	10	1.44	1.04	5.43	5.54	13.71	14.56
	13	1.09	1.02	5.74	5.57	16.88	16.90
	15	0.27	0.27	5.84	5.74	19.29	19.35
7	2	1.04	1.04	3.17	3.24	3.00	3,11
	5	2.05	1.89	4.82	5.29	6.37	6.49
	8	2.17	2.12	4,92	4.74	13.24	13.26
	10	2.47	2.23	4.52	4.81	16.18	16.51
	13	1.99	1,88	6,73	6.74	21.66	21.95
	15	0.83	0.83	8.44	8.60	25.45	25.36
10	2	1.88	1.67	7.44	7.25	6.07	5.93
	5	3.15	2.82	9.50	9.03	11.90	12.11
	8	2.99	3.05	9.05	8.98	14.36	14.28
	10	2.45	3.33	5.81	6.00	17.55	17.57
	13	2.67	2.73	7.78	7.86	21.42	21.17
	15	1.24	1.24	8.65	8.65	28.29	28.29

A strange behaviour for lower velocities was identified in both DBSCAN algorithms, where increasing the objects decreased the tracking errors. This behaviour is explainable by the higher movement restriction of objects belonging to clusters with larger number objects, but further studies are required to further analyse this behaviour. This behaviour has not been identified in the simple NN and NNm algorithms.

Table 6.21 - DBSCAN1 (DB1) and DBSCAN1 (DB2) tracking errors (in percentage) comparison for different number of clusters, objects per cluster, and maximum velocities, with m factor =0.05.

mmd = 0.05							
		Vmax=2		Vmax=5		Vmax= 10	
Clusters	Objects/ Cluster	DB1	DB2	DB1	DB2	DB1	DB2
1	2	0.08	0.12	0.68	0.66	8.68	9.02
	5	0.96	0.96	1.67	1.67	9.75	9.75
	8	0.26	0.24	4.33	4,38	9.54	9.41
	10	0.00	0.00	9.64	9.64	9.14	7.82
	13	0.38	0.34	5.24	5.40	9.55	9.34
	15	0.85	0.85	3.87	3.87	10.49	10.49
3	2	0.00	0.00	3.37	3.30	10.75	10.54
	5	0.00	0.00	5.06	5.02	13.97	14.89
	8	3.13	3.07	4.05	4.09	11.29	12.46
	10	5.45	5.43	3.06	2.81	9.91	9.83
	13	4.88	4.92	3.83	3.83	15.44	15.39
	15	1.99	1.99	3.78	3.77	19.55	19.63
5	2	0.98	0.91	5.44	5.40	13.05	12.69
	5	2.18	2.69	10.72	11.23	20.79	22.28
	8	2.08	1.97	8.52	8.63	20.75	21.52
	10	1.94	2.33	6.42	7.55	16.61	17.37
	13	1.47	1.46	7.12	7.03	19.42	18.68
	15	0.79	0.79	6.49	6.19	20.84	20.82
7	2	2.53	2.36	6.30	6.05	12.91	12.82
	5	4.07	4.20	7.54	7.43	13.78	12.98
	8	3.90	3.87	6.10	6.17	18.03	18.11
	10	3.37	4.06	4.63	5.46	18.29	18.54
	13	3.59	3.25	7.22	6.98	24.45	23.55
	15	2.00	2.00	7.06	7.22	27.59	27.70
10	2	1.73	1.79	7.07	7.41	11.09	11.71
	5	2.02	1.90	8.33	8.41	13.55	14.38
	8	2.27	2.30	7.69	8.75	18.42	18.33
	10	1.81	2.30	6.42	6.43	21.07	21.42
	13	2.24	2.32	8.51	9.19	26.01	25.89
	15	2.76	2.67	9.91	9.98	34.52	34.52

6.3. Dissemination of Results

The results presented in the previous two sections were published in several publications, as shown in Table 6.22, for the relevant publications in journals,

Table 6.**23** for the relevant publications in Book Chapters and Table 6.24 for the relevant publications in Conference Proceedings. The main role relevant to the research work is also presented. Table 6.24 also presents the relevant Workshops and Courses that were attended during this research work.

Table 6.22 – Dissemination results of this research work in Journals and my roles in the publications

Publication	Role
Samuel M. D. Oliveira, Ramakanth Neeli-Venkata, Nadia S. M. Goncalves, João A. Santinha, **Leonardo Martins**, Huy Tran, Jarno Mäkelä, Abhishekh Gupta, Marilia Barandas, Antti Häkkinen, Jason Lloyd-Price, José M. Fonseca Andre S. Ribeiro. 2016. "Increased Cytoplasm Viscosity Hampers Aggregate Polar Segregation in *Escherichia Coli*." *Molecular Microbiology* 99 (4): 686–99. https://doi.org/10.1111/mmi.13257.	Participated in the development of the nucleoid segmentation algorithm
Leonardo Martins, Ramakanth Neeli-Venkata, Samuel M. D. Oliveira, Antti Häkkinen, Andre S. Ribeiro, and José M. Fonseca. 2018. "SCIP: A Single-Cell Image Processor Toolbox." *Bioinformatics* bty505 (June). https://doi.org/https://doi.org/10.1093/bioinformatics/bty505.	Developed the image processing toolbox
Ramakanth Neeli-Venkata, Samuel Oliveira, **Leonardo Martins**, Sofia Startceva, Mohamed Bahrudeen, Jose M. Fonseca, Marco Minoia, and Andre S. Ribeiro. 2018. "The Precision of the Symmetry in Z-Ring Placement in *Escherichia Coli* Is Hampered at Critical Temperatures." *Physical Biology* 15 (5): 1–10. https://doi.org/https://doi.org/10.1088/1478-3975/aac1cb.	Participated in the development of the FtsZ stage classification algorithms and the in the image processing steps
Oliveira, Samuel MD, Nadia SM Goncalves, Vinodh K. Kandavalli, **Leonardo Martins**, Ramakanth Neeli-Venkata, Jan Reyelt, Jose M. Fonseca, Jason Lloyd-Price, Harald Kranz, and Andre S. Ribeiro. "Chromosome and plasmid-borne P LacO3O1 promoters differ in sensitivity to critically low temperatures." Scientific reports 9, no. 1 (2019): 4486.	Participated in the image processing and statistical analysis steps

Table 6.23 – Dissemination results of this research work in Book Chapters

Publication	Role
João Santinha, **Leonardo Martins**, Antti Häkkinen, Jason Lloyd-Price, Samuel M. D. Oliveira, Abhishekh Gupta, Teppo Annila, Andre Mora, Andre S. Ribeiro, and Jose Ribeiro Fonseca. 2015. "ICellFusion: Tool for Fusion and Analysis of Live-Cell Images from Time-Lapse Multimodal Microscopy." In *Biomedical Image Analysis and Mining Techniques for Improved Health Outcomes*, edited by Wahiba Ben Abdessalem Karâa and Nilanjan Dey, 71–99. IGI Global. https://doi.org/10.4018/978-1-4666-8811-7.ch004.	Participated in the development of the cell segmentation and image registration methods.
Leonardo Martins, Pedro Canelas, André Mora, Andre S. Ribeiro, and José Fonseca. 2018. "Generator Platform of Benchmark Time-Lapsed Images Development of Cell Tracking Algorithms: Implementation of New Features Towards a Realistic Simulation of the Cell Spatial and Temporal Organization." In Simulation and Modeling Methodologies, Technologies and Applications. SIMULTECH 2016. Advances in Intelligent Systems and Computing,	Expanded the results from the conference paper and introduced new features, such as cell division.

Vol 676., edited by M. Obaidat, T. Ören, and Y. Merkuryev, 52–74. Springer, Cham. https://doi.org/10.1007/978-3-319-69832-8_4.	

Table 6.24 – Dissemination results of this research work in Conferences and Practical Courses

Publication	Role
EMBO Practical Course, 'Microscopy, Modelling and Biophysical Methods', Heidelberg, Germany	Training in image processing techniques and in simulations techniques, discussion of Thesis Plan and presentation of Poster
11th International Workshop on Computational Systems Biology (11th WCSB), Lisbon, Portugal	Participation as Local Organizer and Abstract published in Conference Proceedings
2015 IEEE 4th Portuguese Meeting on Bioengineering (ENBENG)	Full Paper of preliminary resuls was approved for Oral Presentation and published in Conference Proceedings
DoCEIS 2015 - Doctoral Conference on Computing, Electrical and Industrial Systems	Participation as Local Organizer and presentation of Poster of Thesis Plan
Leonardo Martins, Jose M Fonseca, and Andre S Ribeiro. 2015. "'miSimBa' - A Simulator of Synthetic Time-Lapsed Microscopy Images of Bacterial Cells." In Proceedings - 2015 IEEE 4th Portuguese Meeting on Bioengineering, ENBENG 2015, 1–6. https://doi.org/10.1109/ENBENG.2015.7088854.	Developed all the features in the artificial image generator
Nadia S. M. Goncalves, **Leonardo Martins**, Huy Tran, Samuel M. D. Oliveira, Ramakanth Neeli-Venkata, Jose M. Fonseca, and Andre S. Ribeiro. 2016. "In Vivo Single-Molecule Dynamics of Transcription of the Viral T7 Phi 10 Promoter in *Escherichia Coli*." In *The 8th International Conference on Bioinformatics, Biocomputational Systems and Biotechnologies (BIOTECHNO 2016)*, 9–15.	Participated in the image processing steps
M. Zare, R. Neeli-Venkata, **L. Martins**, S. Peltonen, Ruotsalainen, U., and Andre S. Ribeiro. 2017. "Automatic Classification of Z-Ring Formation Stages at the Single Cell Level in *Escherichia Coli* by Machine Learning." In 4th International Conference on Bioimaging (BIOIMAGING 2017), Book ISBN: 978-989-758-215-8, Porto, Portugal.	Helped in the development of the initial classification algorithms
Canelas, Pedro, **Leonardo Martins**, André Mora, Andre S. Ribeiro, and José Fonseca. 2016. "An Image Generator Platform to Improve Cell Tracking Algorithms - Simulation of Objects of Various Morphologies, Kinetics and Clustering." In Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, 44–55. SCITEPRESS - Science and Technology Publications.	Helped in the development of the initial version of the image simulation toolbox

Chapter 7. Conclusion and Future Work

This section presents the conclusions related to the proposed main research questions and the main hypothesis. The secondary research questions are also answered and discussed based on the implemented research methodologies. Future endeavours related to this research work are also proposed, suggesting how both the image processing and the image simulation toolboxes can be further improved.

7.1. Main Conclusions

It is expected that future endeavours in single-cell biology will continue to focus on the heterogeneity of the spatial-temporal organisation of intracellular components and on better understanding of the combined stochastic functioning of multiple cellular processes, continuing the demand for the development of tailored image processing algorithms that are able to capture the variability of single cell, over the study of a bacterial population.

Going back to the main research question: 'How to design a toolbox capable of simulating models capable of reproducing of realistic morphological and functional experiments of bacterial time-lapsed microscopy images?' To do this, all the available models of the spatial and temporal bacterial cell organization were compiled, along with the morphological and functional descriptions of these processes leading to the development of an image simulation toolbox named 'miSimBa' (Microscopy Image Simulator of Bacterial Cells), which simulated images that reproduced the spatial and temporal organization of *E. coli* cells by modelling realistically cell morphology (shape, size and spatial arrangement), cell growth, cell division and cell motility.

A second platform, named 'Image Tracking Generator', which was mainly developed by Pedro Canelas during his Master Thesis, was implemented with a new feature of cluster creation, and allowed a more generic bacterial cell growth. The validation of the 'Image Tracking Generator' toolbox was made with manual inspection and allowed the creation of 46800 benchmarked datasets of 100 frames in different conditions in order to confirm the features that were implemented. The image simulation toolbox was used to evaluate three tracking algorithms (Simple Nearest-Neighbour, Nearest-Neighbour with Morphology and two variations of the DBSCAN Algorithm), due to its ability of creating specific cell clusters.

The obtained results showed that, for cases with lower maximum velocity, the Simple NN Algorithm was able to track objects even with a significant increase in the number of objects, which validates how the Simple Nearest-Neighbour was able to track *E. coli* cells with accuracy results over 95%, after the intra-modal registration process, as these cells were placed in agarose gel, which limits their movement. It was shown though that in large clusters, even with low maximum velocities, the use of an algorithm such as Nearest-Neighbour with Morphology or the DBSCAN would provide better results. The DBSCAN algorithm showed better performance for a lower number of clusters, while for a the larger amount of cluster, both algorithms (NNm and DBSCAN) performed equally.

The additional research question: 'Which models of biological processes need to be extracted using an Image processing toolbox, in order to create a realistic simulation of the cell spatial and temporal organization?' was answered by implementing an image processing toolbox, aided with machine learning algorithms was in order to study structures of interests that were not previously studied at that level (single-cell and single-molecule).

The developed toolbox, called 'Single Cell Image Processing' (SCIP) [455] was capable of providing assistance to present and future single-cell biology studies, contributing to an increased understanding of relationships between parallel dynamic cellular processes due to its modular-based multi-tasking abilities, allowing multiple structures to be analysed simultaneously, using multi-modal image processing techniques and providing the possibility of characterization of the dynamics of these specific cellular processes: cell division, growth, motility and gene expression), which can then be used to create novel biophysical models that can be introduced in the image simulator. The validation of the SCIP toolbox was still done using a manually labelled 'ground truth' benchmark, which was one the costliest task in terms of time spent during this research work, showing the importance of the development of the simulation toolbox to alleviate significantly this manual task.

The SCIP toolbox was tailored to be used by non-specialists in image analysis or computer science, by performing automatic intra-modal and inter-modal image registration techniques, but also allowed a control-point manual-aided registration process, if the automatic process failed. It also performs cell segmentation using two previously developed algorithms, one based on Otsu thresholding and the Median filter, and the second based on the Gradient Path Labelling (GPL) algorithm and the use of merge and discard classification algorithms to remove or keep the over-segmented objects that results from the implementation of the GPL algorithm. Additional steps were added to both algorithms to further improve the cell segmentation algorithm, by splitting cells using a technique based on the Watershed and distance transform, and the formation of the convex hulls of the objects. The added steps also improved the segmentation algorithm at the pixel-level. A cell tracking algorithm based on a simple Nearest-Neighbour approach was also integrated, to create cell lineages in studies requiring the analysis of time-series.

To segment cellular structures such as the Nucleoids, FtsZ proteins and MinD proteins, two different algorithms were developed, one based on the Gaussian Distribution and one based on the implementation of different Thresholding methods and parameters followed by morphological structuring functions, which was named 'TreshMorph'. The 'TreshMorph' algorithm, with different Thresholding parameters, performed better for each structure of interest than the Gaussian-based algorithm. It is noted that although the Gaussian-based algorithm had lower segmentation scores, it can still be very useful in the development of mathematical models that can describe the spatial and temporal organization of the structures of interest. The GPL algorithm was also used to create seeds for the segmentation of inclusion bodies in Phase-Contrast images. A seed rejection algorithm had to be developed in order to reject the large number of seeds suggested by the GPL algorithm. A previously developed spot detection algorithm was also integrated into the SCIP toolbox. The implemented algorithms were already used in different biological studies, such as the task of identifying the cells with 2 separate nucleoid and characterizing FtsZ Rings in the last stage of development [483].

In summary, the main conclusion of this research work is that both the image processing and the image simulation toolboxes provide a fundamental framework to the support of high-throughput experiments, based on single-cell, single-molecule imaging.

7.2. Future Work

Although the manual validation step was one of the costliest in terms of hours spent during this research work, this work should be alleviated with the continued development of more features in the image simulation toolbox. The main plan is to continue supporting high-throughput experiments of single cell imaging using reliable automated image processing methods and implementing new methods when necessary, increasing the computation speed and enhancing the design of the user experience.

Similar studies to what were presented in [469] can now be extended to proteins of the Min System (MinC, MinD, and MinE) or even other relevant proteins present in the divisome, such as ZapA, ZapB and ZipA. This would allow an even more profound knowledge of the spatial and temporal of the division process, especially at different environmental conditions, such as different temperature, pressure or stress.

It is expected that the simulation toolbox can help future endeavours in the development of new tracking algorithms, cell and structure segmentation algorithms, as it can produce huge amounts of benchmark images that can be used to validate these algorithms without the need of a manually produced benchmark dataset. To do this it will be necessary to introduce a new module that Is capable of generating secondary bodies inside the primary objects, simulating internal cell organelles and structures. It is also necessary to simulate different acquisition systems to generate the unique features of morphological and functional microscopy images, such as texture, signal to noise ratio, illumination problems, etc.

It is also necessary to keep using Machine Learning algorithms in biological studies, as they can classify cellular objects or can be used as a data-mining tool to extract information from large image datasets. Due to this, a Machine Learning module will be added to the image processing toolbox in order to allow an easier access to these algorithms to biology experts, along with a manual segmentation procedure to correct the segmentation of internal structures.

Most of the developed tools can be extended to studies of other bacterial species and even simple prokaryotic cells. To do this, it is necessary to study the differences in cellular processes, the differences of the external features if the available tools can be used to capture those differences or if new methods need to be developed.

Future applications should also be made available as a web-based framework to improve usability from other experts, and to avoid compatibility issues as it would allow an easier access to biology experts to test the implemented methods on their acquired images, even without a deep knowledge of image processing techniques.

References

[1] M.-H. Sung and J. G. McNally, “Live cell imaging and systems biology.,” *Wiley Interdiscip. Rev. Syst. Biol. Med.*, vol. 3, no. 2, pp. 167–82, 2011.

[2] M. Vera, J. Biswas, A. Senecal, R. H. Singer, and H. Y. Park, “Single-Cell and Single-Molecule Analysis of Gene Expression Regulation,” *Annu. Rev. Genet.*, vol. 50, pp. 267–291, 2016.

[3] D. Fusco *et al.*, “Single mRNA Molecules Demonstrate Probabilistic Movement in Living Mammalian Cells,” *Curr. Biol.*, vol. 13, no. 2, pp. 161–167, Jan. 2003.

[4] I. Golding and E. C. Cox, “RNA dynamics in live Escherichia coli cells.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 101, no. 31, pp. 11310–11315, Aug. 2004.

[5] I. Golding, J. Paulsson, S. M. Zawilski, and E. C. Cox, “Real-time kinetics of gene activity in individual bacteria.,” *Cell*, vol. 123, no. 6, pp. 1025–1036, Dec. 2005.

[6] A. Gupta, J. Lloyd-Price, S. M. D. Oliveira, R. Venkata, and A. S. Ribeiro, “In vivo kinetics of segregation and polar retention of MS2-GFP-RNA complexes in Escherichia coli,” *Biophys. J.*, vol. 106, pp. 1926–1937, 2014.

[7] A. Gupta, J. Lloyd-Price, S. M. D. Oliveira, O. Yli-harja, M. Anantha-Barathi, and A. S. Ribeiro, “Robustness of the division symmetry in Escherichia coli and functional consequences of symmetry breaking,” *Phys. Biol.*, vol. 11, no. 6, p. 066005, 2014.

[8] E. J. Stewart, R. Madden, G. Paul, and F. Taddei, “Aging and death in an organism that reproduces by morphologically symmetric division.,” *PLoS Biol.*, vol. 3, no. 2, p. e45, Feb. 2005.

[9] E. Meijering, “Cell Segmentation: 50 Years Down the Road,” *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 140–145, 2012.

[10] N. Bonnet, “Some trends in microscope image processing,” *Micron*, vol. 35, no. 8, pp. 635–653, Jan. 2004.

[11] K. Kruse, “Bacterial Organization in Space and Time,” in *Comprehensive Biophysics*, vol. 7, 2012, pp. 208–221.

[12] P. Hunter, “The paradox of model organisms,” *EMBO Rep.*, vol. 9, no. 8, pp. 717–720, 2008.

[13] B. Wanner, A. Finney, and M. Hucka, “Modeling the E. coli cell: The need for computing, cooperation, and consortia,” in *Systems Biology, Definitions and Perspectives. Topics in Current Genetics, Vol. 13*, vol. 13, no. May, H. V. W. L. Alberghina, Ed. Springer-Verlag Berlin Heidelberg, 2005, pp. 163–189.

[14] M. Hucka *et al.*, “Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML) project,” *Syst. Biol. IEE Proc.*, vol. 1, no. 1, pp. 41–53, Jun. 2004.

[15] M. Kröger, “IECA - International E.coli Alliance - E.coli Database Portal,” 2010. [Online]. Available: http://www.uni-giessen.de/ecoli/IECA/index.php.

[16] M. T. Cabeen and C. Jacobs-Wagner, “Bacterial cell shape,” *Nat. Rev. Microbiol.*, vol. 3, no. 8, pp. 601–10, Aug. 2005.

[17] B. Alberts *et al.*, *Essential cell biology*. Garland Science, 2013.

[18] M. Salton and K. Kim, “Chapter 2. Structure,” in *Medical Microbiology. 4th edition.*, S. Baron, Ed. Galveston (TX): University of Texas Medical Branch at Galveston, 1996.

[19] S. H. Zinder and M. Dworkin, “Chapter 1.7 - Morphological and Physiological Diversity,” in *Prokaryotes*, M. Dworkin, S. Falkow, E. Rosenberg, K.-H. Schleifer, and E. Stackebrandt, Eds. New York, NY: Springer New York, 2006, pp. 185–220.

[20] A. L. Koch, “What size should a bacterium be? A question of scale.,” *Annu. Rev. Microbiol.*, vol. 50, pp. 317–48, Jan. 1996.

[21] F. J. Trueba, E. A. Van Spronsen, J. Traas, and C. L. Woldringh, “Effects of Temperature on the Size and Shape of Escherichia coli Cells,” *Arch. Microbiol.*, vol. 131, no. 3, pp. 235–240, 1982.

[22] J.-V. Höltje, “Cell Walls, Bacterial,” in *The Desk Encyclopedia of Microbiology*, 1st Editio., 2003, pp. 239–250.

[23] U. Henning, K. Rehn, and B. Hoehn, “Cell envelope and shape of Escherichia coli K12.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 70, no. 7, pp. 2033–6, Jul. 1973.

[24] R. Carballido-López and A. Formstone, “Shape determination in Bacillus subtilis.,” *Curr. Opin. Microbiol.*, vol. 10, no. 6, pp. 611–6, Dec. 2007.

[25] J.-V. Höltje, “Growth of the Stress-Bearing and Shape-Maintaining Murein Sacculus of Escherichia coli,” *Microbiol. Mol. Biol. Rev.*, vol. 62, no. 1, pp. 181–203, 1998.

[26] K. C. Huang, R. Mukhopadhyay, B. Wen, Z. Gitai, and N. S. Wingreen, “Cell shape and cell-wall organization in Gram-negative bacteria,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 105, no. 49, pp. 19282–19287, 2008.

[27] K. A. Datsenko and B. L. Wanner, “One-step inactivation of chromosomal genes in Escherichia coli K-12 using PCR products,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 97, no. 12, pp. 6640–5, 2000.

[28] A. Chien, N. S. Hill, and P. A. Levin, “Cell Size Control in Bacteria,” *Curr. Biol.*, vol. 22, no. 9, pp. 1–23, 2013.

[29] U. Henning, “Determination of cell shape in bacteria,” *Annu. Rev. Microbiol.*, 1975.

[30] J. D. Wang and P. A. Levin, “Metabolism, cell growth and the bacterial cell cycle,” *Nat. Rev. Microbiol.*, vol. 7, no. 11, pp. 822–7, Nov. 2009.

[31] K. D. Young, “Bacterial shape: two-dimensional questions and possibilities.,” *Annu. Rev. Microbiol.*, vol. 64, pp. 223–40, Jan. 2010.

[32] G. Lan, C. W. Wolgemuth, and S. X. Sun, “Z-ring force and cell shape during division in rod-like bacteria,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 104, no. 41, pp. 16110–5, Oct. 2007.

[33] J. Fan, K. Tuncay, and P. J. Ortoleva, “Chromosome segregation in Escherichia coli division: a free energy-driven string model,” *Comput. Biol. Chem.*, vol. 31, no. 4, pp. 257–64, Aug. 2007.

[34] K. R. Anderson, N. H. Mendelson, and J. C. Watkins, “A new mathematical approach predicts individual cell growth behavior using bacterial population information.,” *J. Theor. Biol.*, vol. 202, no. 1, pp. 87–94, Jan. 2000.

[35] D. Lauffenburger, “Effects Of Cell Motility And Chemotaxis On Microbial Population Growth,” *Biophys. J.*, vol. 40, no. December, pp. 209–219, 1982.

[36] N. S. Hill, P. J. Buske, Y. Shi, and P. A. Levin, “A Moonlighting Enzyme Links Escherichia coli Cell Size with Central Metabolism,” *PLOS Genet.*, vol. 9, no. 7, p. e1003663, Jul.

2013.

[37] H. C. Berg, *E. coli in Motion*. Springer-Verlag Berlin Heidelberg, 2004.

[38] M. J. Tindall, P. K. Maini, S. L. Porter, and J. P. Armitage, “Overview of mathematical approaches used to model bacterial chemotaxis II: bacterial populations,” *Bull. Math. Biol.*, vol. 70, no. 6, pp. 1570–607, Aug. 2008.

[39] G. H. Wadhams and J. P. Armitage, “Making sense of it all: bacterial chemotaxis.,” *Nat. Rev. Mol. Cell Biol.*, vol. 5, no. 12, pp. 1024–37, Dec. 2004.

[40] N. Mittal, E. O. Budrene, M. P. Brenner, and A. Van Oudenaarden, “Motility of Escherichia coli cells in clusters formed by chemotactic aggregation.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 100, no. 23, pp. 13259–63, Nov. 2003.

[41] E. R. Zhang, L. F. Wu, and S. J. Altschuler, “Envisioning migration: mathematics in both experimental analysis and modeling of cell behavior.,” *Curr. Opin. Cell Biol.*, vol. 25, no. 5, pp. 538–42, Oct. 2013.

[42] F. H. C. Crick, “Central Dogma of Molecular Biology,” *Nature*, vol. 227, no. 5258, pp. 561–563, 1970.

[43] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, 4th editio. Garland Science, USA, 2002.

[44] D. L. Nelson, A. L. Lehninger, and M. M. Cox, *Lehninger principles of biochemistry*. Macmillan, 2008.

[45] S. Busby and R. H. Ebright, “Promoter structure, promoter recognition, and transcription activation in prokaryotes.,” *Cell*, vol. 79, no. 5, pp. 743–6, Dec. 1994.

[46] C. B. Harley and R. P. Reynolds, “Analysis of E. coli promoter sequences.,” *Nucleic Acids Res.*, vol. 15, no. 5, pp. 2343–2361, 1987.

[47] P. L. deHaseth, M. L. Zupancic, and M. T. Record, “RNA polymerase-promoter interactions : the comings and goings of RNA polymerase,” *J. Bacteriol.*, vol. 180, no. 12, pp. 3019–3025, 1998.

[48] W. Ross *et al.*, “A third recognition element in bacterial promoters: DNA binding by the alpha subunit of RNA polymerase,” *Science (80-.).*, vol. 262, no. 5138, pp. 1407–1413, 1993.

[49] R. L. Gourse, W. Ross, and T. Gaal, “UPs and downs in bacterial transcription initiation: the role of the alpha subunit of RNA polymerase in promoter recognition,” *Mol. Microbiol.*, vol. 37, no. 4, pp. 687–695, 2002.

[50] A. Arkin, J. Ross, and H. H. Mcadams, “Stochastic Kinetic Analysis of Developmental Pathway Bifurcation in,” *Genetics*, vol. 149, pp. 1633–1648, 1998.

[51] A. S. Ribeiro, R. Zhu, and S. A. Kauffman, “A general modeling strategy for gene regulatory networks with stochastic dynamics.,” *J. Comput. Biol.*, vol. 13, no. 9, pp. 1630–1639, Nov. 2006.

[52] A. S. Ribeiro, O.-P. Smolander, T. Rajala, A. Häkkinen, and O. Yli-Harja, “Delayed stochastic model of transcription at the single nucleotide level,” *J. Comput. Biol.*, vol. 16, no. 4, pp. 539–553, 2009.

[53] J. Mäkelä, J. Lloyd-Price, O. Yli-Harja, and A. S. Ribeiro, “Stochastic sequence-level model of coupled transcription and translation in prokaryotes,” *BMC Bioinformatics*, vol. 12, no. 1, p. 121, 2011.

[54] L. Martins *et al.*, “Dynamics of transcription of closely spaced promoters in Escherichia coli, one event at a time.,” *J. Theor. Biol.*, vol. 301, pp. 83–94, 2012.

[55] W. R. McClure, “Rate-limiting steps in RNA chain initiation.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 77, no. 10, pp. 5634–5638, 1980.

[56] L. Bai, T. J. Santangelo, and M. D. Wang, “Single-Molecule Analysis of Rna Polymerase Transcription,” *Annu. Rev. Biophys. Biomol. Struct.*, vol. 35, no. 1, pp. 343–360, 2006.

[57] F. Wang, S. Redding, I. J. Finkelstein, J. Gorman, D. R. Reichman, and E. C. Greene, “The promoter search mechanism of E. coli RNA polymerase is dominated by three–dimensional diffusion,” *Nat. Struct. Mol. Biol.*, vol. 20, no. 2, pp. 174–181, 2013.

[58] H. Buc and W. R. McClure, “Kinetics of open complex formation between Escherichia coli RNA polymerase and the lacUV5 promoter. Evidence for a sequential mechanism involving three steps,” *Biochemistry*, vol. 24, no. 11, pp. 2712–2723, 1985.

[59] A. Revyakin, C. Liu, R. H. Ebright, and T. R. Strick, “Abortive initiation and productive initiation by RNA polymerase involve DNA scrunching,” *Science (80-.).*, vol. 314, no. 5802, pp. 1139–1143, 2006.

[60] L. Hsu, “Monitoring abortive initiation,” *Methods*, vol. 47, no. 1, pp. 25–36, 2009.

[61] B. P. Callen, K. E. Shearwin, and J. B. Egan, “Transcriptional interference between convergent promoters caused by elongation over the promoter,” *Mol. Cell*, vol. 14, no. 5, pp. 647–656, 2004.

[62] A. Sanchez, M. L. Osborne, L. J. Friedman, J. Kondev, and J. Gelles, “Mechanism of transcriptional repression at a bacterial promoter by analysis of single molecules,” *EMBO J.*, vol. 30, no. 19, pp. 3940–3946, 2011.

[63] A. J. Griffiths, W. M. Gelbart, J. H. Miller, and R. C. Lewontin, “Chapter 14. Regulation of Gene Transcription,” in *Modern Genetic Analysis*, New York: W. H. Freeman, Ed. 1999.

[64] Y. Taniguchi *et al.*, “Quantifying E. coli proteome and transcriptome with single-molecule sensitivity in single cells,” *Science (80-.).*, vol. 329, no. 5991, pp. 533–538, 2010.

[65] B. Munsky and M. Khammash, “The finite state projection approach for the analysis of stochastic noise in gene networks,” *IEEE Trans. Automat. Contr.*, vol. 53, pp. 201–214, 2008.

[66] A. S. Ribeiro and J. Lloyd-Price, “SGN Sim, a Stochastic Genetic Networks Simulator,” *Bioinformatics*, vol. 23, no. 6, pp. 777–779, 2007.

[67] J. Lloyd-Price, A. Gupta, and A. S. Ribeiro, “SGNS2: A Compartmentalized Stochastic Chemical Kinetics Simulator for Dynamic Cell Populations,” *Bioinformatics*, vol. 28, no. 22, pp. 3004–3005, 2012.

[68] M. N. M. Bahrudeen, S. Startceva, and A. S. Ribeiro, “Effects of Extrinsic Noise are Promoter Kinetics Dependent,” in *Proceedings of the 9th International Conference on Bioinformatics and Biomedical Technology*, 2017, pp. 44–47.

[69] A. S. Ribeiro, A. Häkkinen, H. Mannerström, J. Lloyd-Price, and O. Yli-Harja, “Effects of the promoter open complex formation on gene expression dynamics,” *Phys. Rev. E*, vol. 81, no. 1, p. 11912, Jan. 2010.

[70] A. S. Ribeiro, A. Häkkinen, and J. Lloyd-Price, “Effects of gene length on the dynamics of gene expression,” *Comput. Biol. Chem.*, vol. 41, pp. 1–9, 2012.

[71] A. S. Ribeiro, “Kinetics of gene expression in bacteria — From models to measurements, and back again,” *Can. J. Chem.*, vol. 91, no. 7, pp. 487–494, 2013.

[72] J. Lloyd-Price, H. Tran, and A. S. Ribeiro, “Dynamics of small genetic circuits subject to stochastic partitioning in cell division,” *J. Theor. Biol.*, vol. 356, pp. 11–19, 2014.

[73] A. S. Ribeiro, “Delays as Regulators of the Dynamics of Genetic Circuits,” *Markov Process. Relat. Fields*, vol. 22, no. 3, pp. 573–594, 2016.

[74] J. Lloyd-price *et al.*, “Dissecting the stochastic transcription initiation process in live Escherichia coli,” *DNA Res.*, vol. 23, no. 3, pp. 203–214, 2016.

[75] W. R. McClure, “Mechanism and control of transcription initiation in prokaryotes.,” *Annu. Rev. Biochem.*, vol. 54, pp. 171–204, Jan. 1985.

[76] V. K. Kandavalli, H. Tran, and A. S. Ribeiro, “Effects of σ factor competition are promoter initiation kinetics dependent.,” *Biochim. Biophys. Acta (BBA)- Gene Regul. Mech.*, vol. 1859, no. 10, pp. 1281–1288, 2016.

[77] C. S. D. Palma *et al.*, “A strategy for dissecting the kinetics of transcription repression mechanisms,” in *EMBEC & NBC 2017. EMBEC 2017, NBC 2017. IFMBE Proceedings, vol 65.*, 2018, pp. 1097–1100.

[78] N. S. M. Goncalves, S. Startceva, C. S. D. Palma, M. N. M. Bahrudeen, S. M. D. Oliveira, and A. S. Ribeiro, “Temperature-dependence of the single-cell variability in the kinetics of transcription activation in *Escherichia coli*,” *Phys. Biol.*, vol. 15, no. 2, p. 026007, Jan. 2018.

[79] G. Danuser, “Computer vision in cell biology.,” *Cell*, vol. 147, no. 5, pp. 973–8, Nov. 2011.

[80] R. Eils and C. Athale, “Computational imaging in cell biology.,” *J. Cell Biol.*, vol. 161, no. 3, pp. 477–81, May 2003.

[81] X. Weng and J. Xiao, “Spatial organization of transcription in bacterial cells,” *Trends Genet.*, vol. 30, no. 7, pp. 287–297, 2014.

[82] K. Ritchie, Y. Lill, C. Sood, H. Lee, and S. Zhang, “Single-molecule imaging in live bacteria cells,” *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 368, no. 1611, p. 20120355, Feb. 2013.

[83] C. M. Davis and M. Gruebele, “Labeling for Quantitative Comparison of Imaging Measurements in Vitro and in Cells,” *Biochemistry*, vol. 57, no. 13, pp. 1929–1938, Apr. 2018.

[84] H. Andersson, T. Baechi, M. Hoechl, and C. Richter, “Autofluorescence of living cells,” *J. Microsc.*, vol. 191, no. Pt 1, p. 1—7, 1998.

[85] T. Ha and P. Tinnefeld, “Photophysics of Fluorescence Probes for Single Molecule Biophysics and Super-Resolution Imaging,” *Annu Rev Phys Chem*, vol. 63, no. 2, pp. 595–617, 2012.

[86] O. Shimomura, F. H. Johnson, and Y. J. Saiga, “Extraction, purification and properties of aequorin, a bioluminescent protein from the luminous hydromedusan, Aequorea.,” *J Cell Comp Physiol*, vol. 59, pp. 223–239, 1962.

[87] M. Chalfie, Y. Tu, G. Euskirchen, W. W. Ward, and D. C. Prasher, “Green fluorescent protein as a marker for gene expression,” *Science (80-.).*, vol. 263, no. 5148, pp. 802–805, 1994.

[88] D. C. Prasher, V. K. Eckenrode, W. W. Ward, F. G. Prendergast, and M. J. Cormier,

"Primary structure of the Aequorea victoria green-fluorescent protein," *Gene*, vol. 111, no. 2, pp. 229–233, 1992.

[89] V. Sample, R. H. Newman, and J. Zhang, "The structure and function of fluorescent proteins," *Chem. Soc. Rev.*, vol. 38, no. 10, pp. 2852–2864, 2009.

[90] R. N. Day and M. W. Davidson, "The fluorescent protein palette: tools for cellular imaging," *Chem Soc Rev*, vol. 38, no. 10, pp. 2887–2921, 2009.

[91] D. M. Chudakov, M. V Matz, S. Lukyanov, and K. A. Lukyanov, "Fluorescent Proteins and Their Applications in Imaging Living Cells and Tissues," *Physiol. Rev.*, vol. 90, no. 3, pp. 1103–1163, Jul. 2010.

[92] E. A. Specht, E. Braselmann, and A. E. Palmer, "A Critical and Comparative Review of Fluorescent Tools for Live-Cell Imaging," *Annu. Rev. Physiol.*, vol. 79, no. 1, pp. 93–117, Feb. 2017.

[93] B. N. G. Giepmans, S. R. Adams, M. H. Ellisman, and R. Y. Tsien, "The Fluorescent Toolbox for Assessing Protein Location and Function," *Science (80-.).*, vol. 312, no. 5771, pp. 217–224, 2006.

[94] F. Wu, E. Van Rijn, B. G. C. Van Schie, J. E. Keymer, and C. Dekker, "Multi-color imaging of the bacterial nucleoid and division proteins with," *Front. Microbiol.*, vol. 6, 2015.

[95] J. A. Megerle, G. Fritz, U. Gerland, K. Jung, and J. O. Rädler, "Timing and dynamics of single cell gene expression in the arabinose utilization system," *Biophys. J.*, vol. 95, no. 4, pp. 2103–2115, 2008.

[96] J. K. Fisher, A. Bourniquel, G. Witz, B. Weiner, M. Prentiss, and N. Kleckner, "Four-dimensional imaging of E. coli nucleoid organization and dynamics in living cells," *Cell*, vol. 153, no. 4, pp. 882–895, 2013.

[97] T. Martin, W. S. C., and S. Lucy, "The bacterial nucleoid: A highly organized and dynamic structure," *J. Cell. Biochem.*, vol. 96, no. 3, pp. 506–521, Jun. 2005.

[98] F. R. Blattner *et al.*, "The Complete Genome Sequence of Escherichia coli K-12," *Science (80-.).*, vol. 277, no. 5331, pp. 1453–1462, 1997.

[99] D. J. Clark, R. Ghirlando, G. Felsenfeld, and H. Eisenberg, "Effect of Positive Supercoiling on DNA Compaction by Nucleosome Cores," *J. Mol. Biol.*, vol. 234, no. 2, pp. 297–301, 1993.

[100] B. J. Peter, J. Arsuaga, A. M. Breier, A. B. Khodursky, P. O. Brown, and N. R. Cozzarelli, "Genomic transcriptional response to loss of chromosomal supercoiling in Escherichia coli," *Genome Biol.*, vol. 5, no. 11, p. R87, 2004.

[101] G. J. Pruss and K. Drlica, "DNA supercoiling and prokaryotic transcription.," *Cell*, vol. 56, pp. 521–523, 1989.

[102] S. Deng, R. A. Stein, and N. P. Higgins, "Organization of supercoil domains and their reorganization by transcription," *Mol. Microbiol.*, vol. 57, no. 6, pp. 1511–1521, 2005.

[103] S. Chong, C. Chen, H. Ge, and X. S. Xie, "Mechanism of Transcriptional Bursting in Bacteria," *Cell*, vol. 158, no. 2, pp. 314–326, Jul. 2014.

[104] S. C. Dillon and C. J. Dorman, "Bacterial nucleoid-associated proteins, nucleoid structure and gene expression.," *Nat. Rev. Microbiol.*, vol. 8, no. 3, pp. 185–195, 2010.

[105] L. Postow, C. D. Hardy, J. Arsuaga, and N. R. Cozzarelli, "Topological domain structure of the Escherichia coli chromosome," *Genes Dev.*, vol. 18, pp. 1766–1779, 2004.

[106] J. C. Wang, “DNA Topoisomerases,” *Annu. Rev. Biochem.*, vol. 65, pp. 635–692, 1996.

[107] M. Gellert, “DNA Topoisomerases,” *Annu. Rev. Biochem.*, vol. 50, no. 2, pp. 879–910, 1981.

[108] K. Drlica, “Control of bacterial DNA supercoiling,” *Mol. Microbiol.*, vol. 6, no. 4, pp. 425–433, 1992.

[109] L. F. Liu and J. C. Wang, “Supercoiling of the DNA template during transcription,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 84, no. 20, pp. 7024–7027, 1987.

[110] M. Wery, C. L. Woldringh, and J. Rouviere-Yaniv, “HU-GFP and DAPI co-localize on the Escherichia coli nucleoid,” *Biochimie*, vol. 83, no. 2, pp. 193–200, 2001.

[111] R. S. Nairin, M. L. Dodson, and R. M. Humphrey, “Comparison of ethidium bromide and 4′,6-diamidino-2-phenylindole as quantitative fluorescent stains for DNA in agarose gels,” *J. Biochem. Biophys. Methods*, vol. 6, no. 2, pp. 95–103, 1982.

[112] Y. Matsuzawa and K. Yoshikawa, “Change of the Higher Order Structure in a Giant DNA Induced by 4′, 6-Diamidino-2-phenylindole as a Minor Groove Binder and Ethidium Bromide as an Intercalator,” *Nucleosides and Nucleotides*, vol. 13, no. 6–7, pp. 1415–1423, Jul. 1994.

[113] J. Kapuscinski, “DAPI: a DNA-Specific Fluorescent Probe,” *Biotech. Histochem.*, vol. 70, no. 5, pp. 220–233, Jan. 1995.

[114] S. Bakshi, H. Choi, N. Rangarajan, K. J. Barns, B. P. Bratton, and J. C. Weisshaar, “Nonperturbative Imaging of Nucleoid Morphology in Live Bacterial Cells during an Antimicrobial Peptide Attack,” *Appl. Environ. Microbiol.* , vol. 80, no. 16, pp. 4977–4986, Aug. 2014.

[115] B. Chazotte, “Labeling nuclear DNA using DAPI,” *Cold Spring Harb. Protoc.*, vol. 6, pp. 80–82, 2011.

[116] D. Zink, N. Sadoni, and E. Stelzer, “Visualizing chromatin and chromosomes in living cells,” *Methods*, vol. 29, no. 1, pp. 42–50, 2003.

[117] S. M. D. Oliveira *et al.*, “Increased cytoplasm viscosity hampers aggregate polar segregation in Escherichia coli,” *Mol. Microbiol.*, vol. 99, no. 4, pp. 686–699, 2016.

[118] R. H. Ebright, “RNA Polymerase: Structural Similarities Between Bacterial RNA Polymerase and Eukaryotic RNA Polymerase II,” *J. Mol. Biol.*, vol. 304, no. 5, pp. 687–698, 2000.

[119] R. D. Finn, E. V Orlova, B. Gowen, M. Buck, and M. van Heel, “Escherichia coli RNA polymerase core and holoenzyme structures,” *EMBO J.*, vol. 19, no. 24, pp. 6833–6844, Dec. 2000.

[120] R. Mathew and D. Chatterji, “The evolving story of the omega subunit of bacterial RNA polymerase,” *Trends Microbiol.*, vol. 14, no. 10, pp. 450–455, 2006.

[121] K. S. Murakami, S. Masuda, E. A. Campbell, O. Muzzin, and S. A. Darst, “Structural basis of transcription initiation: an RNA polymerase holoenzyme-DNA complex.,” *Science (80-.).*, vol. 296, no. 5571, pp. 1285–1290, 2002.

[122] M. S. Paget, “Bacterial Sigma Factors and Anti-Sigma Factors: Structure, Function and Distribution,” *Biomolecules*, vol. 5, no. 3, pp. 1245–1265, Sep. 2015.

[123] T. M. Gruber and C. a Gross, “Multiple sigma subunits and the partitioning of bacterial transcription space.,” *Annu. Rev. Microbiol.*, vol. 57, pp. 441–466, 2003.

[124] I. J. Molineux, “The T7 group,” in *The bacteriophages*, Second Edi., R. Calendar, Ed. Oxford University Press, 2005, pp. 277–301.

[125] L. Minakhin *et al.*, “Bacterial RNA polymerase subunit ω and eukaryotic RNA polymerase subunit RPB6 are sequence, structural, and functional homologs and promote RNA polymerase assembly,” *Proc. Natl. Acad. Sci.*, vol. 98, no. 3, pp. 892–897, 2001.

[126] G. Zhang and S. A. Darst, “Structure of the Escherichia coli RNA polymerase $α$ subunit amino-terminal domain,” *Science (80-.).*, vol. 281, no. 5374, pp. 262–266, 1998.

[127] D. G. Vassylyev *et al.*, “Crystal structure of a bacterial RNA polymerase holoenzyme at 2.6 Å resolution,” *Nature*, vol. 417, p. 712, May 2002.

[128] R. Sousa, Y. J. Chung, J. P. Rose, and B.-C. Wang, “Crystal structure of bacteriophage T7 RNA polymerase at 3.3 Å resolution,” *Nature*, vol. 364, p. 593, Aug. 1993.

[129] S. A. Darst, A. Polyakov, C. Richter, and G. Zhang, “Insights into Escherichia coli RNA Polymerase Structure from a Combination of X-Ray and Electron Crystallography,” *J. Struct. Biol.*, vol. 124, no. 2, pp. 115–122, 1998.

[130] K. S. Murakami, “X-ray Crystal Structure of Escherichia coli RNA Polymerase σ(70) Holoenzyme,” *J. Biol. Chem.*, vol. 288, no. 13, pp. 9126–9134, Mar. 2013.

[131] V. L. Tunitskaya and S. N. Kochetkov, “Structural–Functional Analysis of Bacteriophage T7 RNA Polymerase,” *Biochem.*, vol. 67, no. 10, pp. 1124–1135, 2002.

[132] S. A. Darst *et al.*, “Conformational flexibility of bacterial RNA polymerase,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. 7, pp. 4296–4301, Apr. 2002.

[133] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell, “Section 4.4 - The Three Roles of RNA in Protein Synthesis.,” in *Molecular Cell Biology.*, 6th editio., New York: W. H. Freeman, 2008, pp. 119–125.

[134] S. Marguerat and J. Bähler, “Coordinating genome expression with cell size,” *Trends Genet.*, vol. 28, no. 11, pp. 560–565, 2012.

[135] M. J. Fedor and J. R. Williamson, “The catalytic diversity of RNAs,” *Nat. Rev. Mol. Cell Biol.*, vol. 6, p. 399, May 2005.

[136] D. A. Steinhauer and J. J. Holland, “Rapid Evolution of RNA Viruses,” *Annu. Rev. Microbiol.*, vol. 41, no. 1, pp. 409–431, Oct. 1987.

[137] T. T. Weil, R. M. Parton, and I. Davis, “Making the message clear: visualizing mRNA localization,” *Trends Cell Biol.*, vol. 20, no. 7, pp. 380–390, Jul. 2010.

[138] H. E. Johansson, L. Liljas, and O. C. Uhlenbeck, “RNA Recognition by the MS2 Phage Coat Protein,” *Semin. Virol.*, vol. 8, no. 3, pp. 176–185, 1997.

[139] E. Bertrand, P. Chartrand, M. Schaefer, S. M. Shenoy, R. H. Singer, and R. M. Long, “Localization of ASH1 mRNA Particles in Living Yeast,” *Mol. Cell*, vol. 2, no. 4, pp. 437–445, 1998.

[140] J. A. Chao, Y. Patskovsky, S. C. Almo, and R. H. Singer, “Structural basis for the coevolution of a viral RNA-protein complex,” *Nat. Struct. Mol. Biol.*, vol. 15, no. 1, pp. 103–105, 2008.

[141] S. Lange *et al.*, “Simultaneous transport of different localized mRNA species revealed by live-cell imaging,” *Traffic*, vol. 9, no. 8, pp. 1256–1267, 2008.

[142] A.-B. Muthukrishnan *et al.*, “Dynamics of transcription driven by the tetA promoter, one

event at a time, in live Escherichia coli cells.," *Nucleic Acids Res.*, vol. 40, no. 17, pp. 8472–8483, Sep. 2012.

[143] A. Häkkinen, M. Kandhavelu, S. Garasto, and A. S. Ribeiro, "Estimation of fluorescence-tagged RNA numbers from spot intensities.," *Bioinformatics*, vol. Epub ahead, no. of print, pp. 1–8, Jan. 2014.

[144] S. Oehler, E. R. Eismann, H. Krämer, and B. Müller-Hill, "The three operators of the lac operon cooperate in repression.," *EMBO J.*, vol. 9, no. 4, pp. 973–979, 1990.

[145] B. Wu, J. A. Chao, and R. H. Singer, "Fluorescence fluctuation spectroscopy enables quantitative imaging of single mRNAs in living cells," *Biophys. J.*, vol. 102, no. 12, pp. 2936–2944, 2012.

[146] H. Tran, S. M. D. Oliveira, N. Goncalves, and A. S. Ribeiro, "Kinetics of the cellular intake of a gene expression inducer at high concentrations," *Mol. Biosyst.*, vol. 11, no. 9, pp. 2579–2587, 2015.

[147] X. Ma, D. W. Ehrhardt, and W. Margolin, "Colocalization of cell division proteins FtsZ and FtsA to cytoskeletal structures in living *Escherichia coli* cells by using green fluorescent protein," *Proc. Natl. Acad. Sci.*, vol. 93, no. 23, pp. 12998 LP – 13003, Nov. 1996.

[148] W. Margolin, "FtsZ and the division of prokaryotic cells and organelles," *Nat. Rev. Mol. Cell Biol.*, vol. 6, no. 11, pp. 862–871, Nov. 2005.

[149] D. W. Adams and J. Errington, "Bacterial cell division: assembly, maintenance and disassembly of the Z ring," *Nat. Rev. Microbiol.*, vol. 7, p. 642, Sep. 2009.

[150] J. Lutkenhaus, "Assembly Dynamics of the Bacterial MinCDE System and Spatial Regulation of the Z Ring," *Annu. Rev. Biochem.*, vol. 76, no. 1, pp. 539–562, Jun. 2007.

[151] M. Pazos, P. Natale, and M. Vicente, "A Specific Role for the ZipA Protein in Cell Division: Stabilization OF THE FtsZ Protein," *J. Biol. Chem.*, vol. 288, no. 5, pp. 3219–3226, Feb. 2013.

[152] J. Errington, R. A. Daniel, and D.-J. Scheffers, "Cytokinesis in Bacteria," *Microbiol. Mol. Biol. Rev.*, vol. 67, no. 1, pp. 52–65, Mar. 2003.

[153] N. W. Goehring and J. Beckwith, "Diverse Paths to Midcell: Assembly of the Bacterial Cell Division Machinery," *Curr. Biol.*, vol. 15, no. 13, pp. R514–R526, Apr. 2018.

[154] E. Harry, L. Monahan, and L. B. T.-I. R. of C. Thompson, "Bacterial Cell Division: The Mechanism and Its Precison," *Int. Rev. Cytol.*, vol. 253, pp. 27–94, 2006.

[155] J. Männik *et al.*, "Robustness and accuracy of cell division in Escherichia coli in diverse cell shapes.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 109, no. 18, pp. 6957–62, May 2012.

[156] L. J. Wu and J. Errington, "Nucleoid occlusion and bacterial cell division," *Nat. Rev. Microbiol.*, vol. 10, p. 8, Oct. 2011.

[157] A. Touhami, M. Jericho, and A. D. Rutenberg, "Temperature Dependence of MinD Oscillation in Escherichia coli: Running Hot and Fast ," *J. Bacteriol.*, vol. 188, no. 21, pp. 7661–7667, Nov. 2006.

[158] S. G. Addinall, C. Cao, and J. Lutkenhaus, "Temperature shift experiments with an ftsZ84(Ts) strain reveal rapid dynamics of FtsZ localization and indicate that the Z ring is required throughout septation and cannot reoccupy division sites once constriction has initiated.," *J. Bacteriol.*, vol. 179, no. 13, pp. 4277–4284, Jul. 1997.

[159] M. W. Bailey, P. Bisicchia, B. T. Warren, D. J. Sherratt, and J. Männik, “Evidence for Divisome Localization Mechanisms Independent of the Min System and SlmA in Escherichia coli,” *PLOS Genet.*, vol. 10, no. 8, p. e1004504, Aug. 2014.

[160] T. G. Bernhardt and P. A. J. de Boer, “SlmA, a Nucleoid-Associated, FtsZ Binding Protein Required for Blocking Septal Ring Assembly over Chromosomes in E. coli,” *Mol. Cell*, vol. 18, no. 5, pp. 555–564, Apr. 2005.

[161] M. A. Schumacher and W. Zeng, “Structures of the nucleoid occlusion protein SlmA bound to DNA and the C-terminal domain of the cytoskeletal protein FtsZ,” *Proc. Natl. Acad. Sci.*, vol. 113, no. 18, pp. 4988 LP – 4993, May 2016.

[162] S. G. Addinall, E. Bi, and J. Lutkenhaus, “FtsZ ring formation in fts mutants.,” *J. Bacteriol.*, vol. 178, no. 13, pp. 3877–3884, Jul. 1996.

[163] C. D. A. Rodrigues and E. J. Harry, “The Min System and Nucleoid Occlusion Are Not Required for Identifying the Division Site in Bacillus subtilis but Ensure Its Efficient Utilization,” *PLOS Genet.*, vol. 8, no. 3, p. e1002561, Mar. 2012.

[164] E. Galli and K. Gerdes, “FtsZ-ZapA-ZapB interactome of Escherichia coli,” *J. Bacteriol.*, vol. 194, no. 2, pp. 292–302, 2012.

[165] R. Tsukanov *et al.*, “Timing of Z-ring localization in *Escherichia coli*,” *Phys. Biol.*, vol. 8, no. 6, p. 066003, 2011.

[166] S. Rueda, M. Vicente, and J. Mingorance, “Concentration and Assembly of the Division Ring Proteins FtsZ, FtsA, and ZipA during the Escherichia coli Cell Cycle,” *J. Bacteriol.*, vol. 185, no. 11, pp. 3344–3351, Jun. 2003.

[167] M. Zare, R. Neeli-Venkata, L. Martins, S. Peltonen, U. Ruotsalainen, and A. S. Ribeiro, “Automatic Classification of Z-ring Formation Stages at the Single Cell Level in Escherichia Coli by Machine Learning,” in *4th International Conference on Bioimaging (BIOIMAGING 2017), Book ISBN: 978-989-758-215-8, Porto, Portugal*, 2017.

[168] V. W. Rowlett and W. Margolin, “The bacterial Min system,” *Curr. Biol.*, vol. 23, no. 13, pp. R553–R556, 2013.

[169] H. I. Adler, W. D. Fisher, A. Cohen, and A. A. Hardigree, “MINIATURE escherichia coli CELLS DEFICIENT IN DNA,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 57, no. 2, pp. 321–326, Feb. 1967.

[170] Z. Hu and J. Lutkenhaus, “Topological Regulation of Cell Division in E. coli: spatiotemporal oscillation of MinD requires stimulation of its ATPase by MinE and phospholipid,” *Mol. Cell*, vol. 7, no. 6, pp. 1337–1343, Apr. 2001.

[171] Z. Hu, C. Saez, and J. Lutkenhaus, “Recruitment of MinC, an Inhibitor of Z-Ring Formation, to the Membrane in Escherichia coli: Role of MinD and MinE,” *J. Bacteriol.*, vol. 185, no. 1, pp. 196–203, Jan. 2003.

[172] A. Dajkovic and J. Lutkenhaus, “Z Ring as Executor of Bacterial Cell Division,” *J. Mol. Microbiol. Biotechnol.*, vol. 11, no. 3–5, pp. 140–151, 2006.

[173] L. I. Rothfield, Y.-L. Shih, and G. King, “Polar explorers: membrane proteins that determine division site placement,” *Cell*, vol. 106, no. 1, pp. 13–16, Apr. 2001.

[174] A. L. Marston, H. B. Thomaides, D. H. Edwards, M. E. Sharpe, and J. Errington, “Polar localization of the MinD protein of Bacillus subtilis and its role in selection of the mid-cell division site,” *Genes Dev.*, vol. 12, no. 21, pp. 3419–3430, Nov. 1998.

[175] M. Tokunaga, N. Imamoto, and K. Sakata-sogawa, “Highly inclined thin illumination enables clear single-molecule imaging in cells,” *Nat. Methods*, vol. 5, no. 2, pp. 159–161, 2008.

[176] S. M. Singh and A. K. Panda, “Solubilization and refolding of bacterial inclusion body proteins,” *J. Biosci. Bioeng.*, vol. 99, no. 4, pp. 303–310, 2005.

[177] M. Cruts *et al.*, “Null mutations in progranulin cause ubiquitin-positive frontotemporal dementia linked to chromosome 17q21,” *Nature*, vol. 442, p. 920, Jul. 2006.

[178] K. Bersuker, M. Brandeis, and R. R. Kopito, “Protein misfolding specifies recruitment to cytoplasmic inclusion bodies,” *J. Cell Biol.*, vol. 213, no. 2, pp. 229–241, Apr. 2016.

[179] A. B. Lindner, R. Madden, A. Demarez, E. J. Stewart, and F. Taddei, “Asymmetric segregation of protein aggregates is associated with cellular aging and rejuvenation.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 105, no. 8, pp. 3076–3081, 2008.

[180] A. Singh, V. Upadhyay, A. K. Upadhyay, S. M. Singh, and A. K. Panda, “Protein recovery from inclusion bodies of Escherichia coli using mild solubilization process,” *Microb. Cell Fact.*, vol. 14, p. 41, Mar. 2015.

[181] Z. Yang *et al.*, “Highly Efficient Production of Soluble Proteins from Insoluble Inclusion Bodies by a Two-Step-Denaturing and Refolding Method,” *PLoS One*, vol. 6, no. 7, p. e22981, Jul. 2011.

[182] D. C. Williams, R. M. Van Frank, W. L. Muth, and J. P. Burnett, “Cytoplasmic inclusion bodies in Escherichia coli producing biosynthetic human insulin proteins,” *Science (80-.).*, vol. 215, no. 4533, pp. 687 LP – 689, Feb. 1982.

[183] K. Tsumoto, M. Umetsu, I. Kumagai, D. Ejima, and T. Arakawa, “Solubilization of active green fluorescent protein from insoluble particles by guanidine and arginine,” *Biochem. Biophys. Res. Commun.*, vol. 312, no. 4, pp. 1383–1386, 2003.

[184] R. Neeli-venkata, S. Startceva, T. Annila, and A. S. Ribeiro, “Polar Localization of the Serine Chemoreceptor of Escherichia coli Is Nucleoid Exclusion-Dependent,” *Biophys. J.*, vol. 111, no. 11, pp. 2512–2522, 2016.

[185] P. Meyer and J. Dworkin, “Applications of fluorescence microscopy to single bacterial cells.,” *Res. Microbiol.*, vol. 158, no. 3, pp. 187–94, Apr. 2007.

[186] T. S. Gardner, C. R. Cantor, and J. J. Collins, “Construction of a genetic toggle switch in Escherichia coli.,” *Nature*, vol. 403, no. 6767, pp. 339–42, Jan. 2000.

[187] M. B. Elowitz and S. Leibler, “A synthetic oscillatory network of transcriptional regulators.,” *Nature*, vol. 403, no. 6767, pp. 335–8, Jan. 2000.

[188] M. Kandhavelu *et al.*, “In vivo kinetics of transcription initiation of the lar promoter in Escherichia coli. Evidence for a sequential mechanism with two rate-limiting steps.,” *BMC Syst. Biol.*, vol. 5, p. 149, Jan. 2011.

[189] J. Mäkelä *et al.*, “In vivo single-molecule kinetics of activation and subsequent activity of the arabinose promoter.,” *Nucleic Acids Res.*, vol. 41, no. 13, pp. 6544–6552, Jul. 2013.

[190] J. Lloyd-Price *et al.*, “Probabilistic RNA partitioning generates transient increases in the normalized variance of RNA numbers in synchronized populations of Escherichia coli.,” *Mol. Biosyst.*, vol. 8, no. 2, pp. 565–71, Feb. 2012.

[191] J. Lloyd-Price *et al.*, “Asymmetric disposal of individual protein aggregates in Escherichia

coli, one aggregate at a time.," *J. Bacteriol.*, vol. 194, no. 7, pp. 1747–1752, Jan. 2012.

[192] D. L. Coutu and T. Schroeder, "Probing cellular processes by long-term live imaging--historic problems and current solutions.," *J. Cell Sci.*, vol. 126, no. Pt 17, pp. 3805–15, 2013.

[193] D. J. Stephens and V. J. Allan, "Light microscopy techniques for live cell imaging.," *Science*, vol. 300, no. 5616, pp. 82–86, Apr. 2003.

[194] N. Bonnet, "Artificial intelligence and pattern recognition techniques in microscope image processing and analysis," *Adv. Imaging Electron Phys.*, vol. 514, no. 114, pp. 1–77, 2000.

[195] C. A. Glasbey, K. Buildings, E. Eh, and N. J. Martin, "Multimodal microscopy by digital image processing," *J. Microsc.*, vol. 181, no. 3, pp. 225–237, 1996.

[196] S. Bolte and F. P. Cordelières, "A guided tour into subcellular colocalization analysis in light microscopy.," *J. Microsc.*, vol. 224, no. 3, pp. 213–232, Dec. 2006.

[197] E. M. M. Manders, F. J. Verbeek, and A. J. A, "Measurement of co-localization of objects in dual-colour confocal images," *J. Microsc.*, vol. 169, no. 3, pp. 375–382, 1993.

[198] E. M. Manders, J. Stap, G. J. Brakenhoff, R. van Driel, and J. A. Aten, "Dynamics of three-dimensional replication patterns during the S-phase, analysed by double labelling of DNA and confocal microscopy.," *J. Cell Sci.*, vol. 103, no. 3, pp. 857–862, Nov. 1992.

[199] H. Niki and S. Hiraga, "Subcellular distribution of actively partitioning F plasmid during the cell division cycle in E. coli.," *Cell*, vol. 90, no. 5, pp. 951–957, Sep. 1997.

[200] H. Niki, Y. Yamaichi, and S. Hiraga, "Dynamic organization of chromosomal DNA in Escherichia coli," *Genes Dev.*, vol. 14, pp. 212–223, 2000.

[201] F. H. Bahlmann *et al.*, "Endothelial progenitor cell proliferation and differentiation is regulated by erythropoietin Rapid Communication," *Kidney Int.*, vol. 64, pp. 1648–1652, 2003.

[202] E. M. Ozbudak, M. Thattai, H. N. Lim, B. I. Shraiman, and A. Van Oudenaarden, "Multistability in the lactose utilization network of Escherichia coli," *Nature*, vol. 427, no. 6976, pp. 737–740, Feb. 2004.

[203] J. Yu, J. Xiao, X. Ren, K. Lao, and X. S. Xie, "Probing gene expression in live cells, one protein molecule at a time.," *Science (80-.).*, vol. 311, no. 5767, pp. 1600–1603, 2006.

[204] S. Brown *et al.*, "Differential protein expression in Colletotrichum acutatum : changes associated with reactive oxygen species and nitrogen starvation implicated in pathogenicity on strawberry," *Mol. Plant Pathol.*, vol. 9, no. 2, pp. 171–190, 2008.

[205] J. L. Ptacin *et al.*, "A spindle-like apparatus guides bacterial chromosome segregation," *Nat. Cell Biol.*, vol. 12, no. 8, pp. 791–8, Aug. 2010.

[206] R. S. Fischer, Y. Wu, P. Kanchanawong, H. Shroff, and C. M. Waterman, "Microscopy in 3D: a biologist's toolbox," *Trends Cell Biol.*, vol. 21, no. 12, pp. 682–691, Dec. 2011.

[207] A. D. V. and A. V. S. and I. E. V. and N. E. M. and V. S. P. and M. A. Khodorkovskii, "3D super-resolution microscopy of bacterial division machinery," *J. Phys. Conf. Ser.*, vol. 741, no. 1, p. 12066, 2016.

[208] C. Coltharp, J. Buss, T. M. Plumer, and J. Xiao, "Defining the rate-limiting processes of bacterial cytokinesis," *Proc. Natl. Acad. Sci.*, vol. 113, no. 8, pp. E1044--E1053, 2016.

[209] V. W. Rowlett and W. Margolin, "3D-SIM Super-resolution of FtsZ and Its Membrane

Tethers in *Escherichia coli* Cells," *Biophys. J.*, vol. 107, no. 8, pp. L17–L20, Apr. 2018.

[210] A. Le Gall *et al.*, "Bacterial partition complexes segregate within the volume of the nucleoid," *Nat. Commun.*, vol. 7, p. 12107, Jul. 2016.

[211] K. M. Taute, S. Gude, S. J. Tans, and T. S. Shimizu, "High-throughput 3D tracking of bacteria on a standard phase contrast microscope," *Nat. Commun.*, vol. 6, p. 8776, Nov. 2015.

[212] M. Ackermann, S. C. Stearns, and U. Jenal, "Senescence in a bacterium with asymmetric division.," *Science (80-.).*, vol. 300, no. 5627, p. 1920, Jun. 2003.

[213] S. M. Jazwinski, "Growing old: metabolic control and yeast aging.," *Annu. Rev. Microbiol.*, vol. 56, pp. 769–92, Jan. 2002.

[214] L. Ping, B. Weiner, and N. Klecknerr, "Tsr-GFP accumulates linearly with time at cell poles, and can be used to differentiate 'old' versus 'new' poles, in E.coli," *Mol. Microbiol.*, vol. 69, no. 6, pp. 1427–1438, 2009.

[215] J. Winkler *et al.*, "Quantitative and spatio-temporal features of protein aggregation in Escherichia coli and consequences on protein quality control and cellular ageing.," *EMBO J.*, vol. 29, no. 5, pp. 910–23, Mar. 2010.

[216] C. M. Dobson, "The structural basis of protein folding and its links with human disease.," *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, vol. 356, no. 1406, pp. 133–145, Feb. 2001.

[217] J. J. Harding, "Viewing molecular mechanisms of ageing through a lens.," *Ageing Res. Rev.*, vol. 1, no. 3, pp. 465–479, Jun. 2002.

[218] B. Wolozin, "Regulated protein aggregation: stress granules and neurodegeneration.," *Mol. Neurodegener.*, vol. 7, no. 56, pp. 1–12, Jan. 2012.

[219] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, Oct. 2003.

[220] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Article 13, pp. 1–45, Dec. 2006.

[221] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd Editio. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[222] A. H. K. Roeder, A. Cunha, M. C. Burl, and E. M. Meyerowitz, "A computational image analysis glossary for biologists," *Development*, vol. 139, no. 17, pp. 3071–3080, 2012.

[223] S. Uchida, "Image processing and recognition for biological images," *Dev. Growth Differ.*, vol. 55, no. 4, pp. 523–549, May 2013.

[224] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003.

[225] G. Blanchet and M. Charbit, *Digital Signal and Image Processing Using MATLAB*, 1st Editio. ISTE, 2006.

[226] M. Maška *et al.*, "A benchmark for comparison of cell tracking algorithms.," *Bioinformatics*, vol. 30, no. 11, pp. 1609–17, Jun. 2014.

[227] M. Deshmukh and U. Bhosle, "A survey of image registration," *Int. J. Image Process.*, vol. 5, no. 3, pp. 245–269, 2011.

[228] L. G. Brown, "A Survey of Image Registration Techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, 1992.

[229] J. B. A. Maintz and M. A. Viergever, “A survey of medical image registration,” *Med. Image Anal.*, vol. 2, no. 1, pp. 1–36, 1998.

[230] A. A. Goshtasby, *2-D and 3-D Image Registration: For Medical, Remote Sensing, and Industrial Applications*, 1st Editio. Wiley-Interscience, 2005.

[231] B. Glocker, A. Sotiras, N. Komodakis, and N. Paragios, “Deformable Medical Image Registration: Setting the State of the Art with Discrete Methods,” *Annu. Rev. Biomed. Eng.*, vol. 13, no. 1, pp. 219–244, Jul. 2011.

[232] E. Meijering, O. Dzyubachyk, and I. Smal, “Chapter nine - Methods for Cell and Particle Tracking,” in *Imaging and Spectroscopic Analysis of Living Cells*, vol. 504, P. M. B. T.-M. in E. conn, Ed. Academic Press, 2012, pp. 183–200.

[233] J. V. Chapnick *et al.*, “Techniques for multimodality image registration,” in *Bioengineering Conference, 1993., Proceedings of the 1993 IEEE Nineteenth Annual Northeast*, 1993, pp. 221–222.

[234] M. Wyawahare, P. Patil, and H. Abhyankar, “Image Registration Techniques : An overview,” *Int. J. Signal Process. Image Process. Pattern Recognit.*, vol. 2, no. 3, pp. 11–28, 2009.

[235] P. H. S. Torr and A. Zisserman, “Feature Based Methods for Structure and Motion Estimation,” in *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms, ICCV ’99, Corfu, Greece, September 21-22 1999*, 1999, pp. 278–294.

[236] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, “Diffeomorphic demons: Efficient non-parametric image registration,” *Neuroimage*, vol. 45, no. 1, Supplement 1, pp. S61–S72, 2009.

[237] J.-P. Thirion, “Image matching as a diffusion process: an analogy with Maxwell’s demons,” *Med. Image Anal.*, vol. 2, no. 3, pp. 243–260, 1998.

[238] A. Pertsinidis, Y. Zhang, and S. Chu, “Subnanometre single-molecule localization, registration and distance measurements.,” *Nature*, vol. 466, no. 7306, pp. 647–51, Jul. 2010.

[239] A. Shivanandan, H. Deschout, M. Scarselli, and A. Radenovic, “Challenges in quantitative single molecule localization microscopy.,” *FEBS Lett.*, no. June, Jun. 2014.

[240] J. Buss, C. Coltharp, and J. Xiao, “Super-resolution Imaging of the Bacterial Division Machinery,” *J. Vis. Exp.*, no. 71, p. 10.3791/50048 50048, Jan. 2013.

[241] D. I. Cattoni, J.-B. Fiche, A. Valeri, T. Mignot, and M. Nöllmann, “Super-Resolution Imaging of Bacteria in a Microfluidics Device,” *PLoS One*, vol. 8, no. 10, pp. 1–15, 2013.

[242] D. Skea, I. Barrodale, R. Kuwahara, and R. Poeckert, “A control point matching algorithm,” *Pattern Recognit.*, vol. 26, no. 2, pp. 269–276, 1993.

[243] J. A. Parker, R. V Kenyon, and D. E. Troxel, “Comparison of Interpolating Methods for Image Resampling,” *IEEE Trans. Med. Imaging*, vol. 2, no. 1, pp. 31–39, 1983.

[244] A. Goshtasby, “Image registration by local approximation methods,” *Image Vis. Comput.*, vol. 6, no. 4, pp. 255–261, 1988.

[245] T. Nguyen, “Optimal Ground Control Points for Geometric Correction Using Genetic Algorithm with Global Accuracy,” *Eur. J. Remote Sens.*, vol. 48, no. 1, pp. 101–120, Jan. 2015.

[246] E. B. van de Kraats, G. P. Penney, D. Tomazevic, T. van Walsum, and W. J. Niessen,

"Standardized evaluation methodology for 2-D-3-D registration.," *IEEE Trans. Med. Imaging*, vol. 24, no. 9, pp. 1177–89, Sep. 2005.

[247] L. P. Coelho, A. Shariff, and R. F. Murphy, "Nuclear Segmentation In Microscope Cell Images A Hand-Segmented Dataset And Comparison Of Algorithms," in *Proc IEEE Int Symp Biomed Imaging*, 2009, pp. 518–521.

[248] K. S. H. Tainter, U. Taneja, and R. A. Robb, "Quantitative validation of 3D image registration techniques," in *Proc. SPIE 2434, Medical Imaging 1995: Image Processing, (12 May 1995)*, 1995, vol. 2434, pp. 2416–2434.

[249] C. Murtin, C. Frindel, D. Rousseau, and K. Ito, "Image processing for precise three-dimensional registration and stitching of thick high-resolution laser-scanning microscopy image stacks," *Comput. Biol. Med.*, vol. 92, pp. 22–41, 2018.

[250] D. M. W. Powers, "Evaluation : From Precision , Recall And F-Measure To Roc , Informedness , Markedness & Correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[251] J. Rittscher, "Characterization of Biological Processes through Automated Image Analysis," *Annu. Rev. Biomed. Eng.*, vol. 12, no. 1, pp. 315–344, Jul. 2010.

[252] E. Bengtsson, C. Wählby, and J. Lindblad, "Robust cell image segmentation methods," *Pattern Recognit. Image Anal.*, vol. 14, no. 2, pp. 157–167, 2004.

[253] A. J. Hand, T. Sun, D. C. Barber, D. R. Hose, and S. Macneil, "Automated tracking of migrating cells in phase-contrast video," *J. Microsc.*, vol. 234, no. 1, pp. 62–79, 2009.

[254] N. Malpica and C. O. de Solorzano, "Automated Nuclear Segmentation in Fluorescence Microscopy," in *Science, Technology and Education of Microscopy: an Overview. Volume 2 of Microscopy Book Series.*, A. Méndez-Vilas, Ed. Formatex, 2002, pp. 614–621.

[255] P. Vallotton, L. Mililli, L. Turnbull, and C. Whitchurch, "Segmentation of Dense 2D Bacilli Populations," in *2010 International Conference on Digital Image Computing: Techniques and Applications*, 2010, pp. 82–86.

[256] M. E. Sieracki, S. E. Reichenbach, and K. L. Webb, "Evaluation of automated threshold selection methods for accurately sizing microscopic fluorescent cells by image analysis.," *Appl. Environ. Microbiol.*, vol. 55, no. 11, pp. 2762–2772, Nov. 1989.

[257] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electron. Imaging*, vol. 13, no. 1, pp. 13–20, 2004.

[258] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[259] P. Liao, T. Chen, and P. Chung, "A fast algorithm for multilevel thresholding," *J. Inf. Sci. Eng.*, vol. 17, no. 5, pp. 713–727, 2001.

[260] J. Kittler and J. Illingworth, "On threshold selection using clustering criteria," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-15, no. 5, pp. 652–655, 1985.

[261] T. W. Ridler and S. Calvard, "Picture Thresholding Using an Iterative Selection Method," *IEEE Trans. Syst. Man. Cybern.*, vol. 8, no. 8, pp. 630–632, 1978.

[262] M. K. Yanni and E. Horne, "A new approach to dynamic thresholding," in *EUSIPCO'94: 9th European Conf. Sig. Process*, 1994, vol. 1, pp. 34–44.

[263] C. V Jawahar, P. K. Biswas, and A. K. Ray, "Investigations on fuzzy thresholding based on

fuzzy clustering," *Pattern Recognit.*, vol. 30, no. 10, pp. 1605–1613, 1997.

[264] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Comput. Vision, Graph. Image Process.*, vol. 29, no. 3, pp. 273–285, 1985.

[265] P. Sahoo, C. Wilkins, and J. Yeager, "Threshold selection using Renyi's entropy," *Pattern Recognit.*, vol. 30, no. 1, pp. 71–84, 1997.

[266] A. D. Brink and N. E. Pendock, "Minimum cross-entropy threshold selection," *Pattern Recognit.*, vol. 29, no. 1, pp. 179–188, 1996.

[267] C. H. Li and C. K. Lee, "Minimum cross entropy thresholding," *Pattern Recognit.*, vol. 26, no. 4, pp. 617–625, 1993.

[268] H. D. Cheng, Y.-H. Chen, and Y. Sun, "A novel fuzzy entropy approach to image enhancement and thresholding," *Signal Processing*, vol. 75, no. 3, pp. 277–301, 1999.

[269] M. I. Sezan, "A peak detection algorithm and its application to histogram-based image data reduction," *Comput. Vision, Graph. Image Process.*, vol. 49, no. 1, pp. 36–51, 1990.

[270] N. Ramesh, J. H. Yoo, and I. K. Sethi, "Thresholding based on histogram approximation," *IEE Proc. - Vision, Image Signal Process.*, vol. 142, no. 5, pp. 271–279, 1995.

[271] T. Kampke and R. Kober, "Nonparametric optimal binarization," in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, 1998, vol. 1, pp. 27–29 vol.1.

[272] J. Cai and Z.-Q. Liu, "A new thresholding algorithm based on all-pole model," in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, 1998, vol. 1, pp. 34–36 vol.1.

[273] R. Guo and S. M. Pandit, "Automatic threshold selection based on histogram modes and a discriminant criterion," *Mach. Vis. Appl.*, vol. 10, no. 5, pp. 331–338, 1998.

[274] L. Halada, G. A. Ososkov, and P. Slavkovský, "Histogram Concavity Analysis by Quasicurvature," *Comput. Artif. Intell.*, vol. 6, no. 6, pp. 523–533, 1987.

[275] A. Rosenfeld and P. D. La Torre, "Histogram concavity analysis as an aid in threshold selection," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-13, no. 2, pp. 231–235, 1983.

[276] S. Sahasrabudhe and K. Gupta, "A valley-seeking threshold selection," in *Computer vision and image processing*, L. Shapiro and A. Rosenfeld, Eds. Academic Press, Inc., 1992, pp. 55–67.

[277] D. Bradley and G. Roth, "Adaptive Thresholding using the Integral Image," *J. Graph. Tools*, vol. 12, no. 2, pp. 13–21, Jan. 2007.

[278] P. D. Wellner, "Adaptive thresholding for the DigitalDesk," 1993.

[279] A. Hafiane, G. Seetharaman, and B. Zavidovique, "Median Binary Pattern for Textures Classification," in *Image Analysis and Recognition. ICIAR 2007. Lecture Notes in Computer Science, vol 4633. Springer, Berlin, Heidelberg*, M. Kamel and A. Campilho, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 387–398.

[280] M. Spann and A. Nieminen, "Adaptive Gaussian weighted filtering for image segmentation," *Pattern Recognit. Lett.*, vol. 8, no. 4, pp. 251–255, 1988.

[281] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognit.*, vol. 33, no. 2, pp. 225–236, 2000.

[282] J. Bernsen, “Dynamic thresholding of grey-level images,” in *ICPR’86: Proc. Intl. Conf. Patt. Recog*, 1986, pp. 1251–1255.

[283] P. W. Palumbo, P. Swaminathan, and S. N. Srihari, “Document Image Binarization: Evaluation Of Algorithms,” 1986, vol. 0697, no., pp. 697–698.

[284] S. D. Yanowitz and A. M. Bruckstein, “A new method for image segmentation,” in *[1988 Proceedings] 9th International Conference on Pattern Recognition*, 1988, pp. 270–275 vol.1.

[285] F. H. Y. Chan, F. K. Lam, and H. Zhu, “Adaptive thresholding by variational method,” *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 468–473, 1998.

[286] L. Hertz and R. W. Schafer, “Multilevel thresholding using edge matching,” *Comput. Vision, Graph. Image Process.*, vol. 44, no. 3, pp. 279–295, 1988.

[287] K. Ramar, S. Arumugam, S. N. Sivanandam, L. Ganesan, and D. Manimegalai, “Quantitative fuzzy measures for threshold selection,” *Pattern Recognit. Lett.*, vol. 21, no. 1, pp. 1–7, 2000.

[288] L.-K. Huang and M.-J. J. Wang, “Image thresholding by minimizing the measures of fuzziness,” *Pattern Recognit.*, vol. 28, no. 1, pp. 41–51, 1995.

[289] W.-H. Tsai, “Moment-preserving thresolding: A new approach,” *Comput. Vision, Graph. Image Process.*, vol. 29, no. 3, pp. 377–393, 1985.

[290] A. Pikaz and A. Averbuch, “Digital image thresholding, based on topological stable-state,” *Pattern Recognit.*, vol. 29, no. 5, pp. 829–843, 1996.

[291] L. O’Gorman, “Binarization and Multithresholding of Document Images Using Connectivity,” *CVGIP Graph. Model. Image Process.*, vol. 56, no. 6, pp. 494–506, 1994.

[292] R. L. Kirby and A. Rosenfeld, “A Note on the Use of (Gray Level, Local Average Gray Level) Space as an Aid in Threshold Selection,” *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 12, pp. 860–864, 1979.

[293] A. Narendra and A. Rosenfeld, “A Note on the Use of Second-Order Gray-Level Statistics for Threshold Selection,” *IEEE Trans. Syst. Man. Cybern.*, vol. 8, no. 12, pp. 895–898, 1978.

[294] W.-N. Lie, “An efficient threshold-evaluation algorithm for image segmentation based on spatial graylevel co-occurrences,” *Signal Processing*, vol. 33, no. 1, pp. 121–126, 1993.

[295] A. Beghdadi, A. Le Négrate, and P. V. de Lesegno, “Entropic Thresholding Using a Block Source Model,” *Graph. Model. Image Process.*, vol. 57, no. 3, pp. 197–205, 1995.

[296] C. K. Leung and F. K. Lam, “Maximum a posteriori spatial probability segmentation,” *IEE Proc. - Vision, Image Signal Process.*, vol. 144, no. 3, pp. 161–167, 1997.

[297] N. Friel and I. S. Molchanov, “A new thresholding technique based on random sets,” *Pattern Recognit.*, vol. 32, no. 9, pp. 1507–1517, 1999.

[298] P. Maragos and R. Schafer, “Morphological filters--Part I: Their set-theoretic analysis and relations to linear shift-invariant filters,” *IEEE Trans. Acoust.*, vol. 35, no. 8, pp. 1153–1169, 1987.

[299] J. Serra and L. Vincent, “An overview of morphological filtering,” *Circuits, Syst. Signal Process.*, vol. 11, no. 1, pp. 47–108, 1992.

[300] Q. Wu and K. R. Castleman, “9 - Image Segmentation,” in *Microscope Image processing*,

2008, pp. 159–194.

[301] L. Vincent, “Morphological grayscale reconstruction in image analysis: applications and efficient algorithms,” *IEEE Trans. Image Process.*, vol. 2, no. 2, pp. 176–201, 1993.

[302] F. Meyer, “Topographic distance and watershed lines,” *Signal Processing*, vol. 38, no. 1, pp. 113–125, 1994.

[303] H. Ates and O. N. Gerek, “An image-processing based automated bacteria colony counter,” in *2009 24th International Symposium on Computer and Information Sciences*, 2009, pp. 18–23.

[304] A. D. Mora, P. M. Vieira, A. Manivannan, and J. M. Fonseca, “Automated drusen detection in retinal images using analytical modelling algorithms.,” *Biomed. Eng. Online*, vol. 10, no. 59, p. 59, Jan. 2011.

[305] J. Santinha, A. D. Mora, J. Fonseca, N. Gonçalves, and A. S. Ribeiro, “Detection and Segmentation of Nucleoids Based on Gradient Path Labelling,” *IJPMBS - Int. J. Pharma Med. Biol. Sci.*, vol. 4, no. 1, pp. 51–55, 2015.

[306] C. Queimadelas, J. Rodrigues, A.-B. Muthukrishnan, A. D. Mora, A. S. Ribeiro, and J. M. Fonseca, “Segmentation and tracking of Escherichia coli expressing tsr-venus proteins from combined DIC/Fluorescence images,” in *MEDSIP 2012 - 5th International Conference on Advances in Medical Signal and Information Processes*, 2012, pp. 1–2.

[307] C. Jung and J. Scharcanski, “Robust watershed segmentation using wavelets,” *Image Vis. Comput.*, vol. 23, no. 7, pp. 661–669, Jul. 2005.

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

[309] H. T. Yau, Y. K. Lin, L. S. Tsou, and C. Y. Lee, “An Adaptive Region Growing Method to Segment Inferior Alveolar Nerve Canal from 3D Medical Images for Dental Implant Surgery,” *Comput. Aided. Des. Appl.*, vol. 5, no. 5, pp. 743–752, 2008.

[310] R. Adams and L. Bischof, “Seeded region growing.,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, 1994.

[311] S. Lankton, S. Member, and A. Tannenbaum, “Localizing Region-Based Active Contours,” *IEEE Trans. image Process.*, vol. 17, no. 11, pp. 2029–2039, 2008.

[312] S. C. Zhu and A. Yuille, “Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 9, pp. 884–900, 1996.

[313] P. S. Hiremath and P. Bannigidad, “Digital image analysis of cocci bacterial cells using active contour method,” in *2010 International Conference on Signal and Image Processing*, 2010, pp. 163–168.

[314] C. Xu, D. L. Pham, and J. L. Prince, “Chapter 3 - Image segmentation using deformable models,” in *Handbook of medical imaging*, vol. 2, Bellingham, WA: SPIE, 2000, pp. 129–174.

[315] A. Sotiras, C. Davatzikos, and N. Paragios, “Deformable medical image registration: a survey.,” *IEEE Trans. Med. Imaging*, vol. 32, no. 7, pp. 1153–1190, Jul. 2013.

[316] J. Elfring, R. Janssen, and R. van de Molengraft, “Data Association and Tracking: A Literature Survey,” in *ICT Call 4 RoboEarth Project*, 2010.

[317] S. Gu, Y. Zheng, and C. Tomasi, “Efficient visual object tracking with online nearest

neighbor classifier," in *Computer Vision – ACCV 2010. Volume 6492 of the series LNCS*, 2011, pp. 271–282.

[318] A. Gorji and M. B. Menhaj, "Multiple Target Tracking for Mobile Robots Using the JPDAF Algorithm," in *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, 2007, vol. 1, pp. 137–145.

[319] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar Signal Process. IEE Proc. F*, vol. 140, no. 2, pp. 107–113, 1993.

[320] P. Tissainayagam and D. Suter, "Object tracking in image sequences using point features," *Pattern Recognit.*, vol. 38, no. 1, pp. 105–113, Jan. 2005.

[321] P. Vallotton, A. Ponti, C. M. Waterman-Storer, E. D. Salmon, and G. Danuser, "Recovery, visualization, and analysis of actin and tubulin polymer flow in live cells: a fluorescent speckle microscopy study," *Biophys. J.*, vol. 85, no. 2, pp. 1289–1306, Aug. 2003.

[322] X. Yang, H. Li, and X. Zhou, "Nuclei Segmentation Using Marker-Controlled Watershed, Tracking Using Mean-Shift, and Kalman Filter in Time-Lapse Microscopy," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 53, no. 11, pp. 2405–2414, 2006.

[323] F. Bunyak, K. Palaniappan, S. K. Nath, T. I. Baskin, and G. Dong, "Quantitative Cell Motility For In Vitro Wound Healing Using Level Set-Based Active Contour Tracking," *Proceedings. IEEE Int. Symp. Biomed. Imaging*, pp. 1040–1043, Apr. 2006.

[324] L. S. Ong, M. H. Ang, and H. H. Asada, "Tracking of cell population from time lapse and end point confocal microscopy images with multiple hypothesis Kalman smoothing filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010, pp. 71–78.

[325] A. Bhattacharyya, "On a Measure of Divergence Between Two Statistical Populations Defined by Probability Distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–110, 1943.

[326] J. Joyce, "Kullback-Leibler Divergence," in *International Encyclopedia of Statistical Science SE - 327*, M. Lovric, Ed. Springer Berlin Heidelberg, 2014, pp. 720–722.

[327] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Comput. Vis. Image Underst.*, vol. 113, no. 3, pp. 345–352, Mar. 2009.

[328] J. Shi and C. Tomasi, "Good features to track," in *Proceedings CVPR'94., 1994 IEEE Computer Society Conference on. IEEE*, 1994, pp. 593–600.

[329] J. R. Swedlow and K. W. Eliceiri, "Open source bioimage informatics for cell biology," *Trends Cell Biol.*, vol. 19, no. 11–3, pp. 656–660, Nov. 2009.

[330] A. E. Carpenter *et al.*, "CellProfiler: image analysis software for identifying and quantifying cell phenotypes.," *Genome Biol.*, vol. 7, no. 10, p. R100, Jan. 2006.

[331] L. Kamentsky *et al.*, "Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software.," *Bioinformatics*, vol. 27, no. 8, pp. 1179–80, Apr. 2011.

[332] M. Lamprecht, D. Sabatini, and A. Carpenter, "CellProfiler^{TM}: free, versatile software for automated biological image analysis," *Biotechniques*, vol. 42, no. 1, pp. 71–75, Jan. 2007.

[333] C. Wählby *et al.*, "An image analysis toolbox for high-throughput C. elegans assays,"

Nat. Methods, vol. 9, no. 7, pp. 714–716, 2013.

[334] K. Hartwell *et al.*, "Niche-based screening identifies small-molecule inhibitors of leukemia stem cells.," *Nat. Chem. Biol.*, vol. 9, no. 12, pp. 840–8, Dec. 2013.

[335] C. McQuin *et al.*, "CellProfiler 3.0: Next-generation image processing for biology," *PLOS Biol.*, vol. 16, no. 7, pp. 1–17, 2018.

[336] A. E. Carpenter and T. R. Jones, "CellProfiler Website," *Broad Institute of Harvard and MIT*, 2006. [Online]. Available: https://cellprofiler.org/. [Accessed: 25-Jun-2019].

[337] A. Gordon, A. Colman-lerner, T. E. Chin, K. R. Benjamin, R. C. Yu, and R. Brent, "Single-cell quantification of molecules and rates using open-source microscope-based cytometry," *Nat. Methods*, vol. 4, no. 2, pp. 175–181, 2007.

[338] A. Chernomoretz, A. Bush, R. Yu, A. Gordon, and A. C.- Lerner, "Using Cell-ID 1.4 with R for Microscope-Based Cytometry," *Curr. Protoc. Mol. Biol.*, vol. 84, no. 14.18, pp. 1–27, 2009.

[339] H. Shen *et al.*, "Automated tracking of gene expression in individual cells and cell compartments.," *J. R. Soc. Interface*, vol. 3, no. 11, pp. 787–94, Dec. 2006.

[340] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes : Active Contour Models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[341] C.-J. Du, M. Marcello, D. G. Spiller, M. R. H. White, and T. Bretschneider, "Interactive segmentation of clustered cells via geodesic commute distance and constrained density weighted Nyström method," *Cytom. Part A*, vol. 77A, no. 12, pp. 1137–1147, 2010.

[342] F. Piccinini, A. Kiss, and P. Horvath, "CellTracker (not only) for dummies," *Bioinformatics*, vol. 32, no. 6, pp. 955–957, 2015.

[343] J. C. Crocker and D. G. Grier, "Methods of Digital Video Microscopy for Colloidal Studies," *J. Colloid Interface Sci.*, vol. 179, no. 1, pp. 298–310, 1996.

[344] P. Horvath and F. Piccinini, "CellTracker Website," *Hungarian Academia of Sciences Biological Research Centre*, 2015. [Online]. Available: http://www.celltracker.website/about-celltracker.html. [Accessed: 24-Jul-2019].

[345] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images.," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 4, pp. 841–52, Apr. 2010.

[346] C. S. Bjornsson, G. Lin, Y. Al-kofahi, K. L. Smith, W. Shain, and B. Roysam, "Associative image analysis: a method for automated quantification of 3D multi-parameter images of brain tissue," *J. Neurosci. Methods*, vol. 170, no. 1, pp. 165–178, 2009.

[347] Q. Wang, J. Niemi, C.-M. Tan, L. You, and M. West, "Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy," *Cytom. A*, vol. 77, no. 1, pp. 101–110, 2010.

[348] J. Selinummi, J. Seppälä, O. Yli-Harja, and J. Puhakka, "Software for quantification of labeled bacteria from digital microscope images by automated image analysis," *Biotechniques*, vol. 39, no. 6, pp. 859–863, Dec. 2005.

[349] Z. R. Harrold, M. R. Hertel, and D. Gorman-Lewis, "Optimizing Bacillus subtilis spore isolation and quantifying spore harvest purity," *J. Microbiol. Methods*, vol. 87, no. 3, pp. 325–329, 2011.

[350] J. J. Varga, B. Therit, and S. B. Melville, "Type IV pili and the CcpA protein are needed

for maximal biofilm formation by the gram-positive anaerobic pathogen Clostridium perfringens," *Infect. Immun.*, vol. 76, no. 11, pp. 4944–4951, Nov. 2008.

[351] N. D. Gray *et al.*, "The quantitative significance of Syntrophaceae and syntrophic partnerships in methanogenic degradation of crude oil alkanes," *Environ. Microbiol.*, vol. 13, no. 11, pp. 2957–2975, Nov. 2011.

[352] K. Hostanska, M. Rostock, J. Melzer, S. Baumgartner, and R. Saller, "A homeopathic remedy from arnica, marigold, St. John's wort and comfrey accelerates in vitro wound scratch closure of NIH 3T3 fibroblasts," *BMC Complement. Altern. Med.*, vol. 12, no. 1, p. 100, 2012.

[353] C. Schmidt *et al.*, "Biological studies on Brazilian plants used in wound healing," *J. Ethnopharmacol.*, vol. 122, no. 3, pp. 523–532, 2009.

[354] Y. Wang *et al.*, "Quantification of increased cellularity during inflammatory demyelination," *Brain*, vol. 134, no. 12, pp. 3590–3601, Dec. 2011.

[355] Q. Wang, L. You, and M. West, "CellTracer : Software for automated image segmentation and line- age mapping for single-cell studies (Discussion Paper)," 2008.

[356] O. Sliusarenko and J. Heinritz, "High-throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics," *Mol. Microbiol.*, vol. 80, no. 3, pp. 612–627, 2011.

[357] I. Vallet-Gely and F. Boccard, "Chromosomal Organization and Segregation in Pseudomonas aeruginosa," *PLOS Genet.*, vol. 9, no. 5, pp. 1–9, 2013.

[358] G. Demarre *et al.*, "Differential management of the replication terminus regions of the two Vibrio cholerae chromosomes during cell division," *PLoS Genet.*, vol. 10, no. 9, pp. e1004557–e1004557, Sep. 2014.

[359] K. J. Barns and J. C. Weisshaar, "Single-cell, time-resolved study of the effects of the antimicrobial peptide alamethicin on Bacillus subtilis," *Biochim. Biophys. Acta*, vol. 1858, no. 4, pp. 725–732, Apr. 2016.

[360] T. F. Cootes and C. J. Taylor, "Combining point distribution models with shape models based on finite element analysis," *Image Vis. Comput.*, vol. 13, no. 5, pp. 403–409, 1995.

[361] A. Paintdakhi *et al.*, "Oufti: an integrated software package for high-accuracy, high-throughput quantitative microscopy analysis," *Mol. Microbiol.*, vol. 99, no. 4, pp. 767–777, 2016.

[362] J. Young *et al.*, "Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy.," *Nat. Protoc.*, vol. 7, no. 1, pp. 80–8, 2012.

[363] C. Queimadelas, "Automated segmentation, tracking and evaluation of bacteria in microscopy images," Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa, 2012.

[364] S. Chowdhury, M. Kandhavelu, O. Yli-Harja, and A. S. Ribeiro, "Cell segmentation by multi-resolution analysis and maximum likelihood estimation (MAMLE).," *BMC Bioinformatics*, vol. 14 Suppl 1, no. Suppl 10, p. S8, Jan. 2013.

[365] A. Häkkinen, 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*, vol. 29, no. 13, pp. 1708–1709, Jul. 2013.

[366] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression*

Trees. Chapman and Hall/CRC, 1984.

[367] J. Lloyd-price *et al.*, "Dissecting the stochastic transcription initiation process in live Escherichia coli," *DNA Res.*, vol. 23, no. 3, pp. 203–214, 2016.

[368] H. Tran, S. M. D. Oliveira, N. Goncalves, and A. S. Ribeiro, "Kinetics of the cellular intake of a gene expression inducer at high concentrations," *Mol. Biosyst.*, vol. 11, no. 9, pp. 2579–2587, 2015.

[369] A. ul M. Khan, A. Torelli, I. Wolf, and N. Gretz, "AutoCellSeg: robust automatic colony forming unit (CFU)/cell analysis using adaptive image segmentation and easy-to-use post-editing techniques," *Sci. Rep.*, vol. 8, no. 1, p. 7302, 2018.

[370] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Natl. Acad. Sci.*, vol. 93, no. 4, pp. 1591–1595, 1996.

[371] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, 1991.

[372] N. J. Gotelli and B. J. McGill, "Null versus neutral models: what's the difference?," *Ecography (Cop.).*, vol. 29, no. 5, pp. 793–800, 1996.

[373] W. Xiong, Y. Wang, S. H. Ong, J. H. Lim, and L. Jiang, "Learning Cell Geometry Models For Cell Image Simulation : An Unbiased Approach," in *Proceedings of 2010 IEEE 17th International Conference on Image Processing*, 2010, pp. 1897–1900.

[374] S. K. Hahl and A. Kremling, "A Comparison of Deterministic and Stochastic Modeling Approaches for Biochemical Reaction Systems: On Fixed Points, Means, and Modes," *Front. Genet.*, vol. 7, p. 157, 2016.

[375] D. T. Gillespie, "Exact Stochastic Simulation of Coupled Chemical Reactions," *J. Phys. Chem.*, vol. 81, no. 25, pp. 2340–2361, 1977.

[376] D. T. Gillespie, "Stochastic simulation of chemical kinetics.," *Annu. Rev. Phys. Chem.*, vol. 58, pp. 35–55, Jan. 2007.

[377] D. T. Gillespie, "A rigorous derivation of the chemical master equation," *Phys. A Stat. Mech. its Appl.*, vol. 188, no. 1–3, pp. 404–425, 1992.

[378] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *J. Comput. Phys.*, vol. 22, no. 4, pp. 403–434, 1976.

[379] D. J. Wilkinson, *Stochastic modelling for systems biology*. CRC press, 2011.

[380] J. Lloyd-Price, *Simulating stochastic chemical kinetics with dynamic compartmentalization at runtime : Master of Science Thesis*. Tampere: Tampere university of Technology, 2011.

[381] A. Gupta and P. Mendes, "An Overview of Network-Based and -Free Approaches for Stochastic Simulation of Biochemical Systems," *Computation*, vol. 6, no. 1, 2018.

[382] M. A. Gibson and J. Bruck, "Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels," *J. Phys. Chem. A*, vol. 104, no. 9, pp. 1876–1889, 2000.

[383] H. Li and L. R. Petzold, "Logarithmic Direct Method for Discrete Stochastic Simulation of Chemically Reacting Systems," *Tech. Rep.*, pp. 1–11, 2006.

[384] Y. Cao, D. T. Gillespie, and L. R. Petzold, "The slow-scale stochastic simulation

algorithm," *J. Chem. Phys.*, vol. 122, no. 1, p. 14116, 2005.

[385] V. Ulman, D. Svoboda, M. Nykter, M. Kozubek, and P. Ruusuvuori, "Virtual cell imaging: A review on simulation methods employed in image cytometry," *Cytom. Part A*, vol. 89, no. 12, pp. 1057–1072, Dec. 2016.

[386] M. Grigoryan, G. Hostetter, O. Kallioniemi, and E. Dougherty, "Simulation Toolbox for 3D-FISH Spot-Counting Algorithms," *Real-Time Imaging*, vol. 8, no. 3, pp. 203–212, Jun. 2002.

[387] M. E. Ambühl, C. Brepsant, J.-J. Meister, a B. Verkhovsky, and I. F. Sbalzarini, "High-resolution cell outline segmentation and tracking from phase-contrast microscopy images.," *J. Microsc.*, vol. 245, no. 2, pp. 161–70, Feb. 2012.

[388] A. Lehmussola, J. Selinummi, P. Ruusuvuori, A. Niemisto, and O. Yli-Harja, "Simulating fluorescent microscope images of cell populations.," in *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2005, vol. 3, pp. 3153–6.

[389] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja, "Computational framework for simulating fluorescence microscope images with cell populations.," *IEEE Trans. Med. Imaging*, vol. 26, no. 7, pp. 1010–6, 2007.

[390] X. Du and S. Dua, "Segmentation of fluorescence microscopy cell images using unsupervised mining.," *Open Med. Inform. J.*, vol. 4, pp. 41–9, Jan. 2010.

[391] P. Ruusuvuori, A. Lehmussola, J. Selinummi, T. Rajala, H. Huttunen, and O. Yli-Harja, "Benchmark Set Of Synthetic Images For Validating Cell Image Analysis Algorithms," in *Proceedings of the 16th European Signal Processing Conference, EUSIPCO*, 2008.

[392] P. Ruusuvuori *et al.*, "Evaluation of methods for detection of fluorescence labeled subcellular objects in microscope images," *BMC Bioinformatics*, vol. 11, p. 248, Jan. 2010.

[393] A. Lehmussola, P. Ruusuvuori, J. Selinummi, T. Rajala, and O. Yli-harja, "Synthetic Images of High-Throughput Microscopy for Validation of Image Analysis Methods," *Proc. IEEE*, vol. 96, no. 8, pp. 1348–1360, 2011.

[394] D. Svoboda, M. Kasik, M. Maska, and J. Hubeny, "On simulating 3D Fluorescent Microscope Images," in *Computer Analysis of Images and Patterns -12th International Conference, CAIP 2007, Vienna, Austria, August 27-29, 2007. Proceedings*, 2007, pp. 309–316.

[395] D. Svoboda, M. Kozubek, and S. Stejskal, "Generation of digital phantoms of cell nuclei and simulation of image formation in 3D image cytometry.," *Cytometry. A*, vol. 75, no. 6, pp. 494–509, Jun. 2009.

[396] D. Svoboda and V. Ulman, "MitoGen: A Framework for Generating 3D Synthetic Time-Lapse Sequences of Cell Populations in Fluorescence Microscopy," *IEEE Trans. Med. Imaging*, vol. 36, no. 1, pp. 310–321, 2017.

[397] D. V Sorokin *et al.*, "FiloGen: A Model-Based Generator of Synthetic 3-D Time-Lapse Sequences of Single Motile Cells With Growing and Branching Filopodia," *IEEE Trans. Med. Imaging*, vol. 37, no. 12, pp. 2630–2641, 2018.

[398] D. Svoboda and V. Ulman, "Generation of synthetic image datasets for time-lapse fluorescence microscopy," in *ICIAR'12 Proceedings of the 9th international conference*

on Image Analysis and Recognition - Volume Part II, 2012, vol. 7325, pp. 473–482.

[399] V. Ulman and J. Hubeny, “On Generating Ground-Truth Time-Lapse Image Sequences And Flow Fields,” in *In International Conference on Informatics in Control, Automation and Robotics - ICINCO 2007, RA-1, Angers: INSTICC*, 2007, pp. 234–239.

[400] R. Satwik, P. Benjamin, H. Nicholas, A. Steven, and W. Lani, “SimuCell : a flexible framework for creating synthetic microscopy images a PhenoRipper : software for rapidly profiling microscopy images,” *Nat. Methods*, vol. 9, no. 7, pp. 634–636, 2012.

[401] T. Zhao and R. F. Murphy, “Automated learning of generative models for subcellular location: building blocks for systems biology,” *Cytometry. A*, vol. 71, no. 12, pp. 978–90, Dec. 2007.

[402] R. Murphy, “CellOrganizer: Image-derived Models of Subcellular Organization and Protein Distribution,” *Methods Cell Biol.*, vol. 110, pp. 179–93, 2012.

[403] T. Peng and R. F. Murphy, “Image-derived, three-dimensional generative models of cellular organization.,” *Cytom. Part A*, vol. 79, no. 5, pp. 383–91, May 2011.

[404] M. H. Swat, G. L. Thomas, J. M. Belmonte, A. Shirinifard, D. Hmeljak, and J. a Glazier, “Chapter 13 - Multi-scale modeling of tissues using CompuCell3D.,” in *Methods in cell biology*, vol. 110, Elsevier Inc., 2012, pp. 325–366.

[405] F. Graner and J. Glazier, “Simulation of biological cell sorting using a two-dimensional extended Potts model.,” *Phys. Rev. Lett.*, vol. 69, no. 13, pp. 2013–2016, Sep. 1992.

[406] T. E. Buck, J. Li, G. K. Rohde, and R. F. Murphy, “Toward the virtual cell: automated approaches to building models of subcellular organization ‘learned’ from microscopy images.,” *Bioessays*, vol. 34, no. 9, pp. 791–9, Sep. 2012.

[407] R. Satwik, P. Benjamin, H. Nicholas, A. Steven, and W. Lani, “SimuCell : a flexible framework for creating synthetic microscopy images a PhenoRipper : software for rapidly profiling microscopy images,” *Nat. Methods*, vol. 9, no. 7, pp. 634–636, 2012.

[408] Z. Qu, J. N. Weiss, and W. R. MacLellan, “Coordination of cell growth and cell division: a mathematical modeling study.,” *J. Cell Sci.*, vol. 117, no. Pt 18, pp. 4199–207, Aug. 2004.

[409] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. The MIT Press, 2012.

[410] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience New York, NY, US, 2012.

[411] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[412] M. Nixon and A. S. Aguado, *Feature Extraction & Image Processing for Computer Vision*, 3rd ed. Orlando, FL, USA: Academic Press, Inc., 2012.

[413] S. B. Kotsiantis, “Supervised Machine Learning : A Review of Classification Techniques,” *Informatica*, vol. 31, no. 3, pp. 249–268, 2007.

[414] X. J. Zhu, “Semi-supervised learning literature survey,” 2005.

[415] V. Podgorelec, P. Kokol, B. Stiglic, and I. Rozman, “Decision Trees : An Overview and Their Use in Medicine,” *J. Med. Syst.*, vol. 26, no. 5, pp. 445–463, 2002.

[416] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artif. Intell. Rev.*, vol. 39, no. 4, pp. 261–283, 2013.

[417] L. Rokach and O. Maimon, *Data mining with decision trees: theory and applications*, 2nd ed., vol. Vol. 19-. WorWorld Scientific Publishing Co., Inc., 2014.

[418] R. J. Quinlan, “C4.5: Programs for Machine Learning,” 1993.

[419] Y. Wang and S. Xia, “Unifying attribute splitting criteria of decision trees by Tsallis entropy,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2507–2511.

[420] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE Trans. Syst. Man. Cybern.*, vol. 21, no. 3, pp. 660–674, 1991.

[421] X. Chen, M. Wang, and H. Zhang, “The use of classification trees for bioinformatics,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 55–63, 2011.

[422] A. Criminisi and J. Shotton, *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.

[423] Y.-Y. Song and Y. Lu, “Decision tree methods: applications for classification and prediction,” *Shanghai Arch. psychiatry*, vol. 27, no. 2, pp. 130–135, Apr. 2015.

[424] B. E. Boser, T. B. Laboratories, I. M. Guyon, and V. N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers,” in *COLT ’92 Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[425] J. Shawe-Taylor and S. Sun, “A review of optimization methodologies in support vector machines,” *Neurocomputing*, vol. 74, no. 17, pp. 3609–3618, 2011.

[426] J. Weston, C. Watkins, and others, “Support vector machines for multi-class pattern recognition.,” in *7th European Symposium on Artificial Neural Networks Bruges, Belgium, April 21-22-23*, 1999, vol. 99, pp. 219–224.

[427] A. Mathur and G. M. Foody, “Multiclass and Binary SVM Classification: Implications for Training and Classification Users,” *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 241–245, 2008.

[428] L. Nanni and A. Lumini, “An ensemble of support vector machines for predicting virulent proteins,” *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7458–7462, 2009.

[429] K. A. Cyran *et al.*, “Support Vector Machines in Biomedical and Biometrical Applications,” in *Emerging Paradigms in Machine Learning*, 2013, pp. 379–417.

[430] W. S. Noble, “What is a support vector machine ?,” *Nat. Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006.

[431] W. S. Noble, “Support vector machine applications in computational biology,” in *Kernel Methods in Computational Biology*, 2003, pp. 71–92.

[432] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Schölkopf, and G. Rätsch, “Support vector machines and kernels for computational biology.,” *PLoS Comput. Biol.*, vol. 4, no. 10, p. e1000173, Oct. 2008.

[433] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, vol. 398. John Wiley & Sons, 2013.

[434] S. Menard, *Applied logistic regression analysis*, vol. 106. Sage, 2002.

[435] E. Vittinghoff, D. V Glidden, S. C. Shiboski, and C. E. McCulloch, *Regression methods in biostatistics: linear, logistic, survival, and repeated measures models*. Springer Science & Business Media, 2011.

[436] M. E. Shipe, S. A. Deppen, F. Farjah, and E. L. Grogan, “Developing prediction models for clinical use using logistic regression: an overview,” *J. Thorac. Dis.*, vol. 11, no. Suppl 4, pp. S574–S584, Mar. 2019.

[437] V. R. Avali, G. F. Cooper, and V. Gopalakrishnan, “Application of Bayesian logistic regression to mining biomedical data,” *AMIA ... Annu. Symp. proceedings. AMIA Symp.*, vol. 2014, pp. 266–273, Nov. 2014.

[438] S. Liu, M. Lu, H. Li, and Y. Zuo, “Prediction of Gene Expression Patterns With Generalized Linear Regression Model,” *Front. Genet.*, vol. 10, p. 120, 2019.

[439] M. A. Sartor, G. D. Leikauf, and M. Medvedovic, “LRpath: a logistic regression approach for identifying enriched biological groups in gene expression data,” *Bioinformatics*, vol. 25, no. 2, pp. 211–217, Nov. 2008.

[440] Vanlalhruaia, Y. K. Singh, and N. D. Singh, “Binary face image recognition using logistic regression and neural network,” in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 3883–3888.

[441] C. G. Atkeson, A. W. Moore, and S. Schaal, “Locally Weighted Learning,” in *Lazy Learning*, D. W. Aha, Ed. Norwell, MA, USA: Kluwer Academic Publishers, 1997, pp. 11–77.

[442] N. Czink, C. Mecklenbräuker, and G. Del Galdo, “A novel automatic cluster tracking algorithm,” *IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. PIMRC*, pp. 1–5, 2006.

[443] H. S. Khamis, K. W. Cheruiyot, and S. Kimani, “Application of K-Nearest Neighbour Classification in Medical Data Mining,” *Int. J. Inf. Commun. Technol. Res.*, vol. 4, no. 4, p. 8, 2014.

[444] A. Y. Kondratiev, H. Yaginuma, Y. Okada, and D. V Sorokin, “A Method for Automatic Tracking of Cell Nuclei in 2D Epifluorescence Microscopy Image Sequences,” in *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2018, pp. 1–6.

[445] P. Bhuvaneswari and A. B. Therese, “Detection of Cancer in Lung with K-NN Classification Using Genetic Algorithm,” *Procedia Mater. Sci.*, vol. 10, pp. 433–440, 2015.

[446] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” in *2nd Int. Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[447] T. N. Tran, K. Drab, and M. Daszykowski, “Revised DBSCAN algorithm to cluster data with dense adjacent clusters,” *Chemom. Intell. Lab. Syst.*, vol. 120, pp. 92–96, 2013.

[448] M. A. Masood and M. N. A. Khan, “Clustering Techniques in Bioinformatics,” *IJMECS*, vol. 7, no. 1, pp. 38–46, 2015.

[449] M. E. Celebi, Y. A. Aslandogan, and P. R. Bergstresser, “Mining biomedical images with density-based clustering,” in *International Conference on Information Technology: Coding and Computing (ITCC’05) - Volume II*, 2005, vol. 1, pp. 163-168 Vol. 1.

[450] D. R. Edla and P. K. Jana, “A Prototype-Based Modified DBSCAN for Gene Clustering,” *Procedia Technol.*, vol. 6, pp. 485–492, 2012.

[451] A. Sharma and A. Sharma, “KNN-DBSCAN: Using k-nearest neighbor information for parameter-free density based clustering,” in *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 2017, pp.

787–792.

[452] A. Bryant and K. Cios, “RNN-DBSCAN: A Density-Based Clustering Algorithm Using Reverse Nearest Neighbor Density Estimates,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1109–1121, 2018.

[453] J. Santinha *et al.*, “iCellFusion: Tool for Fusion and Analysis of Live-Cell Images from Time-Lapse Multimodal Microscopy,” in *Biomedical Image Analysis and Mining Techniques for Improved Health Outcomes*, W. B. A. Karâa and N. Dey, Eds. IGI Global, 2015, pp. 71–99.

[454] J. Santinha *et al.*, “IMAGE ALIGNMENT AND LINEAGE CONSTRUCTION TOOL TO STUDY SEGREGATION AND PARTITIONING IN DIVISION OF UNWANTED PROTEIN AGGREGATES IN ESCHERICHIA COLI (abstract),” in *Proceedings of WCSB2014*, 2014, p. 30.

[455] L. Martins, R. Neeli-Venkata, S. M. D. Oliveira, A. Häkkinen, A. S. Ribeiro, and J. M. Fonseca, “SCIP: A Single-Cell Image Processor toolbox,” *Bioinformatics*, vol. bty505, Jun. 2018.

[456] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision.* Cengage Learning, 2014.

[457] R. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE Trans. Acoust.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.

[458] A. Häkkinen, 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*, vol. 29, no. 13, pp. 1708–1709, Jul. 2013.

[459] J. J. More, “The Levenberg-Marquardt algorithm: Implementation and theory,” in *Numerical Analysis*, 1978, pp. 105–116.

[460] K. W. Dunn, M. M. Kamocka, and J. H. Mcdonald, “A practical guide to evaluating colocalization in biological microscopy,” *Am. J. Physiol. Physiol.*, vol. 300, no. 4, pp. C723–C742, 2011.

[461] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433–459, Jul. 2010.

[462] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. 2002.

[463] V. Zinchuk, O. Zinchuk, and T. Okada, “Review Quantitative Colocalization Analysis of Multicolor Confocal Immunofluorescence Microscopy Images : Pushing Pixels to Explore Biological Phenomena,” *Acta Histochem Cytochem*, vol. 40, no. 4, pp. 101–111, 2007.

[464] L. Martins, J. M. Fonseca, and A. S. Ribeiro, “‘miSimBa’ - A simulator of synthetic time-lapsed microscopy images of bacterial cells,” in *Proceedings - 2015 IEEE 4th Portuguese Meeting on Bioengineering, ENBENG 2015*, 2015, no. February, pp. 1–6.

[465] P. Canelas, L. Martins, A. Mora, A. S. Ribeiro, and J. M. Fonseca, “An Image Generator Platform to Improve Cell Tracking Algorithms - Simulation of Objects of Various Morphologies, Kinetics and Clustering,” in *Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, 2016, pp. 44–55.

[466] P. Canelas, “Simulation of biologically inspired object movement for the study of object tracking algorithms,” Faculdade de Ciências e Tecnologia (FCT) - Universidade Nova de

Lisboa, 2016.

[467] L. Martins, P. Canelas, A. Mora, A. S. Ribeiro, and J. Fonseca, “Generator Platform of Benchmark Time-Lapsed Images Development of Cell Tracking Algorithms: Implementation of New Features Towards a Realistic Simulation of the Cell Spatial and Temporal Organization,” in *Simulation and Modeling Methodologies, Technologies and Applications. SIMULTECH 2016. Advances in Intelligent Systems and Computing, vol 676.*, M. Obaidat, T. Ören, and Y. Merkuryev, Eds. Springer, Cham, 2018, pp. 52–74.

[468] A. Zapun, T. Vernet, and M. Pinho, “The different shapes of cocci.,” *FEMS Microbiol Rev*, vol. 32, no. 2, pp. 345–60, 2008.

[469] R. Neeli-Venkata *et al.*, “The precision of the symmetry in Z-ring placement in Escherichia coli is hampered at critical temperatures,” *Phys. Biol.*, vol. 15, no. 5, pp. 1–10, 2018.

[470] H. Abdi and L. J. Williams, “Principal Component Analysis,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433–459, 2010.

[471] M. T. Korpela, J. S. Kurittu, J. T. Karvinen, and M. T. Karp, “A Recombinant Escherichia coli Sensor Strain for the Detection of Tetracyclines,” *Anal. Chem.*, vol. 70, no. 21, pp. 4457–4462, 1998.

[472] T. Fawcett, “An Introduction to ROC Analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2014.

[473] M. Styner, C. Brechbuhler, G. Szckely, and G. Gerig, “Parametric estimate of intensity inhomogeneities applied to MRI,” *IEEE Trans. Med. Imaging*, vol. 19, no. 3, pp. 153–165, 2000.

[474] B. S. Reddy and B. N. Chatterji, “An FFT-based technique for translation, rotation, and scale-invariant image registration,” *IEEE Trans. Image Process.*, vol. 5, no. 8, pp. 1266–1271, 1996.

[475] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded Up Robust Features.,” in *Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol 3951.*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[476] E. Rosten and T. Drummond, “Fusing points and lines for high performance tracking,” in *Tenth IEEE International Conference on Computer Vision (ICCV’05)*, 2005, vol. 2, pp. 1508–1515.

[477] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary Robust invariant scalable keypoints,” in *2011 International Conference on Computer Vision*, 2011, pp. 2548–2555.

[478] C. Harris and M. Stephens, “A combined corner and edge detector.,” in *Alvey vision conference, vol. 15, no. 50, 1988*, 1988.

[479] D. Nistér and H. Stewénius, “Linear Time Maximally Stable Extremal Regions,” in *Computer Vision – ECCV 2008. ECCV 2008. Lecture Notes in Computer Science, vol 5303*, 2008, pp. 183–196.

[480] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration.,” in *VISAPP - International Conference on Computer Vision Theory and Applications*, 2009, vol. 2, no. 2, pp. 331–340.

[481] M. Gleicher, “Projective registration with difference decomposition,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997,

pp. 331–337.

[482] A. D. Mora, P. M. Vieira, A. Manivannan, and J. M. Fonseca, “Automated drusen detection in retinal images using analytical modelling algorithms.,” *Biomed. Eng. Online*, vol. 10, p. 59, 2011.

[483] R. Neeli-Venkata *et al.*, “The precision of the symmetry in Z-ring placement in Escherichia coli is hampered at critical temperatures,” *Phys. Biol.*, vol. 15, no. 5, pp. 1–10, 2018.

[484] S. M. D. Oliveira *et al.*, “Chromosome and plasmid-borne PLacO3O1 promoters differ in sensitivity to critically low temperatures,” *Sci. Rep.*, vol. 9, no. 1, p. 4486, 2019.

[485] W.-Y. Loh and Y.-S. Shih, “Split Selection Methods For Classification Trees,” *Stat. Sin.*, vol. 7, no. 4, pp. 815–840, 1997.

[486] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

Annexes

A.1 - SCIP's User Interface - Buttons and Controls

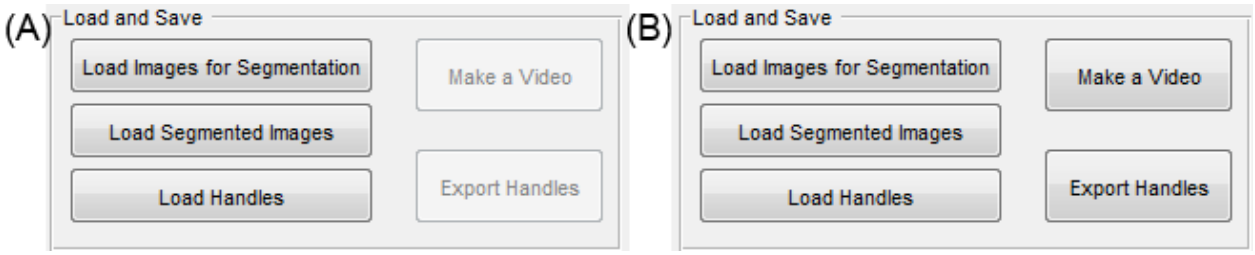


Figure A.1 - Save and Load User Interface: (A) options before loading and (B) options after loading.

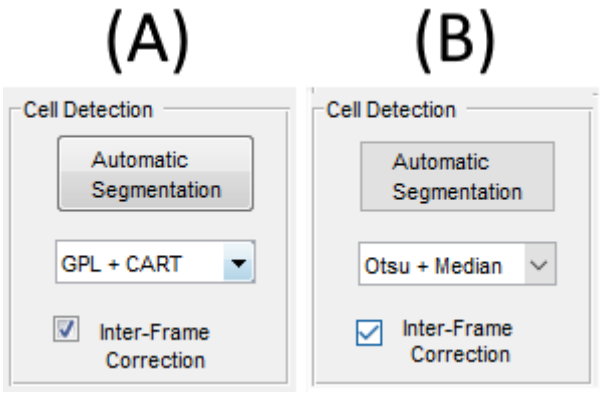


Figure A.2 – Activation of the Cell Segmentation Interface options: (A) 'GPL+CART' (B) 'Otsu + Median'.

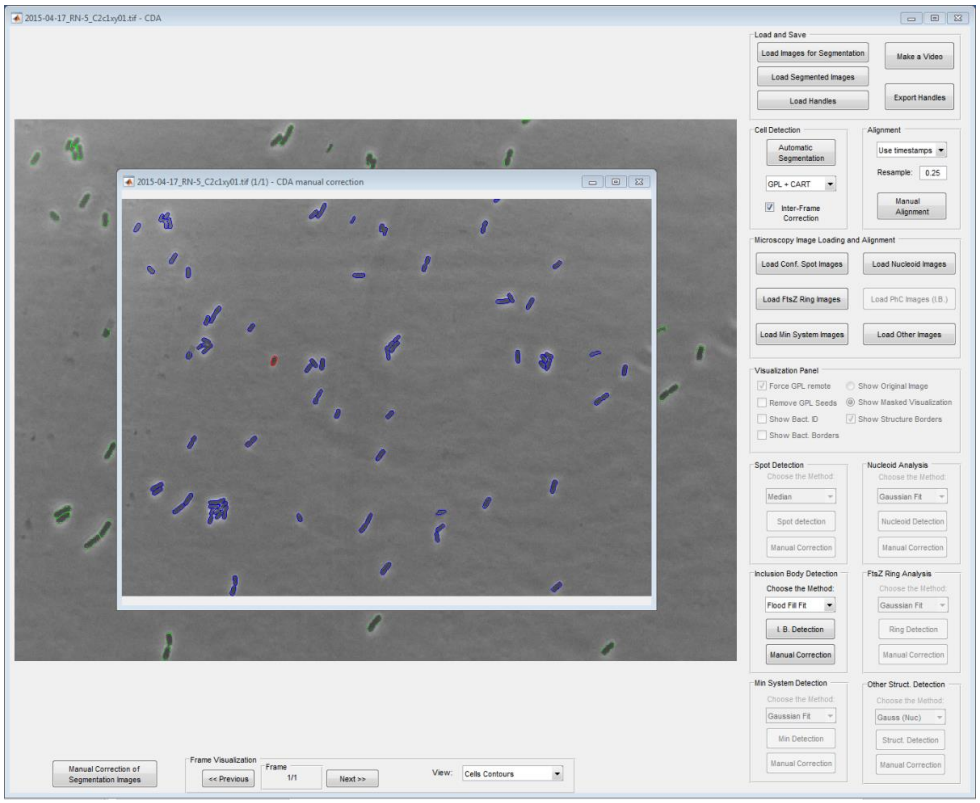


Figure A.3 – Manual Adjustment Window. Blue outlines result from the automatic segmentation, while red outlines are manual adjustments.

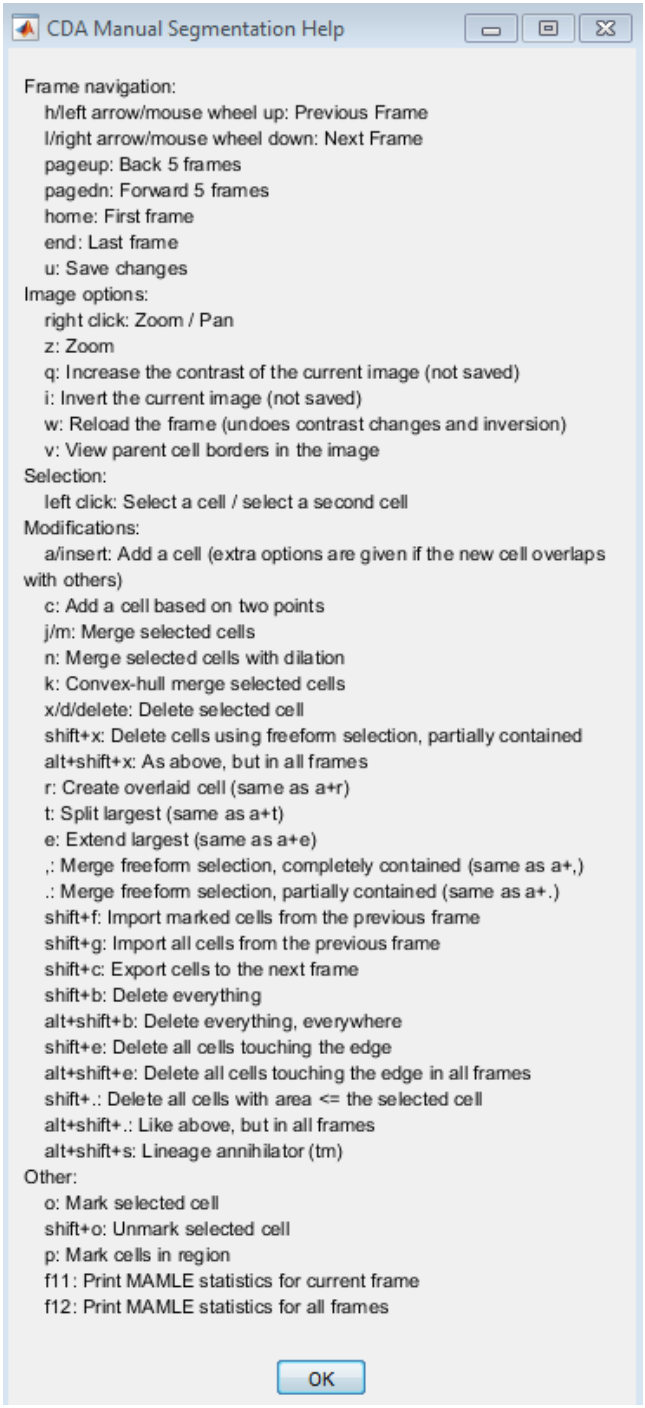


Figure A.4 – 'Help Menu' of the Manual Segmentation

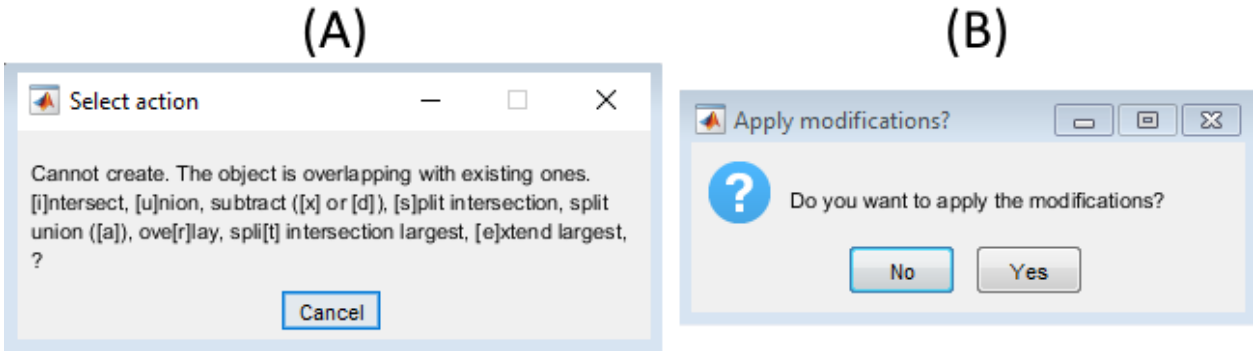


Figure A.5 – Manual Corrections Popups. (A) Options allowed if the new manual segmentation overlaps with an existing cell segment; (B) Popup menu for applying and saving manual corrections

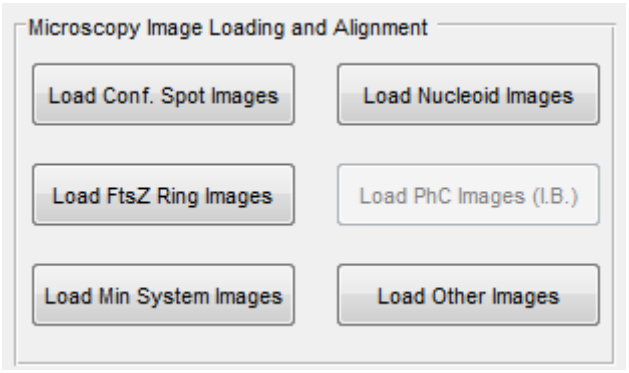


Figure A.6 – Activation of the Microscopy Image Loading Interface with the Load Images for Segmentation Pipeline

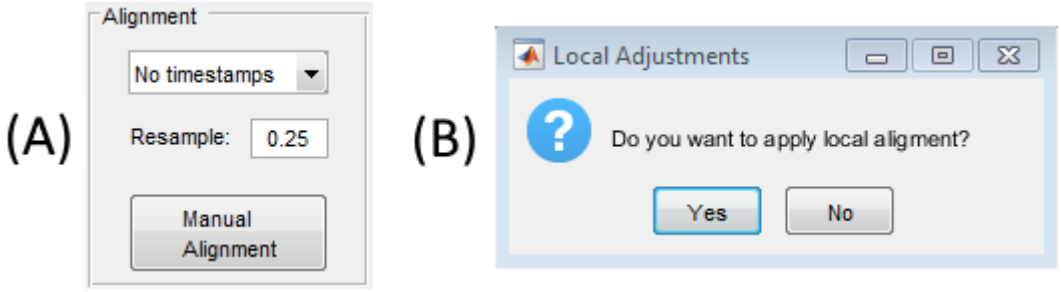


Figure A.7 – Image alignment interface. (A) Activation of the Cell Alignment Interface options (B) Popup for the Execution of Local Adjustments during the Alignment Process.

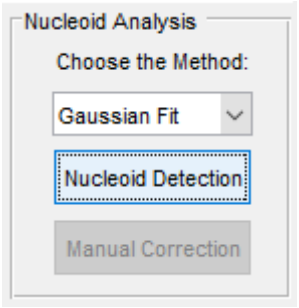


Figure A.8 – Activation of the Gaussian Segmentation method.

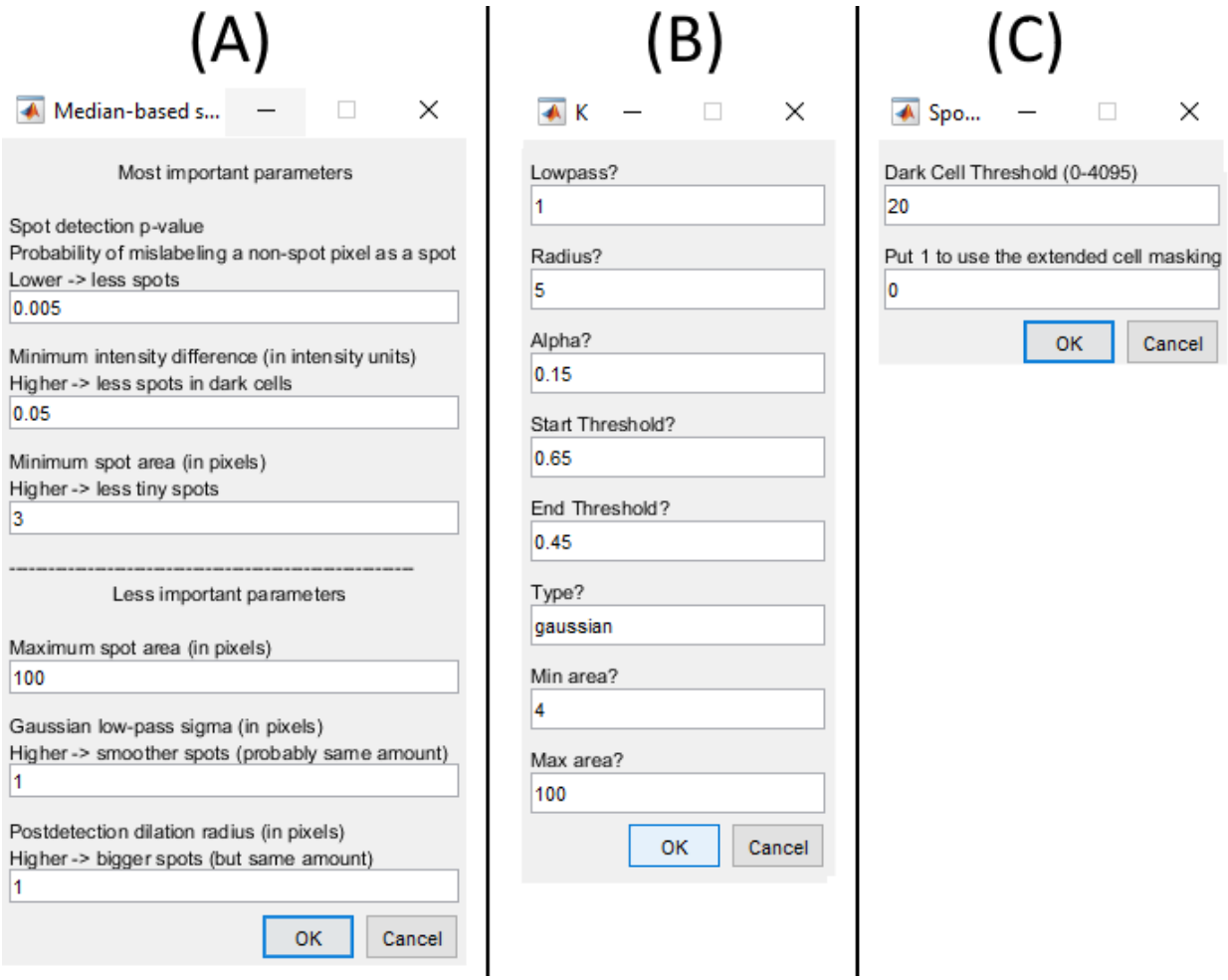


Figure A.9 – Spot detection parameters window: (A) Median Algorithm, (B) Kernel Algorithm and (C) Gaussian Algorithm.

A.2 Confusion Matrices

Table A.1 - Confusion Matrix for nucleoid segmentation. Values are shown for the Gaussian Algorithm with different 'd' parameter values and the 'TreshMorph' Algorithm (TM) with different threshold (T) values. Here 'mean' and 'std' represent the Mean and Standard Deviation of the pixel intensities inside each cell.

Number of cells	Condition Positive	Condition Negative
TM (T = Global Otsu)		
Prediction Positive	2472201	59243
Prediction Negative	901992	4233293
TM (T = Multilevel Otsu)		
Prediction Positive	1388335	2227
Prediction Negative	1985858	4290309
TM (T = mean)		
Prediction Positive	2448241	85397
Prediction Negative	925952	4207139
TM (T = mean + 1/3 std)		
Prediction Positive	3281233	659250
Prediction Negative	92960	3633286
TM (T = mean + 2/3 std)		
Prediction Positive	3154418	392811
Prediction Negative	219775	3899725

TM (T = mean + 1 std)		
Prediction Positive	2940987	207916
Prediction Negative	433206	4084620
TM (T = mean + 4/3 std)		
Prediction Positive	2666275	100529
Prediction Negative	707918	4192007
TM (T = mean + 5/3 std)		
Prediction Positive	2363086	44854
Prediction Negative	1011107	4247682
Gaussian with d = 2		
Prediction Positive	2305718	127402
Prediction Negative	1068475	4165134
Gaussian with d = 3		
Prediction Positive	2387020	111270
Prediction Negative	987173	4181266
Gaussian with d = 4		
Prediction Positive	2353814	96244
Prediction Negative	1020379	4196292
Gaussian with d = 5		
Prediction Positive	2293914	83996
Prediction Negative	1080279	4208540
Gaussian with d = 6		
Prediction Positive	2220521	73761
Prediction Negative	1153672	4218775
Gaussian with d = 7		
Prediction Positive	2164674	66213
Prediction Negative	1209519	4226323
Gaussian with d = 10		
Prediction Positive	1988236	46411
Prediction Negative	1385957	4246125
Gaussian with d = 15		
Prediction Positive	1810807	33425
Prediction Negative	1563386	4259111
Gaussian with d = 20		
Prediction Positive	1740388	33923
Prediction Negative	1633805	4258613

Table A.2 - Confusion Matrix for the detection of FtsZ rings with the Gaussian Segmentation Algorithm (with different 'd' parameter values and the 'TreshMorph' Algorithm based on different threshold values.

Gaussian Algorithm d=2		
Number of cells	**Condition Positive**	**Condition Negative**
Prediction Positive	587173	199818
Prediction Negative	580990	6298748
Gaussian Algorithm d=3		
Prediction Positive	598252	188603
Prediction Negative	569911	6309963

Gaussian Algorithm d=5		
Prediction Positive	628096	200295
Prediction Negative	540067	6298271
Gaussian Algorithm d=7		
Prediction Positive	650016	223066
Prediction Negative	518147	6275500
Gaussian Algorithm d=10		
Prediction Positive	693895	270020
Prediction Negative	474268	6228546
Gaussian Algorithm d=13		
Prediction Positive	714073	300927
Prediction Negative	454090	6197639
Gaussian Algorithm d=15		
Prediction Positive	735695	327075
Prediction Negative	432468	6171491
Gaussian Algorithm d=17		
Prediction Positive	737474	342341
Prediction Negative	430689	6156225
Gaussian Algorithm d=20		
Prediction Positive	754130	372812
Prediction Negative	414033	6125754
TM (Global Otsu)		
Prediction Positive	823613	130789
Prediction Negative	344550	6367777
TM (Multilevel Otsu – Level 2)		
Prediction Positive	280287	11838
Prediction Negative	887876	6486728
TM (Multilevel Otsu – Level 1)		
Prediction Positive	936735	196653
Prediction Negative	231428	6301913
TM (T = mean)		
Prediction Positive	1040150	593052
Prediction Negative	128013	5905514
TM (T = mean - 1/6 stdd)		
Prediction Positive	1073450	799430
Prediction Negative	94713	5699136
TM (T = mean + 1/6 stdd)		
Prediction Positive	997672	440958
Prediction Negative	170491	6057608
TM (T = mean + 2/6 stdd)		
Prediction Positive	943644	327716
Prediction Negative	224519	6170850
TM (T = mean + 3/6 stdd)		
Prediction Positive	883040	247891
Prediction Negative	285123	6250675
TM (T = mean + 4/6 stdd)		
Prediction Positive	818329	189836
Prediction Negative	349834	6308730

Table A.3 - Confusion Matrix for the detection of minD proteins with the Gaussian Segmentation Algorithm (with different 'd' parameter values and the 'TreshMorph' Algorithm based on different threshold values.

Gaussian Algorithm d=5		
Number of cells	**Condition Positive**	**Condition Negative**
Prediction Positive	124998	120856
Prediction Negative	291961	1149161
Gaussian Algorithm d=10		
Prediction Positive	175091	157179
Prediction Negative	241868	1112838
Gaussian Algorithm d=13		
Prediction Positive	182805	163069
Prediction Negative	234154	1106948
Gaussian Algorithm d=14		
Prediction Positive	192697	173166
Prediction Negative	224262	1096851
Gaussian Algorithm d=15		
Prediction Positive	192318	171713
Prediction Negative	224641	1098304
Gaussian Algorithm d=17		
Prediction Positive	**188530**	**170502**
Prediction Negative	**228429**	**1099515**
Gaussian Algorithm d=19		
Prediction Positive	**188107**	**171000**
Prediction Negative	**228852**	**1099017**
Gaussian Algorithm d=20		
Prediction Positive	**189581**	**174553**
Prediction Negative	**227378**	**1095464**
Gaussian Algorithm d=25		
Prediction Positive	**185538**	**177185**
Prediction Negative	**231421**	**1092832**
TM (Global Otsu)		
Prediction Positive	**400262**	**346036**
Prediction Negative	**16697**	**923981**
TM (Multilevel Otsu)		
Prediction Positive	**243306**	**4103**
Prediction Negative	**173653**	**1265914**
TM (T = mean)		
Prediction Positive	**351072**	**18457**
Prediction Negative	**65887**	**1251560**
TM (T = mean – 1/6 stdd)		
Prediction Positive	**386178**	**76520**
Prediction Negative	**30781**	**1193497**
TM (T = mean – 2/6 stdd)		
Prediction Positive	**398782**	**161677**
Prediction Negative	**18177**	**1108340**
TM (T = mean + 1/6 stdd)		
Prediction Positive	**291190**	**5851**
Prediction Negative	**125769**	**1264166**
TM (T = mean + 2/6 stdd)		

Prediction Positive	**233164**	**2378**
Prediction Negative	**183795**	**1267639**

Table A.4 – Confusion Matrix for the detection of Inclusion bodies based on the GPL seed placement and their respective deletion for 3 examples of low, medium and high stress and also the results from joining all examples.

Example 1 – Low Stress (0 mM of NaCl)		
Number of cells	**Condition Positive**	**Condition Negative**
Prediction Positive	25	1
Prediction Negative	10	851
Example 2 - Medium Stress (125 mM of NaCl)		
Prediction Positive	201	3
Prediction Negative	25	1110
Example 3 - High Stress (300 mM of NaCl)		
Prediction Positive	375	13
Prediction Negative	43	2121
Joined all 3 examples		
Prediction Positive	601	17
Prediction Negative	78	4082

A.3 Rules of the Discard and Merge Classifier

Rules for the Discard Classifier

```
/*Terminal Node 1*/
if
(V <= 40.7242 && R <= 0.893248
)
{
   terminalNode = -1; class = 1; probClass0 = 0; probClass1 = 1;
}

/*Terminal Node 2*/
if
(R > 0.893248 && V <= 22.0388
)
{terminalNode = -2; class = 0; probClass0 = 0.984252; probClass1 = 0.015748;
}

/*Terminal Node 3*/
if
(
R > 0.893248 && V > 22.0388 && V <= 40.7242 && LH <= 37.5 && P <= 70.8198
)
{
   terminalNode = -3; class = 0; probClass0 = 1; probClass1 = 0;
}

/*Terminal Node 4*/
if
(
   R > 0.893248 && V > 22.0388 && V <= 40.7242 && LH <= 37.5 && P > 70.8198 )
{
   terminalNode = -4; class = 1; probClass0 = 0; probClass1 = 1;
}
```

```
/*Terminal Node 5*/
if
(
   R > 0.893248 && V > 22.0388 && V <= 40.7242 && LH > 37.5
)
{
   terminalNode = -5; class = 0; probClass0 = 0.736842; probClass1 = 0.263158;
}

/*Terminal Node 6*/
if
(
   V > 40.7242 && R <= 0.904304
{
   terminalNode = -6; class = 1;  probClass0 = 0.0112994; probClass1 = 0.988701;
}

/*Terminal Node 7*/
if
(
   V > 40.7242 &&  R > 0.904304 &&  S <= 1.06049
)
{
   terminalNode = -7; class = 0; probClass0 = 0.833333; probClass1 = 0.166667;
}

/*Terminal Node 8*/
if
(
   V > 40.7242 && R > 0.904304 && S > 1.06049 && P <= 1.40175E+008 && LH <= 26.5
)
{
   terminalNode = -8; class = 0; probClass0 = 1; probClass1 = 0;
}

/*Terminal Node 9*/
if
(
   R > 0.904304 && S > 1.06049 && P <= 1.40175E+008 && V > 40.7242 && V <= 97.2952 && LH > 26.5 &&  LH <= 43.5
)
{
   terminalNode = -9; class = 1; probClass0 = 0.0588235; probClass1 = 0.941176;

/*Terminal Node 10*/
if
(
   R > 0.904304 && S > 1.06049 && V > 40.7242 &&  V <= 97.2952 &&  LH > 43.5 && P <= 64.5772
{
   terminalNode = -10; class = 1; probClass0 = 0; probClass1 = 1;
}

/*Terminal Node 11*/
if
(
   R > 0.904304 && S > 1.06049 && LH > 43.5 && P > 64.5772 && P <= 1.40175E+008 && V > 40.7242 && V <= 61.403
)
{
   terminalNode = -11; class = 1; probClass0 = 0; probClass1 = 1;
}

/*Terminal Node 12*/
if
(
   R > 0.904304 && S > 1.06049 && LH > 43.5 && P > 64.5772 && P <= 1.40175E+008 && V > 61.403 &&    V <= 97.2952
)
```

```
{
   terminalNode = -12; class = 0; probClass0 = 0.441176; probClass1 = 0.558824;
}

/*Terminal Node 13*/
if
(
   R > 0.904304 && S > 1.06049 &&  P <= 1.40175E+008 && LH > 26.5 && V > 97.2952 )
{
   terminalNode = -13; class = 1; probClass0 = 0.0196078; probClass1 = 0.980392;
}

/*Terminal Node 14*/
if
(
   V > 40.7242 && R > 0.904304 &&S > 1.06049 && P > 1.40175E+008
)
{
   terminalNode = -14; class = 0; probClass0 = 1; probClass1 = 0;
}
```

Rules for the Merge Classifier

```
/*Terminal Node 1*/
if
(
   R <= 0.718971 && F <= 1.6604
)
{
   terminalNode = -1; class = 1; probClass0 = 0.040293; probClass1 = 0.959707;
}

/*Terminal Node 2*/
if
(
   R <= 0.718971 && F > 1.6604 && F <= 1.96565 && V <= 1470.62
)
{
   terminalNode = -2; class = 0; probClass0 = 1; probClass1 = 0;
}

/*Terminal Node 3*/
if
(
   R <= 0.718971 && F > 1.6604 && F <= 1.96565 && V > 1470.62 && V <= 10641.7
)
{
   terminalNode = -3; class = 1; probClass0 = 0.0714286; probClass1 = 0.928571;
}

/*Terminal Node 4*/
if
(
   F > 1.6604 && F <= 1.96565 && V > 10641.7 && R <= 0.523484
)
{
   terminalNode = -4; class = 1; probClass0 = 0; probClass1 = 1;
}

/*Terminal Node 5*/
if
(
   F > 1.6604 && F <= 1.96565 && V > 10641.7 && R > 0.523484 && R <= 0.718971
{
   terminalNode = -5; class = 0; probClass0 = 0.833333; probClass1 = 0.166667;
}
```

```
/*Terminal Node 6*/
if
(
   R <= 0.718971 && F > 1.96565 && CC <= 27.5
)
{
   terminalNode = -6; class = 0; probClass0 = 0.8; probClass1 = 0.2;
}

/*Terminal Node 7*/
if
(
   R <= 0.718971 && CC > 27.5 && V <= 9791 && F > 1.96565 && F <= 2.42879
)
{
   terminalNode = -7; class = 1; probClass0 = 0.176471; probClass1 = 0.823529;
}

/*Terminal Node 8*/
if
(
   R <= 0.718971 && CC > 27.5 && V <= 9791 && F > 2.42879
)
{
   terminalNode = -8; class = 0; probClass0 = 1; probClass1 = 0;
}

/*Terminal Node 9*/
if
(
   R <= 0.718971 && F > 1.96565 && CC > 27.5 && V > 9791
)
{
   terminalNode = -9; class = 0; probClass0 = 1; probClass1 = 0;
}

/*Terminal Node 10*/
if
(
   R > 0.718971 && R <= 0.771011 && F <= 1.6804
)
{
   terminalNode = -10; class = 1; probClass0 = 0.230769; probClass1 = 0.769231;
}

/*Terminal Node 11*/
if
(
   R > 0.718971 && R <= 0.771011 && F > 1.6804
)
{
   terminalNode = -11; class = 0; probClass0 = 0.809524; probClass1 = 0.190476;
}

/*Terminal Node 12*/
if
(
   R > 0.771011
)
{
   terminalNode = -12; class = 0; probClass0 = 0.987705; probClass1 = 0.0122951;}
```