Quantitative single-cell analysis using deep learning based smart live-cell microscopy

Von der Fakultät für Elektrotechnik und Informationstechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Johannes Ewald Seiffarth, M.Sc.

aus Kreuztal

Berichter: Professor Dr.-Ing. Johannes Stegmaier
Univ.-Professor Dr. rer. nat. Wolfgang Wiechert

Tag der mündlichen Prüfung 22.10.2025

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Abstract

Living cells form the fundamental building blocks of life on Earth, constructing complex ecosystems in multicellular organisms, such as humans and animals, as well as microbial communities. Understanding the single-cell behavior under dynamic environmental changes is crucial in various fields ranging from investigating cell responses to disease infection, antibiotic treatment, or environmental changes caused by climate change to screening and optimizing strains and process parameters in biotechnology and bioengineering. Microfluidic live-cell imaging (MLCI) has emerged as a vital technology for studying single-cell behavior in space and time under precisely controlled environmental conditions and high-throughput, unlocking spatio-temporal insights into whole cell populations and individual cells. Recent developments in optogenetics enable us to precisely influence and interact with living cells by using light to trigger specific intracellular processes. High-throughput MLCI experiments simultaneously monitor hundreds of independently developing cell populations, generating large-scale imaging datasets – often hundreds of gigabytes – within a single experiment day, posing significant challenges for data analysis. In this setting, exerting targeted influence and interaction on experiment, microscopy, population, and single-cell level requires a real-time understanding of the experiment's progress and decision-making during the experiment. This thesis has two key goals to overcome these challenges: (1) the development of deep learning-based single-cell analysis workflows to efficiently analyze large-scale MLCI datasets and gain detailed insights into single-cell behavior, (2) elevate these analyses into a real-time live-cell imaging operating system to create new smart MLCI experiments that make event-driven decisions and gain unprecedented insights into single-cell interactions.

This dissertation is structured into two major parts: (1) creating the data and deep learning-based analysis foundation for analyzing MLCI experiments and (2) developing smart experimentation with real-time event-driven decision-making and interaction.

In the first part, we build the data and analysis foundation for MLCI experiment datasets. We developed ObiWanMicrobi – a semi-automated cell segmentation and tracking data annotation tool. Using ObiWanMicrobi, we build the "Tracking one-in-a-million" dataset comprising over one million segmented cell masks and their corresponding temporal tracking links. Based on the dataset, we developed experiment-aware tracking metrics that evaluate the influence of experiment parameters, such as the cell population size and imaging rate, on the tracking quality. Leveraging state-of-the-art deep learning-based methods for cell segmentation and tracking, we developed "acia-workflows" integrating a modular yet sequential image and data analysis pipeline for MLCI into accessible workflows. These workflows combine image analysis with subsequent quantitative data analysis, visualization, and text documentation while scaling to high-throughput dataset analysis. We demonstrated that these workflows provide new quantitative insights into single-cell development under dynamic envi-

ronmental conditions and, together with interactive visualizations, offer new perspectives on MLCI experiments by combining information across temporal, spatial, and throughput dimensions.

In the second part, we elevate these analysis capabilities to real-time, enabling smart and event-driven MLCI experiments at high throughput. First, we develop the Digital Microfluidic Chip, a fusion of microfluidic chip design and deep learning-based real-time image analysis, which provides a real-time semantic understanding of microfluidic structures, performs real-time image pre-processing, and enables efficient microscope navigation. We then develop the Just event-driven imaging operating system (JediOS) that introduces a new design paradigm to implement imaging procedures in high-throughput MLCI experiments. JediOS executes one "imaging application" adapting the imaging parameters based on events per observed cell population, and integrates a complete in-silico simulation framework for event-driven decision-making and verification. Combining JediOS with a real-time image analysis pipeline, we implement new MLCI experiment routines and showcase that JediOS offers new opportunities for experiment design. We utilize these event-driven experiment designs for high-throughput screening to characterize the mechanism and sensitivity of specific optogenetic tools, so-called photosensitizers, which release intracellular stress upon excitation, thereby influencing cell development. For the first time, we show that the light energy determines the amount of stress induced into every single cell leading to a specific single-cell development response ranging from slight growth reduction to immediate cell death.

In summary, our contributions combine deep learning image processing with extensive data analysis and visualization to unlock quantitative single-cell analysis analyzing more than one billion cells over the period of this work. The smart live-cell imaging experiment platform utilizes these single-cell insights in real-time for event-driven decision-making, initiating a paradigm shift from passive observation to influence and interaction with living cells in high-throughput MLCI experiments.

Kurzfassung

Lebende Zellen bilden die grundlegenden Bausteine des Lebens auf der Erde und bilden komplexe Ökosysteme in vielzelligen Organismen wie Menschen und Tieren sowie in mikrobiellen Gemeinschaften. Das Verständnis des Verhaltens einzelner Zellen unter dynamischen Umweltveränderungen ist in verschiedenen Bereichen von entscheidender Bedeutung, angefangen bei der Untersuchung von Zellreaktionen auf Krankheitsinfektionen, Antibiotikabehandlungen oder durch den Klimawandel verursachte Umweltveränderungen bis hin zum Screening und zur Optimierung von Stämmen und Prozessparametern in der Biotechnologie. Die mikrofluidische Lebend-Zell-Bildgebung (MLCI) hat sich zu einer wichtigen Technologie für die Untersuchung des Verhaltens einzelner Zellen in Raum und Zeit unter präzise kontrollierten Umweltbedingungen und mit hohem Durchsatz entwickelt und ermöglicht räumlich-zeitliche Einblicke in ganze Zellpopulationen und einzelne Zellen. Jüngste Entwicklungen in der Optogenetik ermöglichen zudem, lebende Zellen präzise zu beeinflussen und mit ihnen zu interagieren, indem Licht eingesetzt wird, um bestimmte intrazelluläre Prozesse auszulösen. Hochdurchsatz-MLCI-Experimente überwachen gleichzeitig Hunderte von sich unabhängig entwickelnden Zellpopulationen und generieren innerhalb eines einzigen Versuchstages große Bilddatensätze – oft Hunderte von Gigabyte –, was erhebliche Herausforderungen für die Datenanalyse mit sich bringt. Zudem erfordert die gezielte Beeinflussung und Interaktion auf Versuchs-, Mikroskopie-, Populations- und Einzelzellebene ein Echtzeitverständnis des Experimentfortschritts. Diese Arbeit hat zwei Hauptziele, um diese Herausforderungen zu bewältigen: (1) die Entwicklung von Deep-Learning-basierten Einzelzell-Analyse-Workflows, um groß angelegte MLCI-Datensätze effizient zu analysieren und detaillierte Einblicke in das Verhalten einzelner Zellen zu gewinnen, (2) die Überführung dieser Analysen in ein Echtzeit-MLCI-Betriebssystem, um neue intelligente MLCI-Experimente zu ermöglichen, die ereignisgesteuerte Entscheidungen treffen und neuartige Einblicke in die Interaktionen einzelner Zellen geben.

Diese Dissertation gliedert sich in zwei Hauptteile: (1) die Schaffung der Grundlagen für Daten- und Deep-Learning-basierte Analysen für die Auswertung von MLCI-Experimenten und (2) die Entwicklung intelligenter Experimente mit ereignisgesteuerter Entscheidungsfindung und Interaktion in Echtzeit.

Im ersten Teil wurde die Daten- und Analysebasis für MLCI-Experimentdatensätze entwickelt. Dafür wurde ObiWanMicrobi entwickelt – ein halbautomatisches Werkzeug zur Zellsegmentierung und Datenannotation. Damit wurde der "Tracking one-in-a-million" Datensatz erstellt, der über eine Million segmentierte Zellmasken und die entsprechenden zeitlichen Tracking-Links umfasst. Auf der Grundlage des Datensatzes wurde eine neue Tracking-Metrik entwickelt, die den Einfluss von Experimentparametern wie der Größe der Zellpopulation und der Bildgebungsrate auf die Tracking-Qualität bewerten. Unter Nutzung von state-of-the-art Deep Learning-basierten Methoden für Zellsegmen-

tierung und -tracking wurden acia-workflows entwickelt, die eine modulare und sequenzielle Bild- und Datenanalyse-Pipeline für MLCI bereitstellen und sich für die Analyse von Datensätzen mit hohem Durchsatz skalieren lassen. Es wurde gezeigt, dass diese Workflows neue quantitative Einblicke in die Entwicklung einzelner Zellen unter dynamischen Umweltbedingungen liefern und zusammen mit interaktiven Visualisierungen neue Perspektiven für MLCI-Experimente bieten, indem sie Informationen über zeitliche, räumliche und Durchsatzdimensionen hinweg kombinieren.

Im zweiten Teil wurde diese Analysefunktionen auf die Echtzeitanalyse erweitert und ermöglicht so intelligente und ereignisgesteuerte MLCI-Experimente mit hohem Durchsatz. Zunächst wurde der Digital Microfluidic Chip entwickelt, eine Kombination aus mikrofluidischem Chipdesign und Deep-Learning-basierter Echtzeit-Bildanalyse, die ein semantisches Echtzeitverständnis mikrofluidischer Strukturen ermöglicht, eine Echtzeit-Bildvorverarbeitung durchführt und eine effiziente Mikroskopnavigation ermöglicht. Anschließend wurde das Just event-driven imaging operating system (JediOS) entwickelt, welches ein neues Designparadigma zur Implementierung von MLCI-Experimenten mit hohem Durchsatz einführt, indem es eine "Bildgebungsanwendung" pro beobachteter Zellpopulation ausführt und ein vollständiges In-silico-Simulationsframework für ereignisgesteuerte Experimentdurchführung und -verifizierung liefert. Durch die Kombination von JediOS mit einer Echtzeit-Bildanalyse-Pipeline wurden ganz neue MLCI-Experimentroutinen entwickelt, die zeigen, dass JediOS neue Möglichkeiten für das Experimentdesign bietet. Diese ereignisgesteuerten Experimentdesigns wurden für das Hochdurchsatz-Screening genutzt, um den Mechanismus und die Empfindlichkeit spezifischer optogenetischer Werkzeuge, sogenannter Photosensibilisatoren (PS), zu charakterisieren, die bei Aktivierung intrazellulären Stress freisetzen und dadurch die Zellentwicklung beeinflussen. Es wurde damit erstmals gezeigt, dass die Lichtenergie die Menge des in jeder einzelnen Zelle induzierten Stresses bestimmt und zu einer spezifischen Einzelzellentwicklung führt, die von einer leichten Wachstumsverringerung bis zum sofortigen Zelltod reicht.

Die Entwicklungen in dieser Arbeit kombinieren die Deep-Learning-Bildverarbeitung mit umfassender Datenanalyse und -visualisierung, um eine quantitative Einzelzellanalyse zu ermöglichen. Über den Zeitraum dieser Arbeit wurden dadurch mehr als eine Millarde Zellen analysiert. Die intelligente Plattform für Lebend-Zell-Bildgebung nutzt diese Einzelzell-Erkenntnisse in Echtzeit für ereignisgesteuerte Entscheidungen und leitet damit einen Paradigmenwechsel von der passiven Beobachtung hin zur Beeinflussung und Interaktion mit lebenden Zellen in Hochdurchsatz-MLCI-Experimenten ein.

Acknowledgements

Conducting this PhD thesis project has been an incredible journey spanning four years, with its share of highs and lows, including a conference journey around the world. Many important people supported me in pushing the frontier of life cell imaging while also maturing as a researcher, friend, partner, and human being. Thus, I am deeply grateful to all the people who played a crucial role in this interdisciplinary endeavor, collaborated with me to advance our vision of groundbreaking life science research, and placed their trust in me. Despite all the fantastic achievements I can display in this work, I went through deep valleys of frustration, bug tracing, experiment failure, and the challenge of starting a thesis during COVID with a 1-year strict home office session. These challenges have been an inherent part of this PhD project and my maturation as a researcher. In the end, I am deeply thankful for all the projects and threads of work that intertwined so successfully, to the people who introduced me to the world of living cells and the challenges they face, and the successful collaborations that created such encouraging results.

In particular, I would like to thank my close colleagues from the Institute of Biotechnology (IBG-1) in the Research Centre Jülich. There is the *microscale* group, which performs all the live-cell imaging experiments and introduced me to the challenges, conducted experiments, and contributed to rich discussions on data analysis for live-cell experiments. I would like to mention Bastian Wollenhaupt, who introduced me to the underlying microfluidic live-cell imaging technology and was an excellent talk partner during my first conference in Berlin. The discussions with him early during my project sowed the seeds for many ideas that I could implement later on. Working with Matthias Pesch was a pleasure, as he played a crucial role in preparing the biological and microfluidic aspects of all the smart experiments presented in this thesis. Moreover, the discussion with him shaped many ”smart” decisions during the development. Nevertheless, he also showed a lot of patience with all the ”not-so-smart” decisions along the way to developing ”smart” experiments, which led to some failed experiments. Without him, advancing the technology and conducting all these experiments would not have been possible in such a short amount of time.

Teaming up with Thomas Drepper and Seung-Hyun Paik led to incredible successful teamwork in developing optogenetic experiments that I would have never imagined before. Moreover, Lennart Witting, Michelle Bund, and Keitaro Kasahara contributed their expertise in life cell imaging and their curiosity about microbial cells in general, challenging me with high demands for image analysis workflows. Not to be forgotten is Agnes Müller-Schröer, who keeps the microfluidic live-cell lab running.

I would also like to thank my research group, the *modsims*. There is Martin Beyß, who contributed his expertise in computer and infrastructure administration. Fredrik Jadebeck and Anton Stratmann have been great office mates, challenging my enthusiasm for using AI. I had great dis-

cussions with Johannes Schmölders and Jan Breuer during bike or car rides to Jülich and with cold beverages after work. An incredible thank you also to my fellow image analysis colleagues Richard Paul and Karina Ruzaeva and all their discussions in our weekly imaging meetup. Thanks also to my masters student, Lukas, for his outstanding contribution to the real-time analysis of microfluidic chip systems and his patience in spending many days in the basement lab developing the interface for controlling the microscope platforms.

I would also like to thank my direct supervisors, Katharina Nöh and Hanno Scharr, for the many discussions and detailed feedback. Especially, Katharina gave me the opportunity to develop and implement new ideas for shaping the domain of microbial life-cell imaging. Many thanks also to Dietrich Kohlheyer for his expertise in microfluidics, to Wolfgang Wiechert for offering this fascinating research position within his institute, and to Johannes Stegmaier for providing me the opportunity for this PhD thesis.

I was extremely fortunate to become associated with HDS-LEE, the Helmholtz School for Data Science in Life, Earth, and Energy, which provided a great exchange platform for PhD students across different research domains, especially in the immediate aftermath of the COVID period. Their support in travel funding enabled me to attend conferences worldwide, where I shared our research, technology, and important insights.

I would also like to thank my collaborators from the Karlsruhe Institute of Technology (KIT), Ralf Mikut, Tim Scherr, Oliver Neumann, Katharina Löffler, Nils Friederich, and Angelo Jovin Yamachui Sitcheu, for their tight collaboration, extensive discussions, and new perspectives. Moreover, I would like to thank Luisa Blöbaum and Alexander Grünberger for their contributions of insights and data to the developed tracking benchmark.

Finally, there are many people in my personal life who have supported me throughout this journey. I would like to express my deepest gratitude to my parents, brothers, and family for their unwavering support throughout this multi-year project and across every stage of my research career. Their encouragement has been a constant source of strength to me. My heartfelt thanks go to my friends - especially Lisa - for their patience and understanding when my work on studying living bacteria demanded so much of my time that I could not spend with them. The last person I would like to thank is my girlfriend, Franzi. I met you near the completion of this project, but your love, energy, and passion made these past months of writing my thesis the happiest of my life.

Preface

Writing this thesis was an interdisciplinary endeavor at the intersection of life sciences, biology, biotechnology, computer sciences, and electrical engineering. Discussions with my colleagues, visiting international conferences, daily work in the lab, and my background in computer science enabled me to conduct this research at the Institute for Biotechnology (IBG-1) at the research center in Jülich. However, following all parts and aspects of this work may be challenging. To break down the interdisciplinary complexity of this work, we provide brief guidance on how to approach and read this thesis.

This thesis is structured into two overarching parts. The first introduces the data science foundation and analysis opportunities that are then extended into a real-time experiment platform for smart microscopy in the second part. Both parts have in common that the intertwined development of deep learning, data science, and microfluidic live cell imaging is crucial to jointly develop this field.

However, the structure and way of reading both parts differ. The chapters in the first part (Chapters 2 to 5) represent a sequential progression, where each chapter builds upon the one before but can be read individually. Each of these chapters covers a specific part of the data analysis in live-cell experiments. In the second part, the chapters are more tightly connected, developing the platform and experiments that utilize real-time information and smart microscopy. Due to this tight coupling, we recommend reading these chapters in sequential order to gain a comprehensive understanding of smart microscopy and its impact on microfluidic live-cell imaging.

If you are primarily interested in the biological insights that are unlocked by this work, we advise you to read Chapter 4 and Chapter 9. If you are interested in the technological advances in smart microscopy in high-throughput experiments, read Chapters 6 to 8.

Lastly, I would like to motivate all the researchers who read this thesis to challenge the claims of this work and stand on the shoulders of giants to further develop and exploit the countless opportunities that spatio-temporal single-cell investigation has to offer for life science research. Do not be afraid when hitting certain boundaries and obstacles, but team up across multiple disciplines to overcome them and better explore the wonderful microcosmos of microbial life that is invisibly all around us.

Published work

The research for this thesis was conducted at the Institute of Bio- and Geosciences: Biotechnology (IBG-1) at the Forschungszentrum Jülich as part of the SATOMI and EMSIG projects [SATOMI ZT-I-PF-04-011, EMSIG ZT-I-PF-04-044]. Parts of this thesis have already been published as preprints, in peer-reviewed journals, at conferences, and in the form of open source software packages. The adaptions from such publications are detailed at the beginning of each chapter. Any thought, methodology, result, conclusion and direct or indirect contribution to this work is considered as properly cited by the following listing of publications:

Articles

- Seiffarth, J.; Scherr T.; Wollenhaupt B.; Neumann O.; Scharr H; Kohlheyer D., Mikut R., Nöh K.: ObiWan-Microbi: OMERO-based integrated workflow for annotating microbes in the cloud. SoftwareX. 2024. `https://doi.org/10.1016/j.softx.2024.101638`
- Seiffarth, J.; Blöbaum, L.; Grünberger, A.; Nöh, K.: Customizable and Interactive Visualizations for Investigating Spatio-Temporal Single-Cell Information. IEEE International Symposium on Biomedical Imaging. 2024. `https://doi.org/10.1109/ISBI56570.2024.10635297`
- Seiffarth, J.; Blöbaum, L.; Paul, R.D.; Friederich, N.; Sitcheu, A.J.; Mikut, R.; Scharr, H.; Grünberger, A.; Nöh, K.: Tracking one-in-a-million: Large-scale benchmark for microbial single-cell tracking with experiment-aware robustness metrics. In BioImage Computing Workshop at the European Conference on Computer Vision (ECCV). 2024. `https://doi.org/10.1007/978-3-031-91721-9_20`
- Seiffarth, J.; Nöh, K.: acia-workflows: Automated Single-cell Imaging Analysis for Scalable and Deep Learning-based Live-cell Imaging Analysis Workflows. *in review.* 2025. `https://doi.org/10.48550/arXiv.2510.05886`.
- Seiffarth, J.; Nöh, K.: PyUAT: Open-source Python framework for efficient and scalable cell tracking. PLOS One. 2025. `https://doi.org/10.1371/journal.pone.0337110`
- Friederich, N.; Jovin Yamachui Sitcheu, A.; Nassal, A.; Yildiz, E.; Pesch, M.; Beichter M.; Scholtes L.; Akbaba, B.; Lautenschlager, T.; Neumann, O.; Kohlheyer, D.; Scharr, H.; Seiffarth, S.; Nöh, K.; Mikut R.: EAP4EMSIG - enhancing event-driven microscopy for microfluidic single-cell analysis. Automatisierungstechnik. 2025 `https://doi.org/10.1515/auto-2025-0018`.

- Paul R.D.; Seiffarth, J.; Rügamer, D.; Scharr, H.; Nöh, K.: How To Make Your Cell Tracker Say" I dunno!". *accepted*. 2025. https://doi.org/10.48550/arXiv.2503.09244

- Kasahara; K., Seiffarth, J.; Stute, B.; von Lieres, E.; Drepper, T.; Nöh, K.; Kohlheyer, D.: Unveiling microbial single-cell growth dynamics under rapid periodic oxygen oscillations. Lab on Chip. 2025. https://doi.org/10.1039/D5LC00065C.

- Witting L.; Seiffarth, J.; Stute, B.; Schulze T.; Hofer, J.M.; Nöh, K.; Eisenhut, M.; Weber, A.P.M.; von Lieres, E.; Kohlheyer, D.: A microfluidic system for the cultivation of cyanobacteria with precise light intensity and CO_2 control: enabling growth data acquisition at single-cell resolution. Lab on a Chip. 2025. https://doi.org/10.1039/D4LC00567H

- Paul, R.D.; Seiffarth, J.; Scharr, H.; Nöh, K.: Robust approximate characterization of single-cell heterogeneity in microbial growth. IEEE International Symposium on Biomedical Imaging. 2024. https://doi.org/10.1109/ISBI56570.2024.10635267

- Kasahara, K.; Leygeber, M.; Seiffarth, J.; Ruzaeva, K.; Drepper, T.; Nöh, K.; Kohlheyer, D.: Enabling oxygen-controlled microfluidic cultures for spatiotemporal microbial single-cell analysis. Frontiers in microbiology. 2023. https://doi.org/10.3389/fmicb.2023.1198170

- Scherr, T.; Seiffarth, J., Wollenhaupt B.; Neumann, O.; Schilling, M.P.; Kohlheyer, D.; Scharr, H.; Nöh, K.; Mikut, R.: microbeSEG: A deep learning software tool with OMERO data management for efficient and accurate cell segmentation. Plos one. 2022. https://doi.org/10.1371/journal.pone.0277601

- Schito, S.; Zuchowski, R.; Bergen, D.; Strohmeier, D.; Wollenhaupt, B.; Menke, P.; Seiffarth, J.; Nöh, K.; Kohlheyer, D.; Bott, M.; Wiechert, W.; Baumgart, M.; Noack, S.: Communities of Niche-optimized Strains (CoNoS)–Design and creation of stable, genome-reduced co-cultures. Metabolic engineering. 2022. https://doi.org/10.1016/j.ymben.2022.06.004

- Sachs, C.C.; Ruzaeva, C., Seiffarth, J., Wiechert, W.; Berkels, B.; Nöh, K.: CellSium: versatile cell simulator for microcolony ground truth generation. Bioinformatics Advances. 2022. https://doi.org/10.1093/bioadv/vbac053

- Rohr, K.; Geinitz, B.; Seiffarth, J., Anbarani, A.; Bernauer, S.; Moch, M.; Tenhaef, J.; Wiechert, W.; Nöh, K.; Oldiges, M.: Insights into the morphology-productivity relationship of filamentous fungi through small-scale cultivation and automated microscopy of Thermothelomyces thermophilus. Biotechnology Progress. 2025. https://doi.org/10.1002/btpr.3528

Conference and symposium talks

- "Deep-Learning Meets Microbial Live-Cell Imaging: Powerful Analysis Workflows from Annotation to Prediction" at the Helmholtz Imaging Conference in Berlin. 2022.

- "Live-cell mAIcroscopy: Cracking the challenge to image living cells with real-time event attention" at the Helmholtz Imaging Conference in Heidelberg. 2024.

- ”Beyond observation in microbial live-cell imaging: Exerting control on microbial populations using real-time AI image analysis and response triggering” at the 7th Machine Learning and AI in Bio(Chemical) Engineering Conference in Cambridge. 2024.
- ”Real-time Closed-loop Microbial Growth Control using Dynamic Photosensitizer Activation” at the 4th Optogenetic Technologies and Applications Conference in Boston. 2024.
- ”Tracking one-in-a-million: Large-scale benchmark for microbial single-cell tracking with experiment-aware robustness metrics” at the BioImage Computing Workshop at the European Conference on Computer Vision (ECCV) in Milano. 2024.

Conference Posters

- ”Understanding Microbial Communities - The Microfluidic Toolbox” at EMBL’s 24th annual PhD Symposium in Heidelberg. 2022.
- ”Automated Life-Cell Imaging meets AI - From Microbial Sensing to Insight and Control at Scale” at the Sense-compute-Response Workshop in Darmstadt. 2023.
- ”Customizable and Interactive Visualizations for Investigating Spatio-temporal Single-cell Information” at the International Symposium for Biomedical Imaging (ISBI) in Athens. 2024.
- ”mAIcroscopy: Democratizing smart AI microscopes for real-time live-cell analysis”. EMBL AI and Biology. 2024.
- ”Smart Microscopy - Exploiting real-time information for microbial growth control”. EPFL Latsis Symposium. 2024.

Code repositories and software packages

- ObiWan-Microbi: `https://github.com/hip-satomi/ObiWan-Microbi`
- SegServe: `https://github.com/hip-satomi/SegServe`
- Interactive Visualizations: `https://github.com/JuBiotech/Supplement-to-Seiffarth-et-al.-ISBI-2024`
- acia-workflows: `https://github.com/JuBiotech/acia-workflows`
- acia-core: `https://github.com/JuBiotech/acia-core`

Datasets

- Tracking one-in-a-million: `https://doi.org/10.5281/zenodo.7260136`
- microbeSEG dataset: `https://doi.org/10.5281/zenodo.6497714`
- Dynamic oxygen conditions: `https://doi.org/10.5281/zenodo.13982746`
- Fluorescence-labeled Co-cultivation: `https://git.nfdi4plants.org/j.seiffarth/bund-et-al_2025`

Contents

Abstract iii

Kurzfassung v

Acknowledgements vii

Preface ix

Published work x

Acronyms xix

Glossary xxii

1 Motivation and background 1
- 1.1 Importance of understanding microbial behavior in complex environments 1
- 1.2 Spatio-temporal single-cell observations at high-throughput 2
- 1.3 From passive observation to real-time influence 4
- 1.4 The Microfluidic live-cell imaging technology 6
  - 1.4.1 Designing and manufacturing microfluidic high-throughput devices 6
  - 1.4.2 Microfluidic live-cell imaging experiments 8
  - 1.4.3 Image and data analysis pipeline 8
  - 1.4.4 Challenges in single-cell imaging and analysis 9
- 1.5 Scope of this thesis 10

I Data foundation and analysis 13

2 ObiWan-Microbi: OMERO-based integrated workflow for annotating microbes in the cloud 15
- 2.1 Motivation and significance 16
- 2.2 Software Description 18
  - 2.2.1 Remote deep learning segmentation execution 18
  - 2.2.2 Efficient browser-based segmentation and tracking annotation in time-lapse images 18
  - 2.2.3 Microservice architecture 20

2.3 Illustrative Examples . . . 20
2.4 Impact . . . 21
2.5 Conclusion . . . 23

3 Tracking one-in-a-million: microbial cell tracking benchmark with experiment-aware metrics 25
3.1 Motivation . . . 26
3.2 Related Work . . . 28
3.3 Benchmark Dataset . . . 29
3.3.1 Data acquisition . . . 30
3.3.2 Semi-automated segmentation & tracking annotation . . . 30
3.3.3 Dataset statistics . . . 30
3.3.4 Implications of imaging interval subsampling . . . 31
3.4 Experiment-aware microbial live-cell tracking metrics . . . 32
3.4.1 Cell tracking metrics . . . 32
3.4.2 *EATM* cell tracking metric for MLCI . . . 34
3.5 Tracking evaluation . . . 35
3.6 Conclusions . . . 37

4 Acia-workflows: automated single-cell imaging analysis for scalable and deep learning based live-cell imaging analysis workflows 39
4.1 Motivation . . . 40
4.2 Analysis pipeline and workflow . . . 43
4.2.1 Loading 2D+t microscopy time-lapse data . . . 43
4.2.2 Cell segmentation . . . 44
4.2.3 Cell tracking . . . 44
4.2.4 Visualization, data analysis and insight generation . . . 45
4.2.5 From pipeline to workflow . . . 45
4.2.6 Scaling to quantitative analysis of high-throughput experiments . . . 46
4.3 Results . . . 46
4.3.1 Measuring microbial population growth . . . 46
4.3.2 Co-culture characterization using fluorescently labeled strains . . . 48
4.3.3 Quantifying temporal responses of individual cells to environment changes . . . 49
4.3.4 Scaling to quantitative insight in high-throughput MLCI . . . 51
4.4 Discussion & Outlook . . . 54

5 Customizable and interactive visualizations for investigating spatio-temporal single-cell information 55
5.1 Motivation . . . 56
5.2 Customizable visualization views . . . 57
5.2.1 The lineage view . . . 57
5.2.2 The descendants view . . . 57
5.2.3 The distribution view . . . 59
5.3 Results & discussion . . . 59
5.4 Conclusion . . . 60

II Smart and event-driven microfluidic live-cell imaging 61

6 Digital Microfluidic Chip: enabling smart microscopy in versatile high-throughput microfluidic chip systems 65
6.1 Motivation . . . 66
6.2 Designing the Swiss Army Knife - a versatile & high-throughput digital microfluidic chip design . . . 68
6.3 Deep learning-based reference marker detection . . . 68
6.3.1 Data recording & annotation . . . 70
6.3.2 Training reference marker detection models . . . 70
6.3.3 Evaluation of reference marker detection . . . 71
6.4 RoI masking and automated error detection . . . 74
6.4.1 Marker pairing and automated error detection . . . 74
6.4.2 Rotation and masking . . . 74
6.4.3 Complete pipeline performance . . . 74
6.5 Fast blueprint map alignment for high-throughput experiments . . . 77
6.5.1 Map alignment . . . 77
6.5.2 The mapping procedure . . . 78
6.5.3 RoI position mapping error . . . 78
6.6 Discussion & Limitations . . . 80
6.7 Conclusion . . . 81

7 JediOS: just event-driven imaging operating system 83
7.1 Motivation . . . 84
7.2 High-throughput live-cell imaging - a multi-tasking perspective . . . 85
7.3 Event-driven imaging language (EDIL) . . . 87
7.3.1 RoI state-machine for a basic imaging experiment (SM1) . . . 88
7.3.2 Adaptive imaging interval (SM2) . . . 89
7.3.3 Optogenetic excitation pulse screening (SM3) . . . 91
7.3.4 Closed-loop optogenetic growth control (SM4) . . . 91
7.4 Scheduling state-machine executions in high-throughput experiments . . . 92
7.4.1 Scheduling the execution of a single state-machine . . . 93
7.4.2 Scheduling the execution of multiple state-machines for high-throughput imaging 94
7.4.3 Scheduling algorithm . . . 95
7.5 Simulating high-throughput experiments on the JediOS platform . . . 96
7.5.1 Modeling cell population development . . . 97
7.5.2 Implementation of the in-silico simulation platform . . . 98
7.6 Simulation studies . . . 99
7.6.1 Simulation study 1 - basic imaging experiments (SM1) . . . 99
7.6.2 Simulation study 2 - Adaptive imaging (SM2) . . . 101
7.6.3 Simulation study 3 - high-throughput light excitation pulse screening (SM3) . 104
7.7 Discussion and Conclusion . . . 106
7.7.1 Asynchronous Execution . . . 107
7.7.2 Continuous integration of software-defined experiment designs . . . 107

7.7.3 Advanced scheduling strategies . . . 108
7.7.4 Mixing state-machine designs . . . 108
7.7.5 From simulation to the real experiment . . . 109

8 Smart live-cell experimentation using JediOS at the example of high-throughput photosensitizer stress induction 111
8.1 Motivation . . . 112
8.2 Implementing the microscopy platform . . . 113
8.3 Real-time image analysis pipeline (RT-IAP) . . . 115
8.4 Smart MLCI experiment application: High-throughput photosensitizer screening . . . 117
8.4.1 Genetically-encoded photosensitizers: An optogenetic tool for light-based intracellular stress induction. . . . 118
8.4.2 Screening photosensitizer sensitivity to dynamic light activation . . . 118
8.4.3 Preventing light scattering and cross-illumination . . . 120
8.5 Smart MLCI experimentation workflow leveraging real-time information and event-driven imaging . . . 120
8.5.1 The five steps of smart MLCI experimentation . . . 121
8.5.2 Step 1 - Map creation . . . 121
8.5.3 Step 2 - Initial screening . . . 121
8.5.4 Step 3 - Initial screening dashboard and suitable RoI selection . . . 122
8.5.5 Step 4 - Conducting the event-driven screening experiment . . . 125
8.5.6 Step 5 - Live-view of the experiment . . . 127
8.6 Discussion and Outlook . . . 131
8.6.1 Limitations . . . 132
8.6.2 Outlook . . . 133
8.7 Summary . . . 134

9 Data-driven analysis of the high-throughput photosensitizer stress induction screening 137
9.1 Measuring single-cell excitation fluorescence . . . 138
9.1.1 Calibrating the excitation pulse energy . . . 138
9.1.2 Excitation pulse energy and average cell fluorescence . . . 139
9.1.3 Heterogeneity in single-cell fluorescence response . . . 140
9.2 Quantifying stress-induction effects on the cell population level . . . 143
9.2.1 Temporal cell population development . . . 143
9.2.2 Quantifying stress-induction impact on the temporal cell population development 143
9.3 Quantifying stress-induction impact at the single-cell level . . . 147
9.3.1 Challenges in single-cell tracking in PS experiments . . . 147
9.3.2 Measuring single-cell stress-induction response . . . 149
9.3.3 Evaluating temporal single-cell stress-induction response . . . 150
9.3.4 Identifying single-cell heterogeneity in the lineage tree . . . 152
9.3.5 Combining lineage and population measures for quantitative single-cell offspring quantification . . . 154
9.4 Discussion & Outlook . . . 155

9.5 Summary 156

10 Discussion and Outlook **157**
10.1 Data Benchmarking & Analysis 157
10.2 Smart MLCI experimentation 158
10.3 Towards cybergenetic growth control using photosensitizers 159
10.4 Screening additional cell influencing factors 161
10.5 The costs of MLCI experimentation 161
10.6 Distributed experimentation in multi-microscope setups 165
10.7 Beyond bacteria towards eukaryotic cells 166
10.8 Autonomous experiment design for high-throughput screenings 166

11 Conclusion **169**

List of Figures **173**

List of Tables **176**

Bibliography **177**

Curriculum Vitae **195**

Acronyms

ACFI average cell fluorescence pixel intensity. 139, 140, 141, 142, 145, 146, 147

`acia` Automated single-cell image analysis. 10, 39, 41, 42, 43, 44, 46, 48, 49, 51, 54, 115

`acia-workflow` `acia-workflow`. 39, 40, 42, 43, 45, 46, 49, 54

AnnUI Annotation User Interface. 17, 18, 19, 20, 22, 173

AOGM Acyclic Oriented Graph Matching. 29, 32, 34

AP Average Precision. 71

ASCFI average single-cell fluorescence intensity. 140, 141, 142, 143, 147, 151, 152, 154, 155

B. subtilis *Bacillus subtilis*. 20, 22

BHI Brain Heart Infusion. 30, 47, 162

C. glutamicum *Corynebacterium glutamicum*. 20, 21, 22, 28, 29, 30, 31, 47, 48, 50, 52, 53, 102, 118, 162

CAD Computer Aided Design. 6, 7, 68, 69, 78, 79

CC Cell Count. 46, 47, 48, 52, 53, 63, 64, 97, 98, 115, 117, 122, 125

CDT Cell doubling time. 150, 151, 152, 174

CGF Cell growth factor. 150, 151, 152, 174

CHO Chinese hamster ovary. 166

CI Continuous Integration. 20, 87, 99

COC Cell offspring count. 154, 155

COT Cell offspring TSCA. 154, 155

CPU Central Processing Unit. 85, 86, 101, 165

CT Complete Tracks. 29

CTC Cell Tracking Challenge. 16, 17, 28, 29, 32, 34, 35, 37

CV Coefficient of Variation. 141, 142

DL deep learning. 16, 17, 18, 19, 20, 21, 22, 23, 26, 28, 173

DLS Deep learning segmentation. 21, 23

DMC Digital Microfluidic Chip. vi, 10, 64, 65, 66, 67, 68, 74, 77, 78, 79, 80, 81, 83, 111, 115, 116, 117, 118, 121, 122, 123, 125, 131, 133, 158, 169

E. coli *Escherichia coli*. 20, 22, 49, 51, 52, 53, 68

EATM Experiment-Aware Tracking Metrics. 32, 34, 35, 36, 37

ECN Experiment Control Node. 115, 116

EDIL Event-driven Imaging Language. 62, 64, 83, 85, 87, 88, 92, 96, 98, 107, 112, 158, 160, 169

EP excitation pulse. 91, 118, 119, 120, 122, 127, 128, 129, 130, 131, 132, 137, 138, 140, 143, 145, 146, 147, 149, 150, 152, 154, 155, 156, 159, 160, 170

EPE excitation pulse energy. 118, 119, 127, 131, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 153, 154, 155, 156, 159, 160, 161, 166, 170

FAIR findable, accessible, interoprable, and reusable. 15, 17, 20, 21, 23

FL fluorescence. 91, 98, 113, 114

FN False Negatives. 34

FP False Positives. 34

FPS Frames per second. 71, 74, 75

GB Gigabyte. 3, 8, 39

GNN Graph Neural Network. 29

GPU Graphical Processing Unit. 18, 117

GT ground truth. 16

HOTA Higher Order Tracking Accuracy. 29

HTC Hybrid Task Cascade. 18

HTLCI High-throughput Live-Cell Imaging. 85, 86, 87

IGR Instantaneous Growth Rate. 51, 52, 92, 93, 122, 123, 124, 125

ILP Integer Linear Program. 29

IoU Intersection over Union. 71

IPTG Isopropyl-β-D-1-thiogalactopyranosid. 118, 119

JediOS Just event-driven imaging operating system. iv, vi, 6, 10, 11, 62, 63, 64, 66, 83, 84, 85, 87, 92, 97, 98, 99, 106, 107, 108, 109, 111, 112, 113, 114, 115, 116, 121, 127, 130, 131, 133, 134, 137, 156, 158, 160, 161, 162, 165, 166, 169

JSON JavaScript Object Notation. 18

KPI Key Performance Indicator. 46

LAP Linear Assignment Problem. 28, 29

mAP mean Average Precision. 71, 72, 73, 75

MC Mitotic Branch Correctness. 29

MD Microfluidic design. 68

MHT Multi Hypothesis Tracking. 29

MLCI Microfluidic Live-Cell Imaging. iii, iv, v, vi, 2, 3, 4, 5, 6, 8, 9, 10, 11, 16, 17, 19, 20, 21, 23, 25, 26, 27, 28, 32, 37, 39, 40, 41, 42, 43, 45, 46, 48, 51, 52, 54, 56, 61, 62, 63, 64, 68, 83, 84, 85, 88, 89, 92, 97, 98, 99, 111, 112, 113, 117, 121, 122, 125, 131, 132, 133, 134, 135, 137, 155, 156, 157, 158, 159, 161, 162, 164, 169, 170, 173

MOTA Multiple Object Tracking Accuracy. 29

MTOS Multi-Tasking Operating System. 86, 87

ObiWan-Microbi Omero-based integrated workflow for annotating microbes in the cloud. 10, 15, 17, 18, 20, 21, 23, 30

OD Optical Density. 30

OFM Open Flexure Microscope. 162, 163, 164

OMERO Open Microscopy Environment Remote Objects. 15, 43, 46

PC phase-contrast. 89, 91, 98, 113, 114

PCDD Predicted cell development delay. 150, 151, 152, 174

PDMS Polydimethylsiloxane. 6, 7, 8, 78

PDT Population doubling time. 145, 146, 148, 174

PGF Population growth factor. 145, 146, 147, 148, 174

PPDD Predicted population development delay. 145, 146, 148, 174

PS photosensitizer. vi, 64, 112, 113, 117, 118, 119, 121, 122, 125, 128, 129, 131, 132, 134, 137, 138, 139, 140, 141, 143, 145, 147, 149, 154, 155, 156, 159, 160, 161, 166, 170, 174

PU Predicted Utilization. 96, 99, 100, 101, 103, 104, 105, 106, 130

RD Request Delay. 94, 95, 99

RM Robustness Metric. 32, 35, 36, 37

RoI region of interest. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 62, 63, 64, 65, 66, 67, 68, 69, 74, 76, 77, 78, 79, 80, 81, 83, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 111, 112, 113, 115, 116, 117, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 140, 143, 144, 147, 150, 155, 158, 159, 160, 161, 162, 163, 164, 165

ROS Reactive oxygen species. 118, 119, 138, 140, 143

RT-IAP Real-time Image Analysis Pipeline. 63, 64, 111, 113, 115, 117, 122, 124, 131, 133

SAK Swiss Army Knife. 66, 67, 68, 69, 70, 74, 76, 80

SAM2 Segment Anything 2. 70

SegMSA Segmentation Microservice Architecture. 17

SegServe Segmentation Service. 17, 18, 19, 20, 21

SM1 Basic imaging state-machine. 88, 89, 93, 94, 99, 100, 101, 102, 104, 109, 120, 125, 127, 129, 174

SM2 Adaptive imaging interval state-machine. 90, 101, 102, 103, 174

SM3 Optogenetic single-cell screening state-machine. 91, 92, 104, 109, 119, 120, 130, 159, 174

SM3⋆ Adapted optogenetic single-cell screening state-machine. 119, 120, 127, 128, 129, 139, 144, 174

SM4 Closed-loop optogenetic growth control state-machine. 92, 93, 161

sMLCI smart Microfluidic Live-Cell Imaging. 6, 113, 118, 120, 121, 127, 131, 132, 133, 137, 140, 158, 159, 161, 162, 164, 165, 166, 167, 170

SOLA LUMENCOR SOLA SE II. 139, 141

SOTA state-of-the-art. v, 9, 10, 28, 40, 41, 42, 43, 44, 157

TCA Total Colony Area. 46, 47, 48, 52, 53, 164

TOIAM Tracking one-in-a-million. 10, 29, 31, 33, 35, 37

TP True Positives. 34

TSCA Total Single-Cell Area. xix, 46, 47, 48, 49, 50, 51, 52, 53, 63, 64, 90, 92, 97, 98, 102, 103, 104, 105, 106, 115, 117, 122, 123, 124, 128, 129, 130, 143, 144, 145, 146, 147, 148, 154, 160

TSP Traveling Salesman Problem. 108

YOLO You Only Look Once. 67, 70, 71, 72, 73, 74, 75, 78

Glossary

dora-rs Dora-rs is a data-flow–oriented robotics framework that models applications as interconnected computational nodes exchanging zero-copy messages based on Apache Arrow, thereby enabling efficient and deterministic communication (`https://dora-rs.ai/`). It provides a language-agnostic, modular, and low-latency execution environment for robotics and AI workloads, facilitating distributed deployment, structured integration of heterogeneous components, and reproducible system composition. 115, 116, 117, 133

NGINX nginx ("engine x") is an HTTP web server, reverse proxy, content cache, load balancer, TCP/UDP proxy server, and mail proxy server. 21

REST API A Representational State Transfer Application Programming Interface (REST API) is an programming interface that lets different software systems communicate over HTTP by following the principles of Representational State Transfer—using standardized operations like GET, POST, PUT, and DELETE. It treats server-side data as resources identified by URLs, enabling stateless, scalable, and predictable interactions between clients and servers. 17, 18, 114, 121, 131, 134

Chapter 1

Motivation and background

Contents

1.1 Importance of understanding microbial behavior in complex environments 1

1.2 Spatio-temporal single-cell observations at high-throughput 2

1.3 From passive observation to real-time influence 4

1.4 The Microfluidic live-cell imaging technology 6

1.5 Scope of this thesis 10

Living cells form the fundamental units of life, constituting the structural and functional basis of the biological systems on Earth. They are ubiquitous, comprising the human body and forming the basis of microbial communities in the environment while playing crucial roles in ecosystems and biotechnological processes. Despite their omnipresence and importance, their behavior remains only partially understood. Their small size, extreme variability, and development in space and time require advanced technology to foster our understanding of this living microcosmos.

1.1 Importance of understanding microbial behavior in complex environments

The human body alone consists of an estimated 30 trillion human cells, forming a fascinating and complex human being. However, they are accompanied by approximately an equal number of microbial cells, forming a complex ecosystem known as the human microbiome [1, 2]. Microbial cells are known to colonize the gut [3], skin [4, 5], respiratory tract [6], and reproductive organs [7, 8], shaping physiological processes and immune responses. These microbial communities are not merely passengers but are essential for processes such as digestion, protection from pathogens, and the development of the immune system [6]. Moreover, bacteria in the human gut microbiome have been linked to the efficacy of cancer treatments such as immunotherapy [9].

Conversely, microbial cells can also cause diseases. Pathogenic bacteria and fungi spread contagiously, disrupt homeostasis, and lead to acute and chronic conditions. Famous examples of such disease-causing microbial organisms are the *Mycobacterium tuberculosis* causing tuberculosis

or *Streptococcus pneumoniae* representing a common cause for pneumonia [10, 11]. Apart from immediate harm, also the balance of diversity in the microbiome plays a pivotal role in maintaining health and preventing disease. On the one hand, low diversity in the gut microbiome has been linked to obesity [12] and inflammatory bowel disease [13]. On the other hand, high microbial diversity has been linked to diseases such as bacterial vaginosis [14]. Consequently, microbial organisms are both life-sustaining and potentially harmful, highlighting the need for building tools that give us a detailed understanding of their behavior and temporal development under certain environmental conditions.

Beyond the importance of microbial organisms for the ecosystem and human health, they are essential in modern sustainable biotechnology. Microbes are harnessed in diverse applications ranging from industrial enzyme production to biofuel generation, bioplastics, and the development of sustainable food and pharmaceutical products. For example, microbial organisms are key for producing vaccines [15], food additives such as vitamins and amino acids [16], and biodegradable polymers [17].

Thus, microbial organisms and communities are of fundamental importance for human health, the Earth's ecosystem, and sustainable biotechnological production. Consequently, fostering the understanding of their behavior is crucial for advancing in all three domains. However, it turns out that investigating and understanding the behavior of microbial cells remains challenging: The complexity and variability in microbial communities, their temporal interactions and reactions to external influences in space and time, the dense cell populations they form, and the high number of individual living cells make understanding their behavior a complex endeavor. Added to that, multiple genetically identical cells forming a cell population have been shown to exhibit remarkably different behaviors. While some cells rapidly divide, others may enter dormancy [18] or initiate stress responses [19]. These differences in single-cell behavior, called single-cell heterogeneity, occur across time and for individual cells, such that single-time-point observations or average bulk measurements often miss the temporal dynamics, requiring long-term tracking with single-cell resolution [20].

To fully characterize the behavior of microbial cells across various influences, it is essential to investigate the behavior of individual living cells over time and space, their development, interactions with other cells, and response to external influences - all of which are dynamic, tightly coupled, and challenging to capture simultaneously. For these investigations, we need a technology that (a) provides precise and dynamic control of the cultivation environment and influencing factors such as cultivation medium, gassing, and optogenetic influences, (b) works at high-throughput enabling the robust and quantitative screening of these influencing factors, and (c) gives single-cell development observations in space and time to investigate temporal cell development, single-cell heterogeneity and complex cell-to-cell interactions.

1.2 Spatio-temporal single-cell observations at high-throughput

In recent years, the development of Microfluidic Live-Cell Imaging (MLCI) combined the spatio-temporal imaging of individual cells and their populations with precise control and high-throughput of modern microfluidic chip systems [19, 21, 22, 23, 24]. Figure 1.1 shows the conduction of such an MLCI experiment: First cells are cultivated and then inoculated into the microfluidic chip cultivation system where cells are trapped in 100s of pico-liter sized cultivation chambers also called regions

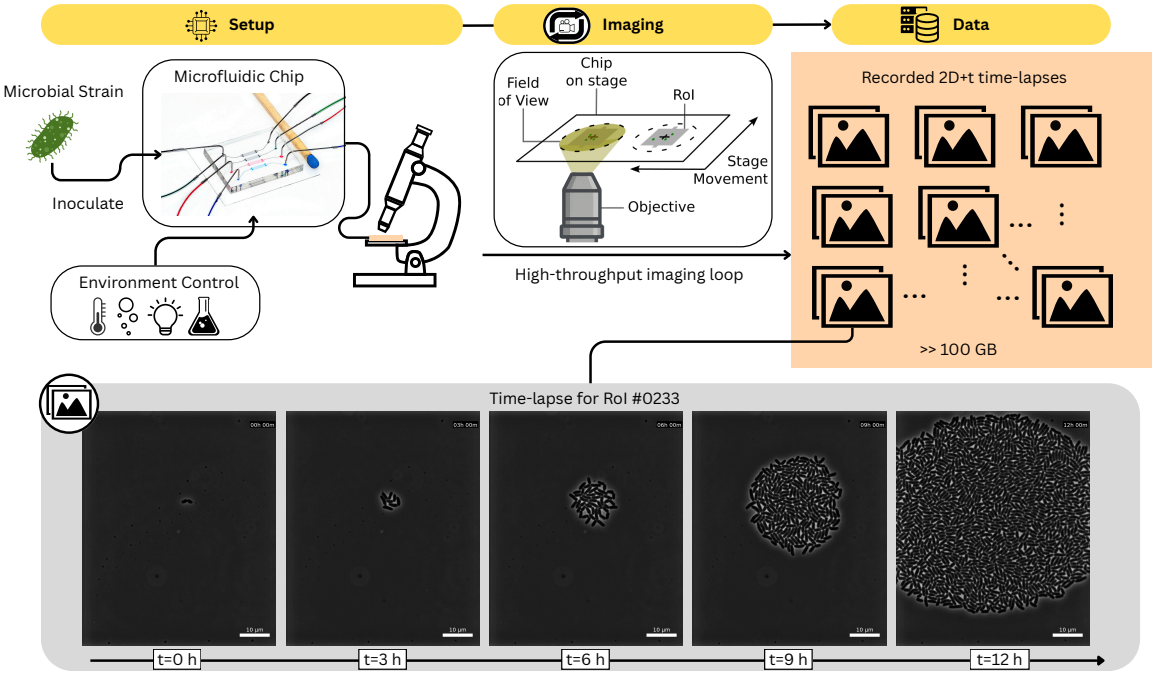

Figure 1.1: **Schematic conduction of a high-throughput MLCI experiment.** A microfluidic chip containing 100s of RoIs is mounted on an automated microscope, microbial cells are inoculated into the microfluidic chip, and cultivated at precisely controlled environment conditions (Setup). In an imaging loop, the microscope continuously captures images of each region of interest by moving the stage (Imaging). The resulting $2D + t$ time-lapse data captures the development of the cell populations in 100s of time-lapses within a high-throughput MLCI experiment leading to hundreds of GB of imaging data (Data). At the bottom, the temporal cell population development within a single RoI is illustrated.

of interest (RoIs). The automated microscope continuously visits the RoI positions and records images, capturing a time-lapse recording of the cell development for each RoI. These time-lapses capture the single-cell development in space and time and are ideally suited to overcome our limited understanding of living cells. However, the high-throughput nature of these experiments leads to hundreds of Gigabytes (GBs) of imaging data recorded in a single experiment per experiment day. With experiments conducted daily at several microscopy setups, these imaging datasets contain a vast treasure trove of insights into microbial life. Exemplary studies have shown that a systematic study of such single-cell behavior in the recorded time-lapses finally unlocks insights into single-cell heterogeneity [19, 22, 25], microbial communities [26], dynamic environment conditions [27], or cellular stress responses [19, 28, 29].

Despite these impressive insights and the large-scale data acquisition using automated microscopy, MLCI has one primary bottleneck: All the observations collected during the experiment are time-lapse imaging sequences. To determine measurements, such as the number of cells or their size, extensive image analysis is required (Figure 1.2). The two major tasks during this image analysis are cell segmentation and tracking. During cell segmentation, individual cells are detected in the recorded microscopy images, and their pixel-precise size is computed. These cell detections on

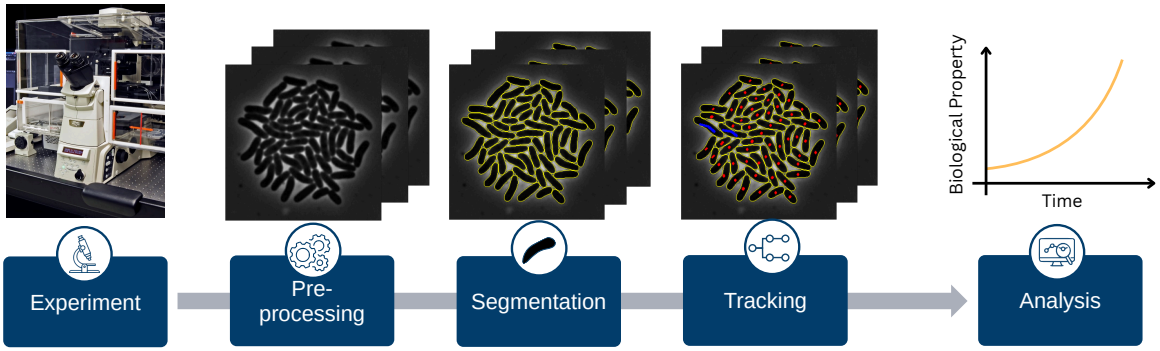

Figure 1.2: **Typical MLCI image analysis pipeline.** In the MLCI experiment, time-lapse sequences for each RoI position are recorded. Each time-lapse sequence is then pre-processed to remove the microfluidic structures (1), segmented to detect every cell instance (2), and tracked to associate cell detections over time and recognize cell movement (red) and cell division (blue). The extracted information is analyzed (4) to extract single-cell insights in extensive data analysis and visualization.

individual images are then connected through time during tracking to reconstruct the temporal development of cells and measure their growth and division behavior. While both tasks have been challenging to address, in recent years, especially the development of deep learning-based methods for segmentation and tracking have changed the field [30, 31, 32, 33]. Sufficiently sized datasets have shown to be crucial to train models on the specific cell morphology, imaging modality and division behavior, and unlock robust, reliable and fast extraction of single-cell measurements.

These single-cell measurements only provide the foundation for generating systematic insights into microbial behavior. Combining and visualizing the extracted measurements across single-cell and population levels is crucial for uncovering and studying the dynamics in single-cell behavior incorporated in the recorded time-lapse data. While software tools such as Omnisegger [34] try to combine the individual analysis steps of segmentation, tracking, and insights generation, they often fail in addressing the large variability in research questions, various data formats, challenges in high-throughput analysis, and are still difficult to use in the interdisciplinary field of life sciences where programming skills vary and accessibility of software is key. With a single-cell analysis platform that overcomes these limitations, systematic high-throughput studies of microbial behavior are finally within reach. Having identified this crucial research gap that hinders the broad application of MLCI, we present solutions for large-scale dataset annotation, and data analysis for MLCI throughout the first part of this thesis. The resulting flexible data analysis platform has been continuously developed, improved and deployed to be used by various users in the lab. For now, this has enabled the analysis of 800+ million cells in more than 2.7 million time-lapse images.

1.3 From passive observation to real-time influence

Despite the immense potential in analyzing single-cell data, most of the conducted MLCI studies are yet purely observational. Thus, the development of cells in cell populations is observed at high-throughput, but without any influence on cell development. Influencing the development of cells

during the experiment would allow us to better study the temporal dynamics hypothesized in complex ecosystems and microbial communities. In recent years, the opportunities to influence cells in microfluidic environments have been improved. For example, Tauber et al. introduced a microfluidic chip system to switch the cultivation medium dynamically [27], Kasahara et al. established a microfluidic chip with precise and dynamic gassing control [35, 36], and optogenetic tools are used to influence cells based on light excitation to, for example, to induce oxidative stress [28] or trigger protein secretion [37]. These increasing opportunities of influence allow us to dynamically alter the environment of the microbial population or trigger specific optogenetic influences while observing the spatio-temporal cell development with single-cell resolution.

Despite these increasing opportunities for influencing living cells, one major limitation remains: typically, MLCI experiments are conducted blindfolded, meaning that during multi-day experiments no measurements are available apart from the recorded images. On the one hand, this leads to a knowledge gap during the experiment, which is especially problematic when technical issues remain unnoticed. Consequently, this can result in the complete loss of valuable single-cell observations. On the other hand, there is no detailed information on the cell population that is influenced, and the effect of the influence remains unclear until the final evaluation after the experiment. For example, triggering influences based on cell population size or development are not yet possible. Especially, closed-loop control applications, as demonstrated by Kumar et al. or Lugange et al. in the field of cybergenetics [38, 37], require real-time readouts to precisely compute the next influence in order to exert closed-loop control on a specific parameter such as cell growth or protein secretion. While such light influences are difficult to utilize in medical therapy and industrial applications due to the abundance of light, the cybergenetics community proposed optogenetics as a rapid prototyping platform to develop control strategies for living cells. Once these control strategies are well understood, they can then be implemented in metabolic controllers inside the cells such that no external light influence is needed anymore [39].

To investigate these control strategies using light-based optogenetic influence on living cells, MLCI is an ideally suited high-throughput screening platform. Light sources are typically available in fluorescence microscope setups, and the high-throughput character of the microfluidic chips promises to accelerate the prototyping and insight generation into optogenetic tools and their effect on living cells. However, conducting such experiments requires a paradigm shift: Instead of passively observing cells during their development, we need to react flexibly to the development of cell populations and trigger influences to control them.

In recent years, smart microscopy has emerged as a field where the imaging routine to observe living cells is no longer static but dynamically adapted during the experiment [40]. Moreover, software platforms for event-driven imaging, such as MicroMator or Cyberscopy, have been developed, where real-time information is extracted from recorded images to trigger events that adjust the imaging routine or even adapt microfluidic parameters such as the medium supply [41, 42]. However, these event-based experiment designs increase the complexity of live-cell imaging experiments and are not designed for high-throughput experimentation. Such high-throughput experiments where many cell populations need to be imaged, analyzed and influenced concurrently come with a utilization and scheduling challenge: the microscope can only be at one RoI position at a time imaging and influencing a cell population and then moves to the next cell population. Moreover, dynamically adapting imaging rates based on the cell development may quickly lead to an over-utilization of the microscope, such that it is not fast enough to image all RoIs in time.

Therefore, we envision a microscopy software platform for MLCI that analyzes recorded images in real-time, simplifies the formulation of event-driven imaging routines that react to real-time information and allows to perform in-silico simulations of high-throughput live-cell experiments in order to verify the correct execution and implementation of the imaging routine and investigate the utilization of the microscope.

Throughout the second part of this thesis, we develop the technological foundation for high-throughput, event-driven imaging experiments by introducing an operating system perspective and a divide-and-conquer strategy to handle the complexity of these experiments. This results in the development of the Just event-driven imaging operating system (JediOS) that allows the simulation of such experiments in-silico and executes them on real microscopy hardware using the same code. Using JediOS, we develop a smart experiment workflow that enables us to greatly enhance the data quality of MLCI experiments, resulting in highly automated experiment conduction while tracking and documenting all decisions. This simplifies the reproducibility and shareability of such smart experiments. Finally, we develop an exemplary smart Microfluidic Live-Cell Imaging (sMLCI) experiment to screen the induction of stress into living cells using optogenetic tools. We utilize the extensive data analysis platform from the first part of this thesis to analyze the response of cells at population and single-cell levels. We are the first to show that the amount of induced stress into each cell is linked to the energy of the light pulses and determines the future development of both the cell population and individual cells.

Throughout this thesis, we combine MLCI technology with microbial domain knowledge, deep learning, and smart microscopy to unlock quantitative, high-throughput insights into living cells under dynamic environmental conditions. These developments denote a paradigm shift in single-cell analysis, moving from passive observation into the active driver's seat of experiments, unlocking completely new smart experiments with autonomous decisions that promise to have a huge impact on the research field of single-cell analysis.

1.4 The Microfluidic live-cell imaging technology

MLCI is the core technology used in this thesis to investigate developing cell populations at high-throughput, single-cell resolution, and precise and dynamic control of environmental influences. Thus, we introduce the integral parts of MLCI used throughout this thesis, covering the design of microfluidic chips, the conduction and analysis of experiments, and the opportunities for influence. While we focus primarily on the digital aspects of the technology that are important for this thesis, a detailed introduction to the MLCI technology in general is given in [21, 43].

1.4.1 Designing and manufacturing microfluidic high-throughput devices

The design and manufacturing of microfluidic chips are key for MLCI as they shape the device used for cultivating and imaging the development of cell populations. Figure 1.3 shows four core steps relevant to this thesis: (1.) the Computer Aided Design (CAD) of the RoI representing the cultivation chamber (violet) and the connection to supply channels (green). (2.) composing 100s to 1000s of these RoIs to a high-throughput microfluidic chip design with medium channels and in- and out-lets (green). (3.) The fabrication of a silicone wafer mold, and (4.) the manufacturing of the disposable microfluidic chips in Polydimethylsiloxane (PDMS) that are then used during the

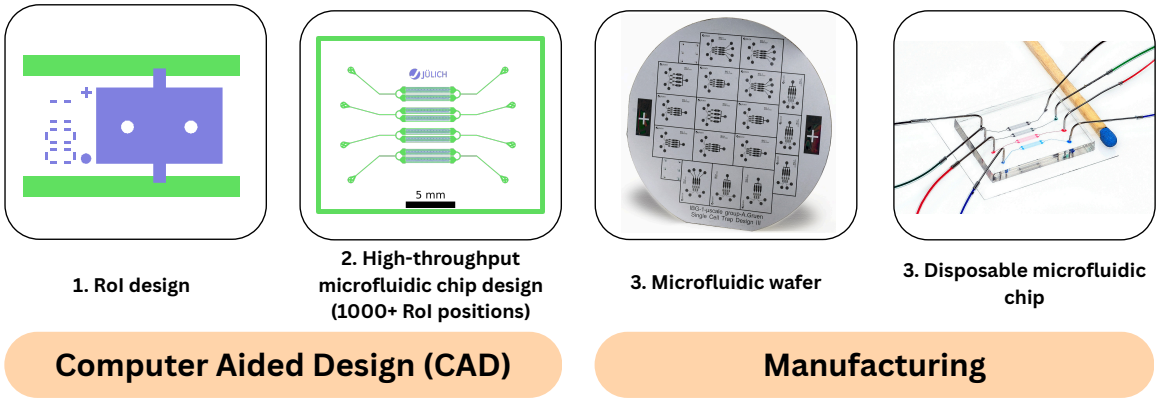

Figure 1.3: **High-throughput microfluidic cultivation chips - From design to manufacturing.** (1.) RoIs are designed with cultivation regions (purple, height approx. 1 μm) and medium supply regions (height approx. 10 μm). (2.) 100s to 1000s of these RoIs form a high-throughput design. This CAD (CleWin5) design contains inlets, outlets, channels (green), and the RoIs for cell cultivation (purple). (3.) Multiple chip designs are combined into a microfluidic wafer design that is manufactured in the Helmholtz Nano Facility. (4.) The wafer serves as a mold to fabricate disposable PDMS chips that are mounted to the microscope and serve as the cultivation platform. The steps (1-2) are purely performed digitally in CAD software, and steps (3-4) involve manufacturing in the lab.

experiment.

For this thesis, especially the first two steps are interesting as they are performed entirely in software using CAD (Figure 1.3). Microfluidic high-throughput chips are designed using Computer Aided Design software in a 2.5D blueprint format (CleWin5). It consists of a 2D sketch denoting supply channels for medium supply (green) and cultivation chambers that are used to trap and cultivate cell chambers (violet), which are referred to as RoIs in the following. Both colors denote different heights of microfluidic structures. While the cultivation chamber is typically manufactured with a height of roughly 1 μm to provide a 2D monolayer for microbial cell growth, the medium channels are usually manufactured with a height of 10 μm, leading to a 2.5D blueprint. The design of the chip also determines the number of RoIs, which in turn determines the maximal throughput of the microfluidic chip and the number of supply channels that allow for cultivations in physically separate and individually supplied channels. In general, arbitrary structures and 2D shapes can be used during the design phase of such microfluidic chip blueprints, giving us large flexibility for creating structured environments. However, the smaller the structures, the more likely they are not correctly formed in the resulting PDMS chip (usually structures smaller than 1 μm become challenging).

The designed microfluidic blueprint is then used to fabricate a silicon wafer. This wafer serves as the mold for multiple microfluidic chips, either of the same or various blueprint designs. The wafer is then repeatedly used to fabricate disposable microfluidic chips using PDMS that are bonded to a glass plate, preparing them for microscopy imaging. While the material and production of the wafer are expensive, the individual PDMS microfluidic chips are disposable and inexpensive. Thus, smart decisions during the design process of the microfluidic chip wafer are crucial such that the resulting

PDMS chips suitable for many MLCI experiments.

1.4.2 Microfluidic live-cell imaging experiments

The disposable microfluidic chip, mounted on a glass plate, is positioned on the stage of an automated microscope. This microscope uses a high-magnification objective (usually 100× magnification) to capture one RoI in its field of view (Figure 1.1). To monitor all RoIs, the microscope moves the microfluidic chip and records images of the individual RoIs at different positions.

The conduction of an MLCI experiment then usually involves the following steps:

1. **Inoculation**: First, the cell suspension is inoculated into the individual supply channels of the microfluidic chip, trapping cells randomly in the RoIs. This procedure is performed manually and needs to be repeated for every microfluidic channel.

2. **RoI selection**: Several RoI positions are selected that contain promising cell populations. The selection is usually performed manually by traversing the microfluidic chip using a joystick and recording the RoI positions.

3. **Imaging loop**: The selected RoI positions are automatically approached one after the other, and images are recorded. These recordings can contain bright-field, phase-contrast, or fluorescence images. The imaging of RoIs is repeated in a loop such that a time-lapse is recorded for every RoI position capturing the development of the cell population. The constant imaging interval defines the time interval between two consecutive images in the recorded time-lapse. The imaging loop is typically performed by third-party software by the microscope vendor, in our case the NIS-Elements software, and the recorded imaging data is then stored on the computer's file system.

1.4.3 Image and data analysis pipeline

The result of such an MLCI experiment is a large amount (10s to 100s of GB) of time-lapse imaging data containing the raw microscopy time-lapses capturing cell populations and microfluidic structures. To extract biological measurements from these time-lapses, an imaging analysis pipeline is required. Figure 1.2 shows the typical MLCI image analysis pipeline consisting of four steps:

1. Pre-processing: During the pre-processing, surrounding microfluidic structures are removed, and the image is cropped to the RoI.

2. Segmentation: During segmentation, individual cell objects are detected in each image, and pixel-precise masks are computed to measure the precise location and extent of each cell.

3. Tracking: During cell tracking, the cell detections are connected through time to reconstruct the life cycle of each cell and identify cell divisions, appearance, or disappearance.

4. Data Analysis & Visualization: During this last step, the extracted measurements in space (segmentation) and time (tracking) are combined into a spatio-temporal view with single-cell resolution across single-cell, population, multi-population, and multi-experiment levels.

In recent years, several tools for live-cell imaging analysis or complete pipelines have been developed. In particular, plugin-based tools like Fiji [44] or napari [45] come with an extensive suite of plugins for segmentation or tracking cells, a broad user base, and good documentation. However, the need to assemble these plugins into a full pipeline for image analysis makes their application in practice challenging, especially for programming beginners in the interdisciplinary field of MLCI. Complete analysis pipelines such as OmniSegger [34], SuperSegger [46], DELTA [47], or CellProfiler [48] usually come with an assembled image analysis pipeline but often only partially cover the four-step analysis pipeline, come with extensive setup overhead, or are designed around a specific set of segmentation and tracking methods limiting their applicability to the wide variety of microbial cells and imaging modalities. Moreover, state-of-the-art (SOTA) segmentation methods, such as Omnipose [30], Cellpose [31], and Contour Proposal Networks [32], as well as tracking methods like Trackastra [33], are evolving rapidly but require modular analysis pipelines that incorporate these models and keep pace with new developments.

1.4.4 Challenges in single-cell imaging and analysis

Based on the conduction of the MLCI experiments and their analysis, we identify two core fields of improvement: (1.) data availability and analysis, and (2.) smart and event-driven microfluidic live-cell imaging based on real-time information.

Benchmarking datasets, such as the Cell Tracking Challenge [49], are crucial for training, improving, and verifying cell segmentation and tracking. However, they often lack time-lapse recordings of microbial cultivations. Thus, a microbial large-scale benchmark is inevitable for training SOTA deep learning methods for the robust and automated analysis of MLCI time-lapses. However, even after extracting segmentation and tracking information, their analysis remains challenging due to the complexity and variety of research hypotheses in MLCI. Some experiments may focus on the growth rates of cell populations, others on fluorescence readouts, while others may apply dynamic conditions and be interested in how living cells respond. The extracted spatio-temporal single-cell information is combined across different levels of granularity, including single-cell, cell population, multi-population, or multi-experiment levels to investigate these different research hypotheses. Serving all these analysis requirements in one tool is challenging and requires a neat combination of SOTA methods for data extraction and flexible data modeling and visualization.

In contrast to the image analysis, the microfluidic experiment conduction still requires many manual steps conducted by the user. For example, the RoI positions are visited manually by the user utilizing a joystick to perform movements with the microscope stage. Thus, selecting and saving RoI positions for the imaging loop involves a high manual workload that scales linearly with the number of concurrently observed RoI positions. However, this discourages high-throughput experiments with 100s to 1000s of RoI positions - one particularly striking advantage of MLCI to collect robust insights through many replicates. Thus, an efficient and smart experiment setup routine is inevitable to perform such high-throughput experiments conveniently.

Apart from the imaging loop setup, the automated microscope conducts the single-cell observation almost blindfolded. Despite the recording of images that can be manually inspected by a human operator, no information is extracted from these images during the experiment. For example, abnormal cell population development is usually only observed during data analysis and often renders the recorded dataset useless. Real-time insights into the running MLCI experiment are essential to

verify its correct conduction and detect and overcome issues early, preventing data loss, especially in multi-day experiments.

While analyzing observations in real-time is usually possible with today's imaging software, exporting the recorded images to the file system, the reaction and adaptation of the imaging routine are usually difficult to implement in third-party imaging software. However, interacting with living cells, reacting to their development, and applying dynamic conditions and influences is key to improving the information extracted from MLCI experiments. Recent applications in cybergenetics [39, 37] have shown that the manipulation of living cells with light is ideally suited for high-throughput MLCI experiments. However, real-time analysis and full software control of the microscope, as well as convenient conduction of such event-driven experiments [41] are inevitable to unlock the high-throughput capabilities of MLCI for such studies but are not yet available.

Throughout this thesis, we address and overcome the mentioned challenges across data analysis and smart MLCI experimentation.

1.5 Scope of this thesis

The remainder of this thesis is structured in two overarching parts (Figure 1.4):

- **Part I - Data Foundation & Analysis**: In **Chapter 2**, we introduce *ObiWan-Microbi* - a semi-automated annotation platform to create large-scale annotations for segmentation and tracking of time-lapse sequences. In chapter **Chapter 3**, we introduce the Tracking one-in-a-million (TOIAM) benchmark dataset and extend existing metrics to make them experiment-aware by incorporating the number of cells and the imaging rate. We show that both experiment parameters have an extensive influence on the tracking performance of SOTA cell tracking methods and should be considered during the experiment design. Both chapters introduce the data foundation for analyzing MLCI data with deep learning methods. In **Chapter 4**, we introduce acia-workflows - a fusion of an extensive image analysis pipeline (`acia`) into Jupyter workflows to unlock extensive and automated image analysis while providing flexible data analysis and visualization capabilities. At the example of three MLCI studies, we demonstrate the capabilities of these workflows and highlight their ability to quantitatively gain insights into living cell behavior using high-throughput experiments. **Chapter 5** extends the analysis capabilities to interactive visualizations that enable the exploration of live-cell imaging data, providing insight into both single-cell and population information and facilitating the development of new hypotheses that can then be quantitatively verified. The tools developed in this first part have been used to analyze more than one billion cell detections.

- **Part II - Smart & event-driven microfluidic live-cell imaging**: In **Chapter 6**, we present the Digital Microfluidic Chip (DMC) concept - a fusion of microfluidic design and deep learning image analysis - to unlock high-throughput imaging with automated RoI position mapping and real-time semantic understanding of microfluidic structures. Therefore, the DMC lays the foundation for real-time insights inevitable for making decisions during a running experiment. In **Chapter 7**, we address the increased complexity of high-throughput for event-driven imaging experiments and develop the JediOS. This software platform performs a paradigm shift for MLCI experiments, introducing a divide-and-conquer approach on

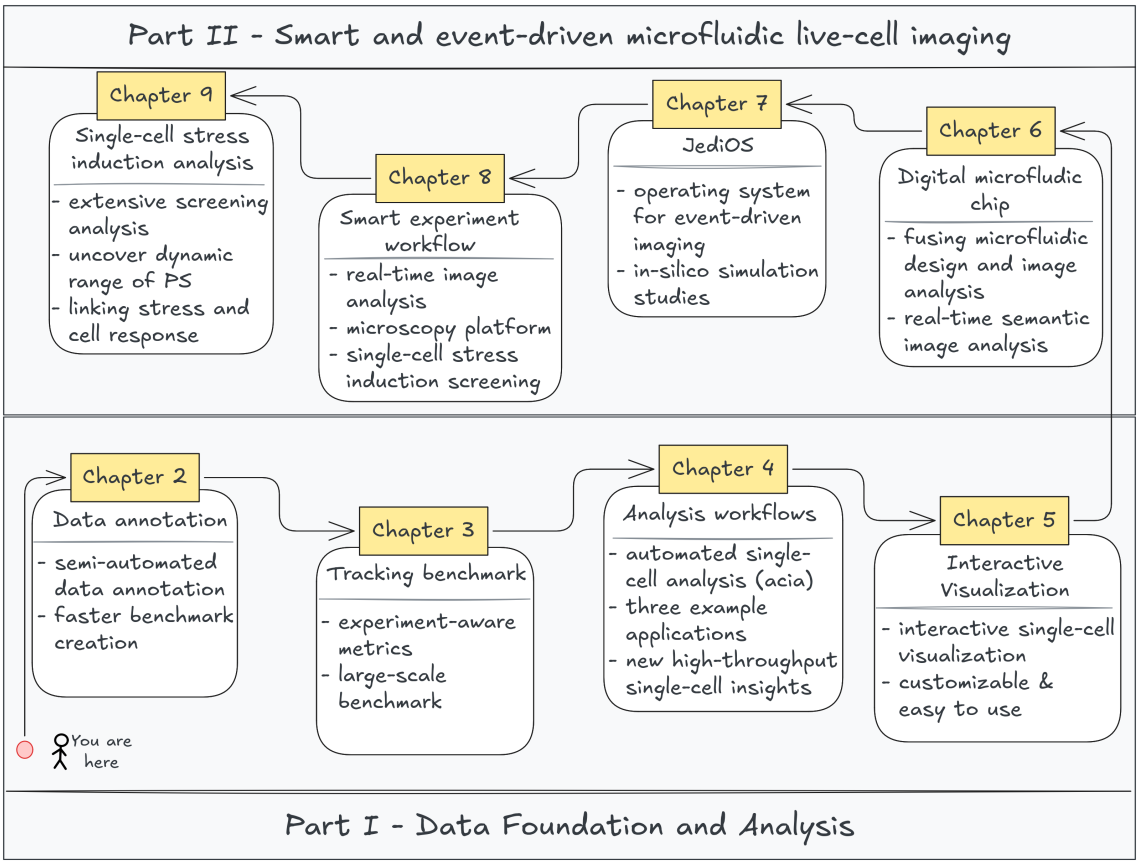

Figure 1.4: **Structure of this thesis.** This thesis is structured into two parts - Data Foundation and Analysis and Smart, and event-driven microfluidic live-cell imaging - with four chapters each.

event-driven imaging where an imaging application controls the imaging routine for a single RoI and the scheduler manages the concurrent execution of 100s of imaging applications on the microscopy platform. Combining this with in-silico simulations, we demonstrate that such event-driven imaging experiments unlock new experiment routines while increasing the throughput and information quality. In **Chapter 8**, we complement JediOS with a real-time analysis pipeline to react to biological readouts during the experiment and showcase its usage for screening the impact of optogenetic-based stress induction. Therefore, we develop a five-step smart MLCI workflow that automates RoI mapping, RoI selection, performs the event-driven high-throughput experiment, and provides real-time insights with customizable dashboards. The resulting smart experiment is highly automated, event-driven, and exemplary applied to the high-throughput screening of optogenetic-based stress induction. In **Chapter 9**, we analyze this optogenetic screening experiment with population and single-cell resolution. We are the first to show the link between the amount of light used for stress induction to the cell population and single-cell response.

The thesis is concluded by an extensive discussion of the achievements and impact of this work and a final conclusion.

Part I

Data foundation and analysis

Chapter 2

ObiWan-Microbi: OMERO-based integrated workflow for annotating microbes in the cloud

Contents

2.1 Motivation and significance 16
2.2 Software Description 18
2.3 Illustrative Examples 20
2.4 Impact 21
2.5 Conclusion 23

Automated microscopy techniques combined with high-throughput microfluidic cultivation systems capture the development of living cells in large-scale imaging datasets. Recent advances in deep learning methods trained on large-scale and precisely annotated data sets have enabled automated processing. However such annotated datasets are often not available for microbial organisms or custom imaging modalities. This lack of datasets is partly caused by the difficulty to annotate segmentation and tracking information for live-cell time-lapse sequences.

To overcome the annotated data bottleneck, particularly in the microbial domain, we present the open-source Omero-based integrated workflow for annotating microbes in the cloud (*ObiWan-Microbi*) platform providing a fast workflow for large-scale annotation of up to hundred thousands of segmentation and tracking annotations in time-lapse imaging data. *ObiWan-Microbi* focuses on easy-to-use semi-automated annotation in the browser eliminating the need for local installation or accelerator hardware, encourages findable, accessible, interoprable, and reusable (FAIR) data management using the Open Microscopy Environment Remote Objects (OMERO) server, and provides convenient collaborative cloud deployment to simplify the creation and long-term development of large-scale annotated data sets. The public availability of such benchmark data sets has the potential to improve data-driven methods, increase comparability among them, and is an essential step towards reliable automated image processing in microbial live-cell microscopy.

The content of this chapter is published in: J. Seiffarth, T. Scherr, B. Wollenhaupt, O. Neumann;

H. Scharr; D. Kohlheyer, R. Mikut, K. Nöh; ObiWan-Microbi: OMERO-based integrated workflow for annotating microbes in the cloud. SoftwareX. 2024. https://doi.org/10.1016/j.softx.2024.101638.

2.1 Motivation and significance

Automated live-cell imaging has become an exceptionally valuable tool for life scientists to investigate the spatio-temporal development of cell populations at the single-cell level [50, 51, 52]. Recent advances in microfluidic systems unlocked the simultaneous cultivation of microbial cultures in hundreds of pico-liter bioreactors on a single lab-on-chip device [22, 53]. In combination with either phase-contrast or fluorescence imaging, MLCI allows performing experiments under various continuous cultivation conditions [54, 55, 21], while observing intracellular processes, such as the production of specific chemical products [56, 57] or the consequences of cellular stress responses [29]. Thus, MLCI turned into a versatile research tool with unique capabilities to elucidate the enigmatic emergence of phenotypic heterogeneity of microbes [58], to shed light into the interactions within microbial communities [59, 60], and promises to become increasingly relevant for modern bioprocess development [61, 21].

In MLCI experiments, high amounts of isogenic cells are repeatedly imaged over multiple generations to be representative of the complexity and variety of microbial behavior and, eventually, to generate conclusive biological insights. In a typical experiment featuring exponentially growing microbes, 300 growth chambers are imaged every five minutes for one day (generating $\approx$ 800 GB), resulting in millions of cell instances. Hence, extracting relevant biological information such as cell sizes, the development of fluorescence in single cells, or growth rates from the acquired time-lapse image sequences requires robust and automated image processing workflows. Two crucial steps in these processing workflows are the identification of individual cells (*cell instance segmentation*) and their association through time (*cell tracking*).

Several advanced deep learning (DL) models have been developed for cell segmentation [62, 63, 31, 30], cell tracking [64, 47], or simultaneous cell tracking and segmentation [65] to establish fully automated image processing. The common key to success in these approaches is the availability of extensive high-quality (i.e., pixel-perfect) ground truth (GT) annotation for effectively training and validating DL models [66, 67]. In the biomedical domain, the collection of such annotated data sets has been established and diverse images and image sequences have been annotated by domain experts and in crowdsourcing efforts [68], with the manual workload reaching up to thousands of work hours [69]. As a result, DL models, trained with these annotated data sets, are now empowering comprehensive image analysis pipelines [70, 71, 72, 69, 73, 47]. A particular example of successful cross-fertilization is the Cell Tracking Challenge (CTC), which provides public benchmarks for biomedical segmentation and tracking methods tailored to eukaryotic cells that push the further development of methods in this field [49]. In contrast, and with only sparse exceptions [74], there is a notable lack of open large-scale MLCI benchmark data sets, holding back similar breakthroughs in the microbial domain [25]. Here, benchmark data capturing the unique challenges of MLCI images – the formation of dense cell colonies, with often simple shapes but frequent cell division, which makes it difficult to identify and track individual cells – are markedly different from existing CTC data sets and, therefore, essential for fine-tuning DL methods and models to microbes and validating automated analysis methods [66].

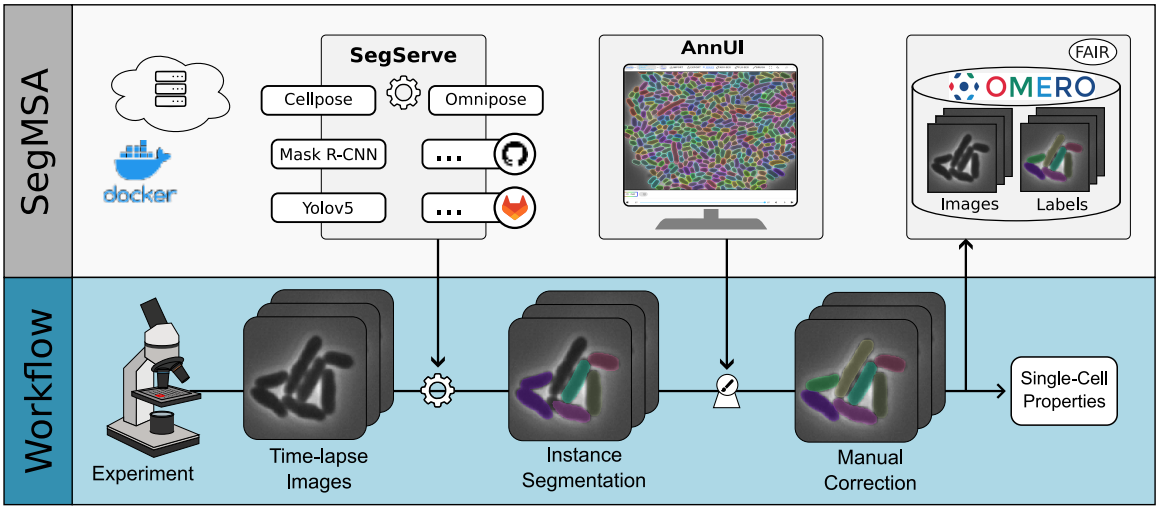

Figure 2.1: ***ObiWan-Microbi* platform overview.** Fully integrated end-to-end workflow for semi-automated image annotation of MLCI time-lapse microscopy in the browser. *ObiWan-Microbi* consists of Segmentation Service (*SegServe*), Annotation User Interface (*AnnUI*), and *OMERO* jointly distributed in the microservice architecture Segmentation Microservice Architecture (*SegMSA*).

Clearly, creating benchmark data resembling the domain-specific MLCI challenges requires an efficient annotation tool for segmentation and tracking, scaling to thousands of cell instances in a single image and thousands of cell tracks per time-lapse sequence. In contrast, in the CTC 2D data sets the maximal number of cell instances in a single image is 651 and the maximal number of divisions in a single sequence is 569. Given these characteristics, establishing a toolset for efficient, at best semi-automated, annotation of microbial image data is key to progress. Specifically, annotation tools need to be simple to operate, require minimal installation effort, and not be overloaded with functionality to ease the entry for and adoption by domain experts. Furthermore, FAIR data management, i.e., meeting findability, accessibility, interoperability and reusability principles, is essential for long-term and continuous development of public benchmark data sets [75].

To address the needs, we developed the *ObiWan-Microbi* software platform that incorporates these features into a single and easy-to-use workflow (Figure 2.1). For *ObiWan-Microbi*, we developed the browser-based annotation software *AnnUI* as key component for large-scale segmentation and tracking annotation in the browser without any local installation. It utilizes the existing *OMERO* server to support FAIR data management, and integrates state-of-the-art DL segmentation methods for semi-automated annotation executed on a remote server using the developed *SegServe* software.

The web app *AnnUI* connects to an *OMERO* data storage and provides an efficient user interface for segmentation and tracking annotation of time-lapse sequences, scaling to hundreds of thousands of annotations and supporting convenient touch or pen devices for high-precision annotation. For scenarios where automated segmentation already yields decent segmentation predictions, we developed *SegServe*, a REST API providing an extendable model zoo for remote DL segmentation execution that eliminates hardware barriers for the end-user, and enables an accelerated semi-automated annotation workflow. *ObiWan-Microbi* combines the existing *OMERO* data management platform, with our newly developed *AnnUI* and *SegServe* tools into a microservice architecture, is open-source, and

comes with detailed video tutorials. Together, *ObiWan-Microbi* provides an efficient workflow for creating large-scale annotated benchmark data sets to push forward reproducible and comparable segmentation and tracking development in the MLCI domain.

2.2 Software Description

The primary objective of *ObiWan-Microbi* is to provide an efficient and user-friendly annotation workflow for MLCI experiments. In order to achieve this, we make three underlying software design decisions: First, we adopted existing DL segmentation methods and developed the service *SegServe* that makes them remotely executable using a REST API. Second, we developed the browser-based user interface *AnnUI*, optimized for efficient large-scale time-lapse annotation and semi-automated annotation, eliminating the need for local software installation or special hardware resources on the user side. Third, we package all software components as containers to make them easy to maintain and deploy them as microservices alongside the *OMERO* platform.

2.2.1 Remote deep learning segmentation execution

In practice, semi-automated annotation using existing DL segmentation methods is often difficult for end-users due to extensive hardware and software requirements. Providing accelerator hardware such as Graphical Processing Units (GPUs) to all users and managing software versions is very cost and time intensive. Thus, we developed *SegServe*, a REST API designed to execute DL segmentation methods with GPU acceleration remotely on a central compute server. *SegServe* utilizes *FastAPI* to provide the REST API backend, but it also comes with an automatically generated user interface. The instance segmentation method is defined by its git URL, version number, and parameters. Together with the image, this information is sent via a REST API request, the method is installed and executed on the remote server, and the resulting instance segmentation is returned in JavaScript Object Notation (JSON) format. The DL segmentation methods are bundled in git repositories and installed automatically with all their software dependencies using *MLFlow* (https://mlflow.org/) dependency management and *anaconda* environments (https://www.anaconda.com/). The on-demand installation guarantees reproducible code execution for different model or software versions and secures adaptability to future DL segmentation methods. Moreover, the software installations are cached and reused for all subsequent segmentation requests. We integrated three off-the-shelf DL segmentation methods for microbial image analysis, namely Cellpose [31], Omnipose [30] and HTC [63] trained on simulated image data [76]. In addition, Yolov5 is available for general bounding-box object detection [77].

2.2.2 Efficient browser-based segmentation and tracking annotation in time-lapse images

To unlock the efficiency of semi-automated annotation workflows, we developed the user-friendly web app *AnnUI*. The application is based on the Angular and Ionic frameworks (https://angular.io/, https://ionic.io/) and is implemented using Typescript (https://www.typescriptlang.org/). *AnnUI* connects to the *OMERO* data backend and allows for navigating image data. A comprehensive feature set for both manual and semi-automated segmentation annotation of time-lapse sequences

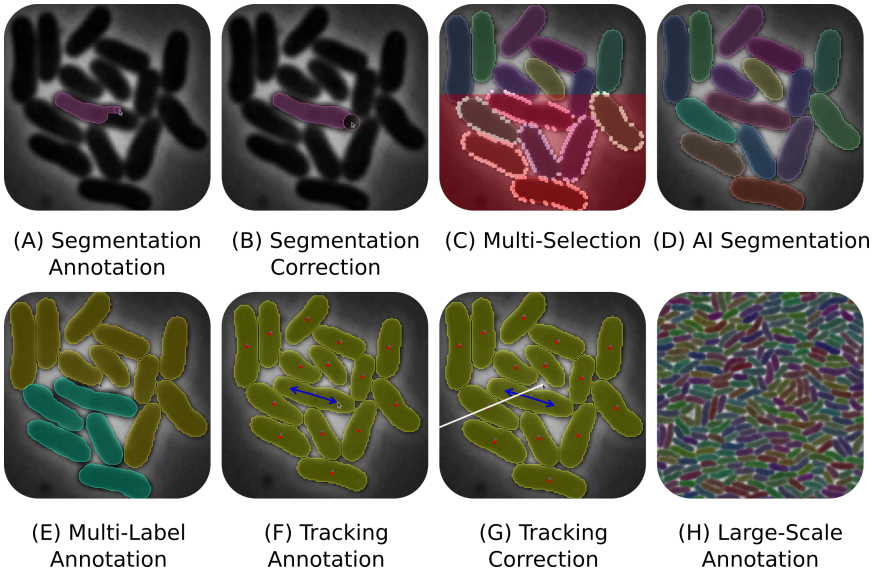

Figure 2.2: **Overview of *AnnUI* functionality:** Manual segmentation annotation (A); correction using the brush tool (B); multi-selection for artifact removal (C); DL segmentation proposals for semi-automated annotation and manual error correction (D); multi-label annotation (E); manual tracking annotation by drawing linking edges between segmentation annotations in consecutive images (F); the removal of tracking links using a track cutting tool that allows drawing a cut (white line) and removes all intersecting track links (G); visualization of a large amount of detected cells with random color for instance separation (H).

leverages the capabilities of the *SegServe* service. Therefore, using *AnnUI* across large and interdisciplinary research teams is simplified as no local software installation and maintenance is needed.

AnnUI provides an optimized tool set for convenient and fast manual annotation, including a brush tool for drawing and correcting segmentation annotations, a multi-selection tool to remove annotations in a rectangular area, and multi-label annotations for complex imaging data (Figure 2.2). The data model records every annotation action separately, providing undo or redo functionality. Therefore, mistakes during the annotation process can be corrected without losing progress. In addition, keeping track of all annotation actions allows quantifying the amount of manual annotation work needed for every single image in a time-lapse sequence. To prevent annotation loss due to network connectivity problems, the data model is automatically synchronized with the *OMERO* backend using *RxJS* events (https://rxjs.dev/).

For high-precision annotation, *AnnUI* supports touch inputs and can be used with professional pen devices or consumer tablets. For semi-automated annotation, three established DL segmentation and one bounding box detection model are provided for prediction, followed by a correction using the manual annotation tools. To integrate custom DL segmentation approaches or fine-tuned hyperparameters, *AnnUI* allows deriving new model configurations for the *SegServe* API.

Based on the cell instance annotation, *AnnUI* facilitates the creation of tracks for individual cells, explicitly focusing on minimizing the number of mouse clicks and improving the quality of the manual annotations. Tracking cells in MLCI is very challenging due to frequent division, dense cell colonies, and the uniform visual appearance. Thus, it is often necessary to incorporate the temporal

context of a cell and its neighborhood to make the correct manual tracking links. We integrated a tracking workflow that emphasizes this temporal context. Upon selecting a cell instance, the subsequent image in the time-lapse is automatically presented to enable the selection of the next cell instance in the track. This sequential procedure is continued until the end of the cell cycle, i.e., from cell appearance to division or disappearance, and gives the user the temporal context of the cell development (Figure 2.2 FG). The cell track is completed with a cell division or disappearance annotation. Furthermore, already annotated cell tracks are visualized using a gray arrow for the movement in the previous frame and blue or red arrows for the movement towards the next frame for dividing and non-dividing cells, respectively (Figure 2.2 F).

To annotate all cell tracks, we incorporated a quick finder method highlighting the next cell instance that still needs to be tracked. Furthermore, any erroneous tracks or links can easily be corrected using the undo/redo functionality or the cutting tool, which allows users to draw a line to remove all intersecting tracking links (Figure 2.2).

2.2.3 Microservice architecture

To facilitate FAIR data principles, *ObiWan-Microbi* utilizes the *OMERO* service for image data handling and annotation storage, including user and access management. Thus, software compatible with *OMERO*, such as *Fiji* [44] or *Napari* [78] can directly use the annotations for further analysis, for example, to extract single-cell properties.

ObiWan-Microbi combines *AnnUI*, *SegServe*, and *OMERO* into a Docker-based microservice architecture (Figure 2.3). The modularization into containers and the definition of their network interactivity allow for a convenient installation procedure of the full platform with only three commands in the command line. During development, errors in the setup are detected using Continuous Integration (CI) pipelines that automatically build and test the individual containers. Furthermore, the microservice architecture allows the integration of custom containers, adapting the platform to existing infrastructure, or deploying it in the cloud.

All in all, *ObiWan-Microbi* establishes a workflow tailored to MLCI data and designed for efficiency, usability and scalability to support large-scale annotation efforts. Its extensible microservice architecture and the remote execution of new DL models keeps installation and future maintenance efforts minimal.

2.3 Illustrative Examples

We present the functionality of the *ObiWan-Microbi* annotation with two examples. First, we demonstrate automated DL segmentation on microbial images to underline the importance of having a model zoo to choose from when performing semi-automated annotation. Second, we showcase large-scale data sets that have already been created using the platform.

During semi-automated annotation, the workload for manual correction highly depends on the performance of existing DL segmentation models. Figure 2.4 shows the segmentation results displayed in the *AnnUI* interface for exemplary phase-contrast images of three established microbial organisms, namely *Escherichia coli* (*E. coli*), *Corynebacterium glutamicum* (*C. glutamicum*), and *Bacillus subtilis* (*B. subtilis*). We observe that *Omnipose* performs the best pre-annotation for a follow-up manual correction, while *HTC* and *Cellpose* require higher manual annotation efforts. Us-

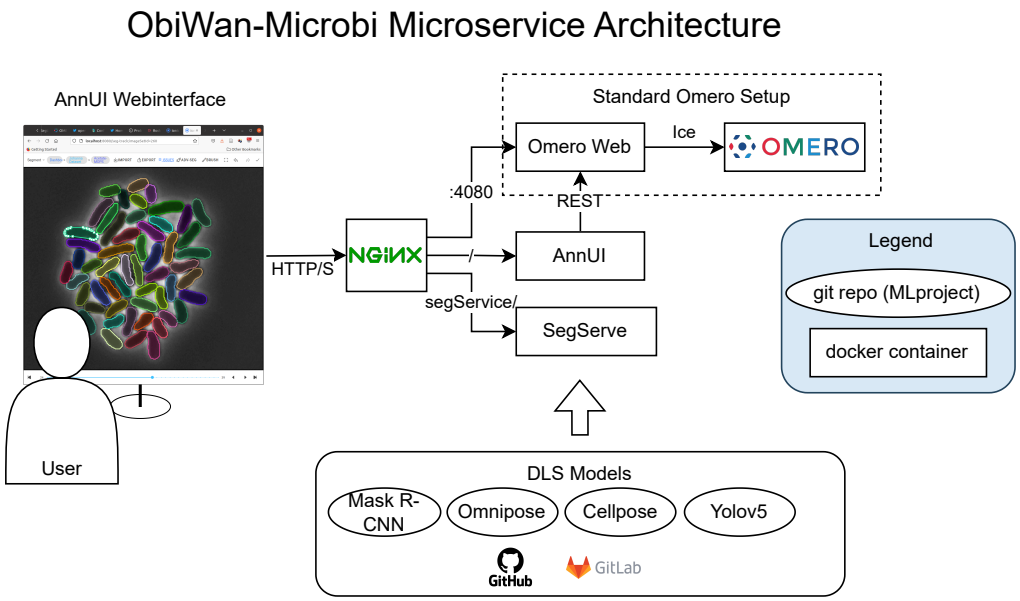

Figure 2.3: The *ObiWan-Microbi* microservice architecture: An NGINX reverse proxy organizes the HTTP traffic between the microservices. The *SegServe* service downloads and installs DL segmentation models on demand from git repositories, for example, hosted on GitHub or GitLab.

ing *ObiWan-Microbi*, users are empowered to qualitatively assess the best working DL segmentation approach for their specific use cases, concentrating annotation efforts on erroneous segmentation only. We have achieved annotation speeds of more than 200 cells per minute using the semi-automated segmentation workflow.

The shortcomings of existing DL segmentation methods can be reduced by iteratively retraining models on newly annotated data, following the human-in-the-loop principle. Due to its annotation capabilities and interoperability, *ObiWan-Microbi* is already used within the *microbeSEG* software [79] that complements the workflow by generating training data and retraining of DL segmentation methods on cropped images. Annotated segmentation data (`https://doi.org/10.5281/zenodo.6497714`) and trained DL segmentation models (`https://doi.org/10.5281/zenodo.7221151`) are made publicly available.

Furthermore, the fast semi-automated workflow of *ObiWan-Microbi*, handling thousands of annotations per image, has already been used for large-scale and complete MLCI time-lapse annotation. A first data set for *C. glutamicum* experiments including 1.4 million cell instances and more than $29k$ cell tracks in five time-lapse sequences is also made publicly available (`https://doi.org/10.5281/zenodo.7260136`) and serves as a strong baseline for developing new automated methods and validating their suitability for automated application.

2.4 Impact

ObiWan-Microbi provides the complete annotation workflow of segmentation and tracking tailored for time-lapse MLCI. Its support of FAIR data management, state-of-the-art Deep learning segmentation (DLS) methods, and collaborative annotation makes it especially useful for interdisciplinary

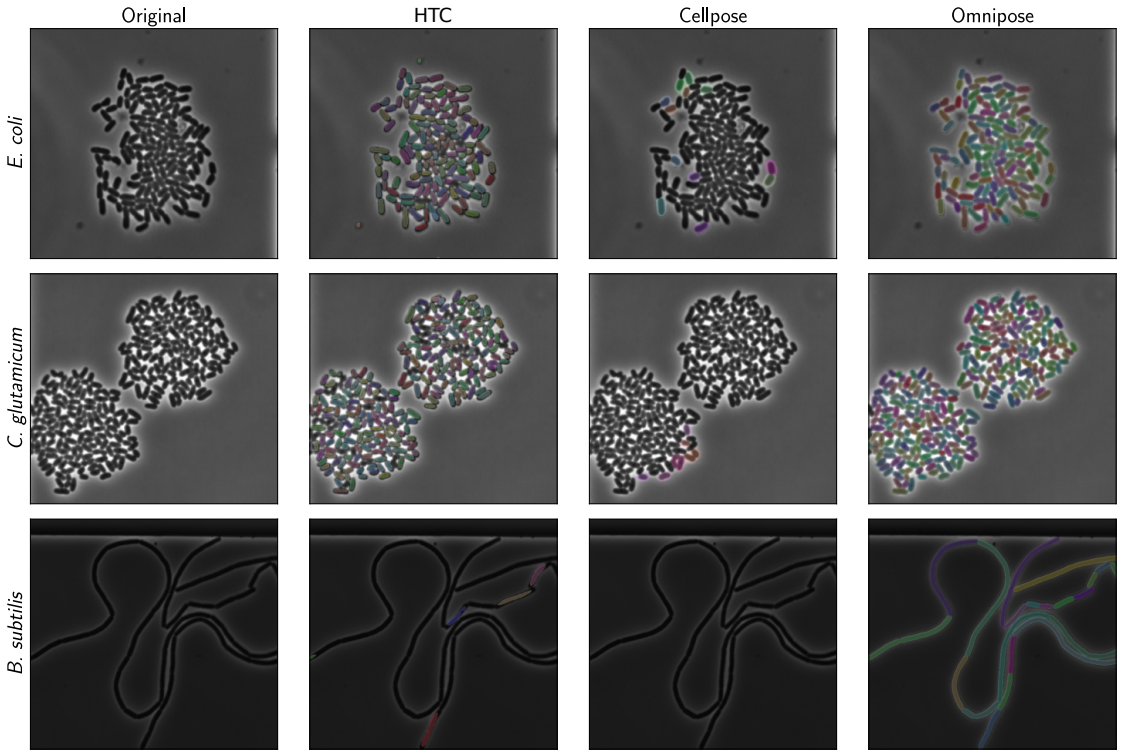

Figure 2.4: Application of DL segmentation methods *HTC*, *Cellpose* and *Omnipose* inside the *AnnUI* web interface to microfluidic live-cell image data containing *E. coli*, *C. glutamicum* and *B. subtilis* cells.

image processing and large-scale annotation efforts. Using and integrating existing best-practices and methods ensures interoperability with existing downstream processing (e.g. *Fiji*). The easy and reproducible usage of DLS methods removes time-consuming manual setups, allowing to focus on fast and precise annotation.

While initially developed and optimized for MLCI annotation, the data-integrated workflow of *ObiWan-Microbi* has the potential to be used in other imaging domains, including plant images, biomedical cell images, material sciences, or satellite images. Due to its expendable DLS architecture, even tailored deep learning solutions for these domains can be incorporated, and the multi-label annotation aids in performing complex image annotation. These domains also benefit from FAIR data usage and collaborative annotation opportunities for precise image analysis and large-scale benchmark data set generation. However, given the particular application field (microbes), currently, the workflow is based on closed-polygon annotations, limiting its ability to annotate complex shapes, holes or disconnected components.

2.5 Conclusion

ObiWan-Microbi is a collaborative annotation platform providing an efficient workflow for creating high-precision annotations for segmenting and tracking in microbial time-lapse imaging. Its focus on efficient large-scale annotation, encouragement of FAIR data management, expandable DL segmentation methods, and minimal user hardware requirements simplifies annotation efforts in interdisciplinary research domains. Based on growing annotated data sets, combined with human-in-the-loop workflows, *ObiWan-Microbi* provides the foundation for iteratively improving data and methods and pushing publicly available benchmark data sets for the training and validating novel methods in the MLCI domain. As we have recently seen in the biomedical field, this development is essential for establishing reliable and fully automated image analysis pipelines and unlocking the full potential of time-lapse MLCI for investigating biological phenomena.

Chapter 3

Tracking one-in-a-million: microbial cell tracking benchmark with experiment-aware metrics

Contents

3.1 Motivation **26**
3.2 Related Work **28**
3.3 Benchmark Dataset **29**
3.4 Experiment-aware microbial live-cell tracking metrics **32**
3.5 Tracking evaluation **35**
3.6 Conclusions **37**

Tracking the development of living cells in live-cell time-lapses reveals crucial insights into single-cell behavior and presents tremendous potential for biomedical and biotechnological applications. In MLCI, a few to thousands of cells have to be detected and tracked within hundreds of growing cell colonies. While the challenge of tracking cells is heavily influenced by the experiment parameters, namely the imaging interval and maximal cell number, suitable microbial tracking benchmarks are not widely available in MLCI and the effect of these parameters on the tracking performance is not yet quantified.

Therefore, we present the largest publicly available and annotated dataset for MLCI, containing more than 1.4 million cell instances, 29k cell tracks, and 14k cell divisions. With this dataset at hand, we generalize existing tracking metrics to incorporate relevant imaging and experiment parameters into experiment-aware metrics. These metrics reveal that current cell tracking methods crucially depend on the choice of the experiment parameters, where their performance deteriorates at high imaging intervals and large cell colonies. Thus, our new benchmark quantifies the influence of experiment parameters on the tracking quality, and gives the opportunity to develop new data-driven methods that generalize across imaging and experiment parameters. The benchmark dataset is publicly available at `https://zenodo.org/doi/10.5281/zenodo.7260136`.

The content of this chapter is published in: Seiffarth, J.; Blöbaum, L.; Paul, R.D.; Friederich, N.; Sitcheu, A.J.; Mikut, R.; Scharr, H.; Grünberger, A.; Nöh, K.: Tracking one-in-a-million: Large-scale benchmark for microbial single-cell tracking with experiment-aware robustness metrics. Bioimaging workshop at the European Conference on Computer Vision. 2024. https://doi.org/10.1007/978-3-031-91721-9_20.

3.1 Motivation

Detecting objects, segmenting their visual appearance into pixel-precise masks and tracking their movement through time is a fundamental challenge of computer vision providing crucial scene understanding necessary for autonomous driving [80], pedestrian management [81], sports or robotics [80]. Especially, in biomedical imaging, tracking the development of individual living cells allows gaining insights into the basic principles of life and diseases. For instance, single-cell tracking allows studying virus infections [82], pathogenic bacteria [83], cell aging [47], and cell interactions [84, 29] at the single-cell level. In particular, MLCI is a technology that performs high-throughput screening of the temporal developments of individual cells (see Figure 3.1). Herein, living microbial cells are introduced into microfluidic chip devices and trapped within thousands of micrometer-sized microfluidic structures called cultivation chambers. Within these structures, the cells grow in monolayers while their temporal development is recorded using automated microscopy. The microscope scans the cultivation chambers one by one, takes an image and repeatedly performs this within a loop, recording a time-lapse that captures the temporal development of the independent cell colonies. As a result, a single MLCI experiment records dozens of time-lapses and produces hundreds of gigabytes of raw imaging data. Clearly, automated segmentation and tracking methods are essential to extract information from the time-lapse images and to gain insights into microbial colony development and single-cell behavior.

However, tracking living microbial cells presents unique challenges distinct from those usually encountered in general object tracking. First, living microbial cells divide frequently, with division times ranging from a few minutes to hours. In MLCI experiments, few cells grow exponentially into dense and large colonies with up to thousands of cells captured in a single microscopy image (Figure 3.1E) while their total number is limited by the size of the cultivation chamber. Second, microbial cells in phase-contrast microscopy are visually hard to distinguish, making it hard to track them by their appearance (see Figure 3.1D). Third, the time-lapse recordings are affected by parameters such as the choice of the imaging interval between two consecutive phase contrast images and the number of concurrently monitored cultivation chambers. Moreover, both parameters are interdependent: lower imaging intervals usually simplify the tracking challenge but enforce shorter movement cycles of the microscope and, consequently, limit the number of concurrently monitored cultivation chambers. Notably, the imaging procedure itself may influence the growth behavior of the cells. While phase contrast imaging is not considered to impact microbial growth, fluorescence imaging may lead to phototoxicity and -bleaching effects [86]. Therefore, MLCI experiments are usually conducted with relatively low imaging rates. This leads to frequent cell divisions and larger cell displacement between consecutive frames. Consequently, we argue that MLCI not only needs robust and highly automated cell tracking, but also informed choices of experiment parameters, namely the imaging interval and the upper cell count limit.

In recent years, the rapid development in DL methods has driven segmentation and tracking

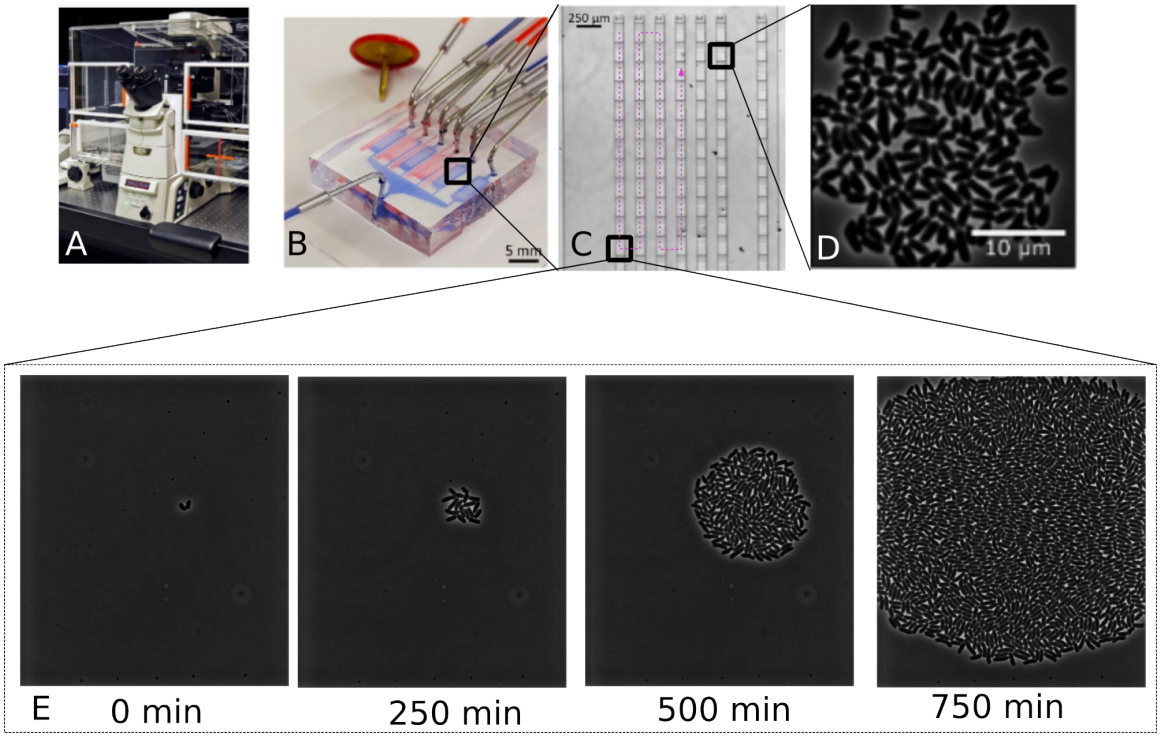

Figure 3.1: **Data acquisition in Microfluidic Live-Cell Imaging**. A microfluidic cultivation device is mounted to an automated microscope (A, B). The device contains hundreds to thousands of rectangular cultivation chambers (C) that are imaged one-by-one, moving the microscope stage and capturing images at a series of time-points (D). This imaging-and-movement loop, indicated by the purple line (C), is repeated with a pre-set imaging interval and leads to time-lapse recordings capturing the temporal development of the cell colony from a few cells up to several thousands (E). The images in (E) depict a 80×90 μm region. B and C are adapted from Blöbaum *et al.* [85].

methods [87, 88, 89, 90, 91, 92, 93, 94]. Moreover, the increased availability of large-scale datasets such as ImageNet [95], KITTI [80], CityScapes [96], SA-1B [94], and LAION-5B [97] has shown to be a crucial driver for method development. Within the life sciences, we have seen a similar rapid development of methods. DL segmentation approaches have been developed [30, 31, 60, 32, 98], driven by annotated datasets [99, 31, 67, 100, 101, 69]. For cell tracking, special methods have been developed that incorporate cell divisions[101, 102, 64, 103, 33, 65, 49]. However, public datasets for cell tracking [99, 101, 104] primarily focus on microscopy images with eukaryotic cells. In contrast to microbial cells, eukaryotes usually show more distinctive visual features, less frequent cell division and the images contain fewer cell instances. For microbial cells, only few tracking datasets [47] are available, making it difficult to train data-driven tracking methods and benchmarking suitable experiment parameters. Due to the lack of datasets, the importance of the experiment parameters for high-quality automated tracking has not been investigated before.

Therefore, we establish a novel benchmark for cell tracking in MLCI and extend existing tracking metrics with experiment parameters to quantify their effect on the tracking performance. Our contributions are threefold: (1) we introduce a new annotated time-lapse dataset, recorded with low imaging interval for the segmentation and tracking of *C. glutamicum* cells containing roughly 1.4 million cell masks and about 14k cell divisions. (2) We introduce experiment-aware metrics that extend existing metrics and incorporate the choice of the imaging interval and the maximum number of cells into the evaluation. (3) We evaluate SOTA tracking methods using our devised metrics across a broad range of experiment parameters. We thereby show that the performance of SOTA tracking algorithms deteriorates, especially at lower imaging rates and higher cell counts. Notably, this fact that has not yet been quantified by the CTC community. Therefore, our benchmark represents a step forward to towards fully automated and robust data-driven microbial single-cell tracking and raises awareness about the importance of experiment parameters for cell tracking in MLCI experiments.

3.2 Related Work

Benchmark datasets. Benchmark dataset for cell segmentation cover different cell tissues, morphologies and imaging modalities [67, 100, 101, 31]. The availability of additional tracking information for full time-lapse recordings is much less common. The cell tracking challenge combines datasets of various cell types and provides partial dense segmentation and full tracking information [99, 49]. Schwartz *et al.* introduced the DynamicNuclearNet dataset containing roughly 600K segmented nuclei instances with roughly 2k cell divisions[101]. Van Vliet *et al.* provide a dataset of six genetic variants of *Escherichia coli* in 39 time-lapse videos containing roughly 100k cell instances and 9k cell divisions. Anjum *et al.* introduce the CTMC challenge dataset contains roughly 2 million cell detections within 2.9k cell tracks and 457 cell division events [104]. Their segmentation annotation is restricted to bounding boxes and on average 13 cells are visible within a microscope image.

Tracking Methods. In the predominant tracking-by-detection scheme, cells are first detected in the microscopy time-lapse and then linked across frames to build biologically valid cell tracks. Thus, tracking-by-detection is usually formulated as a graph problem, where nodes represent the cell detections at specific points in time and edges link cell detections through time. Therefore, Jaqaman *et al.* [102] formulated tracking as a Linear Assignment Problem (LAP) where cell detections are linked into segments, which are then linked to incorporate cell division events. Theorell *et al.* [64]

extended this approach into Multi Hypothesis Tracking (MHT), where cell linking costs are derived from biological models and tracking predictions are sampled using a particle filter approach. In contrast, Löffler *et al.* [103] formulated a coupled minimum cost flow and correct segmentation errors during the tracking. All these methods require hand-tuned parameters to compute the costs of linking cells.

Data-driven approaches promise to derive these linking costs purely from training data. Ben-Haim *et al.* [105] use a Graph Neural Network (GNN) to predict cell linking costs and utilize features computed from contrastive visual embeddings. However, cell divisions are performed heuristically among the cells' neighborhood. Similarly, Schwartz *et al.* [101] utilize a graph attention network to predict linking costs used for building a LAP. Gallusser *et al.* [33] extract cell detection features and use a transformer network to predict the linking costs. The tracking graph is constructed using a greedy scheme or optimizing an Integer Linear Program (ILP) for minimizing linking costs. O'Connor *et al.* [47] predict the dense evolution of cell mask to the next time point for every cell instance using a U-Net architecture. Cells are then linked by their mask overlap. While Gallusser *et al.* and O'Connor *et al.* have been using microbial datasets, the other data-driven approaches have focused on tracking eukaryotic cells.

Tracking metrics. General object tracking metrics such as the Multiple Object Tracking Accuracy (MOTA) [106] and Higher Order Tracking Accuracy (HOTA) [107] have been established for tracking a wide range of objects, but lack the consideration of object division crucial in cell tracking. Thus, specialized tracking metrics have been developed and established in cell tracking, especially by the CTC [99]. Herein, we distinguish technical and biological tracking metrics. Technical tracking metrics such as the *TRA* and *LNK* are based on the Acyclic Oriented Graph Matching (AOGM) [108]. The AOGM is a weighted sum of costs for the minimal set of operations to transform the predicted segmentation and tracking into the ground truth segmentation and tracking. While the *TRA* metric scores both segmentation and tracking errors, the *LNK* metric solely rates errors in cell linking. Biological metrics are motivated by biological events that are of special interest. For instance, the Complete Tracks (CT) measures the number of completely correctly reconstructed cell tracks from the first detection of a cell to its division or disappearance. Moreover, the Mitotic Branch Correctness (MC) rates the quality of reconstructing cell division events.

3.3 Benchmark Dataset

In this chapter, we present a new MLCI benchmark dataset for microbial single-cell tracking, briefly termed TOIAM. The dataset consists of microscopy time-lapses of growing *C. glutamicum* that show a characteristic 'snapping' division behavior: Cells continuously grow over time, increasing their size, and suddenly divide into two, introducing cell movement. This sudden cell movement transfers to neighboring cells, introducing sudden cell movement across the entire cell population, adding another challenge to cell tracking. The images are recorded using phase contrast imaging at low imaging intervals of one image per minute. The recorded microscopy frames were annotated with cell segmentation masks and tracking information using a semi-automated workflow. We highlight the special characteristic of microbial datasets that are crucial to consider for robust cell tracking.

3.3.1 Data acquisition

MLCI experiments are usually carried out in three steps [21]: First, the microbial organism is cultivated in a so-called preculture until reaching a certain biomass measured by the Optical Density (OD). Second, the cell suspension is introduced into a microfluidic cultivation chip, trapping individual cells within the cultivation chambers. Third, medium supply is connected to the microfluidic chip and the imaging routine is started recording hundreds of cultivation chambers at a specific imaging interval.

In our case, we cultivated *C. glutamicum* (ATCC 13032) in Brain Heart Infusion (BHI) medium at 30 °C. From an overnight preculture, the main culture has been inoculated the next day starting with an OD600 of 0.05 and grown at 120 rpm to an OD600 of 0.25. A microfluidic chip has been fabricated according to Täuber *et al.* [27], and fixed to the microscope's stage. The main culture cells were transferred to monolayer cultivation chambers (height of 720 nm) on the microfluidic chip within the inoculation procedure. Constant medium flow through the microfluidic device has been provided by pressure driven pumps with a pressure of 100 mbar on the medium reservoir.

The time-lapse phase contrast images of five cultivation chambers have been recorded every minute using an inverted microscope (Nikon Eclipse Ti2) with a 100x oil immersion objective and a DS-QI2 camera (Nikon) at 15 % relative DIA-illumination intensity and 100 ms exposure time. The recording procedure has been performed for 800 minutes, leading to a total of $4,000$ recorded microscopy images. The recorded images provide a spatial resolution of 0.072 micrometers per image pixel in both spatial dimensions.

3.3.2 Semi-automated segmentation & tracking annotation

To provide high quality annotation for the large number of recorded images with a limited amount of manual annotation workload, we decided to first perform an initial segmentation and tracking using Omnipose [30] and PyUAT [109], respectively. The result was subsequently corrected by an expert in the annotation tool *ObiWan-Microbi* [110]. For cell segmentation, we focused on providing annotations for every single-cell. Therefore, over- and under-segmentation, false positive and false negative segmentations were corrected. Based on the corrected segmentation, the tracking edges were manually checked and corrected. Table 3.1 shows the number of manual corrections carried out for the different time-lapse recordings. Only few manual segmentation corrections were performed. For cell tracking, the majority of corrections had to be performed towards the end of the time-lapse sequences, where the cell count and divisions events increased substantially and cells leave the field of view at the left and right image borders.

3.3.3 Dataset statistics

The annotated dataset contains more than 1.4 million densely annotated cell instances, 29k cell tracks, and 14k cell divisions (see Table 3.2). Thus, our data set is large enough to be split into meaningful train, validation, test splits. Moreover, Figure 3.2 shows that all five time-lapses show similar temporal developments of cell numbers, divisions and disappearances. Nevertheless, cell count and, thereby, the number of cell divisions are exponentially increasing throughout the time-lapses (Figure 3.1E) leading to a temporal imbalance. For instance, the large number of cells at the end of the time-lapses leads to 50 % of the overall division events occurring in the last 100 minutes

Table 3.1: Amount of manual tracking correction actions to correct errors in the semi-automated annotation workflow. A correction action is the addition and deletion of a tracking link or adding, deleting and editing of segmentation masks.

	Time-lapses					
	0	1	2	3	4	Total
Segmentation	143	319	202	1087	25	1,776
Tracking	773	1,463	4,057	1,823	1,111	9,227

Table 3.2: Statistics for five time-lapse sequences in the benchmark dataset and its split into train, validation and test sets. The table shows the number of densely annotated segmentation masks (cell instances), cell tracks and cell divisions.

Split	Time-lapse Index	# Images	# Cell Instances	# Cell Tracks	# Cell Divisions
Train	0	800	238,364	4,918	2,448
Train	1	800	292,070	6,137	3,053
Train	2	800	327,832	6,884	3,428
Val	3	800	292,995	6,184	3,064
Test	4	800	264,011	5,740	2,844
Total	**0,1,2,3,4**	**4,000**	**1,415,272**	**29,863**	**14,837**
Train	0,1,2	2400	858,266	17,939	8,929

of the time-lapse recordings. Moreover, the number of cell disappearances is much higher towards the end of the time-lapse recordings as the colony exceeds the size of the cultivation chamber and cells leave the field of view (Figure 3.2B, C).

3.3.4 Implications of imaging interval subsampling

The temporal imbalance of microbial time-lapse datasets in Figure 3.2 is amplified by the choice of the imaging interval. Recording the TOIAM dataset with a low imaging interval allows us to simulate higher intervals by considering only every k-th recorded image. We call this subsampling with a factor $k \in \mathbb{N}$ in the following. Due to the imaging interval of one minute, a subsampling of k leads to an imaging interval of k minutes. Using the subsampling procedure, we created subsampled datasets with higher imaging intervals.

Figure 3.3 shows that changing the imaging interval has a tremendous impact on the structure of the dataset and, therefore, on the challenge to track the living cells. Time-lapses with simulated higher imaging intervals contain fewer images, but the number of cell divisions and cell appearances stays constant. Thus, the number of cell divisions and disappearances between consecutive frames increases strongly with higher imaging intervals (Figure 3.3C, D). More cell divisions between consecutive frames lead to much larger cell displacements due to the 'snapping' cell division of *C. glutamicum* (see Figure 3.3E). However, also the frequency of cell divisions increases. While at an imaging interval of one minute roughly 1 % of cell links are cell divisions, higher imaging intervals

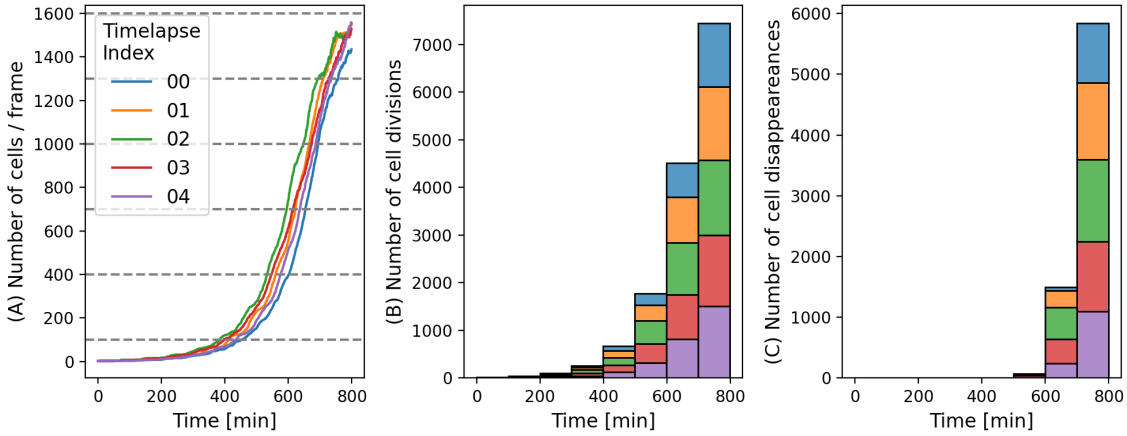

Figure 3.2: **Temporal imbalance of the five microbial time-lapse recordings of the benchmark dataset.** We measured the temporal development of cell count (A), cell divisions (B), and cell disappearance events (C) for each time-lapse. Cell division and disappearance events are grouped into bins of 100 minutes. The dashed lines in (A) indicate cell count limits (100, 400, 700, 1,000, 1,300, 1,600).

lead to a strong increase, with up to 34 % of the cell links being divisions at an imaging interval of 40 minutes (fig. 3.3F). Thus, having a good estimate on the frequency of cell divisions, for example based on previous experiments, is crucial.

3.4 Experiment-aware microbial live-cell tracking metrics

For a benchmark, suitable metrics are essential that rate the quality of the method performance and serve as an objective comparison tool. However, we have shown that MLCI time-lapses have unique characteristics that make the application of existing metrics difficult. Therefore, we present two new metrics that build on top of the well-established CTC metrics but introduce experiment awareness: First we incorporate the influences of experiment parameter choices, i.e. the imaging interval and maximum number of cells. We decompose the existing metrics along these experiment parameters and term these Experiment-Aware Tracking Metrics (*EATM*). Second, we summarize the performance of tracking metrics across a wide range of these parameters within a single Robustness Metric (*RM*).

3.4.1 Cell tracking metrics

The CTC introduces several metrics for measuring the quality of a tracking prediction in comparison with ground truth information [99]. The *TRA* and *LNK* metrics are based on the AOGM score that determines the minimum number of operations needed to convert a predicted tracking result into its corresponding ground truth [108]. These operations include corrections of over- and under-segmentation and the addition or removal of tracking links. Each of these operations is awarded a constant penalty cost, and the AOGM gives their weighted sum.

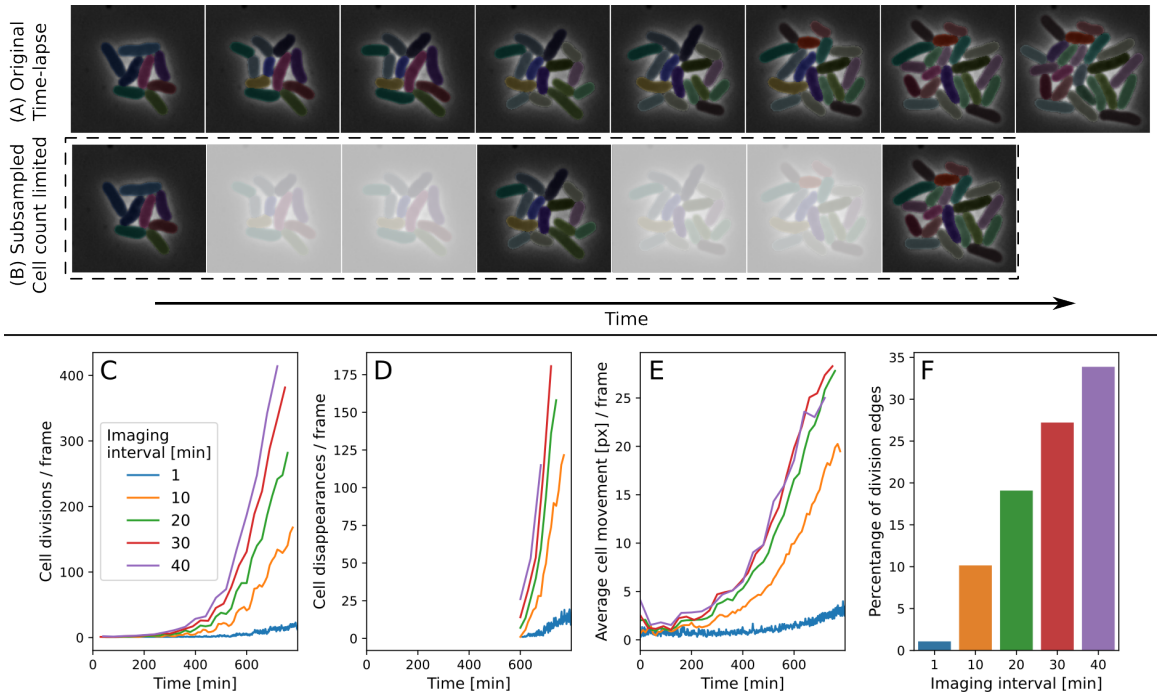

Figure 3.3: **Subsampling and cell count limiting of time-lapse sequences**. (A) shows an excerpt of an MLCI time-lapse. (B) shows an exemplary subsampling with a factor of 3 and truncation at a cell count limit of 21 leading to 3 frames in total. Grayed out images denote frames removed due to subsampling, the dashed box denotes the cell count limit. (C-E) shows the temporal changes in the number of cell division, disappearance and movement per microscopy frame when subsampling to different imaging intervals. The curves have been exponentially smoothed. (F) shows the percentage of division events in contrast to non-division links. (C-F) show data from the TOIAM test split.

The *TRA* metric is the AOGM, normalized to $[0, 1]$:

$$TRA = 1 - \frac{\min\left(AOGM, AOGM_0\right)}{AOGM_0} \tag{3.1}$$

where $AOGM_0$ is the $AOGM$ of an empty tracking graph (no nodes, no edges). The *LNK* metric scores the quality of the tracking:

$$LNK = 1 - \frac{\min\left(AOGMA, AOGMA_0\right)}{AOGMA_0} \tag{3.2}$$

where $AOGMA$ denotes the $AOGM$ where only edge operation costs are considered and the $AOGMA_0$ denotes the $AOGMA$ of the ground truth graph without edges, respectively.

Moreover, the *DIV* and *CT* measure the amount of correctly reconstructed 'biological events' that are cell divisions (*DIV*) and complete cell tracks (*CT*) [99]. Therefore, for both types of events True Positives (TP), False Positives (FP), and False Negatives (FN) are computed, while precision and recall are summarized in the F1-score

$$F_1 = \frac{2}{1/precision + 1/recall} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{3.3}$$

The F1-score for cell divisions and complete tracks are termed *DIV* and *CT* in the following.

3.4.2 *EATM* cell tracking metric for MLCI

Existing tracking metrics, such as those introduced by the CTC, have proven to be useful to rate the tracking quality. However, they do not consider the specific characteristics of microbial time-lapses that are crucially influenced by experiment parameter choices. Thus, we extend the existing metrics and decompose them in to the devised *EATM* tracking metric, that considers the imaging interval and the maximal number of living cells.

Let $S = \{S_1, \ldots, S_L\}$ be the given segmentation ground truth of a time-lapse with $L \in \mathbb{N}$ frames. Then, some tracking-by-detection method T predicts a tracking graph $G = (V, E)$ containing the segmentation detections as nodes, $V = \bigcup_{l \in \{1,\ldots,L\}} S_l$, and links between cell detections as tracking edges, $E \subseteq V \times V$. We define a tracking metric to be a function $m(\cdot, \cdot)$ that compares a predicted tracking graph $\hat{G} = T(S)$ to the ground truth tracking graph $G^\star$ using a metric m with normalized score:

$$0 \leq m(\hat{G}, G^\star) \leq 1. \tag{3.4}$$

The *TRA*, *LNK* and *DIV* metrics defined before are such metrics.

First, we make such a metric sensitive to experiment parameters. Therefore, we evaluate the metric on temporally subsampled and cell count limited versions of the time-lapses (Figure 3.4). Let $k \in \mathbb{N}$ be the subsampling parameter for reducing the temporal resolution. Let $N_{max} \in \mathbb{N}$ be the cell count limit, that is, the maximal number of cells within a single image of the time-lapse. Then a time-lapse is truncated to the last frame where the cell count does not exceed the limit N_{max} (Figure 3.3B). We denote the subsampled and truncated segmentation information with $S|^k_{N_{max}}$ and the ground truth tracking graph with $G^\star|^k_{N_{max}}$, respectively. Then we define the *EATM* version $\tilde{m}$ of m

$$\tilde{m}^k_{N_{max}}(T, S, G^\star) := m\left(T(S|^k_{N_{max}}), G^\star|^k_{N_{max}}\right) \tag{3.5}$$

that evaluates the metric m on tracking prediction for the subsampled and truncated segmentation $T(S|^k_{N_{max}})$ with the subsampled and truncated ground truth tracking $G^\star|^k_{N_{max}}$.

Second, we define a new RM to summarize the robustness of the tracking algorithm across a wide range of imaging intervals and cell count limits. We define a set of subsamplings $SF \subset \mathbb{N}$ and maximum cell counts $MC \subset \mathbb{N}$. The robustness metric RM of a metric m measures the normalized frequency that the $EATM$ of m surpasses a given threshold $\vartheta \in [0, 1]$:

$$RM(m, \vartheta, SF, MC) := \frac{1}{|SF| \cdot |MC|} \sum_{k \in SF} \sum_{mc \in MC} \mathbb{1}\left[\tilde{m}^k_{mc}(T, S, G^\star) \geq \vartheta\right], \tag{3.6}$$

where $\mathbb{1}[\,\cdot\,]$ is the indicator function.

The selected parameters within the parameter sets SF and MC specifically guide the meaning of the RM. For example, focusing on robustness across a wide range of imaging rates requires the choice of a wide range of subsampling factors (SF). Similarly, focusing on robustness across various numbers of cells that can be tracked concurrently requires a larger MC. Moreover, the size of the parameter sets controls the granularity of the metric evaluation. Evaluating many subsampling and maximum cell count parameters results in numerous evaluations, each requiring the execution of the tracking method for the dataset. Thus, the size of the parameter sets balances granularity with metric computation time.

The selection of the threshold ϑ sets the lower limit of the tracking metric that is still considered a "suitable" tracking. Whether the tracking quality is suitable depends on the underlying tracking metric and the goal of the tracking (e.g., reconstructing all cell links correctly). We choose a threshold of 80 % and 90 % to guarantee high and very high tracking quality.

3.5 Tracking evaluation

We evaluated the performance of tracking methods on our new dataset using the $EATM$ and RM metrics. For the comparison, we selected three representative tracking methods: The `Distance` method provides a baseline using pure distance information for linking costs and predicting links using a greedy scheme. The `LAP` method uses mask overlap for linking costs and optimizes the linking cost between consecutive frames in an LAP [102]. The `Trackastra` [33] method is the best performing tracker according to the current CTC leaderboard. In our evaluation, `Trackastra` represents the data-driven methods for predicting linking costs.

We evaluated all three tracking methods with various imaging intervals and cell count limits on the TOIAM test split. Figure 3.4 shows resulting heatmaps of the $EATM$ based on the DIV metric. Across all three tracking methods, the DIV metric decreases for larger imaging intervals as well as higher cell limits, indicating deteriorating performance. The evaluation shows that higher cell numbers, more frequent cell divisions, and lower temporal resolution make the cell tracking task notably more difficult.

Among all three tracking methods, the `Distance` method performs worst across all experiment parameters. Therefore, making cell linking inferences based purely on distances, is not suitable for MLCI, especially when using a greedy scheme. The `LAP` method performs slightly better in the DIV metric at lower imaging intervals, benefiting from the more informative overlap costs and the non-greedy `LAP`. However, its performance drops rapidly when many cells are present and the imaging interval is increased leading to larger cell movement and, therefore, limiting the usefulness of

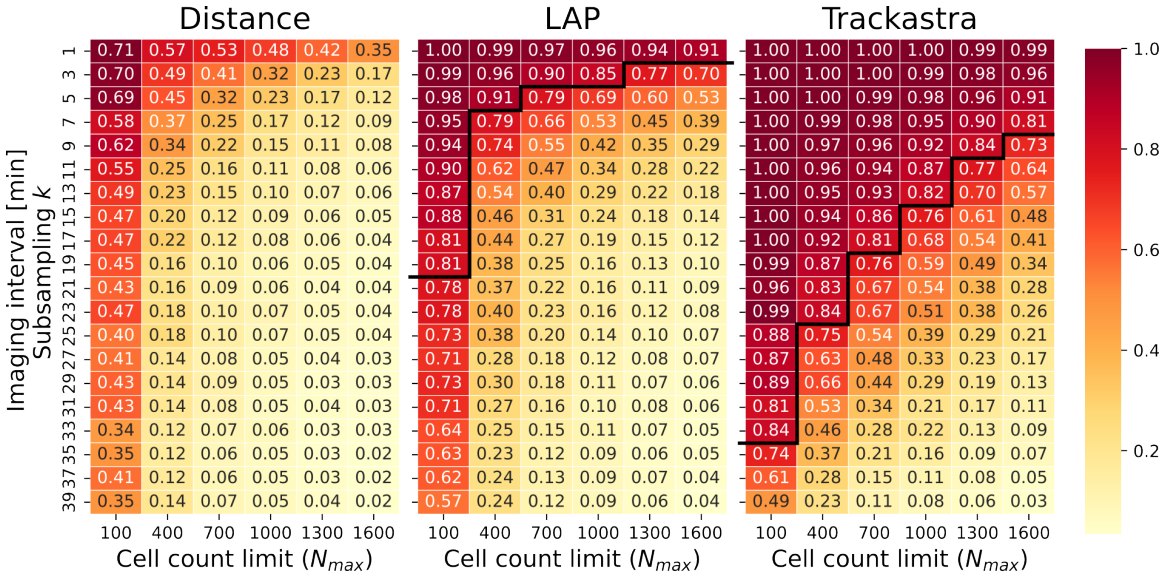

Figure 3.4: The *EATM* based on the *DIV* metric measured across different imaging intervals and cell count limits. The black line marks the region surpassing the 80 % threshold (*RM*@0.8). The higher the value of the *DIV* metric, the better is the reconstruction of the cell divisions, with a value of 1 meaning perfect cell division reconstruction. Evaluations have been performed on the test split.

Table 3.3: Robustness metric version of the *TRA*, *LNK*, and *DIV* metrics evaluated on the test split using *RM* thresholds of 80 % and 90 %. The *RM* metric has been computed over the subsampling factors (*SF*) and cell count limits (*MC*) used in Figure 3.4.

Method	RM@0.8			RM@0.9		
	TRA ↑	*LNK* ↑	*DIV* ↑	*TRA* ↑	*LNK* ↑	*DIV* ↑
`Distance` (greedy)	**1.00**	0.16	0.00	0.46	0.12	0.00
`LAP`	**1.00**	0.23	0.16	0.47	0.16	0.11
`Trackastra` (greedy)	**1.00**	**0.50**	**0.45**	**0.81**	**0.42**	**0.32**

overlap costs. In contrast, `Trackastra` shows much stronger *DIV* scores across the various experiment parameters. The transformer network trained on the training split seems to learn patterns that generalize to higher imaging intervals. Thus, `Trackastra` robustly performs division reconstruction at various experiment parameters, opening the opportunity to monitor more cultivation chambers concurrently, which is important in the context of high-throughput MLCI-based screening applications.

While the *EATM* heatmaps give detailed insights into the methods' performance, they only visualize a single tracking metric. To summarize and compare the methods across different metrics, we summarize the robustness of the tracking method to the experiment parameters using the *RM* metric. Table 3.3 shows the *RM* score for the *TRA*, *LNK*, and *DIV* metrics using a threshold of 80 % and 90 %. For the *DIV* metric, the *RM* score is also visualized by the black line in Figure 3.4.

We observe that the *TRA* metric is not sensitive enough to give robustness insights when ground truth segmentation masks are provided. The correct segmentation will always lead to scores above

0.8 in all experiment parameters, yielding a misleading robustness score of 1. The *RM*s of *LNK* and *DIV* are more sensitive and provide insights for both robustness thresholds. `LAP` shows low results in *LNK* and *DIV* metrics, highlighting that it is only suitable for tracking cells at low imaging intervals and small cell colony sizes. The `Trackastra` method shows robustness up to 50 % at an 80 % threshold and, therefore, allows performing automated tracking at various experiment settings. Across all evaluations, the *DIV* metric is consistently lower than the *LNK* metric, indicating that cell division reconstruction is more difficult than linking non-dividing cells. This, underlines the larger focus on predicting cell divisions in MLCI datasets.

3.6 Conclusions

In this chapter, we have presented a new benchmark for cell tracking in MLCI with increased experiment awareness in metric ratings. The presented TOIAM dataset is the largest publicly available for MLCI in terms of annotated cell masks, cell tracks and cell divisions. Moreover, we have highlighted that MLCI data comes with unique challenges due to exponentially growing cell colonies and frequent cell divisions. These challenges are strongly influenced by experiment parameters such as the imaging interval and the maximum number of cells per frame. To capture these influences in appropriate metrics, we extended existing metrics towards experiment-awareness (*EATM*) and summarize them in a robustness metric (*RM*). We have shown that these *EATM*s and *RM*s give crucial insights into the practical suitability of tracking methods across a wide range of experiment parameters. Thus, our efforts aim to closely integrate method development and experiment design, and to open a stringent approach for experimenters to make informed decisions about their experiment parameter choices.

For now, our TOIAM dataset is limited to a single type of microbe cultivated within a single experiment. Therefore, we are looking forward to extending the dataset to other cell types and cultivation conditions and also apply the introduced metrics to other existing datasets in the future. Moreover, the evaluation of the tracking methods with erroneous segmentation and imperfect image data as well as the use of other biologically motivated metrics, such as the CT or BIO metrics from the CTC, is crucial for further evaluating their practical applicability. Adapting tracking methods to the challenging imaging conditions in MLCI, for example, by tuning hyper-parameters or establishing temporal subsampling during the training procedure might lead to more robust cell tracking methods.

Summarizing, our large-scale benchmark represents a step forward towards robust data-driven microbial single-cell tracking and facilitates tight integration of experiment parameters and tracking method development using experiment-aware metrics.

Chapter 4

Acia-workflows: automated single-cell imaging analysis for scalable and deep learning based live-cell imaging analysis workflows

Contents

4.1 Motivation **40**
4.2 Analysis pipeline and workflow **43**
4.3 Results **46**
4.4 Discussion & Outlook **54**

Live-cell imaging technology enables detailed spatio-temporal characterization of living cells at single-cell resolution, which is critical for advancing biomedical and bioprocess research. High-throughput setups with 10s to 100s of parallel cell cultivations offer the potential for robust and reproducible insights. However, these insights are hidden within 100s of GBs of time-lapse imaging data recorded per experiment. Thus, developing automated workflows to extract single-cell information is crucial for leveraging the capabilities of live-cell imaging and gaining quantitative insights into single-cell dynamics. We here present `acia-workflow`, the joint development of the automated live-cell imaging analysis (`acia`) Python library and its usage in analysis workflows (`acia-workflow`) that assemble the image analysis pipeline, software dependencies, documentation, and visualizations into a single Jupyter Notebook leading to accessible, reproducible and scalable analysis workflows. We demonstrate the analysis and customization capabilities of `acia-workflow`s in three case studies that investigate various MLCI experiments, ranging from growth rate comparisons, and co-culture analysis, to precise, minute-resolution quantitative analyses of the response dynamics of individual cells to changing oxygen conditions.

The content of this chapter is available as a pre-print: Seiffarth, J.; Kasahara, K.; Bund, M.; Lückel, B.; Paul, R.D.; Pesch, M.; Witting, L.; Bott, M.; Kohlheyer, D.; Nöh, K. acia-workflows: Auto-

mated Single-cell Imaging Analysis for Scalable and Deep Learning-based Live-cell Imaging Analysis Workflows. arXiv. 2025. `https://doi.org/10.48550/arXiv.2510.05886`. The `acia-workflow` software and case studies are available open-source at `https://github.com/JuBiotech/acia-workflows`.

4.1 Motivation

Live-cell imaging is at the forefront of investigating the dynamic behavior of living cells, fostering our understanding of cancer treatment [111], protein secretion [112, 113], diseases [114, 115], single-cell heterogeneity [116, 117] and biofilm formation [118, 119]. Integrating automated live-cell imaging with disposable, high-throughput microfluidic devices enables the phenotyping of single cells and their development into cell populations in precisely controlled environments [120, 121, 122]. These MLCI setups simultaneously record numerous cell populations within a single experimental run at constant or time-varying conditions. Their picoliter scale and high-throughput nature promise resource-efficient yet reliable single-cell insights by increasing sample size while reducing the number of experimental runs. Thus, MLCI is ideally suited for studying dynamic biological processes that occur over short timescales, delineating the growth behavior of single cells over many generations [123, 35, 27], and the interplay of cell-to-cell interactions [26, 59], including the emergence of single-cell heterogeneity [124].

All these diverse studies rely on extracting single-cell information from MLCI time-lapse data and analyzing the spatio-temporal development of cells and their populations. However, these single-cell measurements are hidden in the imaging data and need to be extracted from 10s to 100s of gigabytes of time-lapse data, demanding highly automated image analysis pipelines.

Figure 4.1A shows the typical five steps of a MLCI analysis pipeline: First, the time-lapse imaging data is loaded, and then cell objects are detected within every microscopy image using instance segmentation. These cell objects are tracked across images to reconstruct their temporal context, followed by the extraction of biological single-cell features, such as cell size, fluorescence, or age. Finally, the extracted single-cell features are used to derive and visualize live-cell insights. Although the sequential steps of the pipeline are well defined, it remains essential to determine the criteria that constitute an effective image analysis pipeline in practice. Therefore, we have distilled the most important five key abilities arising from the practical user experience of deploying MLCI pipelines: accessibility, customizability, modularity, scalability, and shareability.

These ability requirements arise from the different pipeline steps and general challenges in deploying software in a research lab. MLCI data comes in various formats, usually depending on the utilized microscopy platform and vendor software. It contains the recorded imaging channels, such as phase contrast, bright field, and fluorescence, as well as additional metadata. To process these different formats, the users need the ability to customize the imaging and metadata loading process. Thus, we define customizability as one key ability.

Cell segmentation [30, 31, 32, 125, 79] and tracking [33, 102, 47] methods have seen rapid development in recent years. Especially advances in deep learning approaches trained on growing benchmark datasets yield high-quality and automated single-cell results for various imaging modalities and cell morphologies [125, 31, 101, 100]. To transfer these advances and continuous developments into practical application, the user needs the ability to quickly test different SOTA methods, adapt their parameters, and easily integrate upcoming improved methods. Therefore, modularity with clearly

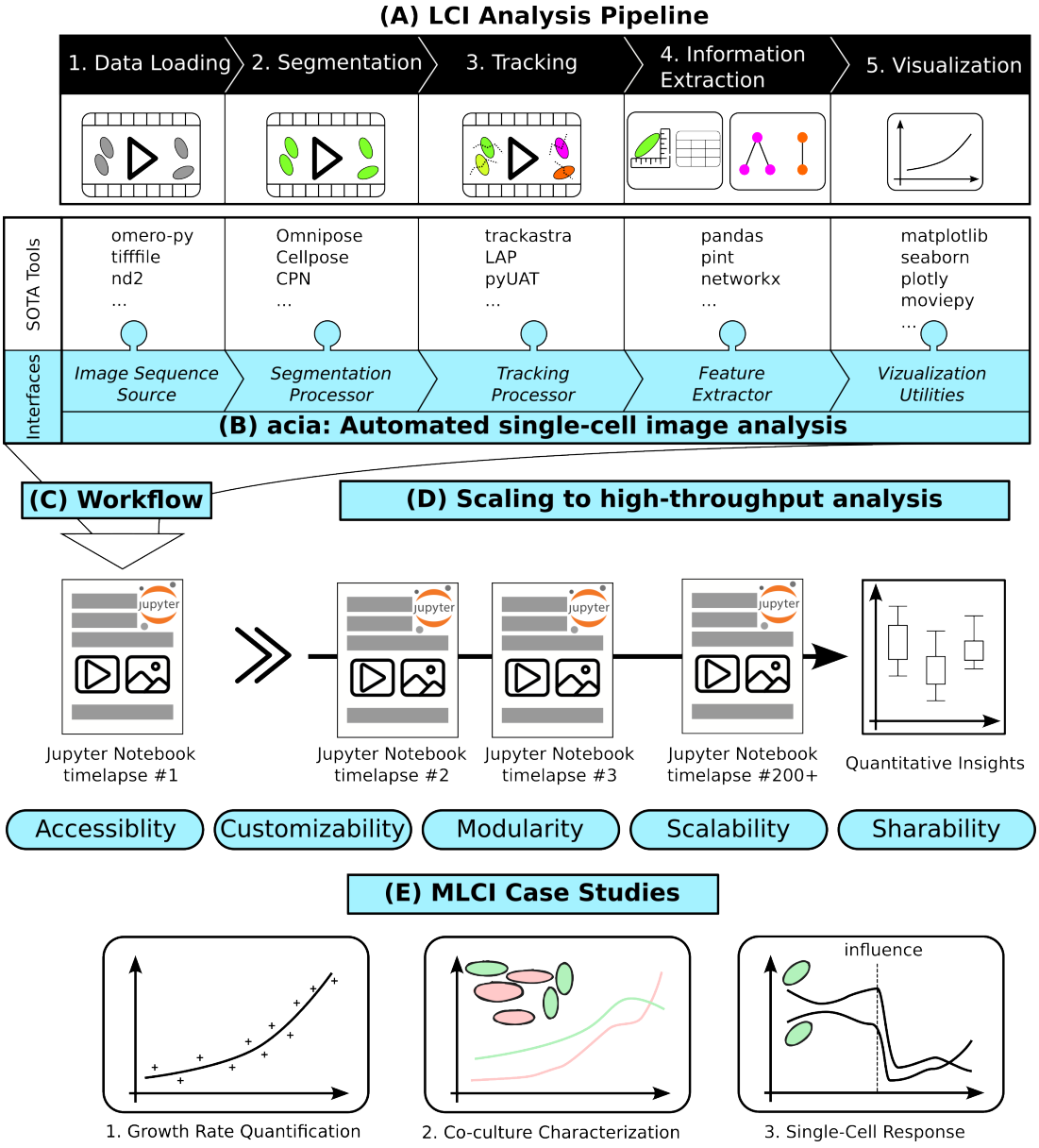

Figure 4.1: **Acia-workflows overview.** We implement the five-step MLCI analysis pipeline (A) as modular components within the `acia` library (B), utilizing SOTA methods and existing Python libraries. These modular steps are implemented sequentially within a single Jupyter Notebook, fusing code, documentation, software dependencies, and visualizations (C). These notebooks are scaled to high-throughput experiments with numerous time-lapse recordings to gain quantitative insights (D). We develop three workflows for everyday applications in MLCI (E) and showcase the five key abilities: accessibility, customizability, modularity, scalability, and shareability.

defined interfaces is crucial for cell segmentation and tracking.

The single-cell information arising from segmentation and tracking results lays the foundation for biological insights. However, these insights highly depend on the research question. Some users may be particularly interested in large cells, others in cells with specific fluorescence markers, and yet others may be specifically interested in the behavior of dying, fast-growing, or abnormal cell behavior. Therefore, we recognize the need for high customization in single-cell feature extraction and visualization to match the demands of ever-changing research questions. Combining complex spatio-temporal single-cell information across single-cell, population, and experiment levels is crucial to answering them.

Once such a pipeline is developed and extracts single-cell insights, it remains critical to quantify the variability of the measured effects and patterns of single-cell behavior. Consequently, such experiments are typically carried out in a high-throughput manner, capturing MLCI data from multiple independent cell populations. For a user, it is important to quickly roll out the pipeline across multiple replicates and gather the information to quantify the variability in the extracted insights. Thus, we denote scalability as another key ability.

Alongside customizability, modularity, and scalability, we require two additional capabilities that arise from deploying such pipelines in our research lab: accessibility and sharability. We define good accessibility as having low-entry barriers for users with little to no programming experience, without complex software installation procedures or specific hardware requirements. The entry to the analysis software should be as easy as opening a website in the browser to democratize its usage among interdisciplinary researchers. Moreover, the usage of such software is boosted by the ability to share and reproduce analysis results among teams and fellow researchers. Sharing enables the verification of extracted insights and the continuous improvement of the analysis pipeline. Thus, developing and improving such analysis pipelines becomes a joint community effort.

In recent years, numerous analysis pipelines have been developed [47, 126, 48, 46, 34, 71]. However, these pipelines are typically centered around a specific segmentation and tracking approach, which limits their modularity and makes application across various imaging modalities and cell morphologies challenging. In contrast, plugin-based analysis tools such as Fiji [44] and napari [45] provide a wide range of segmentation and tracking approaches [127, 33, 128, 129] and come with tremendous visualization capabilities. However, users need to assemble their own pipelines from scratch, making these tools especially useful for advanced users but challenging for beginners.

To address our identified five key abilities for good MLCI image analysis pipelines and to overcome the limitations of existing pipelines, we present `acia-workflow` - an MLCI analysis pipeline that integrates SOTA tools into modular, customizable, accessible, scalable, and sharable workflows. To achieve this, `acia-workflow` follows two complementary strategies. First, we develop the `acia` Python library to implement modular interfaces (Figure 4.1B). These interfaces enable us to integrate SOTA segmentation and tracking methods and chain all pipeline steps into a sequential execution. Moreover, modules defined by interfaces can be quickly swapped in and out, making changes in the segmentation or tracking methods convenient. Second, we assemble these modular components into Jupyter Notebooks, fusing the Python code with rich documentation, visualization, and the automated installation of all software dependencies, into a single document that we term a "workflow" (Figure 4.1C). Such a workflow contains general steps such as segmentation and tracking and can be tailored to subsequent quantitative single-cell analyses performed based on the extracted single-cell information. The resulting workflow examples are highly accessible, as

Jupyter Notebooks can be executed directly in a web browser, run on platforms that already include deep learning accelerator hardware (e.g., Google Colab), and are compatible with high-performance computing systems [72]. Containing software dependencies, code, and documentation in a single document, our workflows increase sharability, reproducibility, and customizability [130]. Finally, we design the `acia-workflow` to be easily applied to multiple time-lapse sequences, thereby achieving high scalability and unlocking quantitative insights (Figure 4.1D).

In this chapter, we develop the modular components and underlying concepts of the `acia` Python library. To showcase the five key abilities, we develop analysis workflows for analyzing three MLCI experiments: Quantifying population growth rates, performing co-culture characterization, and measuring single-cell responses to changes in the cultivation conditions (Figure 4.1E). We demonstrate the scalability of the workflows by applying them to multiple replicates without manual changes to the workflow code. To underline accessibility, sharability, and reproducibility, our developed example workflows are available online and can be reproduced directly in a web browser using Google Colab or Binder.

4.2 Analysis pipeline and workflow

Analyzing MLCI time-lapses requires a multi-step pipeline, where different modules handle image format, segmentation, tracking, cell feature extraction, and visualization or insight generation (Figure 4.1A). We design these modules within the `acia` pipeline with the previously derived five key abilities in mind. We here focus on integrating existing SOTA methods and libraries throughout all pipeline steps and add spatio-temporal visualizations and video renderings. The pipeline is implemented in Jupyter Notebooks, which we refer to as workflows, software dependencies, documentation, visualizations, video renderings, and custom code, all within a single traceable document. This fusion in a single document makes the workflow reproducible and accessible via execution in the web browser. Additionally, these workflows are designed to scale to high-throughput imaging experiments without requiring manual code changes.

4.2.1 Loading 2D+t microscopy time-lapse data

Live-cell microscopy time-lapse data comes in various formats: the popular bio-formats framework supports up to 160 different image and metadata formats (`http://www.openmicroscopy.org/bio-formats/`). Handling and processing all these different formats is a challenging task. Thus, we define a common data interface for 2D+t live-cell time-lapse data represented in a $T \times H \times W \times C$ shaped vector, denoting the time (T), spatial imaging (H, W), and channel (C) dimensions. To process the data, implement the interface `ImageSequenceSource` that provides an iterator along the temporal dimension, yielding $H \times W \times C$ images as `NumPy` arrays [131]. Thus, the user can utilize existing libraries to load the specific imaging format and convert it into an $T \times H \times W \times C$ `NumPy` array. In addition, an implementation is provided to stream image stacks from OMERO servers [132]. With `acia`, loading 2D+t time-lapses boils down to a few lines of Python code, no matter whether it is stored locally or in an online OMERO database.

4.2.2 Cell segmentation

The first step in extracting single-cell information from 2D+t sequences is segmentation, i.e., the pixel-precise detection of the individual cells (Figure 4.1A). Numerous deep learning approaches have been developed in recent years, including `Cellpose` [31], `Omnipose` [30], and `CPN` [32], which show excellent performance for specific cell morphologies and imaging modalities. To achieve high-quality segmentation, `acia` focuses on the modular integration of these SOTA approaches. Therefore, we define a `SegmentationProcessor` interface in Python that receives a 2D+t time-lapse (`ImageSequenceSource`) and computes a segmentation overlay (`Overlay`) containing cell instances, represented as cell masks or cell contours in polygon format (Figure 4.2B). We provide implementations of `Cellpose`, `Omnipose`, and `CPN` to test different segmentation models quickly. This modularity enables the user to select the most suitable method for the cell morphology under study and the imaging modality at hand. The interface is generic, allowing for the implementation and integration of future cell segmentation methods into the pipeline.

4.2.3 Cell tracking

Similar to cell segmentation, many cell tracking approaches have been developed [33, 102, 47, 125]. The most common is the tracking-by-detection paradigm, where cell segments are first detected in the images and then linked through time. Thus, `acia` defines the `TrackingProcessor` interface that uses the 2D+t image sequence (`ImageSequenceSource`) and the segmentation information (`Overlay`) and computes the resulting tracking lineage as a graph structure (`networkx DiGraph`, `https://github.com/networkx/networkx`). In this graph structure, every detected cell instance is represented as a node, and edges link them through time (Figure 4.2C). The linked cell detections form so-called "tracklets" that cover an entire cell cycle, from cell birth to its division or disappearance. These tracklets are given unique labels and are stored in a tracklet graph, where a node represents the entire cell cycle and edges indicate cell divisions (Figure 4.2D). We implement the interface for three tracking methods classic `LAP` [102], transformer-based `Trackastra` [33], and a Bayesian cell tracker `PyUAT` [133]. As for segmentation models, adding custom tracking implementations is straightforward.

Unit-aware spatio-temporal cell feature extraction

Segmentation and tracking information are the basis for extracting spatio-temporal single-cell features. These features include morphological information, such as cell location, area, length, perimeter, and fluorescence, as well as cell lifecycle readouts, including age, birth, and division sizes. The interpretation of the extracted spatio-temporal features highly depends on the experiment metadata, such as the spatial resolution of the microscope's camera, the objective used, or the imaging interval. Therefore, we implemented unit-aware cell feature extractors using automated unit computation based on the `pint` (`https://github.com/hgrecco/pint`) library. The segmentation features are extracted for every single cell detection and stored in a unit-aware `pandas` [134] data frame (Figure 4.2B). The tracking features are extracted for every single cell cycle and stored in a second `pandas` table (Figure 4.2E). The information in both tables is linked using the "label" column, assigning a unique number to every cell tracklet.

Using established `pandas` data structures simplifies custom data handling, manipulation, and

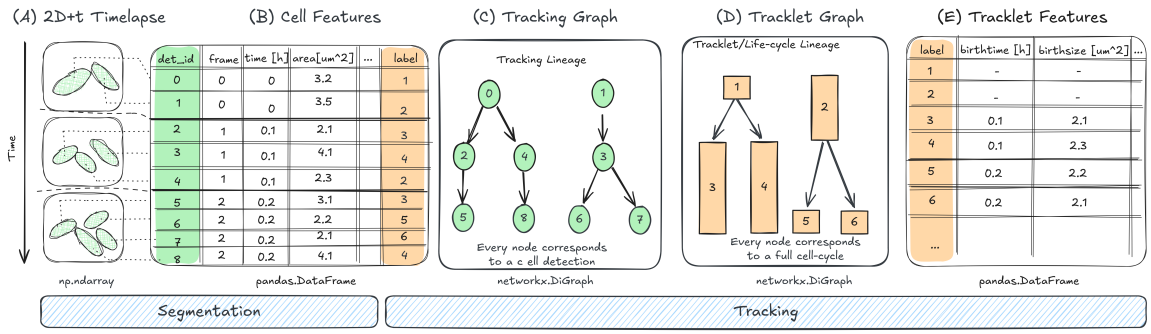

Figure 4.2: **Data extraction with `acia-workflow`. A:** The 2D+t time-lapse is segmented to extract cell instances. **B:** Spatial single-cell features are extracted with units. **C & D:** Cell instances are associated through time to build the tracking and tracklet lineages. **E:** Temporal single-cell features are extracted with units and linked to the spatial information using the "label" key. All data is stored in common Python data structures of the `NumPy`, `networkx`, and `pandas` libraries.

visualization. The attached physical units are used in downstream computations and facilitate early detection of calculation or calibration errors throughout the analysis pipeline.

4.2.4 Visualization, data analysis and insight generation

The information contained in MLCI time-lapse data is challenging to interpret as it spans both spatial and temporal dimensions, with varying levels of detail at single-cell, cell colony, or experiment levels. Thus, adding custom analyses (e.g., growth models or statistical analyses) and visualizations is crucial for generating insights into the live-cell imaging data. Jupyter Notebooks and single-cell feature representation in tabular Pandas formats enable the utilization of rich visualization tools available in Python, such as Matplotlib [135], seaborn [136], or Plotly (Plotly Technologies Inc., Montreal, Canada, https://plot.ly).

Besides static representations, video rendering and replay capabilities using `moviepy` (`https://github.com/Zulko/moviepy`) for immediate quality control of segmentation and tracking results is critical for any interpretation. The direct information feedback allows users to adapt or design custom filters, such as removing artifacts by filtering for a physiologically sensible range of cell sizes. We also provide rendering functionality for cell lineages with attached single-cell information such as cell size or single-cell growth rates. Combining these videos with charts on the development of individual cells, colonies, and cell lineage leads to comprehensive insights into the MLCI data. Interactively exploring single-cell data allows for the development of new ideas, uncovering patterns, and raising hypotheses for subsequent quantitative verification through custom data manipulations and visualizations [137].

4.2.5 From pipeline to workflow

With all these modules having clearly defined interfaces and utilizing common data formats, we assemble their sequential execution into a single Jupyter Notebook. We implement the pipeline, including software dependencies, documentation, and visualizations, and term the resulting notebook a workflow. Sharing this workflow with the imaging data is easy and allows the reproduction of the

complete image analysis pipeline results [130]. These workflows are executed in a web browser, making the image analysis accessible and easy to customize. On the other hand, these notebooks are compatible with high-performance computing setups and also used on the expert level [72].

4.2.6 Scaling to quantitative analysis of high-throughput experiments

Analyzing a single MLCI time-lapse using a workflow provides detailed insights into single-cell development, but it contains limited information about their biological heterogeneity and variability. High-throughput MLCI concurrently observes 10s to 100s of cell populations in a single experiment, bearing the potential to measure heterogeneity and variability by quantitative analysis across multiple replicates and cell populations. To apply the same analysis workflow automatically to all these recorded cell populations, we introduce the concept of scaling workflows (Figure 4.1D): We parameterized the notebook by its unique time-lapse identifier (e.g. file system path or OMERO ID) and utilize the `papermill` (`https://github.com/nteract/papermill`) library to automatically adjust the parameter in the notebook and apply the workflow for multiple time-lapses. The scaling procedure is executed in a separate notebook with parallel execution of the workflows, and the resulting Jupyter Notebook executions are stored within subfolders, documenting every analysis step, visualization, and insight. Within this scaling notebook, the single-cell data is gathered across all analyzed time lapses and used to quantify heterogeneity and variability across numerous cell populations.

4.3 Results

To demonstrate the key capabilities of `acia-workflow`, we develop three analysis workflows for publicly available MLCI datasets and extend the analyses beyond the questions answered in their original publications. Thus, our workflows highlight their suitability for analyzing diverse research questions, shedding new light on microbial behavior, and providing new quantitative insights into single-cell development.

4.3.1 Measuring microbial population growth

In biotechnology, the microbial growth rate is a major Key Performance Indicator (KPI) for evaluating the performances of novel strains under applied cultivation conditions [35, 138]. In MLCI, the biomass (population) growth rate is derived indirectly from the temporal change in cell number or cell area. However, no consensus exists across the literature on which type of growth measure to prefer. Therefore, we present a workflow that computes both - cell number and area-based growth rates - fully automatically, visualizes them, and highlights the advantages and disadvantages of each measurement approach.

In its simplest form, the temporal development of growing cell populations is characterized by counting the cells within a population over time, a task that is sometimes still performed manually [139]. Using `acia`'s automated deep learning-based instance segmentation, the Cell Count (CC) and pixel-precise area of all cells are automatically extracted for every image. We measure the temporal development of the Total Single-Cell Area (TSCA) along with the Total Colony Area (TCA) of the whole cell colony (Figure 4.3).

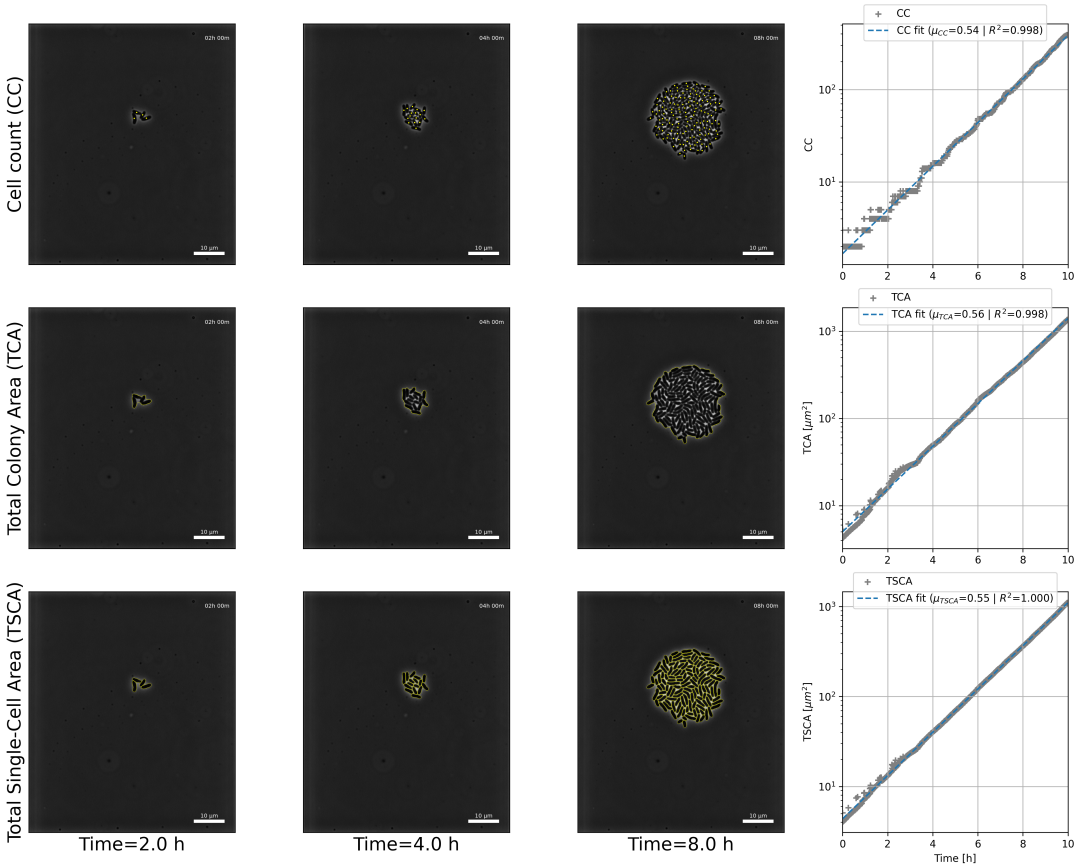

Figure 4.3: **Three variants to quantify colony growth rates:** cell count (CC, upper row), total colony area (TCA, middle row), total single-cell area (TSCA, lower row) measured for *C. glutamicum* wildtype cultivated in BHI growth medium under constant flow conditions. The growth rates are inferred from time-series data (crosses) and fitted (dashed) using an exponential growth model (right column).

For this case study, we investigate the MLCI time-lapses of the cultivation of *C. glutamicum* under constant environmental conditions [125]. Due to the continuous supply of growth medium, we expect exponential growth in cell population size. We develop a workflow that performs single-cell segmentation using `Omnipose`, filters for specific cell sizes to remove artifacts and extracts CC, TCA, and TSCA quantities. We then fit a linear model in the log space to these time series to determine the three associated exponential growth rates, μ_{CC}, μ_{TCA}, and μ_{TSCA}, respectively.

Figure 4.3 shows the extraction of the different quantities from the time-lapse data and the resulting temporal development and model fits. As expected for this dataset, the cell populations exhibit exponential growth, supported by high R^2 scores, and all three growth measures show very similar growth rates. However, the CC quantity shows discrete steps leading to oscillations around the exponential growth model. These oscillations are caused by synchronous cell division, especially within the regime of low cell numbers [140, 141, 142, 143] and become less notable with higher cell numbers. Also, the TSCA shows an oscillating behavior, but much less pronounced than the CC. However, the TCA quantity includes the area between the cells and, therefore, depends on the packing density of the cell colony. In comparison, the TSCA measures the total single-cell area and shows the smoothest alignment between measured data and the exponential model.

Both area-based measures (TCA and TSCA) are relatively robust regarding over-segmentation, as the area of the instance segmentation is summarized to a population area. In contrast, the CC measure is highly affected by under- and over-segmentation and should only be used when high-quality segmentation information is available. Comparing all three measures, as in our developed workflow, enables validating their suitability for the MLCI time-lapse dataset under study.

4.3.2 Co-culture characterization using fluorescently labeled strains

For characterizing co-culture development, it is crucial to measure their precise temporal strain-specific composition [26, 59]. The single-cell measurements in MLCI are ideally suited for that: labeling the bacterial strains with different fluorescence markers allows distinguishing cell strains by their fluorescence signals. We develop an analysis workflow that precisely measures the composition and temporal development of co-cultures and quantifies the growth rates of the individual sub-populations.

For this case study, we utilize an MLCI co-culture dataset with two *C. glutamicum* strains encoding either mVenus or E2-Crimson fluorescence proteins (SI 1, 2). The dataset is publicly available at `https://git.nfdi4plants.org/j.seiffarth/bund-et-al_2025`. As these two strains only differ in the expressed fluorescence protein and are cultivated at constant medium supply, we expect to observe exponential growth with no notable differences in the growth rates of the two strains.

We perform the segmentation using `Omnipose` and use the fluorescence `FeatureExtractor` implementation of the `acia` library to extract the mean fluorescence intensity for every cell detection. We customize the analysis pipeline by removing non-fluorescent particles (impurities and segmentation artifacts), determine their median cell fluorescence, and apply a k-means clustering to cluster the detected cells into two clusters (mVenus vs. E2-Crimson) using `scipy` [144]. Figure 4.4A shows excerpts of the time-lapse where the cell outlines are colored based on their fluorescence label.

As expected, the two *C. glutamicum* strains show exponential growth indicated by the straight regression line in the log-space (Figure 4.4B) while the determined TSCA growth rates of the two

sub-populations differ slightly (0.46 compared to 0.49 1/h). More interestingly, the analysis reveals temporal dynamics in both the average cell area and fluorescence signals (Figure 4.4C-E). The average cell area of both sub-populations shows a trend toward an average size of roughly 2 μm^2 for both populations. The smaller colony (red) shows stronger fluctuations originating from synchronous cell development and division. These fluctuations are reduced when the number of cells increases. At the same time, the average fluorescence pixel intensities measured inside the cells increase throughout the experiment (Figure 4.4E). We attribute this to the accumulation of maturation of fluorescence proteins [145]. In summary, the dynamics of cell size and fluorescent intensity must be considered when interpreting fluorescence measurements.

4.3.3 Quantifying temporal responses of individual cells to environment changes

For now, we have extracted insights into the dynamics of cell populations but have not yet investigated and compared the behavior and response of individual cells. For this, in addition to cell segmentation, tracking of all cells across the time-lapses is required to extract spatio-temporal information.

In our third case study, we leverage `acia`'s capabilities to extract single-cell tracking information to study the impact of oxygen availability on living cells. Anaerobic conditions are ubiquitous across diverse environments, ranging from the human gut to large-scale bioreactors, and are known to influence growth performance. However, the extent of this impact has been difficult to quantify and remains largely unknown. Kasahara et al. developed a microchip device to perform and collect MLCI data of *E. coli* under precisely controlled and fast (roughly 10 seconds) oxygen shifts [35]. They observed that the *E. coli* populations respond within minutes to the removal of oxygen by reducing their TSCA growth rate (Figure 4.5A). In our case study, we develop an `acia-workflow` that extends this analysis to the single-cell level, confirming that this effect is also present for single cells. Moreover, we investigate the consistency of the cells' responses in terms of timing and adaption to these alternating oxygen conditions.

To this end, we developed an analysis notebook using `Omnipose` segmentation followed by cell tracking using the `trackastra` cell tracker. To remove artifacts, we first correct for over-segmentation, only investigate cells that are observed over their entire life cycle (i.e., birth to division), and use `acia`'s `FeatureExtrator` to extract and quantify the single-cell area. As a reference, Figure 4.5A shows the temporal TSCA development, reproducing the results reported by Kasahara et al. [35]. Phases of 21% oxygen are colored green, and phases of 0% oxygen are colored red. In their study, Kasahara et al. observed a substantial decrease in TSCA growth upon entering the anaerobic phase (e.g., at $t = 1.5\ h$). However, the TSCA's population view on the data obscures differences in individual cell responses within the same population, which may differ substantially in their strength and timing.

Tracking individual cells provides insights into the temporal response of each cell. Figure 4.5C shows the lineage tree reconstructed by the cell tracking. The lineage illustrates the development from a single cell within the population (left) to an increasing number of cells resulting from cell divisions, as indicated by branchings in the lineage. Based on this lineage, we select tracked cells that have been observed before and after the entry of the anaerobic phase at $t = 1.5\ h$. Figure 4.5B shows the development of five of these cells experiencing the switch from aerobic to anaerobic conditions

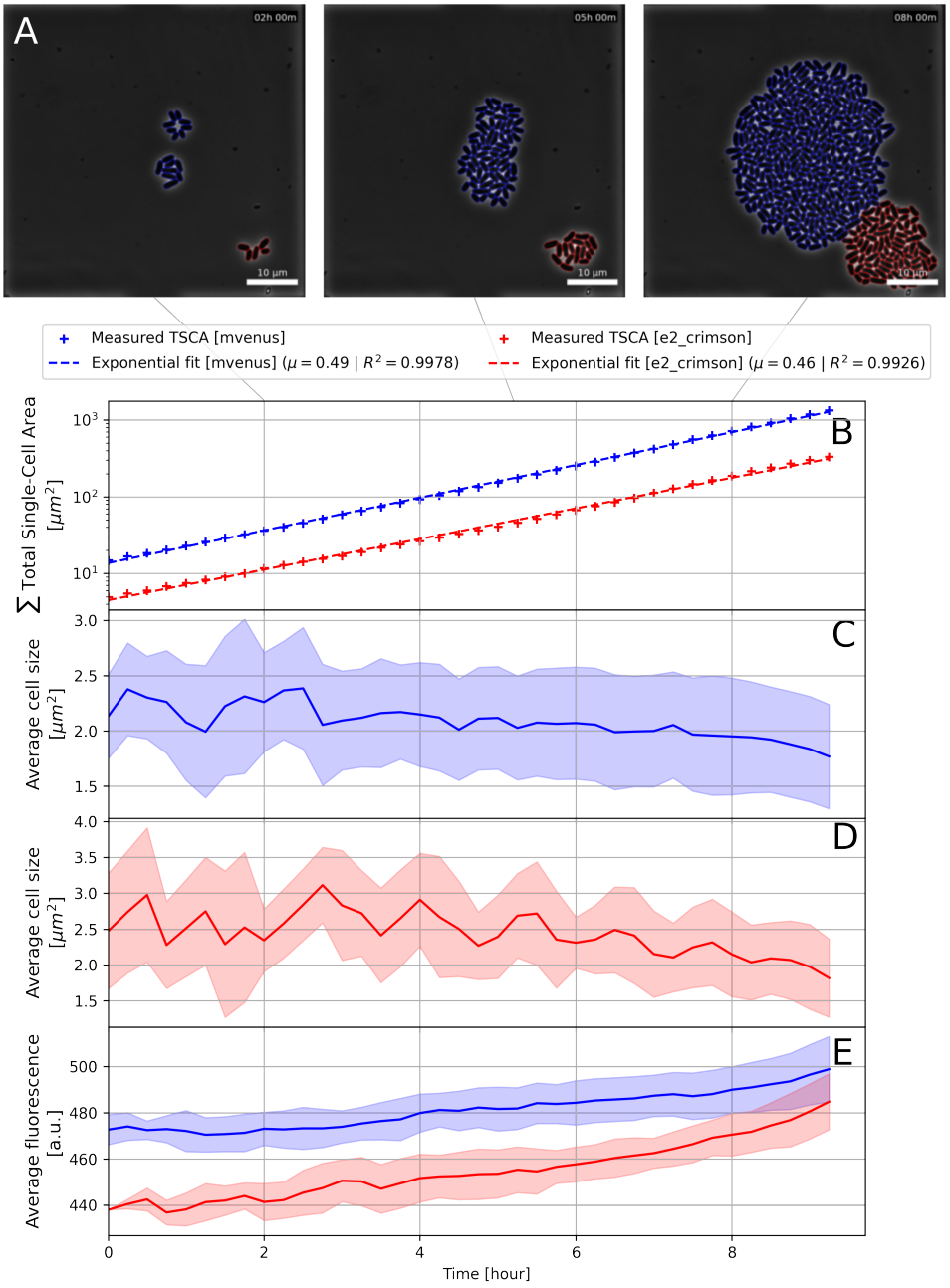

Figure 4.4: **Co-culture cultivation with fluorescence-based strain labeling.** Two *C. glutamicum* strains with E2-Crimson (red) and mVenus (blue) simultaneously cultivated in `CGXII` medium. (A) Snapshots of the time-lapse with cell outlines colored red and blue depending on their fluorescent label. (B) Temporal development of the measured TSCA of the two sub-populations (crosses) and the fit of an exponential growth model (dashed). (C-D) Average cell area distribution of the sub-populations, with a variance tube (one standard deviation) in light colors. (E) Average fluorescence of the labeled strains (blue/red) with variances (one standard deviation).

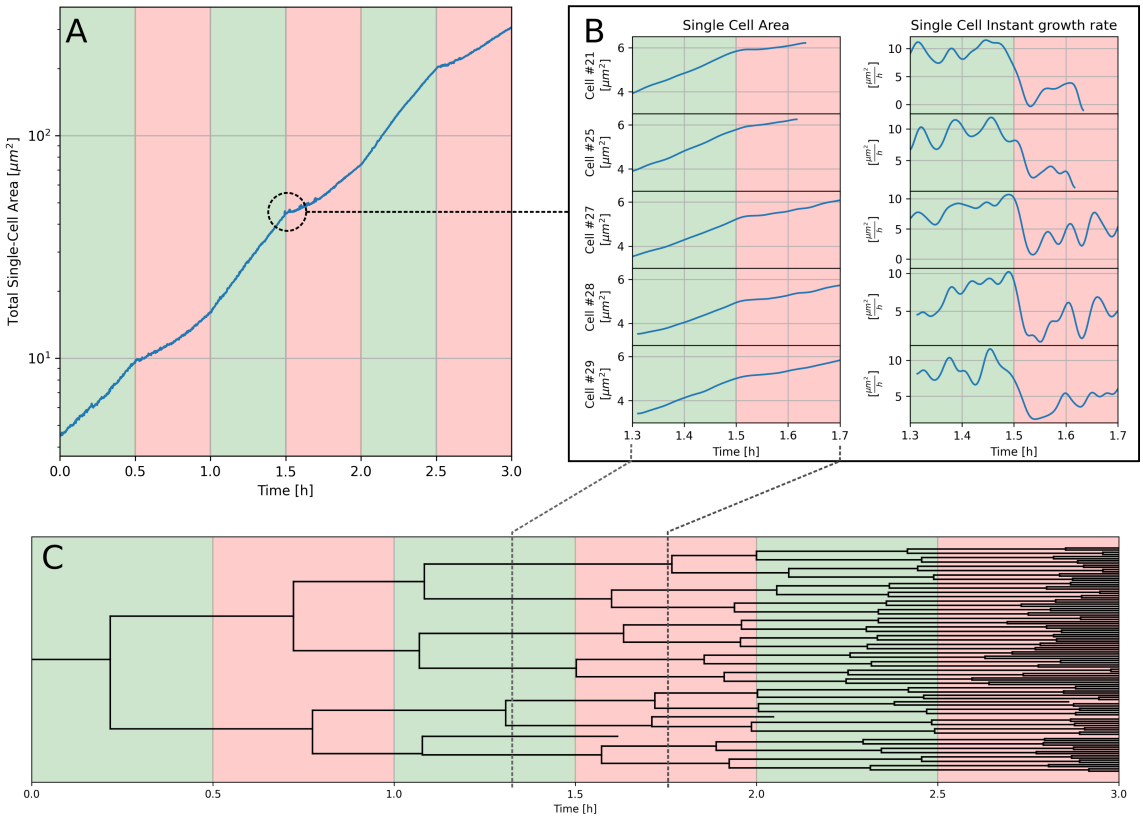

Figure 4.5: **Single-cell insights under oxygen shifts.** (A) TSCA development of *E. coli* as it has been measured by Kasahara et al. [35] at alternating aerobic (21% O_2, green) and anaerobic (0% O_2, red) cultivation conditions. (B) Single-cell area development and instantaneous growth rate for five individual cells experiencing the switch from the aerobic (green) to anaerobic (red) cultivation conditions at $t = 1.5\ h$. (C) Lineage tree generated with automated cell tracking.

($t = 1.5\ h$). We measure their individual area and the Instantaneous Growth Rate (IGR), i.e., the time derivative of the single-cell area (SI 3). Interestingly, the IGR shows a strong and rapid decrease shortly after the shift from aerobic to anaerobic cultivation conditions ($t > 1.5\ h$) and reaches a local minimum within minutes after the oxygen change. Interestingly, this rapid response to the change in oxygen availability is remarkably consistent across all five cells.

4.3.4 Scaling to quantitative insight in high-throughput MLCI

The presented analysis workflows have extracted detailed single-cell information embedded in a single MLCI time-lapse. To assess the variability of the single-cell behavior and the significance of the insights, we exploit the high-throughput capabilities of MLCI, recording time-lapses of multiple cell populations and analyzing them using `acia`'s workflow scaling functionality. Here, the developed workflows are automatically rolled out to all replicates of an experiment, and quantitative results are summarized and visualized.

In our first case study, we have been interested in the types of measurements used to determine growth rates. By analyzing multiple time-lapses, we would now like to investigate the variability in these growth rate measurements. Figure 4.6A shows the quantitative analysis of population growth measures with N=5 replicates for CC, TCA, and TSCA measures, including the growth rates derived by regressing the measurements with an exponential model (top to bottom). The temporal development of all three quantities is well-described by the exponential model, and all three growth measures give similar growth rates across all replicates with small variations in the range of 0.52 h^{-1} to 0.57 h^{-1}. Moreover, the analysis confirmed a difference between count-based and area-based population quantification for low cell counts: the two area-based measurements (TCA, TSCA) show almost no fluctuations compared to the CC measure, which shows a step-wise increase due to synchronous cell divisions. Moreover, replicate "03" consistently shows the lowest growth rate across all three measurements and may qualify as an outlier. However, more replicates are necessary to make this assessment. In summary, scaling the growth measurement workflow to all five time-lapses enables the quantification of variation in measured growth rates and the identification of potential outliers.

In our second case study, we have demonstrated the capability to precisely measure the growth rates of co-culture populations, using the example of mVenus and E2-Crimson labeled *C. glutamicum* strains. This raises the question of whether there is a systematic difference in their growth rates due to the two fluorescence reporters. To quantify the variation in growth rate measures, we analyzed multiple cultivations.

Figure 4.6B shows the quantitative analysis of the fluorescence label strains with N=8 replicates for the TSCA (top) and extracted growth rate (bottom). The temporal development of the TSCA shows consistent exponential growth (linear in log-space) for all 16 cell populations. Based on the TSCA development, we measured the growth rate of all populations and quantified the variability and average growth rate for mVenus and E2-Crimson populations (Figure 4.6B, bottom). Our analysis yields an average TSCA growth rate of 0.477 $h^{-1} \pm 0.026$ for mVenus and 0.473 $h^{-1} \pm 0.030$ for the E2-Crimson strains, respectively. Despite the slightly higher average growth rate of mVenus under consideration of the variability in the measured growth rates, we conclude that the growth rates do not show a systematic difference. Thus, encoding the two fluorescence proteins does not lead to a measurable difference in growth rates in our MLCI experiments.

For the last case study, we have already shown that individual cells of a cell populations react consistently within minutes to the abundance of oxygen by reducing their IGR. To ensure that this effect occurs consistently across different cell populations, we have rolled out the analysis workflow across all $N = 5$ replicates available in the dataset. We analyzed the response of *E. coli* upon switching from the aerobic to the anaerobic phase in the middle of the experiment ($t = 1.5\ h$). Employing single-cell tracking, Figure 4.6C shows the single-cell area development (top) and the instantaneous growth rate on the single-cell level (bottom) for (K=30) cells. As expected, cell growth is faster in the green phase (aerobic) than in the red phase (anaerobic). Moreover, all individual cells show variations in their IGR (bottom) before entering the anaerobic phase. Upon entering the anaerobic phase, however, we confirm an immediate steep drop in the IGR for all cells across all five replicates. After the steep drop at the beginning of the anaerobic phase, the cells again show large variations in their IGR. In summary, despite the large fluctuations in single-cell growth rates (IGR) at constant oxygen conditions, all cells respond by a very consistent and fast decrease of their IGR to the removal of oxygen from their environment.

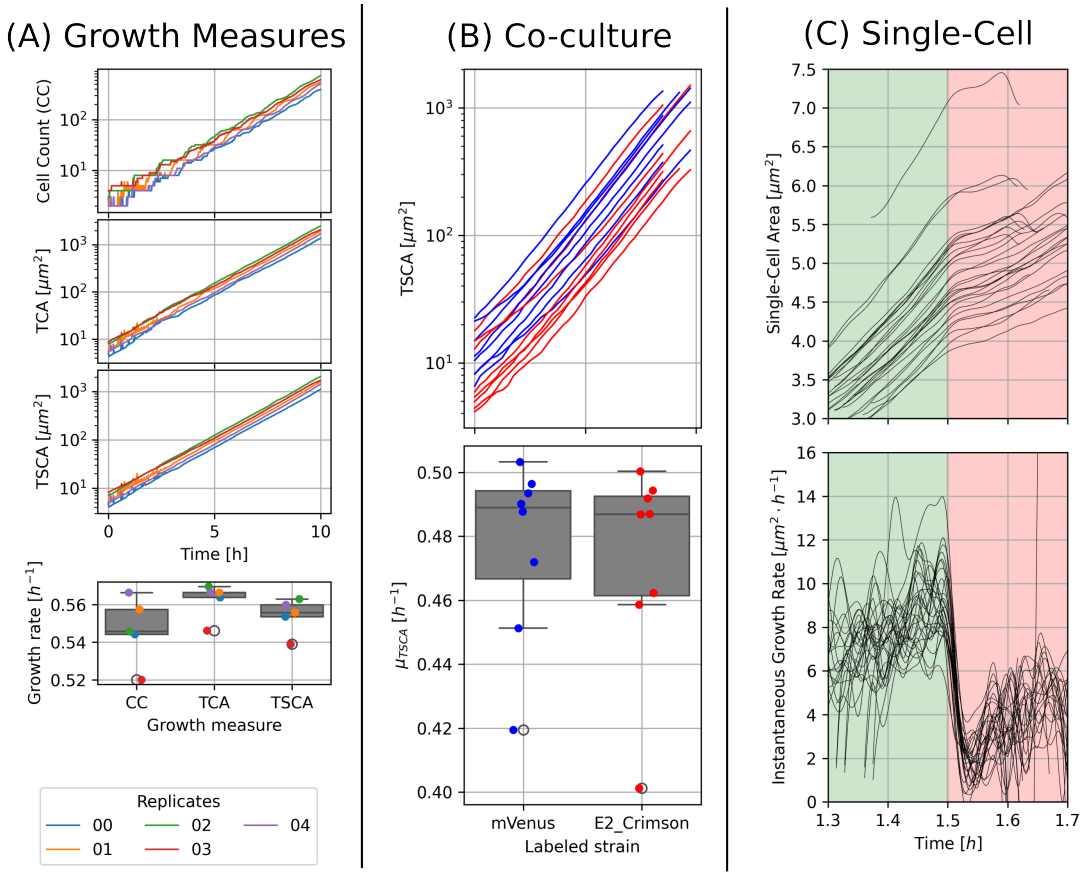

Figure 4.6: **Scaling MLCI analysis to multiple replicates.** (A) shows CC, TCA, and TSCA development (top to bottom) for (N=5) *C. glutamicum* replicates. The growth rate distribution for all three measures is shown at the bottom. Measurements for the replicates are shown in different colors. (B) shows the *C. glutamicum* co-culture TSCA development (top) and exponential growth rate quantification based on TSCA (bottom) for (N=8) replicates for mVenus (blue) and E2-Crimson (red) labeled strains. (C) shows the single-cell area development (top) and instantaneous growth rates (bottom) of *E. coli* in aerobic (green) and anaerobic phase (red) for (N=5) replicates and (K=30) individual cells.

4.4 Discussion & Outlook

MLCI poses high requirements on the flexibility, robustness, and scalability of analysis workflows to reliably extract insights from time-lapse data, including the emerging biological heterogeneity. With `acia-workflow`, we present the combination of the `acia` Python library and Jupyter notebooks into workflows that provide a modular analysis pipeline with extensive customization, are easy to use, fully reproducible, and scale to high-throughput screenings. We have demonstrated these capabilities with three case studies, where the implemented workflows yield new insights into the behavior of single cells and their variability.

For now, `acia` is primarily designed to process 2D+time time-lapses, however, we plan to implement modules for 3D segmentation and tracking methods in the future to extend the analysis capabilities to 3D image sequences. Moreover, our case studies focused on the analysis of bacterial cells. However, `acia` is not limited to bacterial cells, and we plan to extend the suite of example notebooks to other cell morphologies and imaging modalities.

Customizing the `acia-workflow` notebooks requires users to write Python code. While writing code for domain scientists may come with a steep learning curve, large language models are effective in writing short Python snippets, e.g., for custom visualizations or data handling [146]. Thus, we believe that the accessibility of Jupyter Notebooks, which are executed online in a web browser, outweighs the challenges of learning to write concise Python code. Moreover, the code precisely documents every step in the pipeline, allowing others to share and reuse existing snippets.

By publishing these `acia-workflows`, we have lowered the entry barrier for MLCI analysis while demonstrating customizability and reproducibility for high-throughput analyses. The automated analysis workflows represent an important step toward democratizing MLCI and even more general live-cell imaging analysis workflows, unlocking the potential of single-cell insights for both life and data scientists.

Chapter 5

Customizable and interactive visualizations for investigating spatio-temporal single-cell information

Contents

5.1 Motivation **56**

5.2 Customizable visualization views **57**

5.3 Results & discussion **59**

5.4 Conclusion **60**

Modern live-cell imaging and machine-learning based image processing methods shed light on the secrets of the behavior and characteristics of individual cells including their growth and development into colonies. However, the amount and complexity of spatio-temporal live-cell data makes gaining insights notoriously difficult. Therefore, we present novel visualizations that combine spatial and temporal information in interconnected and highly customizable view components. We demonstrate the flexibility of these components by designing experiment-specific visualizations at the example of three live-cell imaging data sets and gain insights into cell division, population formation and dynamic growth rates. The visualizations are implemented in Python notebooks and publicly available at `https://github.com/JuBiotech/Supplement-to-Seiffarth-et-al.-ISBI-2024`.

The content of this chapter is published in: Seiffarth, J.; Blöbaum, L.; Grünberger, A.; Nöh, K.: Customizable and Interactive Visualizations for Investigating Spatio-Temporal Single-Cell Information. IEEE International Symposium on Biomedical Imaging. 2024. `https://doi.org/10.1109/ISBI56570.2024.10635297`.

5.1 Motivation

MLCI is a powerful analysis method for single-cell biology. For instance, single-cell information fosters our understanding of the formation of cell colonies in auxotrophic co-cultures [26], supports drug development [147], and allows investigating the effects of dynamic environmental conditions on the cell growth behavior [36, 148]. In recent years, a suite of computational tools has become available to extract the underlying single-cell information from live-cell imaging sequences. Most notably, machine learning-based cell segmentation [60] and tracking methods [49] are nowadays capable of extracting spatio-temporal information for every single cell within an experiment. This way, rich information sets emerge, containing cell locations, their movement, and migration over time, which is to be aligned with biosensor expression profiles and cell division events. Bringing together such spatial and temporal information layers to gain insights is challenging due to the size, complexity, and detail of live-cell imaging data sets.

Interactive visualizations serve as an invaluable tool to explore spatio-temporal information of live-cell imaging and combine various information layers at different levels of detail. Software tools such as *Vizardous* [149], *CeLaVi* [150], and *CellTrackVis* [151] offer spatio-temporal data visualizations and provide crucial insights into live-cell datasets. *Vizardous* makes spatio-temporal single-cell information available by combining the visualization of the cell lineage tree with cellular characteristics, including cell length, area, and fluorescence signals. The *CeLaVi* platform introduced visualizations in the web browser to simplify user access and easily explore cell lineages and their spatial arrangement in three dimensions. The tool adds information layers by incorporating cell IDs, lineage coloring, lineage tree pruning, and branch-length information. *CellTrackVis* also utilizes the web browser, comes with a Python backend, and adds tight interconnection between multiple view components to study spatio-temporal single-cell information effectively.

All three tools emphasize the importance of interactive visualization for exploring single-cell information and show that access barriers have been reduced by using installation-free web browser applications. However, a critical limitation persists: while the tools' pre-configured views are interactive, they present a fixed set of information and need more capacity for customizing the displayed information. Thus, adding a custom view component to show specific information crucial for investigating an experiment requires extensive programming and is often impractical for users. However, this need for customization occurs frequently and is highly dependent on the research question of the experiment. For instance, investigations that focus on the temporal development of individual cells and their lineage need lineage tree visualizations and temporal depiction of specific cell characteristics such as cell size or biosensor response. In other experiments, the spatial distribution of cells and its influence on their sub-populations are of major interest. These experiments benefit from visualizing spatial arrangements, cell division behavior, and the number of descendants over time. Lastly, when rare events or developments in cell properties are in focus, visualizations that focus on interesting subsets using, for instance, filters for cell size or age are useful. These three examples show the vastly different investigation targets in live-cell imaging and underline the demand for customizable visualizations for precisely investigating the information of interest.

Therefore, we developed novel interactive visualizations for investigating single-cell information, combining ease of use in the web browser using customizable Python notebooks and interconnect view components. These components include lineage trees, segmentation overlays for microscopy images, and the temporal development of single-cell analysis. The notebooks utilize the visualization

frameworks *Plotly* [152] to visualize single-cell information in interactive charts and the *Dash* [153] library to organize them in layouts and react to user inputs. To demonstrate the flexibility of this approach, we selected three different live-cell data sets from the biomedical and biotechnological domains, designed custom visualizations, and gained exciting insights into the data sets. Our proposed interactive visualizations simplify the exploration of complex spatio-temporal single-cell data, address the lack of customization capabilities of existing visualization tools, and provide important templates for experimentalists to tailor experiment-specific visualizations. Integration with *Colab*[1] and *Binder* [154] facilitate reproduction and simplify the usage of the developed notebooks.

5.2 Customizable visualization views

For exploring the single-cell information from different points of view, we developed three visualizations: the (A) *lineage view*, the (B) *descendants view*, and the (C) *distribution view*. All three views are implemented in Jupyter notebooks utilizing the *Ploty* and *Dash* libraries. The single-cell information is loaded into pandas [155] tables, and the tracking lineage is stored in a directed graph in networkx [156]. These established libraries allow for easy adaptation to custom data formats. All three views contain different visualization components and are designed to explore experiment-specific single-cell information.

5.2.1 The lineage view

To explore the spatio-temporal information of single cells in an interactive visualization, we combine the lineage tree visualization and the spatial arrangement of the individual cells, their segmentation, and the microscopy image. Figure 5.1A shows the lineage view composed of four components. The upper left side shows the lineage component, where nodes represent cell detections. Edges and branches illustrate the cell connection through time into trajectories and cell division events. The size of the nodes is proportional to the cell area and is customizable to other single-cell properties stored in the *Pandas* table. On the right side, we show the recorded microscopy image together with the segmentation of the currently selected cell (red). The left bottom part shows the development of the single-cell size over time. The three visualization components are interconnected and clicks on lineage nodes, cell instances, or data points within the cell property development lead to an update of the selected cell and the displayed information. In addition, at the bottom right, a video shows the complete cell track of the currently selected cell highlighted on the microscope image.

5.2.2 The descendants view

Investigations of the influence of spatial locations on the development of single cells are often interested in the size of the sub-populations originating from cells at different spatial locations. Figure 5.1B shows the interactive visualization designed for exploring the development of individual cells and their descendants by combining their segmentation and tracking information. The upper left two images show the segmented microscope images at an early (left) and later (right) time that is selected using the slider below the images. Clicking on a cell in the early or late image leads to a color-coded selection of the ancestor (left) and descendant cells (right). The temporal development

[1] `https://colab.research.google.com/`

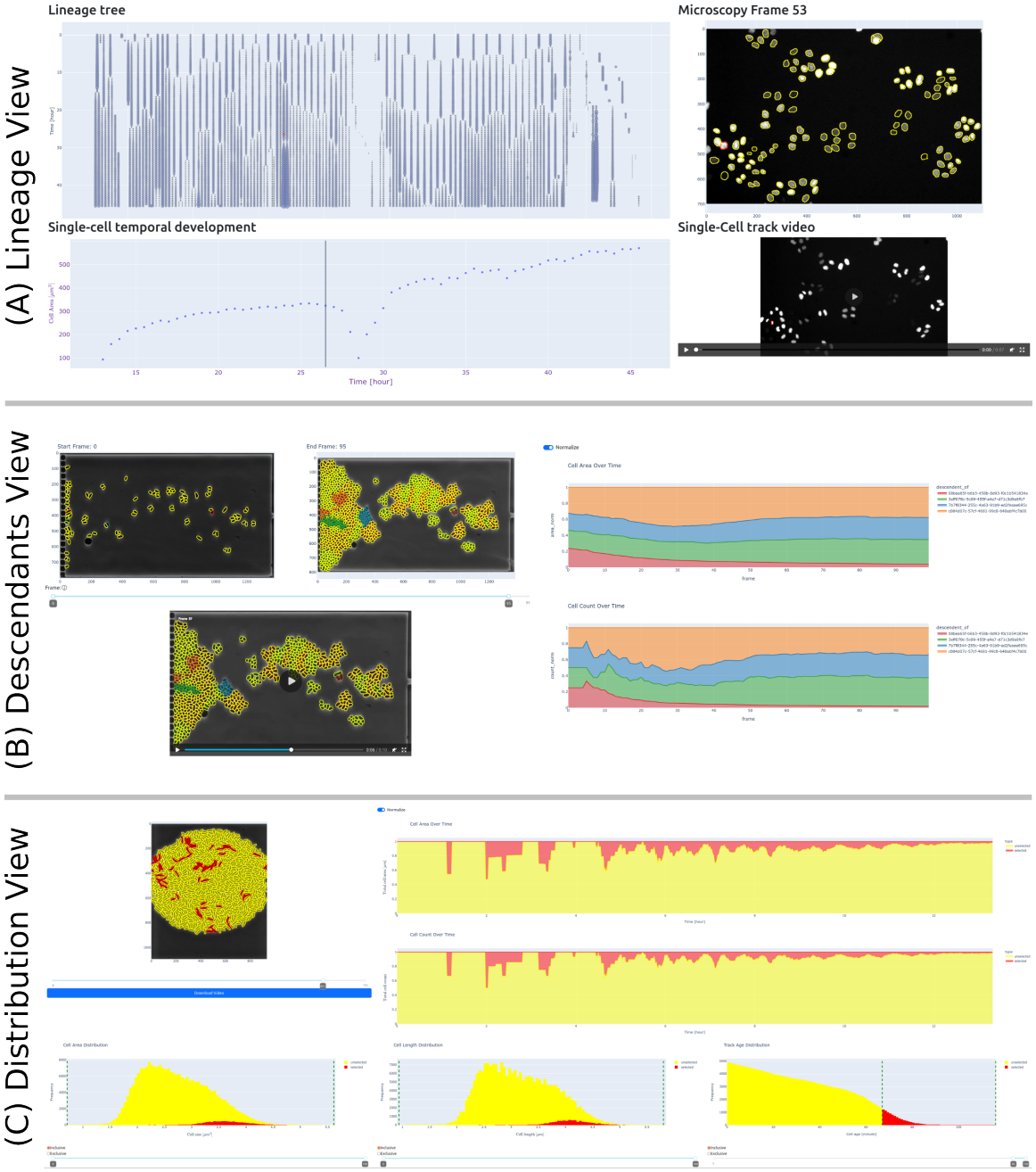

Figure 5.1: **Screenshots of the developed interactive visualizations.** (A) Visualization of the lineage tree and selection of individual cell detections highlighted in the microscopy image. The cell size is mapped to the lineage tree node size and visualized over time. (B) The descendants view allows selecting cells at a specific point in time and investigating their progeny's development. Selected cells and their progeny are color-coded and combined with the temporal development of their size and count proportions. (C) The distribution view is designed to explore large data sets. Hence, single-cell properties, such as cell area, length, and age, are used to filter interesting cells and to visualize the arrangement of interesting cells in the microscopy image as well as the amount of interesting cells over time.

of the descendants between the two selected time points is depicted in a video generated on demand. At the right, additional information about the temporal development of area and cell counts of the selected cell descendants are visualized in the same color coding to explore the proportions of the selected cell populations.

5.2.3 The distribution view

Visualizing temporal and spatial information in the form of lineage trees and images becomes demanding for large-scale data sets. Browsing a lineage tree with hundreds of thousands of nodes and branches is not informative and likely exceeds the computational capabilities and human perception. To explore data sets of this scale, we provide the visualization in Figure 5.1C that allows browsing the distribution of single-cell characteristics focusing on cells of interest, for instance, especially large or old cells, and explore their spatial and temporal occurrence. Therefore, the upper left component shows the microscopy image at a given time with selected (red) and unselected (yellow) cells. The slider below is used to screen different time points of the live-cell experiment. At the bottom, the user gets three different options to filter the distribution of the single-cell characteristics using spatial properties such as cell size and length and temporal properties such as cell age. As an example, we visualize the distributions of these three cell characteristics together with a range slider to select a lower and upper quantile that updates the cell selections. By default, the cells having characteristics between the lower and upper quantiles are highlighted in red in the images and histograms (inclusive mode). When selecting the exclusive mode, cells having characteristics outside these quantiles are highlighted. The filtering is beneficial for finding and investigating extreme cell age, area, or length cases. Using customizable Python notebooks allows the user to integrate additional cell characteristics distributions, for instance, fluorescence intensity values, tailoring the investigation of single-cell properties in large-scale time-lapse recordings.

5.3 Results & discussion

We choose three relevant data sets from the fields of biomedical research and biotechnological process development to demonstrate the customizability of the developed interactive visualizations and their broad applicability to investigate and detect patterns in single-cell development: First, a publicly available HeLa cell data set recorded in the context of investigating cell division genes [157]. Second, a microbial co-cultivation using genome-reduced strains with auxotrophic dependencies for establishing new bio-based production processes [26], and third, the most extensive available tracking data set featuring the soil bacterium *C. glutamicum* with a total of 238k cell detections and close to 5k cell tracks [158]. Figure 5.1A-C shows the application of the three developed views to the three datasets in the aforementioned order.

For investigating the HeLa cell data set, we choose the lineage view as it combines cell lineage and cell track visualization (Figure 5.1A). Therefore, we reconstruct the cell instance segmentation and tracking information from silver truth annotations provided by the Cell Tracking Challenge [49]. In the lineage view, cell development and divisions can be observed, for example, to investigate cell behavior before and after cell division and to study their morphology in microscopy images. We observe that HeLa cells contract before cell division and change their morphology in the microscopy image. However, the lineage tree shows that not all cell contractions lead to cell division events.

The second dataset shows cells growing within an auxotrophic co-culture, where cells are expected to show different growth behaviors depending on the spatial composition of their neighborhood. Cell clusters of both strains should benefit from cross-feeding, while clusters with a single strain might suffer from resource limitations, decreasing the cell growth. In order to investigate this effect in the descendants view, we have segmented and tracked cells using *ObiWan-Microbi* [110]. The descendants view allows selecting and comparing individual cells and monitoring their development into cell clusters (Figure 5.1B). We recognize a wide variation of cell growth and progeny formation, and the proportions of selected sub-populations vary over time. However, we find clusters containing only a single strain that grow into a large progeny and clusters of mixed strains with only a small number of offspring. Larger co-culture datasets with higher temporal resolution are necessary to further investigate these effects.

Such high temporal resolution data sets often become extensive in the number of cell observations and tracks. The third data set represents such a large data case, and in order to explore and filter the data set for interesting cells, we utilize the distribution view (Figure 5.1C). Despite the large number of cell observations and tracks, the view interactively provides filtered single-cell information and allows exploring the cell characteristics of cell size, length, and age. By filtering the data set for the age of cells, that is, the number of minutes a cell trajectory is observed, the visualization reveals that cells become especially old during the first half of the cultivation time. The proportion of these old cells decreases towards the end of the cultivation. Further age filtering indicates that cells divide younger and grow faster at the end of the cultivation despite being exposed to constant environmental conditions. However, these findings are based on filtering a single live-cell experiment and require quantitative verification.

All three presented visualizations uncover interesting information in their application and, therefore, underline the importance of interactive data exploration. In particular, their implementation in Jupyter Notebooks makes them usable as templates for further customization and adaptation to investigate specific research questions. Although these customizations require training in the Python programming language and the *Plotly* and *Dash* libraries, the underlying Python code is also useful for further quantitative analyses and hypothesis validation. For now, the presented visualizations are restricted to 2D+time data and not yet suitable for 3D+time live-cell recordings. However, the customizability of the notebooks simplifies the integration of 3D visualizations and custom data analyses in the future.

5.4 Conclusion

In this chapter, we have introduced novel visualizations that utilize interconnected and interactive view components to explore in-depth information about live-cell experiments in customizable Python Notebooks. We have demonstrated that these visualizations reveal crucial insights across several live-cell imaging data sets. Their customizability is critical to gaining experiment-specific insights, and the interconnection of view components allows analyzing spatio-temporal single-cell data at different information layers. Our publicly available example notebooks allow to interactively explore the recorded data sets and gain fundamental insights into experiment data. The developed notebooks provide the foundation for future development and customization to new data sets and visualization challenges in live-cell imaging.

Part II

Smart and event-driven microfluidic live-cell imaging

The previous four chapters (Part I) have established the data foundation and analysis for high-throughput MLCI experiments, gaining single-cell insights at different levels of granularity. The analysis system has been deployed during its development in our lab, analyzing more than 800 million cell detections in roughly 2.5 million microscopy time-lapse images to this date. Moreover, it has been a cornerstone of systematically gaining single-cell insights in several publications [26, 36, 35, 123, 142].

All these data analyses are performed after the MLCI experiment, leaving a time gap between the experiment conduction and the insight generation. The high automation and scaling of the developed analysis workflows have drastically reduced this time gap. However, it does not change the fact that the high-throughput MLCI experiments are typically conducted blindfolded: Imaging data is continuously recorded, but apart from manual inspection of the images, no quantitative insight into the cell development is available during the experiment. Moreover, MLCI experiments are usually purely observational: After initiating the imaging routine, we become passive observers of the development of the cells. Thus, the availability of real-time information on cell development and the opportunity to use this information to adapt the observation routine or dynamically change influencing factors bears immense opportunities to investigate the complex dynamic behavior of living cells.

In recent years, the concept of smart microscopy has emerged as a first step: smart microscopy experimentation bears the potential to utilize real-time information derived from image recordings during the experiment to adapt the imaging routine [40]. This allows us to make autonomous decisions during the experiment, for example, to adapt the imaging routine to better image the underlying living sample and minimize imaging-based damage, such as phototoxicity [40]. This idea of using real-time information can be extended to exert influence on the living cells. Especially the development of advanced optogenetic tools allows the influence of intracellular processes using light exposure - ubiquitous in fluorescence microscopy imaging systems [28]. This light-based influence led to the emergence of the field of cybergenetics, a concept for implementing closed-loop control of cell populations [159, 39]. In such a closed-loop system, the cell population is continuously monitored, and real-time information is used to compute a specific influence to achieve a certain cell behavior. Lugange et al. have extended this to a high-throughput microfluidic system, controlling protein secretion in hundreds of individual *E. coli* cells trapped in mother machines [37].

The ability to develop such smart imaging routines using microscopy platforms has been additionally accelerated by the development of high-quality deep learning tools for cell segmentation [32, 30, 31] and tracking [33], improvements in open-source microscopy control using Micro-Manager [160] and Pycro-Manager [161], and advances in the precise and local influence of living

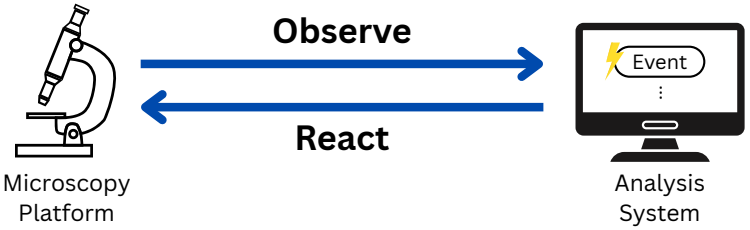

Figure 5.2: **Event-driven imaging for smart microscopy experiments.** Event-driven imaging systems continuously observe the living sample using the microscopy platform, analyze the images and trigger events as a reaction to these real-time readouts. These events may alter the imaging routine or change other experiment parameters to better investigate the living sample.

cells, especially using optogenetics [162]. In combination with high-throughput MLCI, this unlocks completely new opportunities in screening the dynamic influence on single-cell behavior in many cell populations or studying closed-loop control strategies at a high-throughput level [37].

However, processing real-time information and adapting imaging routines increases the complexity of the experiment designs, making them extremely demanding to conduct. In recent years, event-driven microscopy software platforms have been developed (Figure 5.2). Platforms such as Cyberscopy [41] and MicroMator [42] extract real-time information and process it as events to make decisions during live-cell experiments. Both tools have demonstrated their flexibility across a wide range of applications, including phototoxicity screening, adaptive imaging rates, microfluidic environment adaptation, and model predictive control of cell populations. These applications highlight the immense potential of making microscopy routines event-driven and ”smart”.

Despite the advances of available tools, event-driven microscopy in high-throughput MLCI experiments remains challenging. Real-time image analysis of flexibly designed microfluidic RoIs is challenging as chip structures vary and may be indistinguishable from living cells. Moreover, the high-throughput concept in MLCI increases the complexity of designing such experiments, as it involves handling hundreds of cell populations concurrently, making it very difficult for developers and users to design custom imaging routines. Thus, we need an imaging platform that simplifies designing event-driven imaging routines that react to real-time biological information while operating in high-throughput mode, allowing us to scale to the many independently developing cell populations.

In this second part of this thesis, we will step-by-step develop the tools for such a nouveau platform. In particular, we introduce an entirely new perspective on high-throughput MLCI by drawing an analogy to modern computer operating systems. This perspective allows us to implement so-called ”imaging applications” that define the event-driven imaging routine specific to each cell population and RoI on the microfluidic chip. These imaging applications are then all executed with the JediOS. Figure 5.3 illustrates the concept of this operating system, where multiple independent, event-driven imaging applications are executed simultaneously. Each imaging application contains the event-driven imaging logic expressed in the Event-driven Imaging Language (EDIL) - a state-machine-based formalization of events and imaging requests in the form of states and transitions. JediOS then provides a scheduler to orchestrate the concurrent execution of all independent imaging applications and manages the execution of imaging requests on the imaging platform. Importantly, JediOS supports real microscopy platforms and an in-silico simulation platform such that entire

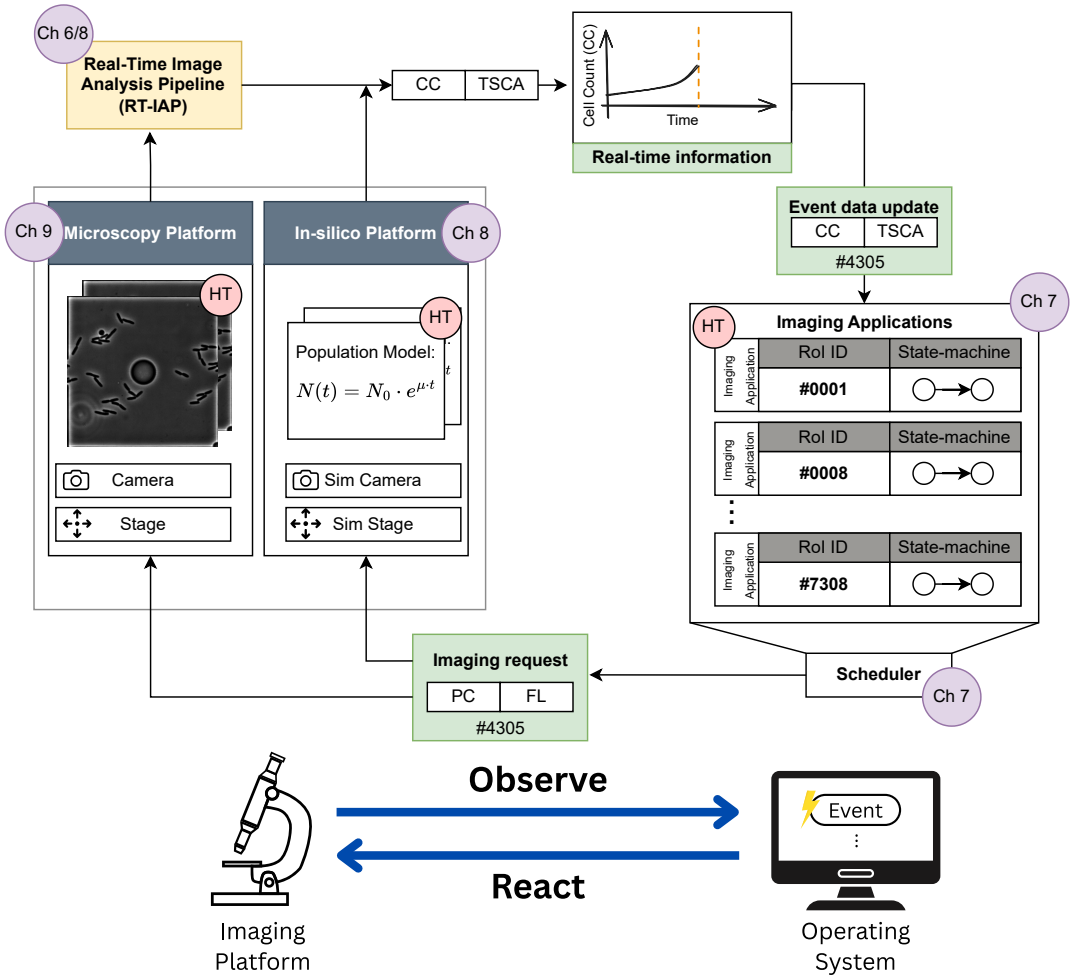

Figure 5.3: **Schematic visualization of the JediOS**. JediOS unlocks the event-driven imaging approach for high-throughput live-cell imaging across versatile imaging platforms. Therefore, it combines imaging applications that implement imaging routines for individual RoIs, imaging platforms that record images, and a real-time image analysis pipeline that extracts real-time information. For high-throughput imaging, there is one imaging application per RoI composed of the RoI ID and a state-machine that manages the imaging routine. The scheduler then combines multiple imaging applications into a high-throughput experiment orchestrating their imaging requests. These imaging requests are executed on an imaging platform, either the real microscopy platform or an in-silico simulation platform that only simulates an MLCI experiment. Real-time information about the cell development is then either extracted by the Real-time Image Analysis Pipeline (RT-IAP) or simulated by population models. These real-time readouts, such as CC or TSCA, are then used to trigger events in the individual imaging applications. The chapters describing the development of the individual components are shown in purple. "HT" marks components that exist for every RoI and cell population.

experiments can be simulated. The real-time information on the cell population development, such as CC and TSCA, is computed by the RT-IAP or the in-silico models and is used within events to update the state-machines driving the imaging applications.

This operating system perspective on high-throughput MLCI unlocks completely new opportunities in designing, simulating, and executing such experiments. In the following chapters, we develop its building blocks and verify these opportunities in simulations. Furthermore, we demonstrate the capabilities of JediOS in a real experiment, screening optogenetic influence on living cells. We achieve a simpler experiment design and much higher throughput in the screening for optogenetic influence parameters, and combine this into a smart experiment routine that automates the experiment conduction and simplifies its repetition.

The remainder of Part II is structured as follows, where each chapter builds upon the achievements of the previous chapters:

- **Chapter 6** deals with the development of the DMC that fuses digital microfluidic chip design with real-time deep learning-based image analysis. The DMC unlocks the ability to transfer the semantic understanding of microfluidic structures from the digital blueprint to the recorded image, thus allowing for the real-time analysis of images with arbitrarily complex microfluidic structures. Moreover, the DMC provides a precise map of all RoI positions of a high-throughput chip, finally making the high-throughput capabilities of modern microfluidic chips available.

- **Chapter 7** introduces the JediOS. This operating system is based on the principal event-driven concept already introduced by Cyberscopy and MicroMator but generalizes the approach to high-throughput experimentation and employs a divide-and-conquer paradigm to perform simple imaging programming for each RoI and cell population independently. Developing the EDIL allows the design of state-machines that contain the logic of imaging and event reaction. Moreover, we develop and implement an in-silico platform to simulate and test entire high-throughput experiments before moving into the lab. At several simulation examples, we demonstrate the potential of JediOS to utilize the high-throughput of MLCI more efficiently and to unlock entirely new experiments with closed-loop control.

- **Chapter 8** utilizes the DMC and JediOS to develop and implement a smart MLCI experiment routine and demonstrates its capabilities at the example of a high-throughput screening of light-based stress induction on living cells. In particular, we introduce a general five-step procedure to build a foundation for any smart MLCI experiment, highlighting the advances of a fully software-based experiment design, the flexibility in experiment design, and the advantage of real-time information for live insights and experiment conduction.

- **Chapter 9** then employs a detailed single-cell analysis of the high-quality data recorded in the previous chapter. Herein, we combine information at single-cell, population, and multi-population levels to gain insights into the sensitivity of cells to light-based stress induction using photosensitizers (PSs). We are the first to systematically quantify the influence of such PSs with single-cell resolution across various light energies and show that the utilized light energy is linked to the future development of each living cell, making PSs a promising candidate for closed-loop cell growth control.

Chapter 6

Digital Microfluidic Chip: enabling smart microscopy in versatile high-throughput microfluidic chip systems

Contents

6.1 Motivation 66
6.2 Designing the Swiss Army Knife - a versatile & high-throughput digital microfluidic chip design 68
6.3 Deep learning-based reference marker detection 68
6.4 RoI masking and automated error detection 74
6.5 Fast blueprint map alignment for high-throughput experiments . . . 77
6.6 Discussion & Limitations 80
6.7 Conclusion 81

All presented results for now are based on imaging data recorded in purely observational experiments. Thus, the single-cell analysis is performed after the experiment. However, to make smart decisions during the experiment based on real-time information that influence living cells or recognize failures of experiments, real-time data analysis is essential. Gaining single-cell information in real-time during the experiment has the potential to employ smart microscopy routines and move from purely observational experiments to experiments with influence and control. However, real-time image analysis in microfluidic live-cell imaging is challenging: the raw microscopy images contain versatile microfluidic structures that form the regions of interest for cultivating the cell populations. Distinguishing microfluidic structures from living cells may be difficult and requires a real-time semantic understanding of the microfluidic chip design.

In this chapter, we introduce the DMC that fuses the digital microfluidic blueprint with the recorded microscopy images, building a real-time semantic understanding. We perform this fusion by placing reference markers and unique IDs next to every RoI on the microfluidic chip design,

detect them with deep learning object detection, and align the microfluidic RoI blueprint with the microscopy image. This fusion leads to three core contributions: (1) We designed the Swiss Army Knife (SAK) microfluidic chip with eight RoI designs, resulting in a high-throughput chip with a total of 1164 RoI positions. (2) We use the DMC to efficiently build a map of all RoI positions on this microfluidic chip. (3) We apply the semantic knowledge of the RoIs to mask microfluidic structures in the microscopy images in real-time. This combination of hardware design and deep learning software in the DMC enables real-time processing and analysis of living cells within each RoI, laying the foundation for smart live-cell experimentation on the JediOS platform (Chapter 7).

The developments presented in this chapter are led by me, but were achieved with the extensive support of my master student, Lukas Scholtes, and my colleague, Matthias Pesch. The DMC research presented in this chapter has been performed under the supervision of Hanno Scharr, Dietrich Kohlheyer, and Katharina Nöh. It is about to be submitted as a manuscript with open-source software.

6.1 Motivation

Smart and event-driven microscopy experiments have immense potential in making decisions during the experiment, for example, adapting the imaging routine or applying influence and exerting control over living cells [40, 41, 37]. However, to make smart decisions during the running experiment, a robust real-time understanding of cell development is needed. Despite the advancements in microscopy and microfluidic technologies, conducting high-throughput live-cell imaging experiments and gaining real-time insights remains challenging in practice. High-throughput experiments often suffer from long imaging setup times: In traditional imaging software, every RoI position is manually visited by the experiment operator, its position is saved, and the process is repeated for every RoI in the imaging loop. Thus, the setup time increases linearly with the number of RoIs on the microfluidic chip. With hundreds to thousands of RoIs in modern microfluidic systems, this becomes a time-intensive procedure of several hours and discourages the exploitation of high-throughput at the cost of losing quantitative insights. Moreover, gaining insights into living cells in real-time requires instance segmentation of the image, that is, the pixel-precise identification of every cell instance within the recorded image. The emergence of powerful deep learning-based cell segmentation [30, 31, 32] provides promising computational efficiency and segmentation quality. However, the design of complex and patterned microfluidic structures makes distinguishing them from living cells quite challenging [163, 120, 164]. Thus, typically simple box-shaped RoI geometries are used, and surrounding structures are removed in a semi-automated process [37].

These challenges - extended setup times, manual steps within the image processing, and challenging image segmentation - not only discourage researchers from leveraging the full capacity of modern microfluidic chips but also impede the development of real-time, ”smart” microscopy systems capable of dynamic decision-making during the experiment as they rely on high-quality real-time measurements.

To address these challenges and bring real-time insights and high-throughput experiments into practical application, we developed the DMC platform, a fusion of microfluidic chip blueprint information with fast deep-learning-based image analysis (Figure 6.1). This platform leverages the structural knowledge embedded in digital microfluidic blueprints to locate RoIs efficiently and projects the

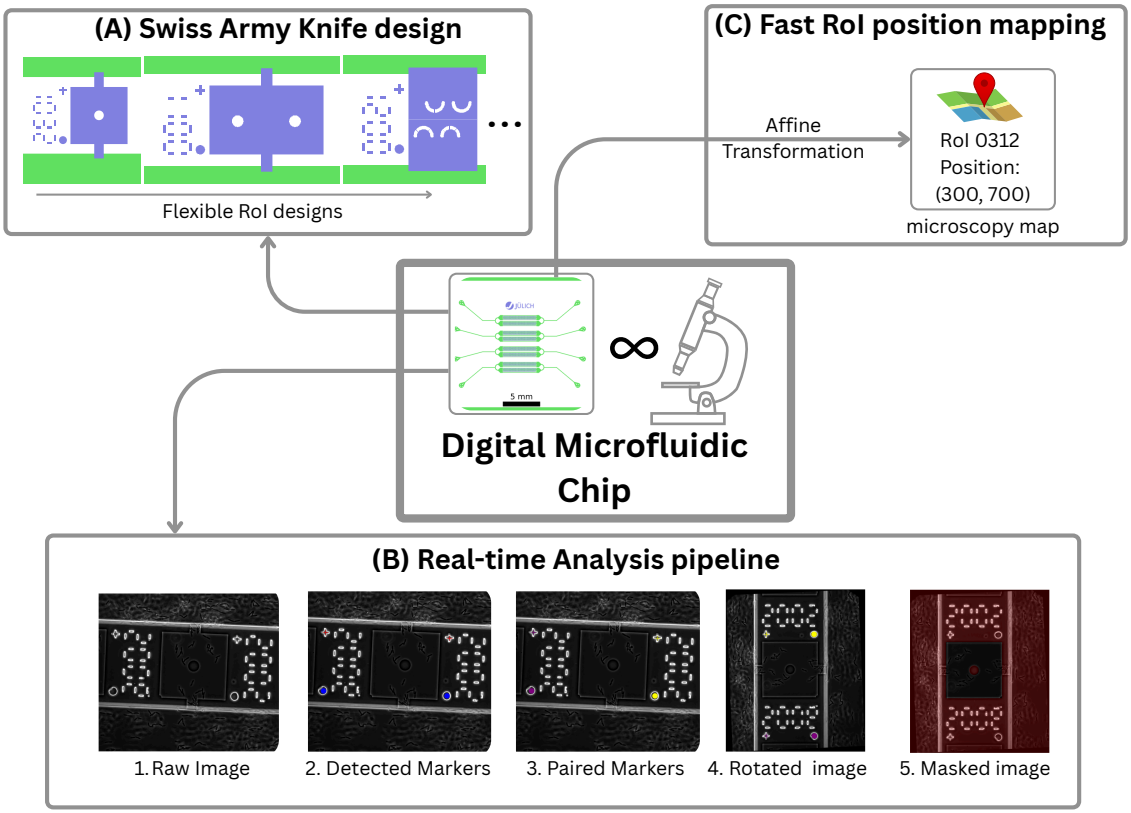

Figure 6.1: **The digital microfluidic chip.** The digital microfluidic chip fuses microfluidic design with deep learning image analysis to create a semantic understanding of the microscope images. The DMC unlocks flexible RoI design with unique IDs and reference markers (A), real-time RoI masking to distinguish microfluidic structures from living cells (B), and fast RoI position mapping by computing an affine transformation from the blueprint to the microscope's coordinate system (C).

semantic understanding of the blueprint onto the recorded images. Our contribution is three-fold:

- *Swiss Army Knife DMC* (Figure 6.1A): We develop a microfluidic chip design containing eight different RoI designs to cultivate bacterial organisms. We term this design the SAK as it provides a versatile test-bed for evaluating our DMC platform and allows us to investigate the growth of bacteria in various designs of structured microfluidic cultivation chambers.

- *Real-time RoI masking*: (Figure 6.1B): We use the knowledge of the RoI position and its layout in the microfluidic blueprint to create a mask of the microfluidic structures and align it to the recorded image using microfluidic reference markers. For automated marker detection, we train and evaluate fast deep learning-based object detection models (You Only Look Once (YOLO)). This method detects markers in sub-seconds with sub-micron precision, and the subsequent masking works with arbitrary RoI shapes that are directly extracted from the microfluidic blueprint design.

- *Fast RoI position mapping* (Figure 6.1C): We align the microfluidic blueprint coordinate sys-

tem with the microscope's stage coordinate system. We achieve this by computing an affine transformation based on visiting and recording three RoIs using manual microscopy positioning. Thus, we have a time-constant imaging setup for locating an arbitrarily large number of RoI positions.

We demonstrate and verify the capabilities of the Digital Microfluidic Chip platform by conducting high-throughput cultivations of *E. coli* in the SAK microfluidic chip. This experiment showcases the platform's ability to handle diverse biological samples, highlights its potential for widespread application in microbiology, and underlines the advantage of fast mapping and real-time analysis of versatile RoI structures.

6.2 Designing the Swiss Army Knife - a versatile & high-throughput digital microfluidic chip design

Microfluidic chip systems are designed in a CAD software before their manufacturing and their application in MLCI (Section 1.4). In order to turn a traditional microfluidic chip design into a DMC, we perform two essential steps: First, we place two reference markers, a cross and a circle, at specific distances next to every RoI of the microfluidic design (Figure 6.1A). Second, we add a unique ID number next to every RoI between the reference markers. Experimenters use these IDs to navigate on the microfluidic chip (e.g., during experiment setup) when operating at high magnifications and small fields of view. Carrying out these two changes for all the RoIs in the Microfluidic design is fully automated, but it requires following specific design rules during CAD design.

To demonstrate the flexibility in microfluidic design and automated masking with complex structures, we have designed an exemplary high-throughput microfluidic chip with more than a thousand RoIs. At the same time, we use eight different but common shapes motivated by popular designs and in microbial MLCI [165, 19]. Figure 6.2 shows the blueprint design of this high-throughput chip that we term the SAK. We adapted the high-throughput design of Grünberger et al. [19], distribute the RoIs across four different media channels that are physically separated, and arrange them into a grid (Figure 6.2A). Figure 6.2B shows the close-up of a medium channel. The eight different RoIs designs differ in shape and size and contain inner structures, making the SAK a perfect test-bed for evaluating the real-time masking of arbitrarily shaped RoI designs (Figure 6.2C).

After manufacturing this chip, we performed two MLCI experiments cultivating *E. coli* organisms and collected a large dataset of these organisms within the SAK.

6.3 Deep learning-based reference marker detection

For masking arbitrarily shaped RoIs within microscopy images, it is essential to transfer the knowledge of the microfluidic chip design to the recorded microscopy images. Therefore, we align both using reference markers that are integrated into the DMC design. Precise and rapid detection of these microfluidic reference markers is crucial for accurate RoI masking. Thus, we train deep learning-based detection and segmentation models and evaluate their capabilities of detecting the reference markers fast and precisely.

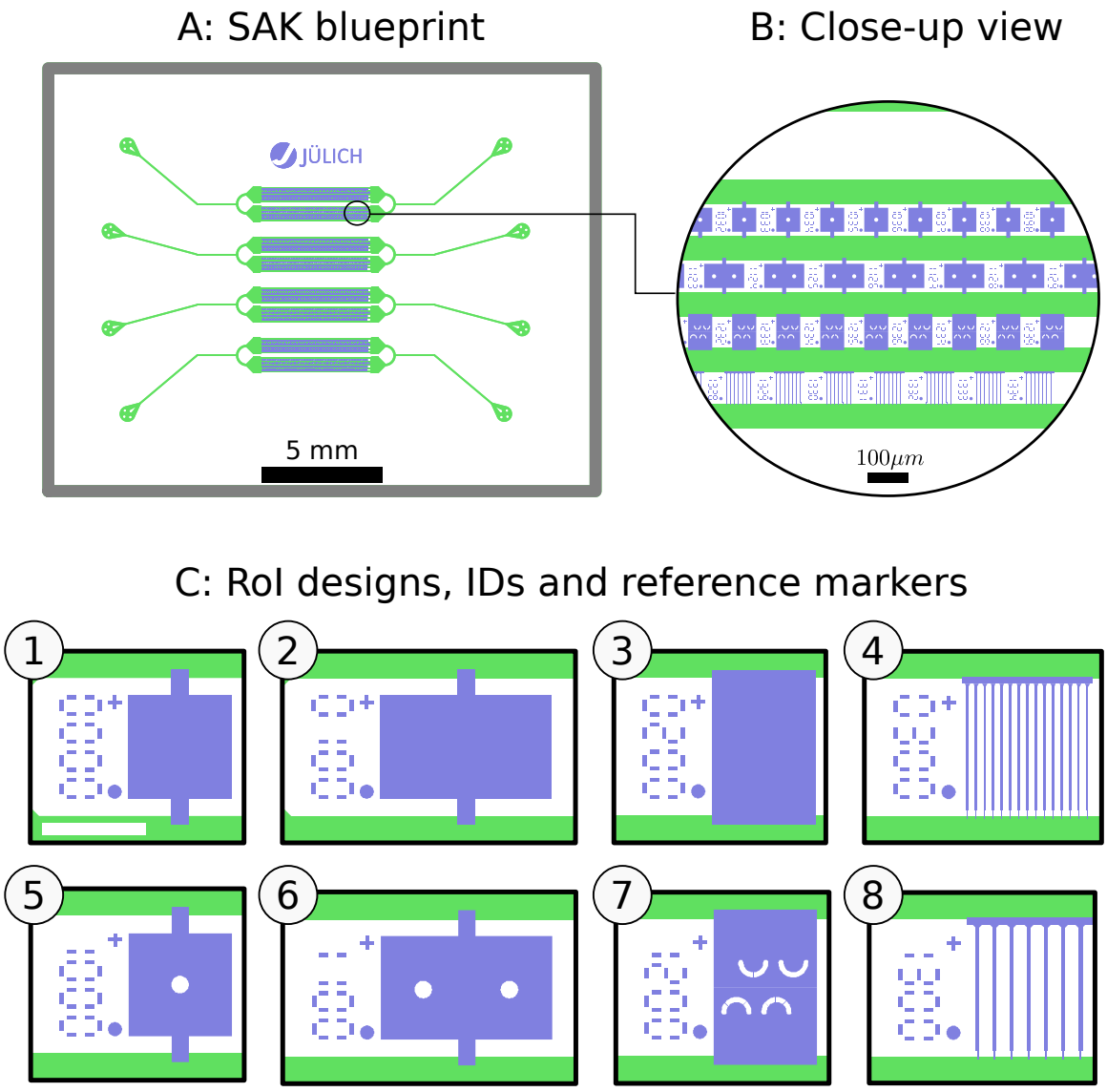

Figure 6.2: **CAD design of the Swiss Army Knife.** (A) Denotes the view of the SAK CAD design of the full microfluidic chip, including in- and out-lets. (B) shows a close-up of the grid layout within a single microfluidic channel. The individual RoIs are placed in rows (with the same RoI design) and columns (with different RoI designs), each assigned a unique ID and reference markers (+ and •) . (C) the detailed view of the eight different RoI designs. The white scale bar in (C1) denotes 60 μm and applies to C1-8.

6.3.1 Data recording & annotation

In order to perform this evaluation, we randomly selected a subset of the recorded images consisting of 220 images showing all eight chamber designs of the SAK. These images have a resolution of 2560×2160 pixels with a spatial pixel size of 0.066 $\mu m/pixel$. We performed instance segmentation annotation of reference markers in a semi-supervised manner: The images were imported into the LabelStudio annotation platform, where a Segment Anything 2 (SAM2) model was used to predict a pixel-precise segmentation mask of the reference markers, guided by interactive point annotations [166]. Only if the predicted segmentation masks were of insufficient quality they been manually corrected. Defect or largely invisible markers (e.g., at the edge of the image) and images with focus loss were not annotated.

In total, 358 cross markers and 356 circle markers were annotated in the 220 images, and the resulting dataset is split into training (110 images), validation (26 images), and test datasets (84 images).

6.3.2 Training reference marker detection models

Based on this annotated dataset, we train and evaluate deep learning-based object detection models. We selected the You Only Look Once object detector family for marker detection, as it is ideally suited for real-time applications and provides multiple parameters to choose a trade-off between detection quality and speed [167]. The primary four parameters of the YOLO model family are:

- the YOLO model version: the YOLO object detection family has been continuously developed and provided in several different versions. We utilize versions 5 (v5), 8 (v8), and 11 (11). Usually, higher version numbers are expected to provide a better quality of object detection. Versions 8 and 11 also contain models, for instance, segmentation.

- YOLO model variant: the YOLO models come in different variants, determining the model size and the overall amount of trainable parameters in the model. We utilize the smallest (n), medium (m), and largest (x) model variants.

- training task: the YOLO models have initially been developed for object detection but have recently also extended to instance segmentation. Thus, we use and train the models on two different training tasks. The first task is object detection (short: det), i.e., predicting bounding boxes around the objects of interest. The second task is instance segmentation (short: seg), i.e., predicting an additional instance segmentation mask for each detected object of interest.

- image resolution: the YOLO models use a fixed input image resolution for faster image processing. Images with different resolutions are resized such that the longer edge matches the model resolution. We utilized two fixed image resolutions, 640p and 1280p.

All these parameters influence the detection quality and speed of the models. To identify the most suitable YOLO model parameter combination and study the impact of different model configurations on the marker detection performance, we have trained YOLO models with all combinations of the four parameters using the ultralytics framework (`https://docs.ultralytics.com/`).

6.3.3 Evaluation of reference marker detection

We evaluated all trained models on the annotated test dataset. Figure 6.3 shows the bounding box detection results for all four parameter dimensions evaluated using the mean Average Precision (mAP) metric. The mAP measures the accuracy of predicted bounding boxes across different object classes - in our case, cross and circle markers. It is computed by calculating the Average Precision (AP) for each class on the precision-recall curve at various Intersection over Union (IoU) thresholds and then averaging the AP across all classes [168]. Thus, the higher the mAP and the closer to the maximum of 1, the better the object detection performance.

Using the mAP, we investigate the effect of model parameters on the quality of bounding box predictions. Interestingly, the YOLO models using the segmentation task and models (triangles) show a much higher mAP and are the only models that exceed the 90% mAP threshold (black dashed line). We attribute this to larger model size and a more informative segmentation loss on this rather small training dataset. The YOLO versions 8 and 11 usually perform very similarly, with version 11 usually slightly ahead. Increasing the resolution (4 times more pixels per image) leads to a slight improvement in mAP across all model variants. Increasing the model size by using a larger model variant also slightly increases the mAP. Especially models with an mAP higher than 90% show that these YOLO models are very well suited for robustly detecting the reference markers in the test dataset.

While high mAP values indicate that the models are capable of detecting the markers and predicting their bounding boxes, we are specifically interested in the reference marker center. This predicted marker center is later used to align the microscopy image with the digital blueprint (Figure 6.1B). Therefore, we measured the absolute errors on the predicted marker center computed from the bounding box. Figure 6.4A shows the 95-th quantile of the absolute marker center error in pixels and micrometers on the test set for the different YOLO model parameter configurations.

We observe the same trend as before: segmentation models, models with higher resolution, and larger model sizes lead to better marker detection, indicated by lower marker center errors. Interestingly, the best-performing models show 95th percentile prediction errors of just a few pixels, resulting in errors in the sub-micron range. These small errors make further improvement of these models challenging, as the marker center prediction error already approaches the precision limit of human annotators.

Apart from the error in predicting the marker center, the speed or throughput of these models is crucial for their real-time application. The speed is measured by the number of image frames that can be processed every second and denoted as Frames per second (FPS). Figure 6.4B shows the FPS evaluated on the test set for the YOLO models. We used a batch size of 1 during the inference to simulate a real-time acquisition and processing of the microscopy images.

We have identified four key insights: (1) the segmentation models are much slower than their detection counterparts due to the additional computation of the object masks, (2) larger model variants lead to slightly slower processing, (3) the higher resolution roughly halves the FPS, (4) and the v8 version usually shows slightly higher FPS than the 11 version.

In total, the slowest model processes more than 15 microscopy images per second, unlocking real-time marker detection. Finally, we select the YOLOv8, variant m, segmentation model in the 640p resolution as it provides a good combination of low marker center prediction error and fast processing with more than 40 FPS. The selected model is marked in Figure 6.4 with the red circle.

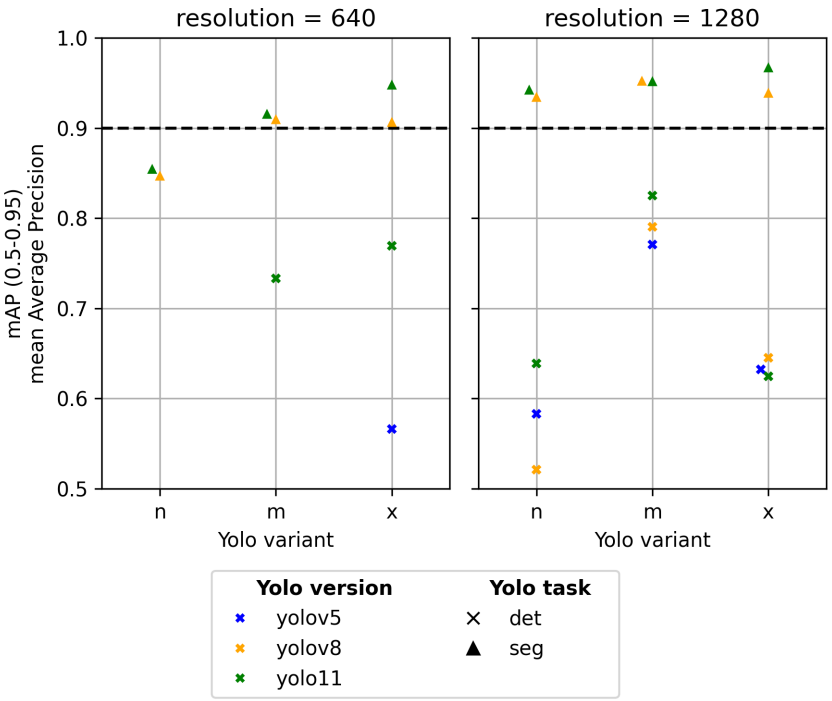

Figure 6.3: **Evaluation of the YOLO model configurations and marker detection quality.** Models with the four parameters (YOLO version, YOLO variant, YOLO task and resolution) are evaluated using the mAP metric. The YOLO version and task are denoted by color and marker style. The YOLO model variant is shown on the x-axis and the resolution in the left (640p) and right (1280p) plot. Each point represents a trained model with a specific parameter set evaluated on the test set. The dashed black lines indicate the 90% mAP. Models with an mAP below 0.5 are not shown.

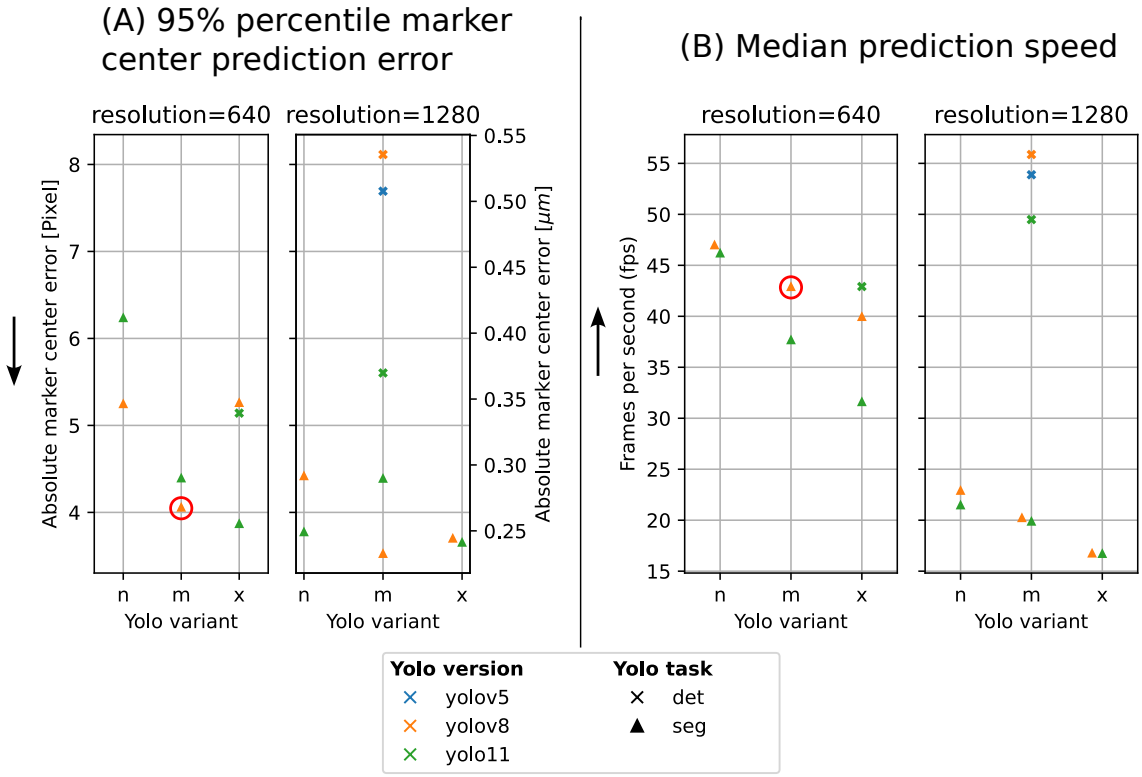

Figure 6.4: **Evaluation of the YOLO model configurations for detecting microfluidic reference markers.** (A) shows the 95% percentile absolute marker center error in pixels and μm for all YOLO model configurations evaluated on the test dataset. Lower values are better and indicate a better suitability for precise marker center prediction. (B) shows the prediction speed of the YOLO model configurations evaluated on the test dataset with a batch size of 1. Higher values are better and indicate a better suitability for real-time processing. Only models with an mAP higher than 0.75 are shown. The red circle denotes the selected YOLO model.

6.4 RoI masking and automated error detection

After selecting a suitable YOLO model for marker detections, we can utilize these marker detections to align DMC blueprint and microscopy information as shown in (Figure 6.1B). We perform this process in three steps: marker pairing, image rotation, and masking, which are described in more detail below.

6.4.1 Marker pairing and automated error detection

After the markers are detected by the YOLO model, we group them in corresponding pairs of a cross and a circle, as there should be two markers for every RoI. We perform the pairing based on the pairwise distance of the marker detections and only confirm pairs that have a marker distance specified in the microfluidic chip design (Figure 6.1B3). Typically, two or four markers are visible in the field of view of the microscopy image, resulting in a single or two marker pairs.

The blueprint information on the marker pair distance enables us to detect errors in marker detection automatically. If fewer than two markers are detected in the image, we flag this image as potentially at risk of a false negative (FN) prediction. However, during live-cell imaging, this can also be caused by deteriorated imaging conditions (e.g., focus loss) or an incorrect imaging position that does not contain the RoI or the reference markers. In both these scenarios, the knowledge of the DMC is used to automatically identify challenging images or errors in the analysis pipeline.

6.4.2 Rotation and masking

Based on the detected marker pairs, the image is first rotated so that the image axes align with the blueprint coordinate axes (Figure 6.1C4). The blueprint information of the RoI design is then used to compute a mask for the microfluidic structures and is aligned to the image using the reference marker positions (Figure 6.1C5).

6.4.3 Complete pipeline performance

Figure 6.5 shows the FPS measured across the full pipeline. Compared to pure marker detection, the full pipeline results in slower processing of 8 to 12 FPS across different YOLO versions, tasks, variants, and resolutions. Therefore, the difference in processing speed of the marker detection models becomes less noticeable. The selected model remains among the fastest evaluated on the complete pipeline and is suitable for real-time processing of microscopy image recordings.

Applying this selected model in the real-time pre-processing pipeline enables the computation of masks for the microfluidic structures across all RoI designs of the SAK chip. Figure 6.6 shows the resulting rotated and masked microscopy images (rectangular) and close-ups (circular) of the SAK microfluidic chip at one point in time of a live-cell experiment. The close-ups reveal that the mask precision is very high, especially in matching well with inner structures (5-7) or fine structures, such as those of the mother machines (4 and 8). Especially, the mother machines in 4 have a height of just $1\mu m$. Thus, their precise masking requires sub-micron accuracy in marker detection and masking.

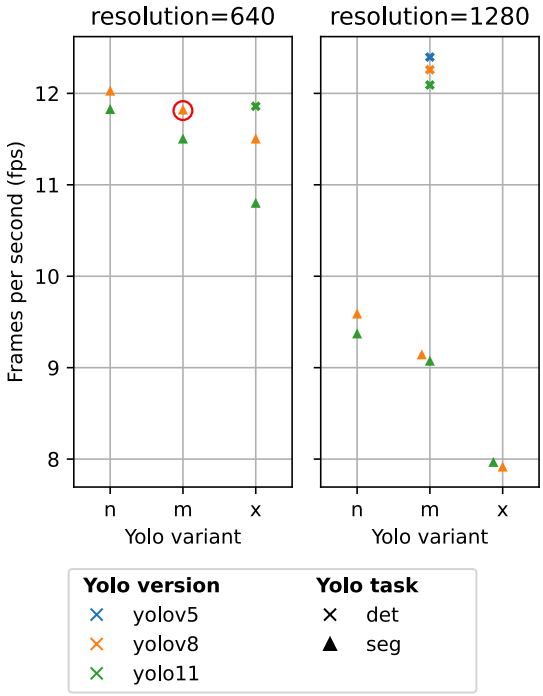

Figure 6.5: **Full RoI masking pipeline speed.** The average FPS measured on the test dataset across the full masking pipeline, including marker detection, pairing, image rotation, and mask alignment, for models across YOLO versions, tasks, variants, and resolutions. Models with an mAP below 75 % are omitted. The red-marked data point denotes the selected model.

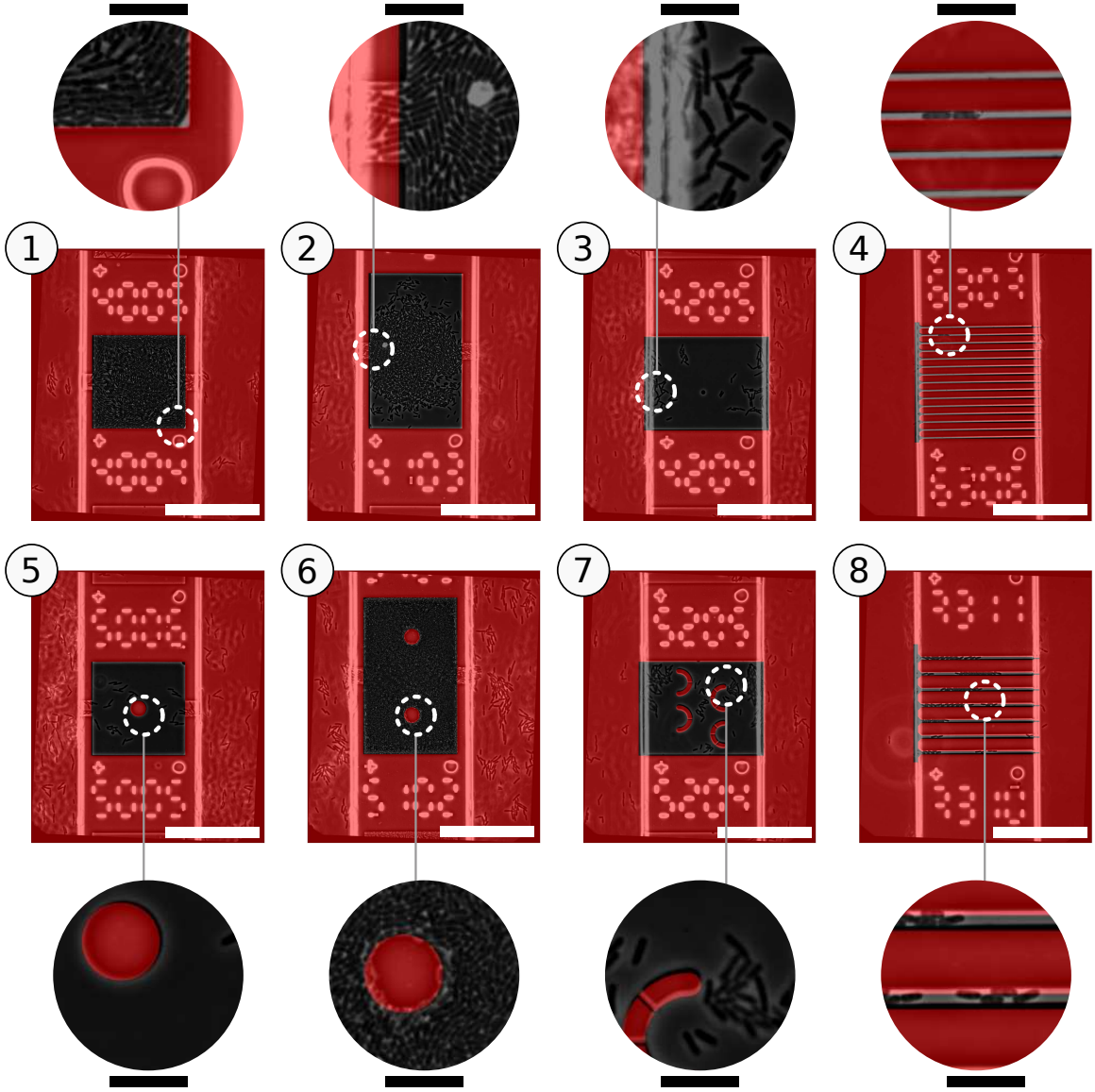

Figure 6.6: **Masked RoIs on the SAK.** 1-8 shows microscopy images of the eight different RoI designs used on the SAK. The overlay mask is visualized in red and aligned on the phase-contrast image by using the real-time analysis pipeline. For each RoI image, the white circle denotes the location of the 5-fold zoomed-in close-up view. The black bars denote the 10 micron scale bar for the close-ups, and the white bars denote the 60 micron scale bar for the masked images.

6.5 Fast blueprint map alignment for high-throughput experiments

While high-throughput experimentation allows for the recording of many replicates, it traditionally comes with long experiment setup times: Every RoI position is visited manually by the experiment operator, its position is saved, and the process is repeated for every RoI. This manual procedure often takes hours increasing the setup time proportionally with the number of RoIs on the microfluidic chip and discouraging the exploitation and systematic studies of hundreds of RoIs.

However, the microfluidic blueprint already contains a map with the precise positioning of all RoIs. For now, this information has not been used to determine the RoI positions on the microscope stage. Thus, using the DMC, we propose a setup routine that aligns the blueprint map with the microscopy stage coordinate system within a constant time of roughly 5 minutes. This map alignment yields all the RoI positions in the microscope's coordinate system.

For this setup procedure, we manually move to three RoIs, record their positions on the microscope stage, their IDs, and the raw microscopy images. We use these three RoI points to compute an affine transformation between the blueprint map and microscope coordinate systems (see 6.5.1). Then, we can transform any position between the coordinate systems, especially the position of all the RoIs, in order to compute all their positions on the microscopy stage.

6.5.1 Map alignment

To align the blueprint map with the microscope's coordinate system, it is necessary to establish a transformation T that maps coordinates from the blueprint system to the microscope stage coordinate system. This transformation is assumed to be affine, allowing for the incorporation of rotation, translation, shear, and scaling between the coordinate systems. The affine transformation $T : \mathbb{R}^3 \to \mathbb{R}^3$ of a point $\mathbf{p} = (x_1, y_1, 1) \in \mathbb{R}^3$ in homogeneous coordinates is then represented by a matrix-vector product:

$$T(x) = \mathbf{A} \cdot \mathbf{p} \tag{6.1}$$

$$= \underbrace{\begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A} \in \mathbb{R}^{3\times 3}} \cdot \underbrace{\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}}_{\mathbf{p} \in \mathbb{R}^3}, \tag{6.2}$$

where $\mathbf{A} \in \mathbb{R}^{3\times 3}$ is the transformation matrix that needs to be determined. Herein, a_{11}, a_{12}, a_{21}, and a_{22} define rotation, scaling and shear and t_1 and t_2 define the translation.

Given a set of three corresponding positions in the digital blueprint coordinate system denoted by $\mathbf{X}_{blueprint}$ and the microscopy stage coordinate system denoted by $\mathbf{X}_{stage}$

$$\mathbf{X}_{blueprint} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}, \mathbf{X}_{chip} = \begin{bmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \\ 1 & 1 & 1 \end{bmatrix}, \tag{6.3}$$

we denote the transformation between the corresponding points by the following equation system:

$$\begin{aligned} \mathbf{A} \cdot \mathbf{X}_{blueprint} &= \mathbf{X}_{chip} \\ \Leftrightarrow \mathbf{A} &= \mathbf{X}_{chip} \cdot \mathbf{X}^{-1}_{blueprint} \end{aligned} \tag{6.4}$$

that is used to determine the transformation matrix $\mathbf{A}$. The inverse of $\mathbf{X}_{blueprint}$ can be computed if the matrix has full rank, i.e., when the positions are linearly independent.

6.5.2 The mapping procedure

To collect these coordinate pairs, the microscope operator manually navigates to a RoI, enters its ID, records an image, and repeats the process for three distinct RoIs. To ensure linear independence, the selected RoI positions must not lie on a straight line. The marker center detection is then used to accurately calculate the center position of each RoI in the microscope's coordinate system. The affine transformation parameters are computed by solving the linear equation system using NumPy's `linalg` module. This transformation is subsequently applied to determine the positions of all RoIs within the microscope stage coordinates. The entire procedure of manually navigating to the RoIs using the joystick and recording three images takes approximately five minutes to complete.

6.5.3 RoI position mapping error

We evaluate this procedure by visiting three corners of a DMC design, recording their coordinates, IDs, and images, and computing the coordinate system transformations. Figure 6.7 shows (A) the map in the blueprint coordinate system that is extracted from the DMC's CAD design and (B) the transformed map in microscope stage coordinates. The three corresponding RoIs are marked with an orange triangle, green diamond, and red "x" in both maps. We observe that there is primarily an almost 90 degree rotation between the two coordinate systems. However, we also observe that it is not precisely 90 degrees. This is caused by the manual placement of the microfluidic chip, which does not fully align with the microscopy stage's coordinate system.

To evaluate the precision of the RoI positions in the resulting map in microscopy stage coordinates, we establish a validation procedure: We move to all the RoI positions of the map using the microscope, take an image, and compare the expected marker positions (based on the map information) with the actual detected marker positions using the YOLO model. We define the difference between the expected and detected marker positions as the absolute RoI position error. This error is composed of three major factors: the affine transformation, the microscope's stage movement, and the marker detection.

Figure 6.8 shows a map of all the RoIs (A) and the color-coded positioning errors as well as a distribution of these errors (B). We observe that all RoI positions are computed with an error of under 20 micron. Moreover, the median error is below 10 micron, and the 90-th percentile error of 16.04 μm underlines the high precision of the map estimation procedure based on visiting only three RoI positions.

The spatial mapping error (Figure 6.8A) reveals that the positioning errors do not accumulate during the microscopy stage movements. An accumulation could be caused by movement errors of the microscopy stage that visits the RoIs one by one, leading to a gradual increase of the error. Moreover, the marker detection error has shown to be in the sub-micron range. Thus, the measured error should be primarily determined by inaccuracies in the computed map positions. Moreover, certain RoI positions are missing in the map due to a failure of the marker detection such that no RoI position error could be computed. This is most likely caused by errors during the molding of the PDMS chip.

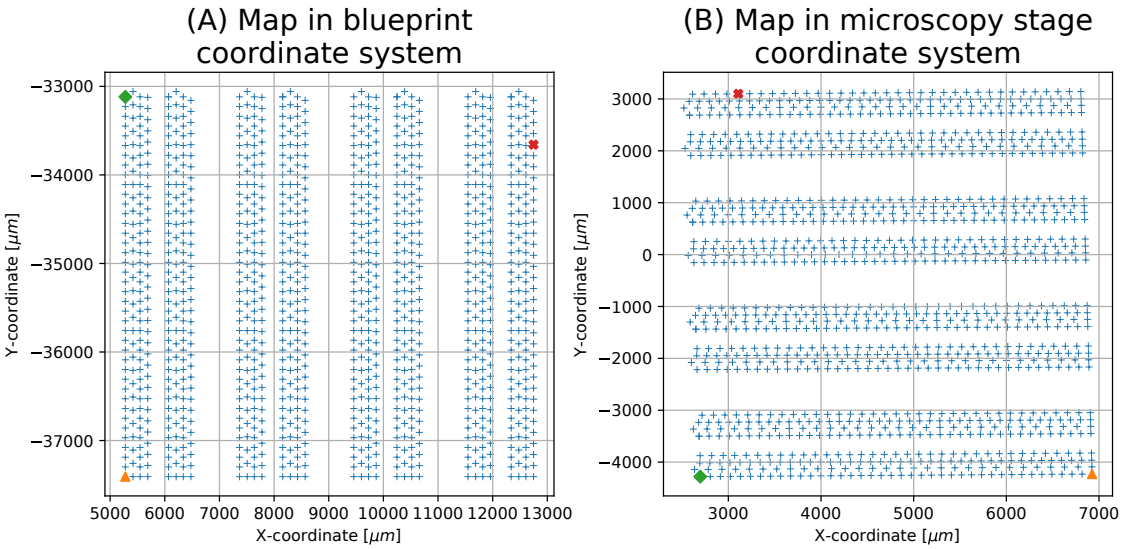

Figure 6.7: **Map computation procedure.** The microfluidic blueprint map (A) is extracted from the DMC's CAD design. Every cross denotes a RoI position. In the map creation procedure, three RoI positions are manually approached using the microscopy's joystick. These three RoI positions are marked with an orange triangle, green diamond, and red "x" in both the blueprint map (A) and the microscopy map (B). Based on these corresponding RoI positions, the transformation between both coordinate systems is computed to determine the positions of all other RoIs in the microscopy stage coordinate system (B) denoted by the blue crosses.

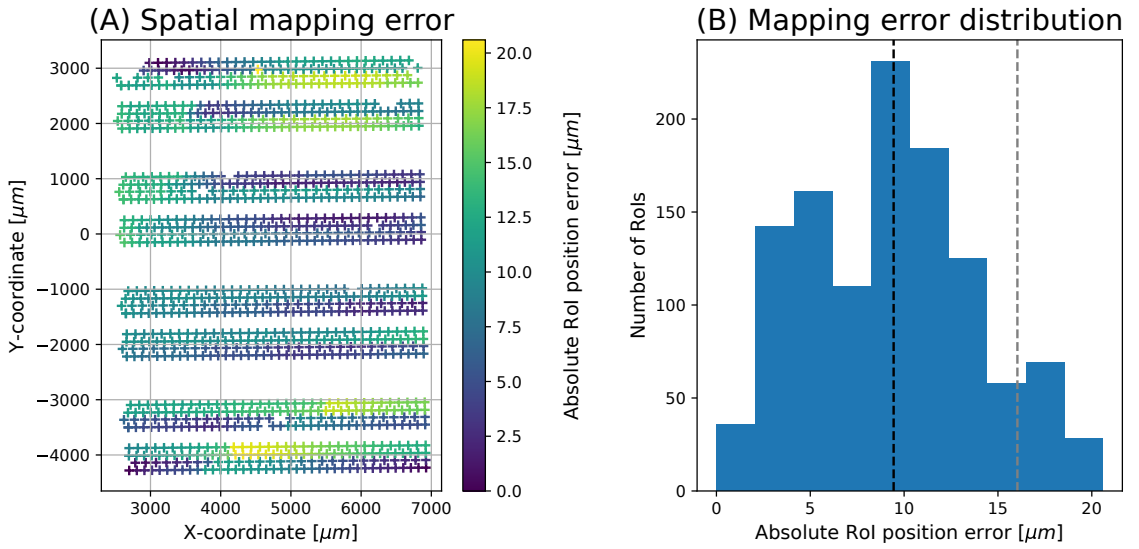

Figure 6.8: **Map RoI position errors.** (A) shows the spatial map of the microfluidic chip where each cross denotes a RoI position. The color denotes the absolute RoI position error, which is the difference between the expected and detected marker position in the image for each RoI. Missing RoI positions are caused by a failure of the marker detection. (B) shows the distribution of the absolute RoI position error. The median (9.44 μm) and 90th percentile (16.04 μm) are shown as black and gray dashed lines.

6.6 Discussion & Limitations

Our implementation of the SAK chip demonstrates the versatility of the DMC concept, particularly in high-throughput monolayer cultivation chips. In combination with the pre-processing pipeline, the DMC allows for the precise masking of arbitrarily shaped RoIs. This is inevitable for real-time cell information extraction used in smart microscopy routines, and it also speeds up the RoI cropping that is necessary before any image analysis. Moreover, the fast mapping procedure finally unlocks the high-throughput of microfluidic chips and provides precise RoI positions across the full microfluidic chip.

For now, the presented DMC system is constrained to RoI designs that fit within the camera's field of view. To address larger cultivation structures, markers can facilitate initial positioning correction, enabling the microscope to capture multiple images that are subsequently stitched together. When the microscope's positioning system achieves sufficient precision in the local neighborhood, the masking algorithm remains applicable to the composite image.

Another challenge encountered was accurately masking small structures, such as mother machines, with structural heights as narrow as one micrometer. The difficulties in precisely masking these small structures arise from limitations in both chip fabrication and limits in marker detection and annotation precision. In our experimental setup, a width of 1 micron corresponds to roughly 15 pixels. Consequently, even a few pixel deviations in the annotation, which then propagate to the prediction, lead to notable errors in the resulting masking. A potential solution to these inaccuracies involves placing more than two markers around each RoI. With these, an overdetermined equation

system can be solved using a least squares approach, is more robust against detection inaccuracies and defect markers, and has the potential to mitigate the effect of the annotation limit.

While the DMC concept currently focuses on 2D applications, it can be extended to 3D cultivation scenarios, such as biofilm cultivation [119] or organ-on-chip devices [169]. However, herein, the focus plane of the reference markers on the microfluidic chip is important. Thus, a more elaborate imaging routine is required first to identify this focus plane and then perform the map estimation or imaging.

6.7 Conclusion

In this chapter, we introduced the Digital Microfluidic Chip, an intelligent fusion of high-throughput microfluidic device design and real-time image analysis. The DMC offers flexible designs of RoI designs, including inner microfluidic structures, an efficient and consistent mapping procedure that scales to thousands of RoIs, and a fully automated, real-time image processing pipeline. This pipeline effectively masks microfluidic structures with sub-micron accuracy, providing a semantic understanding of the recorded microscopy image.

Therefore, the DMC represents a key step toward the seamless integration of microfluidic chip design with deep learning-based image processing, addressing practical challenges in high-throughput live-cell experiments. Access to real-time information also opens new possibilities for smart microscopy applications. Through our open-source software, we aim to equip the microfluidic live-cell imaging community with DMCs and facilitate continued and open developments.

Chapter 7

JediOS: just event-driven imaging operating system

Contents

7.1 Motivation **84**

7.2 High-throughput live-cell imaging - a multi-tasking perspective . . . **85**

7.3 Event-driven imaging language (EDIL) **87**

7.4 Scheduling state-machine executions in high-throughput experiments **92**

7.5 Simulating high-throughput experiments on the JediOS platform . . **96**

7.6 Simulation studies **99**

7.7 Discussion and Conclusion **106**

Using the DMC unlocks real-time information about the cell population development in every RoI on the microfluidic chip. This provides us with the opportunity to make decisions during the experiment based on the biological development of the observed cell populations, which is typically implemented in event-driven live-cell imaging frameworks. However, the high-throughput nature of MLCI requires making decisions for every developing cell population, drastically increasing the complexity of an event-driven experiment.

In this chapter, we present the JediOS - a platform that introduces a new operating system perspective for high-throughput live-cell imaging. Therefore, we first introduce this new perspective on high-throughput MLCI and highlight its commonalities with a computer operating system. We present the EDIL as a programming language for event-based imaging to create so-called imaging applications for each RoI on the microfluidic chip following a divide-and-conquer strategy. JediOS then deals with the execution of all these imaging applications on the imaging platform by using a scheduler. We integrate these developments into a comprehensive simulation platform that enables the simulation of complete live-cell experiments in-silico. In several simulation studies, we demonstrate that the JediOS platform provides crucial insights for experiment designs, achieves significantly higher throughput, and offers the opportunity to precisely define the experiment routine for every RoI independently.

7.1 Motivation

Smart microscopy experiments make use of real-time information to make decisions during the imaging experiment to adapt the imaging routine [40, 41, 42], trigger living cells based on events [41] or influence and control the living cells [37, 39]. High-throughput MLCI experiments are especially promising in this field as they allow the interaction and influence of many independent cell populations and, thus, collect large datasets giving insights into different interaction or control strategies, especially suitable for machine learning-based analysis [37]. The downside of such high-throughput experiments is the increase in complexity of managing the imaging routine, real-time information processing, and possible influencing actions for all cell populations concurrently. Existing platforms, such as Cyberscopy [41] and MicroMator [42], extract real-time information and process it as events to alter the live-cell experiment. However, they are not specifically designed to handle high-throughput experiments where decisions are made for hundreds of cell populations. Thus, implementing such experiment logic on these platforms results in a very complex and difficult-to-follow software implementation. The lack of an extensive in-silico simulation framework and the complexity in designing high-throughput event-driven experiments discourages their usage in everyday MLCI experiments.

While event-driven imaging experiments are inherently more complex than their static counterparts due to the asynchronous information processing and triggering of events, this chapter introduces a new divide-and-conquer perspective to master the complexity and make such experiments accessible to a broad user base. Instead of developing a single monolithic software that handles the imaging of all concurrently observed cell populations, we propose a divide-and-conquer approach and first develop an event-driven imaging routine for each cell population. We refer to the execution of such an imaging routine as an imaging application. These imaging applications deal with events and the recording of images for a single cell population, drastically reducing the complexity compared to a high-throughput experiment. We introduce the concept of an imaging operating system that concurrently executes all these imaging applications, thereby combining them into a high-throughput experiment. Therefore, users and developers do not need to consider the details of the high-throughput experiment specifically but can focus on developing imaging applications that observe, influence, or control cell populations. We combine this operating system with a comprehensive in-silico simulation platform that enables verification of the programming of individual imaging applications, particularly their response to real-time readouts, as well as their execution and combination in high-throughput experiments.

While this change in perspective toward divide-and-conquer design and operating system execution may seem like a minor detail at first glance, the implications and simplifications in designing and testing event-driven live-cell experiments arising from this view are groundbreaking. In this chapter, we demonstrate that this perspective on high-throughput experiments provides a detailed understanding of the interdependence between microscope utilization, the number of concurrently investigated RoIs, and imaging rates.

In the following, we introduce and investigate these new perspectives in detail and develop the JediOS platform outlined in Figure 5.3. In detail:

1. we analyze the requirements of such an imaging software and identify strong parallels with multi-tasking operating systems and this perspective on high-throughput imaging

2. based on this finding, we develop the components of the Just event-driven imaging operating system:
    - the imaging application that contains a state-machine and is associated to a RoI by its ID. The state-machine is formulated in the EDIL to handle the event-driven imaging routine for an individual RoI.
    - the scheduler that orchestrates the asynchronous execution of multiple imaging applications on the imaging platform.
    - a simulation framework that enables the in-silico simulation of entire high-throughput imaging experiments, including biological models, to simulate event-based interaction.
3. finally, we demonstrate the capabilities of the JediOS platform in simulation studies of high-throughput experiments, showcasing the benefits of flexible experiment design while also increasing data quality and throughput.

This chapter focuses on the software development of JediOS and its verification in in-silico simulations. The next chapter integrates a real-time image analysis pipeline and demonstrates the advantages of JediOS at the example of a real MLCI experiment.

7.2 High-throughput live-cell imaging - a multi-tasking perspective

In smart High-throughput Live-Cell Imaging (HTLCI), we would like to design individual imaging routines for each of the 10s to 100s of RoIs containing independently developing cell populations. When we think of the challenges to implement and execute such an experiment on the imaging platform, we often think of the physical movement of the microscope and the challenge to organize this movement to record all RoIs with a sufficient imaging rate. We refer to this view as the "monolithic view," as it involves defining a single imaging routine that orchestrates the image recording of all RoIs. In the following, we will introduce a divide-and-conquer view on this challenge, where we first define the imaging routine in the form of an imaging application for a single cell population within a single RoI. In a second step, we assemble this into a complete high-throughput scenario with many RoIs. To better understand this divide-and-conquer approach, we draw an analogy between multi-tasking systems of modern computers and high-throughput imaging systems.

On modern computers, users run multiple applications concurrently, such as an email client, a video call, or a text processing application. However, the user does not need to worry about the concurrent execution of these programs and can focus primarily on working with these applications. Neither the developers of these applications deal with the details of the concurrent execution. They primarily focus on implementing the application logic in a programming language of their choice (e.g., Python, Java, C++, ...). The execution of this application logic on the hardware of the computer system is then orchestrated by the operating system (Windows, Linux, ...) dealing with the details such as hardware drivers or allocating time on the Central Processing Unit (CPU) using a scheduler (Figure 7.1A). The benefits of this approach are apparent: neither the application developer nor the user needs to worry about the details of execution but can spend their time effectively developing or using these applications. Moreover, the implementation of the software logic and its execution on the

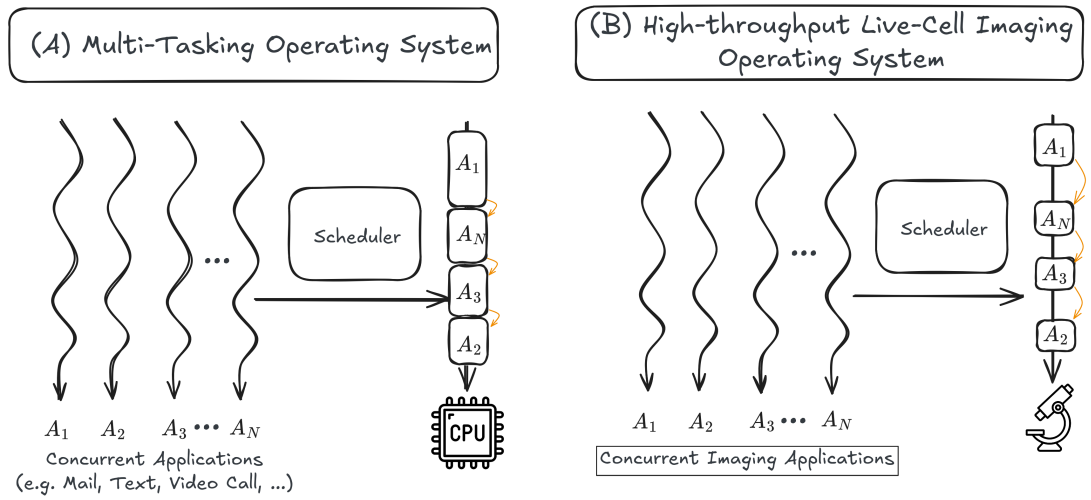

Figure 7.1: **The high-throughput imaging system - a multi-tasking perspective. (A)** Modern multi-tasking operating systems allow to run multiple applications concurrently - all executing their independent logic denoted by the threads A_1 to A_N. A scheduler performs sequential execution, allocating compute time on the Central Processing Unit and switches between applications to distribute the compute time. These switches (context switches), denoted by an orange arrow, happen in milliseconds so that the end user usually does not notice them and has the illusion of concurrent execution. **(B)** shows the envisioned high-throughput live-cell imaging operating system. A set of imaging applications (A_1 to A_N) requests imaging actions and processing events based on real-time image readouts or timeouts. A scheduler is used to orchestrate the execution of the imaging requests on the imaging platform in sequential order. It switches between RoI positions to distribute the imaging time accordingly. A switch denoted by the orange arrows involves the physical movement of the microscopy stage and a refocus procedure that can take up to a second.

underlying hardware are disentangled. Thus, executing the same applications on a different computer with different hardware is possible. Furthermore, improvements in the underlying operating system can accelerate the execution of the applications without breaking their logic. We call such an operating system as described above a Multi-Tasking Operating System (MTOS) as it allows the execution of multiple tasks or applications concurrently.

This basic understanding of an MTOS allows us to transfer its perspective on HTLCI experimentation and draw the analogy depicted in Figure 7.1B. In the MTOS, many different applications run concurrently and request compute resources that are limited, for example, because only a single CPU core is available. Similarly, in HTLCI many imaging applications (one for each RoI and cell population) run concurrently requesting imaging resources, for example, to record the next phase-contrast image of the time-lapse. However, the imaging resource is limited, usually to a single microscope. For the MTOS, the operating system provides a scheduler that rapidly switches between the execution of different applications, ensuring that they all receive appropriate computation time and creating the illusion for the user of concurrent execution. We envision a similar operating system for HTLCI, where the operating system's scheduler switches between serving the imaging requests of the imaging applications, ensures that all record images are captured at the requested time points,

and creates the illusion that all RoIs are observed concurrently. The main difference here is that the switches in HTLCI require a physical movement from one RoI position to another, thus taking place in seconds rather than in microseconds in an MTOS system.

Based on this analogy, we can also transfer the benefits of an MTOS to HTLCI. First, the imaging application developer can primarily focus on programming the desired imaging routines without caring too much about their specific execution in a high-throughput scenario. Moreover, transferring the experiment to another microscopy setup becomes easier as the operating system handles access to the underlying hardware using modular driver implementations.

This operating system perspective on HTLCI also highlights a challenge that is often neglected but inherent to high-throughput imaging: The interdependence between throughput and microscopy imaging capacity. The microscope platform comes with a limit on the overall imaging rate. Thus, if we would like to observe many RoIs concurrently with a high imaging rate (low imaging interval), we bear the risk of "over-utilizing" the imaging capacity of the microscope. This leads to a "stuttering" of the imaging applications very similar to the "stuttering" observed when executing too many concurrent applications on a computer. A view on the "task manager" then often reveals that the system is running beyond its compute resources, slowing down all applications. In HTLCI, this "stuttering" is reflected in lower imaging rates (higher imaging intervals) than intended by the imaging application. This over-utilization phenomenon can occur especially in event-driven imaging where we might alter imaging rates dynamically, for example, based on biological events. Thus, an operating system for HTLCI requires a "task manager" that displays the utilization of the imaging platform and alerts developers to potential resource limitations.

Another considerable advantage of thinking in terms of imaging applications and their formalization in a programming language is their in-silico testability. Modern software testing routines such as unit tests and CI bear the potential to verify the correct execution, especially with regard to real-time events. Fusing imaging applications with cell development models would provide us with the ideal test-bed to verify their logic, reaction to biological events, and even simulate complete high-throughput experiments in-silico to rule out any potential design flaws or over-utilization effects.

In the following, we develop the three key ingredients of such a high-throughput imaging OS that we name the Just event-driven imaging operating system or short JediOS outlined in Figure 5.3:

1. We define the EDIL, a simple programming language that allows the design of complex imaging applications and reactions to real-time events.

2. We introduce a scheduler that orchestrates the concurrent execution of multiple imaging applications.

3. We combine both developments into a simulation platform where models simulate the development of cell populations. We showcase simulations of individual imaging applications, as well as complete HTLCI experiments.

7.3 Event-driven imaging language (EDIL)

Computer applications are typically formalized in a programming language, providing users with a significant amount of flexibility in expressing the logic of their application. This logic encompasses

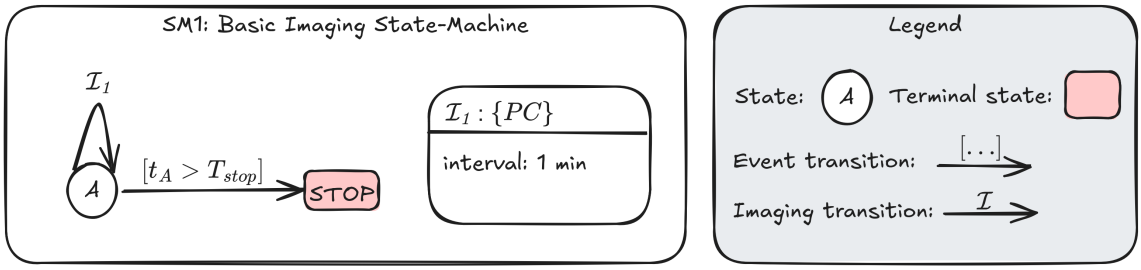

Figure 7.2: **Basic imaging state-machine (SM1) and legend.** The state-machine (left) consists of two states A and $STOP$ and two transitions. The first transition from state A to state A is an imaging transition and executes the imaging action $\mathcal{I}_1$ at an interval of 1 minute. The second transition is a conditioned event transition and is executed as soon as the state-machine spent more than T_{stop} time in state A. The short notation of this condition is $[t_A > T_{stop}]$. The execution of the state-machine starts in the initial state A. The legend explains the components of EDIL state-machine (right).

mathematical computations, as well as interfacing with inputs, stored data, network resources, and user events, such as clicks. In comparison, imaging applications are relatively simple: Their primary goal is to execute an imaging routine that records images of the developing cell population using a specific set of imaging actions (e.g., phase-contrast, fluorescence) and an imaging interval. Based on events that contain real-time information, this imaging routine can be adapted.

To keep the implementation of such imaging applications simple and easy to visualize, we formalize their logic using finite state-machines (called state-machines in the following). We call this formalization the EDIL inspired by the definition of regular languages using automata [170].

State-machines are frequently used to model the behavior of systems with a discrete set of states, such as a traffic light. The state-machine consists of a set of discrete states and transitions. The state-machine is in one state at a time, called the ”active state”, and inputs trigger the transitions from one state to another. Thus, a state-machine with an initial state and a list of inputs allows the execution of the logic encoded in the state-machine design.

For a better understanding of how this state-machine formalization helps designing event-based imaging applications, we introduce the basic concepts and design philosophy behind EDIL through examples of increasing complexity. These examples demonstrate the imaging of a single cell population and reacting to its development in theory. The execution and scheduling of a single and multiple state-machines within an MLCI experiment is introduced in Section 7.4.

7.3.1 RoI state-machine for a basic imaging experiment (SM1)

The basic MLCI experiment images a set of RoIs with a constant imaging interval and terminates after a given period of time. Figure 7.2 shows the state-machine design for such a RoI imaging application that we call SM1. To better understand the design, we will walk through its structure and execution step by step.

The state-machine consists of two states A and $STOP$ representing the running observation (state A) and its termination (state $STOP$). The execution starts in the initial state A, marked by the incoming arrow. The other two arrows denote transitions between the states. In the EDIL, we

use two types of transitions: imaging transitions and event transitions.

The first transition $A \xrightarrow{\mathcal{I}_1} A$ is an imaging transition. Each imaging transition specifies an interval, i.e., the time between two executions of the transition. As soon as the interval elapses, we perform the transition. Attached to the transition is the imaging action $\mathcal{I}_1$. In our example, performing the transition results in recording a phase-contrast image.

The second transition $A \xrightarrow{[t_A > T_{stop}]} STOP$ is a conditioned event transition. Real-time data updates, such as the elapsing of timers or real-time biological information, trigger event transitions. When a transition is conditioned, it can only be executed if all conditions are satisfied. The duration that the state-machine spent in state A is shortly denoted by t_A and the brackets are a short notation for a condition that needs to be satisfied in order to perform the transition. Therefore, as soon as cell population development has been observed for a more extended period of time than T_{stop}, SM1 executes the second transition and moves to state $STOP$. State $STOP$ is a final state (colored red) and, therefore, ends the execution of the state-machine. In the following, final states are only indicated by the red-colored rectangle.

The execution of a state-machine results in a sequence of states and transitions that we call the execution sequence. The states indicate the state of the current imaging application and the transitions denote actions such as image recordings or state changes. The execution of SM1 results in the following execution sequence:

$$\underbrace{A \xrightarrow{\mathcal{I}_1} A \xrightarrow{\mathcal{I}_1} A \xrightarrow{\mathcal{I}_1} \ldots \xrightarrow{\mathcal{I}_1} A}_{\text{Observation period of duration } T_{stop}} \xrightarrow{t_A > T_{stop}} \underbrace{STOP}_{\text{Final state}} \tag{7.1}$$

The execution sequence shows that we achieve the desired time-lapse recording for the basic imaging experiment of a single RoI: For a specified observation period, the imaging action $\mathcal{I}_1$ is executed and records phase-contrast images of a single RoI with a regular imaging interval. As soon as the observation period ends, the condition $[t_A > T_{stop}]$ becomes true, and the state-machine moves to the final state. Thus, investigating the execution sequence allows us to verify the correct imaging logic.

7.3.2 Adaptive imaging interval (SM2)

We have already shown the need for low imaging intervals are necessary to track cells in MLCI experiments (Chapter 3). However, recording each RoI with a low imaging interval throughout the complete experiment massively constrains the throughput, i.e., the number of concurrently monitored RoIs. For example, an imaging interval of 1 minute and an image acquisition time of 2 seconds per image allows for recording 30 RoIs concurrently. This is roughly 5% of the RoI positions available within a modern high-throughput microfluidic chip design with up to 1280 RoIs. Interestingly, the benchmark in Chapter 3 also revealed that the suitable imaging interval for good tracking depends on the number of cells within the population. For tracking a small population of cells (e.g., less than 100), an imaging interval of 20 minutes was sufficient. Such a long imaging interval allows us to concurrently record up to 600 RoIs utilizing roughly 50% of the microfluidic chip's capacity. On the other hand, tracking larger cell populations required shorter imaging intervals. Thus, we aim to design an imaging application that adapts the imaging interval according to the number of cells within the cell population, thereby providing effective cell tracking capabilities while maintaining high throughput.

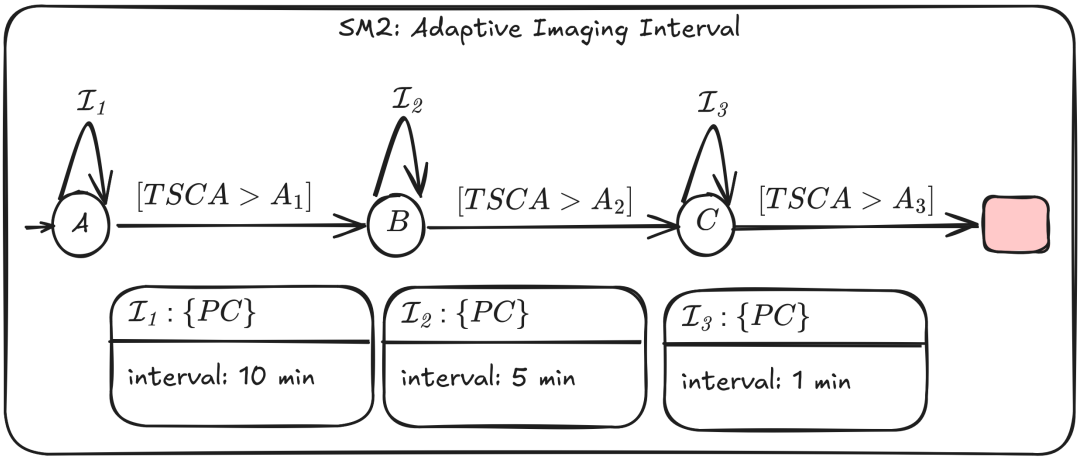

Figure 7.3: **Adaptive imaging interval state-machine (SM2).** The state-machine consists of four states A, B, C, and $STOP$. Three image routines $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3$ with decreasing imaging intervals of $10, 5$ and 1 minute are used to record the development of the cells in states A to D. Three ascending TSCA thresholds $0 < A_1 < A_2 < A_3$ are used in event transitions to sequentially move through the states as soon as the cell population exceeds a specific size. Thus, SM2 gradually shortens the imaging interval in response to growing cell populations.

Figure 7.3 shows the state-machine design of such an imaging application that adapts the imaging interval depending on the size of the cell population. We call this state-machine SM2. The state-machine consists of four states A, B, C, and $STOP$. We introduce three image actions $\mathcal{I}_1, \mathcal{I}_2$, and $\mathcal{I}_3$ with decreasing imaging intervals of $10, 5$, and 1 minute. We introduce three ascending TSCA thresholds $0 < A_1 < A_2 < A_3$ that are used as conditions in event transition to consecutively move to the next state as soon as the cell population exceeds a certain size. Thus, SM2 gradually shortens the imaging interval in response to the growing cell population.

The SM2 starts its execution in the initial state A and records images at an interval of 10 minutes. As soon as the cell population size (TSCA) exceeds the first threshold A_1, the state-machine transits to state B and starts recording images at an interval of 5 minutes. Similarly, the machine moves to state C as soon as the population size exceeds the threshold A_2 and applies the image routine $\mathcal{I}_3$, recording images at an interval of 1 minute. Upon being in state C, the state-machine transits into the $STOP$ state as soon as cell population size exceeds the threshold A_3.

The execution of state-machine SM2 results in the following execution sequence:

$$\underbrace{A \xrightarrow{\mathcal{I}_1} \ldots \xrightarrow{\mathcal{I}_1} A}_{\textit{interval: 10 min}} \xrightarrow{TSCA>A_1} \underbrace{B \xrightarrow{\mathcal{I}_2} \ldots \xrightarrow{\mathcal{I}_2} B}_{\text{interval: 5 min}} \xrightarrow{TSCA>A_2} \underbrace{C \xrightarrow{\mathcal{I}_3} \ldots \xrightarrow{\mathcal{I}_3} C}_{\text{interval: 1 min}} \xrightarrow{TSCA>A_3} STOP. \tag{7.2}$$

The event transitions between states are conditioned on the development of the cell population. Thus, the total length of the execution sequence is also determined by the biological development of the cell population. In this example, we used three states A to C with three imaging intervals and thresholds. However, the imaging application can be easily extended to a finer gradual adaption of the imaging interval by introducing more states, image routines, and thresholds.

7.3.3 Optogenetic excitation pulse screening (SM3)

The previous state-machine examples have focused on event-based recording and observation of the development of cell populations. However, influencing the development of living cells promises to unlock many opportunities to better unterstand their spatio-temporal development. Thus, the next imaging application is designed to influence cells at a certain point of time and observe their development response at high imaging rates in the immediate aftermath of the influence. As an exemplary influence we select an optogenetic tool that allows to induce stress into living cells using light-based excitations. Using excitation pulse (EP)s, i.e., a short exposure to a specific wavelength of light, the light influence on cell populations is precisely defined and local to each RoI. To measure and understand the optogenetic influence of such excitation pulses on the single-cell level, we design an imaging application with three primary objectives: First, we trigger excitation pulses with parameterizable light intensity and exposure duration that illuminates the complete RoI . Second, we perform the excitation pulse when the cell population reaches a specific size to rule out population-size-dependent behavior. Third, we measure the immediate response on the single-cell level by using a specifically low imaging interval directly after the EP. Notably, the imaging action $\mathcal{I}_2$ performing the EP does not specify an imaging interval as it is only executed once during the transition to state B_1. We denote this type of imaging action as a "one-shot."

Figure 7.4 illustrates the state-machine design SM3 which achieves all three objectives. At the beginning of the experiment, the imaging is performed with a constant and moderate imaging interval (state A). As soon as the population size exceeds the threshold A_{start}, the state-machine moves to state B_0 and performs the EP using imaging action $\mathcal{I}_2$. This imaging action records a phase-contrast (PC) image followed by a fluorescence (FL) image. The light intensity and exposure duration of the fluorescence imaging define the EP that provokes the optogenetic effect. Both are parameters of the state-machine, turning the design into a parameterized state-machine. After the EP, the state-machine moves to state B_1 and performs phase-contrast imaging with an interval of 30 s to sample the immediate cell responses with higher temporal resolution. After 20 minutes, the state-machine continues to B_2 and performs the remaining imaging with the original imaging routine I_1. When a certain population size is reached (A_{stop}), the observation of the RoI is terminated.

Parameterizing this state-machine design with the parameters FL^{int} and FL^{dur} that determine the EP light intensity and exposure duration allows us to execute this state-machine with different parameters for different RoIs and precisely screen and measure their impact of the optogenetic effects at high-throughput (objective 1). Choosing different cell population thresholds A_{start} to trigger the excitation pulse allows us to investigate the influence of the population size and guarantees a sufficient number of living cells during the excitation (objective 2). Finally, we achieve high temporal resolution directly after the excitation pulse by adapting the imaging frequency (objective 3).

7.3.4 Closed-loop optogenetic growth control (SM4)

In the last state-machine example, we extend the influence into a closed-loop control of cell population growth following the idea of cybergenetic control. For this, we assume to have a modified microbial strain that is equipped with optogenetic mechanisms to inhibit and stimulate growth using two different light wavelengths. Using these two optogenetic mechanisms, we would like to precisely control the growth rate of the organism such that the cell population develops with a specific target growth rate.

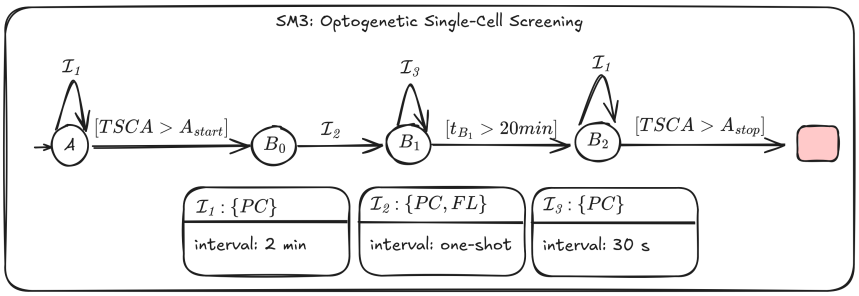

Figure 7.4: **Optogenetic single-cell screening state-machine (SM3).** SM3 observes the cell development until a certain initial population size is reached ($[TSCA > A_{start}]$), then applies an optogenetic influence ($\mathcal{I}_2$), switches to a high imaging rate observation in state B_1 to record the immediate cell development response, moves to a post-influence observation in state B_2, and terminates when the cell population grows too large ($[TSCA > A_{stop}]$). Parameters not depicted in the sketch are the intensity and duration of the fluorescence imaging action FL called FL^{int} and FL^{dur}, respectively.

Figure 7.5 shows the state-machine SM4 for the close-loop control imaging application. Initially, the population development is observed using phase contrast imaging (state A). As soon as there are more than 25 cells in the RoI ($[CC > 25]$), the state-machine transitions to state B and continues the phase-contrast imaging while continuously measuring the IGR μ_t of the cell population (see Section 4.3.3). As soon as the measured growth rate falls below the target growth rate ($[\mu_t < \mu_{target}]$), the machine moves to state C and moves back to state B by executing the $\mathcal{I}_{stim}$ imaging action stimulating the cell growth. If the growth rate is higher than the target growth rate ($\mu_t > \mu_t arget$), the state-machine moves to state D and back to state B by executing the $I_{inhibit}$ imaging action to inhibit the cell growth. To give the living cells time to respond to the optogenetic influence, we integrate a rate limit on moving to states C and D of $[t_B > 5min]$ (not shown in the state-machine figure). The state-machine terminates when the cell population size TSCA exceeds the threshold A_{stop} ($[TSCA > A_{stop}]$).

7.4 Scheduling state-machine executions in high-throughput experiments

To this end, we have formalized the individual imaging applications using EDIL in the form of state-machines. These state-machines contain the complete imaging application logic, including imaging actions and event reactions for a single RoI hosting a single cell population. In an MLCI high-throughput experiment, numerous of such RoIs are investigated such that one state-machine needs to be executed concurrently per RoI, all requesting resources for imaging. In order to orchestrate the execution of these state-machines and allocate imaging time on the microscope, we introduce the second important ingredient of the JediOS platform: the scheduler (see Figure 5.3).

The goal of the scheduler is to orchestrate the execution of a set of imaging applications - each associated with a RoI position and state-machine - in a high-throughput experiment and perform the requested imaging actions at the specified intervals. In the following, we introduce the con-

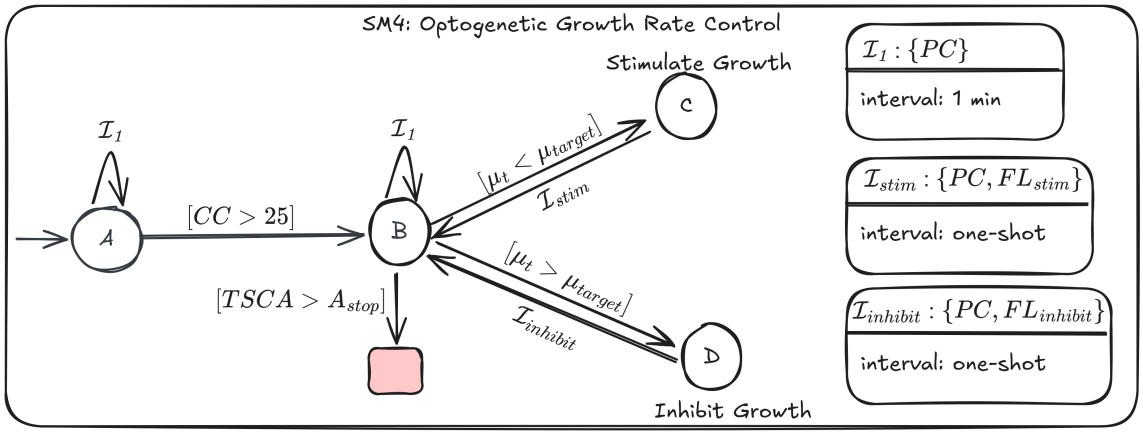

Figure 7.5: **Closed-loop optogenetic growth control state-machine (SM4).** The state-machine continuously observes cell population growth and stimulates or inhibits cell growth using optogenetic influences to bring the IGR μ_t close to the target growth rate μ_{target}. Parameters not depicted in the state-machine are the intensity and duration of the fluorescence imaging action for stimulation FL_{stim} and inhibition $FL_{inhibit}$.

cept of executing a state-machine and present a simple scheduling strategy for high-throughput experimentation.

7.4.1 Scheduling the execution of a single state-machine

The state-machine formalizes the imaging logic for a single RoI in terms of states, imaging, and event transitions. In the examples before, we have already introduced their execution in terms of an execution sequence. For scheduling, we now take a view of the temporal aspect of executing a state-machine.

The state-machine performs actions using the two transition types. Data updates, such as the arrival of new real-time information or the passage of time, trigger event transitions. Their execution is usually very fast and is neglected during the scheduling. The imaging transitions are triggered when the imaging interval elapses, creating an imaging request for recording images on the microscopy hardware. Capturing these images requires exclusive access to the microscopy hardware during their execution. Therefore, the scheduler primarily allocates imaging time on the microscopy hardware to individual state-machines. Figure 7.6 shows the execution of a single state-machine SM1 on two timelines. The upper timeline (SM_1) displays the imaging requests made by the state-machine to record images using the imaging action $\mathcal{I}_1$ at a specified interval. The imaging requests are created by performing imaging transitions during the execution of the state-machine. Each imaging request has a certain target time denoted by $t^\star$ and an expected execution duration denoted by $\tau(\mathcal{I}_1)$ (e.g., 2 seconds). The lower timeline (scheduling S) denotes the scheduled execution of the imaging routine on the imaging platform.

The scheduler continuously executes the state-machine, performing event and imaging transitions. Once an imaging transition has been performed and an imaging request is created, the scheduler places the imaging request on its schedule $\mathcal{S}$, allocating a specific time slot for its execution on

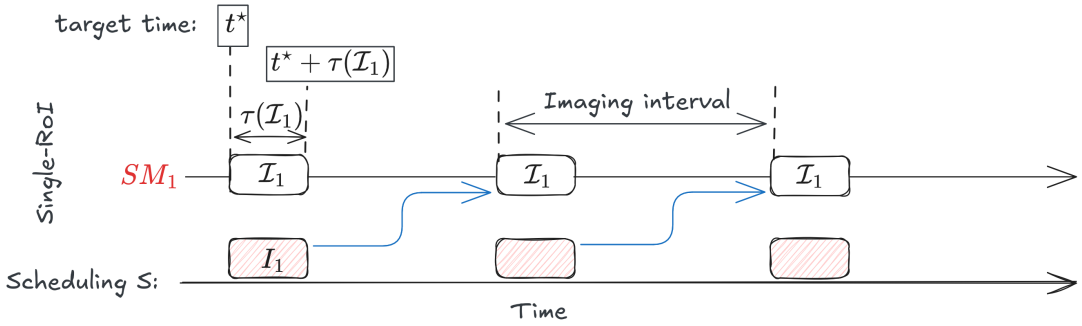

Figure 7.6: **The scheduled execution of a single state-machine.** The imaging transitions of the state-machine SM_1 request images using imaging action $\mathcal{I}_1$ at a target time $t^\star$ (timeline SM_1). $\tau(\mathcal{I}_1)$ denotes the execution duration of the imaging action I_1 on the microscopy hardware. The scheduler computes a sequential execution schedule of these image requests and allocates the corresponding time slots on the microscopy hardware (scheduling S). As soon as the scheduler performs an imaging request, the state-machine registers the next imaging request with a new target time (blue arrow). The repeated request for images leads to the recording of a microscopy time-lapse at the specific imaging interval.

the imaging platform. After finishing the imaging request, the scheduler informs the state-machine, and the new target time $t^\star$ for the next imaging request is computed on the SM_1 timeline (denoted by the blue arrow). In this simple example with a single state-machine, both timelines are fully synchronized, i.e., as soon as an imaging request is created, it is directly placed on the schedule to be executed on the imaging platform. Thus, computing the scheduling is trivial.

7.4.2 Scheduling the execution of multiple state-machines for high-throughput imaging

For high-throughput imaging, multiple state-machines are executed concurrently, performing individual imaging applications on the different RoIs of the microfluidic chip. This concurrent execution of multiple state-machines comes with one difficulty: If multiple state-machines request images at the same time, their requests cannot be fulfilled at their specific target time. Instead, the scheduler needs to decide when to place the imaging request in the schedule to compute a sequential execution order on the microscopy hardware. Figure 7.7 shows such an example of three SM1 machines that are concurrently executed, making imaging requests at the same time at the beginning of the experiment denoted in their timelines SM_1 to SM_3.

The scheduler cannot serve these simultaneous requests at their requested target time. Instead, the scheduler needs to find a sequential execution order for the requests leading to delays denoted by the dashed lines for SM_2 and SM_3. We call these delays the Request Delay (RD) of the imaging request. However, for the second round of imaging requests, no more delays occur. This is due to the dynamic creation of the imaging requests: When the scheduler has executed an imaging request for a state-machine, the new target time for the next request is computed based on the imaging interval. Thus, the scheduling timeline S maintains the requested imaging interval for all three state-machines. However, the timelines of the creation of the imaging requests (e.g., SM_1 and SM_2)

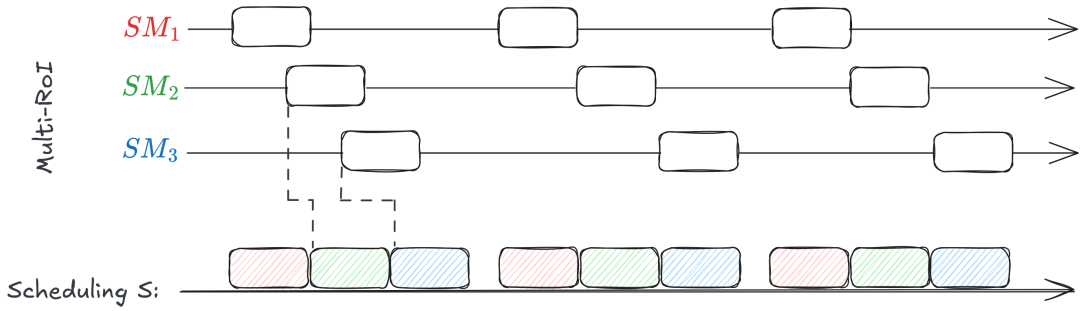

Figure 7.7: **Scheduled execution of multiple state-machines on a single microscope.** Three state-machines SM_1, SM_2, and SM_3 are executed concurrently. The scheduler computes a sequential execution order (lower timeline) of their imaging requests (upper three timelines) called the scheduling S. Due to overlapping imaging requests at the beginning of the experiment, delays are introduced in their execution, marked by dashed lines.

show longer intervals between the imaging requests as defined in the state-machine.

7.4.3 Scheduling algorithm

To formalize the computation of the schedule for executing imaging requests on the imaging platform, we introduce a simple scheduling algorithm inspired by the earliest-deadline first algorithm often used in real-time operating systems [171]. First, we formally define the delay of an image request.

Definition 1: Request Delay (RD)

Let $t^\star$ be the target execution time of an imaging request. Let t bet the current time. The Request Delay (RD) is defined as the time difference between the requested time point $t^\star$ and the current time t:

$$RD(t, t^\star) := t - t^\star. \tag{7.3}$$

The RD is negative if the requested target time for the imaging action is in the future ($t < t^\star$). The RD is positive if the requested target time for the imaging action is in the past ($t > t^\star$). We call this an overdue imaging request.

The goal of the scheduler is to execute all image requests with an RD of 0, so just at the requested target time. As this is not always possible (see Section 7.4.2), we define the scheduling strategy to execute the imaging request with the highest positive RD first. If all imaging requests have a negative RD and are not yet due, the scheduler idles and does not execute any action on the microscopy hardware. The algorithm for this strategy is shown in Algorithm 1.

One critical question for the scheduling is whether there exists a sequential scheduling such that all image requests can be served in time. While answering this question in general is challenging, we can approximate an answer by investigating the utilization of the microscope hardware.

Algorithm 1 Image request scheduling strategy

$\mathbb{T} \leftarrow \{t_1^\star, \ldots, t_K^\star\}$ # Target execution times of imaging requests
$t \leftarrow gettime()$
while At least one state-machine not yet in final state **do**
  $\mathbb{RD} \leftarrow \{RD(t, t_i^\star) | for\ i = 1, \ldots, K\}$ # Compute all RDs
  $i^\star \leftarrow argmax_{i \in \{1,\ldots,K\}} \mathbb{RD}_i$ # Compute maximal RD
  if $\mathbb{RD}_{i^\star} \geq 0$ **then**
    Move stage to corresponding RoI position
    Execute imaging request with index $i^\star$
    Update corresponding state-machine
  end if
  $\mathbb{T} \leftarrow$ Update imaging requests
  $t \leftarrow gettime()$ # Update time
end while

Definition 2: Predicted Utilization (PU)

Let $\mathbb{I} = \{IR_1, \ldots, IR_K\}$ be the set of K imaging requests at time t. Let λ_i and $\tau(IR_i)$ be the imaging interval and execution time of the image request IR_i, respectively.

We define the Predicted Utilization (PU) of the microscope to be the percentage of time the microscope will utilize to fulfill the imaging requests at their imaging interval:

$$PU(\mathbb{I}) := \sum_{i=1}^{K} \frac{1}{\lambda_i} \cdot \tau(IR_i). \tag{7.4}$$

The lower limit of the PU is 0 and corresponds to no active imaging requests and a microscope that is completely idle. Formally, the PU has no upper limit. However, values of more than 100% indicate that there are more imaging requests than the microscope can serve in time. In particular, there cannot exist a sequential scheduling that serves all imaging requests at their target time points. We call this scenario "over-utilization" and discuss this in detail within our simulation studies (Section 7.6).

The PU is an approximate measure of the actual microscopy utilization, as the interval of imaging requests may change due to the processing of events based on real-time information. Thus, the PU is an approximate measure for the microscope's utilization at a specific point in time and over a short time window.

7.5 Simulating high-throughput experiments on the JediOS platform

In the previous chapters, we have introduced the EDIL to design state-machines for executing complex imaging applications (Section 7.3) and a scheduler that orchestrates the execution of these imaging applications in a high-throughput scenario (Section 7.4). In this chapter, we combine these two ingredients with basic cell population models to simulate the development of cell populations

within RoIs (Section 7.5.1) and turn JediOS into a complete in-silico simulation platform for high-throughput MLCI experiments outlined in Figure 5.3. These in-silico simulations are crucial for evaluating the more complex event-driven implementation of imaging applications, understanding the interdependence between high-throughput and limited imaging resources, and verifying the correct implementation of the JediOS platform. In this chapter, we first present the cell population models and then describe the implementation of the JediOS in-silico simulation platform.

7.5.1 Modeling cell population development

Modeling and simulating the temporal cell population development is inevitable to test the correct implementation of event-driven imaging applications that react to the cell population development. Thus, we need models that describe the development within a RoI. Such a model is then queried by a phase-contrast imaging action and performs a forward simulation to provide a biological read-out (e.g., CC or TSCA) at a given point in time (Figure 5.3).

To simulate temporal cell population development, we utilize exponential growth models. Exponential growth is frequently observed in MLCI experiments with continuous medium supply and, thus, provides a solid and simple baseline for describing temporal cell population development (see Chapter 4).

Definition 3: The exponential cell count model

Let $N_0 \in \mathbb{N}$ be the initial number of cells within a cell population and $\mu \in \mathbb{R}$ their exponential growth rate. The number of cells at a time $t \in \mathbb{R}$ is then defined as

$$N(t) = N_0 \cdot e^{\mu \cdot t}. \tag{7.5}$$

Image-based readouts of the cell count rely on counting their number and are natural numbers. Thus, we define an integer read-out version $\hat{N}(t)$ of the cell count model

$$\hat{N}(t) = \lfloor N(t) \rfloor \in \mathbb{N}. \tag{7.6}$$

In the following, $\hat{N}(t)$ is used to compute the real-time readouts of cell counts in cell population models.

Similar to the number of cells, the TSCA of the cell population is simulated with an exponential model. For the sake of simplicity, we assume that the average cell size $\bar{s}$ within the cell computation is constant and known. Then, the TSCA of the cell population is directly derived from the exponential cell count model.

Definition 4: The exponential population area model

Let $\bar{s}$ be the average cell size within a cell population, and let $N(t)$ describe its cell count. Then the TSCA of the cell population is defined as:

$$A(t) = N(t) \cdot \bar{s}. \tag{7.7}$$

7.5.2 Implementation of the in-silico simulation platform

The underlying idea of the in-silico simulation framework is to design and test high-throughput MLCI experiments on the JediOS platform without the need for real microscopy hardware. Once the tests are completed, the same code for the imaging application logic should then be executed on the real microscope hardware. This is essential to keep the number of error sources low and quickly transition from the experiment design to its execution. The main difference between a simulated experiment and a real experiment from the perspective of an imaging application developer is the low-level implementation of stage movements and imaging actions handled by the operating system. All other implementations, including the experiment setup, event processing, and scheduling, remain the same.

The design of the simulation platform is visually outlined in Figure 5.3 and formulated as pseudo-code in Algorithm 2. Within a high-throughput experiment JediOS concurrently executes a set of imaging applications, each consisting of a state-machine formalized in EDIL and a RoI ID that uniquely associates it to a RoI position and cell population within the microfluidic chip. The scheduler of the operating system then orchestrates the sequential execution of the imaging requests as described in Section 7.4.2. These imaging requests, for example, phase-contrast or fluorescence imaging, need to be executed on the imaging platform. Here we distinguish between the real microscopy platform and the in-silico simulation. For the real experiment, the image requests execute the movement of the stage to the RoI location and record an image that needs to be processed to extract real-time information on population development. The details of the underlying implementation are introduced in Chapter 8.

In the in-silico simulation, the imaging requests are simulated. Thus, they query the cell population model associated with the RoI ID of the imaging application. The model then simulates readouts, for example, CC or TSCA, which are then used to trigger events in the corresponding RoI state-machine. During the experiment simulation, this procedure is repeated until either all state-machines have reached a final state or a timeout is triggered.

Algorithm 2 In-silico simulation of high-throughput event-driven imaging.

```
imaging_applications                                  # Set of (RoI ID: str, state-machine)
scheduler                                             # The JediOS scheduler
models                                                # Model for each RoI identified by their ID.
while At least one imaging application's state-machine not yet in final state do
    ia ← scheduler.next(imaging_applications)         # Compute scheduling
    rt_info ← ia.perform(models)                      # Perform the next imaging request
    Update time                                       # Update the time passed during the imaging requests
    ia.update(rt_info)                                # Update the state-machine with real-time information
end while
```

Moreover, for each image request, we simulate the time it takes to execute it on microscopy hardware. Therefore, we simulate the passing time for the two primary actions: the stage movement for approaching a RoI position and the duration of an imaging action. Both durations can be measured on the actual microscopy platform to use realistic values. In our simulations, we assume 1 second for every stage movement, as well as 1 second for recording a phase-contrast image. The duration of fluorescence imaging is composed of the exposure duration and a constant time duration

of 0.5 seconds per requested image.

We implemented the in-silico simulation outlined in Algorithm 2 in Python. For the implementation of the state-machines, we utilize the `pytransitions` library (`https://pypi.org/project/transitions`) that allows to implement the state-machine logic in Python based on their graphical representation of states and actions introduced in this chapter.

During the closed-loop in-silico simulation, we test and verify an entire experiment design before executing it on the actual imaging hardware. In particular, such high-throughput experiment simulations can be directly integrated into a CI pipeline. This helps to continuously verify the expected experiment execution and also verifies the current implementation of the JediOS platform. These continuous tests bear the potential to highlight breaking changes in future software versions or errors in the implementation, thereby reducing possible error sources before executing the experiment on a real microscope with living cells. With the increasing complexity of imaging applications, it is crucial to reduce the costs and time consumption associated with failed experiments.

In addition, the simulation platform enables us to simulate and measure the microscopy utilization, providing a better understanding of the interdependence between the number of concurrently monitored RoIs (throughput) and the design of the state-machine and its imaging parameters (e.g., imaging rates). Based on RD and PU metrics measured during the simulation, over-utilization of the microscope can be detected even in very complex experiment designs.

7.6 Simulation studies

In the following, we utilize the JediOS simulation platform to validate our state-machine designs completely in-silico before their deployment on the actual microscope setup. Moreover, we highlight the specifics of high-throughput experimentation and discuss the phenomena of over-utilization and asynchronous execution.

7.6.1 Simulation study 1 - basic imaging experiments (SM1)

The first state-machine (SM1) is designed to execute the basic experiment in MLCI, namely the recording of phase-contrast images at multiple RoI positions with a constant imaging rate, terminating after a specified observation period (Figure 7.2). This simple example is ideally suited to verify the basics of our simulation framework and introduce the measurements obtained during the simulation, especially with regard to selecting a suitable number of concurrently observed RoIs and preventing over-utilization.

Thus, we perform a simulated high-throughput experiment with $15, 30$, and 45 concurrent executions of the SM1 machine on the simulation platform. The imaging interval is set to 1 minute and the maximum observation period to 5 hours. Figure 7.8 shows the summary of this simulation, where every column denotes a high-throughput simulation with a specific number of RoIs, and the two rows report two metrics that are measured during the simulation.

The first row shows the temporal state distribution chart, i.e., the share of state-machines with a specific active state at a given point in time. In this example, all machines start in state A and remain in this state throughout the entire imaging experiment. Thus, 100% of the state-machines are in state A, indicated by the blue region. At the end of the experiment, the state-machines start to switch to the terminal state $STOP$.

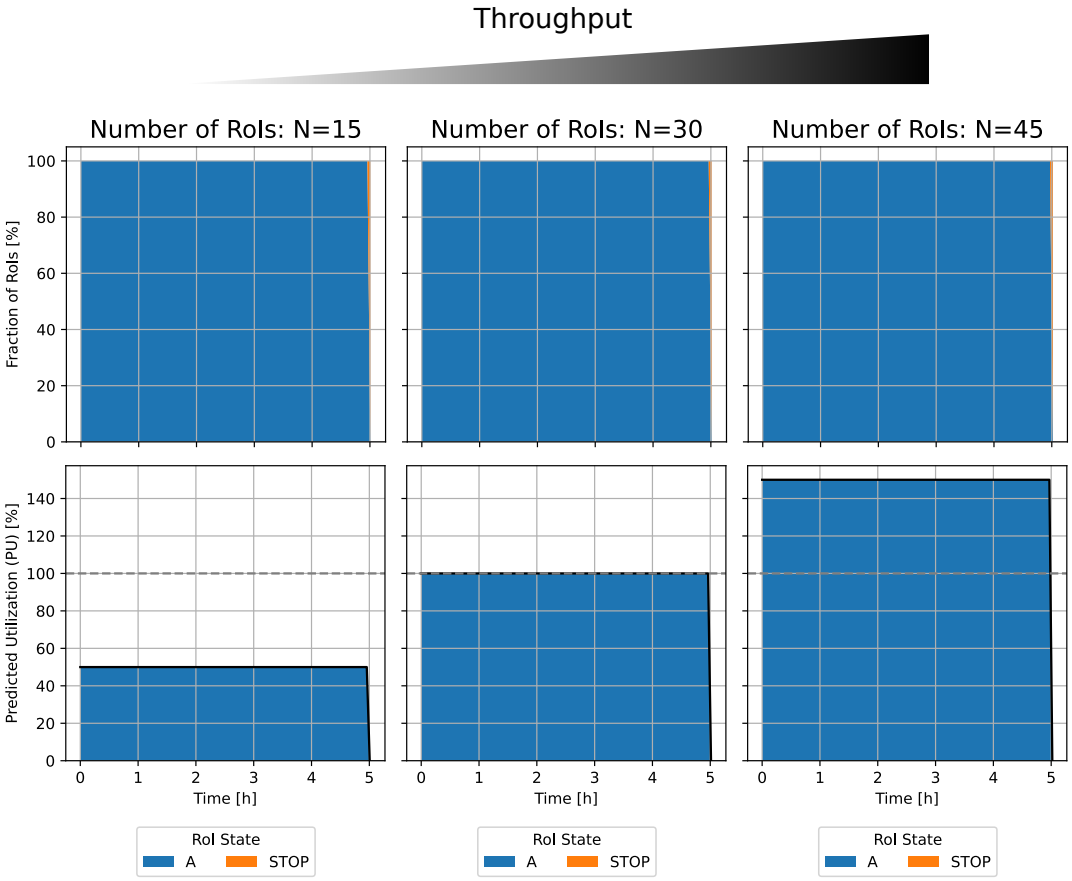

Figure 7.8: **High-throughput simulation of standard imaging experiment (SM1).** The columns denote high-throughput simulations with $15, 30$, and 45 concurrently running state-machines. The rows denote the two measured metrics: the temporal distribution of active state-machine states (top) and the predicted microscope utilization (bottom). The temporal distribution of active states is constant for a long time as all state-machines remain in initial state A (blue). At the very end of the experiment the state-machines terminate and move to the $STOP$ state (orange). The PU is measured during the simulation and is constant for a long time as the imaging intervals are constant. At the end of each experiment it drops to 0 as all state-machines terminate. A higher number concurrently observed RoIs leads to a proportional increase in the PU. The colors (blue, orange) denote the contribution of the individual states to the measured metrics.

The second row shows the PU of the microscope in all three high-throughput scenarios with a PU of 100% highlighted by a gray dashed line. The PU is very similar to the CPU utilization that you would see in your computer's "task manager". PUs below 100% indicate that the that the microscope idles from time to time when all imaging requests have been handled. A utilization of 100% indicates that the microscope is busy all the time recording images, and a utilization larger than 100% indicates that there are more imaging requests than the microscope can serve in time, eventually leading to delays and longer intervals between images than requested. In this simulation, the imaging applications record images at a constant imaging interval throughout the complete experiment. Thus, the utilization remains constant throughout the experiment and only decreases at the end, when the state-machines start to transition into the $STOP$ state and stop requesting images. However, the number of concurrently observed RoIs changes the utilization of the microscope, as it affects the number of images requested. In particular, with an imaging interval of 1 minute and the duration of moving to the RoI position and recording an image of 2 seconds, the theoretical maximum of concurrently observed RoIs without delays is 30.

Our simulation study confirms this theoretic maximum. Monitoring $15, 30$, and 45 RoIs leads to a utilization of $50\%, 100\%$ and 150% of the microscope indicated by the solid black line. Moreover, we illustrate the contribution of each state to the PU in its respective color below the black line. In SM1, there is only a single state with an imaging transition. Therefore, the state A (colored blue) is the only one contributing to the PU, leading to a complete blue coloring.

A utilization below 50% will lead the scheduler to move to an idle state, taking breaks from imaging in 50% of the experiment time. A utilization of 100% means that the microscope is busy all the time, continuously moving to RoI positions and recording images. The case of 150% utilization ($N = 45$ RoIs) is specifically interesting: utilizations above 100% show that the scheduler cannot serve all the imaging requests by the RoI machines at their target time leading to the "over-utilization" effect. This will lead to delays in the imaging, causing a longer imaging interval than requested by the state-machines. In fact, in the case of concurrently observing 45 RoIs, the imaging interval will rise to 1.5 minutes due to the over-utilization of the microscope's imaging capacities.

While "over-utilization" leads to longer imaging intervals than defined in the state-machine image routines, this does not break the experiment execution. The scheduler continues to execute imaging requests, and the microscope continuously records images. Thus, the primary effect of over-utilization is the increase in requested imaging intervals, which may make the extraction of meaningful biological information difficult.

7.6.2 Simulation study 2 - Adaptive imaging (SM2)

We presented the state-machine SM2, motivated by the need to adapt the imaging rate to the development of the biological population, thereby achieving a better temporal resolution for larger cell populations, particularly to enhance tracking performance. In this case study, we incorporate biological population models into the simulation and highlight that the variety in initial cell counts and growth rates leads to an asynchronous temporal execution of the state-machines. We demonstrate that this asynchronous execution bears the potential to perform experiments with a larger number concurrently observed RoIs (higher throughput) compared to standard experiments with a constant and high imaging rate.

To simulate such an experiment adapting the imaging based on biological readouts, we combine

the simulation of the state-machine execution with biological models introduced in Section 7.5.1. Interestingly, the inoculation process, the procedure that seeds cells into the RoIs in a real experiment, leads to very diverse initial numbers of cells in each RoI. Moreover, cell populations typically exhibit a range of growth rates. To model both effects, we sample the initial number of cells N_0 and the growth rate μ for each RoI population model from uniform distributions:

$$N_0 \sim U(1, 40), \tag{7.8}$$

$$\mu \sim U(0.4, 0.8) \cdot 1/h. \tag{7.9}$$

The parameters of the uniform distributions depend on the initial filling of RoIs after the inoculation procedure and the variation in growth rates, here approximated for *C. glutamicum* populations. These parameters are then used in the exponential population model

$$N(t) = N_0 \cdot exp(\mu \cdot t), \tag{7.10}$$

to simulate the temporal development of each cell population.

Figure 7.3 shows the state-machine design SM2. We specify TSCA thresholds ($A_1 = 15\%, A_2 = 30\%, A_3 = 60\%$) in percent of the total RoI area and the sequentially decreasing imaging intervals of $10, 5$, and 1 minute. We compare the experiment with adapted imaging rates to an experiment with constant imaging rates of 1 minute (SM1) that can observe a maximum of 30 RoIs at the highest imaging rate (Section 7.6.2). Figure 7.9 shows the simulation of the adaptive imaging rate experiment at high-throughput for $30, 60$, and 90 RoIs.

In contrast to the previous simulation results, we added the first row that shows the TSCA development for the cell populations in each RoI simulated by exponential cell populations models. Due to the exponential cell population model, all lines are linear in the log space, and the growth rate determines their slope. Sampling the initial cell count N_0 and growth rate μ from the uniform distributions leads to different initial values (time $t = 0\ h$) and slopes of the TSCA developments. The main difference for the TSCA development across the high-throughput simulations is the number of RoIs determining the number of cell population developments that are concurrently recorded.

The second row shows the temporal distribution of the active state-machine states. All state-machines start in state A such that initially 100% of the state-machines are in state A. At the end of the experiment, all state-machines are in the final state $STOP$. However, during the execution of the experiment, events based on the developing cell population area lead to an asynchronous transition of the state-machines through the states B, C, and $STOP$. This transition is visible as a front in the temporal active state distribution (second row). Interestingly, the state-machines are never all in state B or C at the same time but are usually distributed across the different states. The first state-machines already finish execution after 5 hours, whereas most other state-machines have not yet entered state B. This effect is caused by event-based state transitions, where the difference in the initial number of cells and growth rates leads to "asynchronous execution". This "asynchronous execution" effect is caused by the asynchronous execution of the imaging application that depends on the development of the underlying cell population determines its progress. Thus, a high initial number of cells and rapid growth lead to an early transition to states B, C, and $STOP$, resulting in a relatively short execution period. Importantly, a terminated state-machine does not consume any microscopy imaging resources. Therefore, these can be spent on other RoIs that are still worth observing.

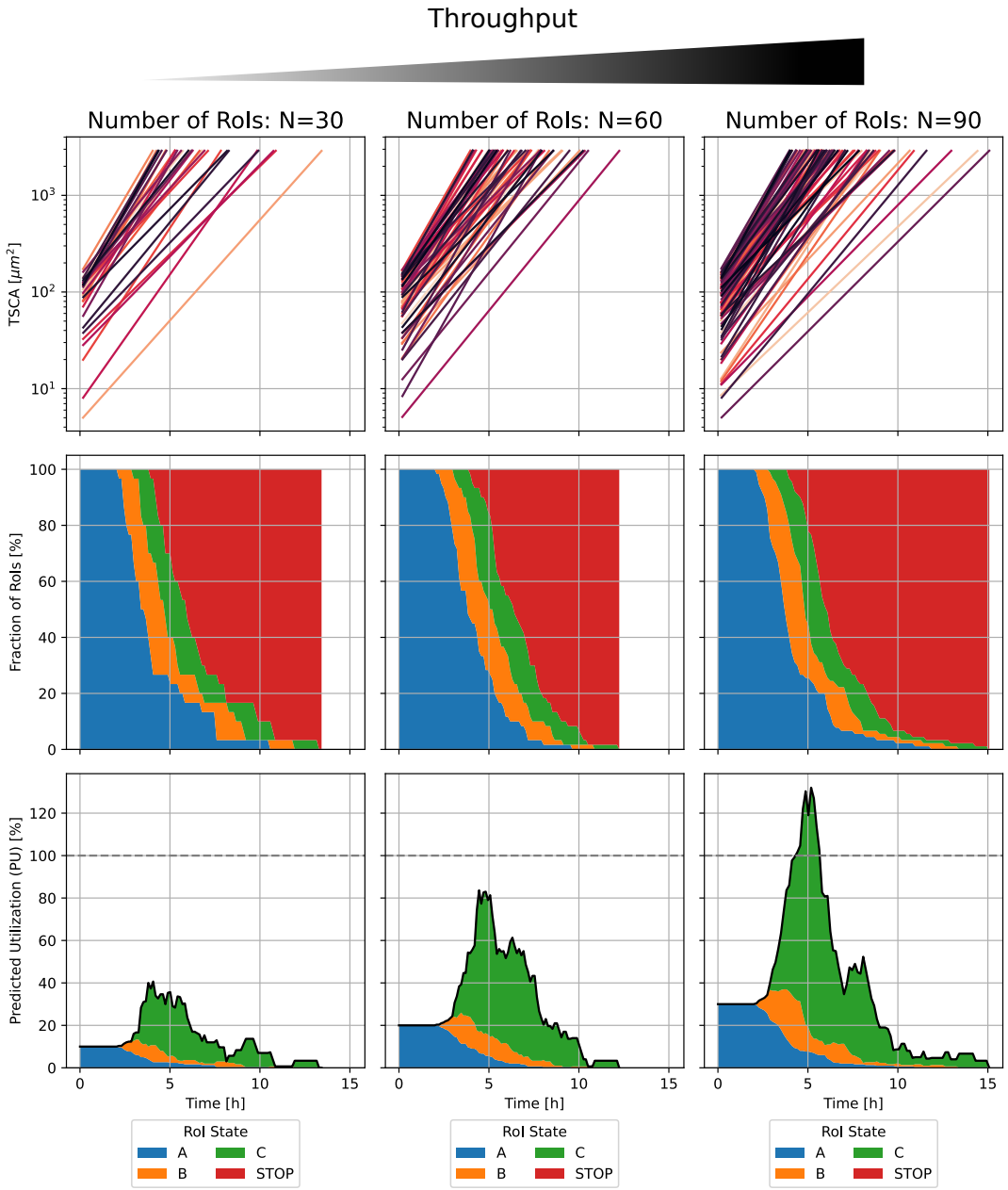

Figure 7.9: **High-throughput simulation of adaptive imaging rates (SM2).** Every column shows simulations with different numbers of RoIs ($N = 30, 60, 90$) while the rows show three metrics of the simulation. The first row shows the simulated TSCA development of the individual cell populations in logarithmic space. The growth rate of the cell populations corresponds to the slope of the lines. The second row shows the temporal distribution of active state-machines states. The bottom row shows the PU (black line) and the contribution of different states. The coloring of the states is the same as in the second row. The dashed gray line indicates 100% utilization.

The benefits of asynchronous execution become apparent in the measurements of the PU shown in the bottom row. We have shown that a standard experiment with constant imaging rates allows us to observe at most $N = 30$ RoIs concurrently at an imaging interval of 1 minute, leading to a full utilization of the microscopy (Section 7.6.2). In contrast, the experiment with adaptive imaging rate and the same number of monitored RoIs ($N = 30$) barely exceeds a microscope utilization of more than 40% (left column). The utilization is low due to the asynchronous execution of the state-machines: they are never all in the same state at the same time, specifically in state C, which records with a low imaging interval (one minute) and consumes the largest amount of microscopy resources. Moreover, some state-machines may have already finished their observation, freeing the imaging resources for other state-machines. To better understand this effect, we have decomposed the contributions of each state to the PU. In particular, state C (green) with an imaging interval of 1 minute leads to the highest contribution as it requires the most microscopy resources. This leads to a rapid increase and spikes in the PU when many state-machines move to state C at the same time. If all state-machines were in state C at the same time, this would lead to a PU of 100%, similar to the SM1 study. However, the dynamic adaptation of the imaging rate based on the development of the cell population leads to a maximal PU of roughly 40%.

Doubling the number of RoIs ($N = 60$, center column) leads to a maximal PU of 80%. Thus, the adaptive choice of imaging rates allows us to observe twice the number of cell populations compared to a basic experiment with a constant imaging rate. Only the third scenario, with a total number of $N = 90$ concurrently observed RoIs, leads to PU above 100%, indicating that it is impossible to serve all imaging requests within their specified intervals. The large contribution of state C, where images are requested every minute, becomes apparent in this last high-throughput simulation. Nonetheless, observing $N = 90$ RoIs is already a three times higher throughput compared to the basic non-adaptive experiment.

7.6.3 Simulation study 3 - high-throughput light excitation pulse screening (SM3)

Our third simulation study investigates the state-machine design SM3 that has been specifically developed to screen the effects of an excitation pulse and its immediate impact on cell population development. Herein, the goal is to specifically record the aftermath of the excitation pulse at high temporal resolution (imaging interval of 30 seconds) while still being able to perform these experiments at high-throughput to screen many excitation pulse parameters. In this study, we are specifically interested in the implications of the microscope utilization and how many RoIs can be observed concurrently with such a design.

In contrast to previous simulation studies, we integrate the influence of the excitation pulse into the cell population model. For now, we model a simple light-based stress induction into the cell population and approximate the influence by a constant reduction in the growth rate of 0.3 h^{-1}.

We performed the simulation study in four high-throughput scenarios with $N = 15, 30, 45, 60$ concurrently executed state-machines. A basic imaging experiment with a constant imaging interval of 30 seconds has the capacity to record 15 RoIs at 100% microscope utilization. Figure 7.10 shows the simulation results for the four scenarios organized in columns of an increasing number of RoIs and measures TSCA development, temporal active state distribution and microscope utilization (in rows, top to bottom).

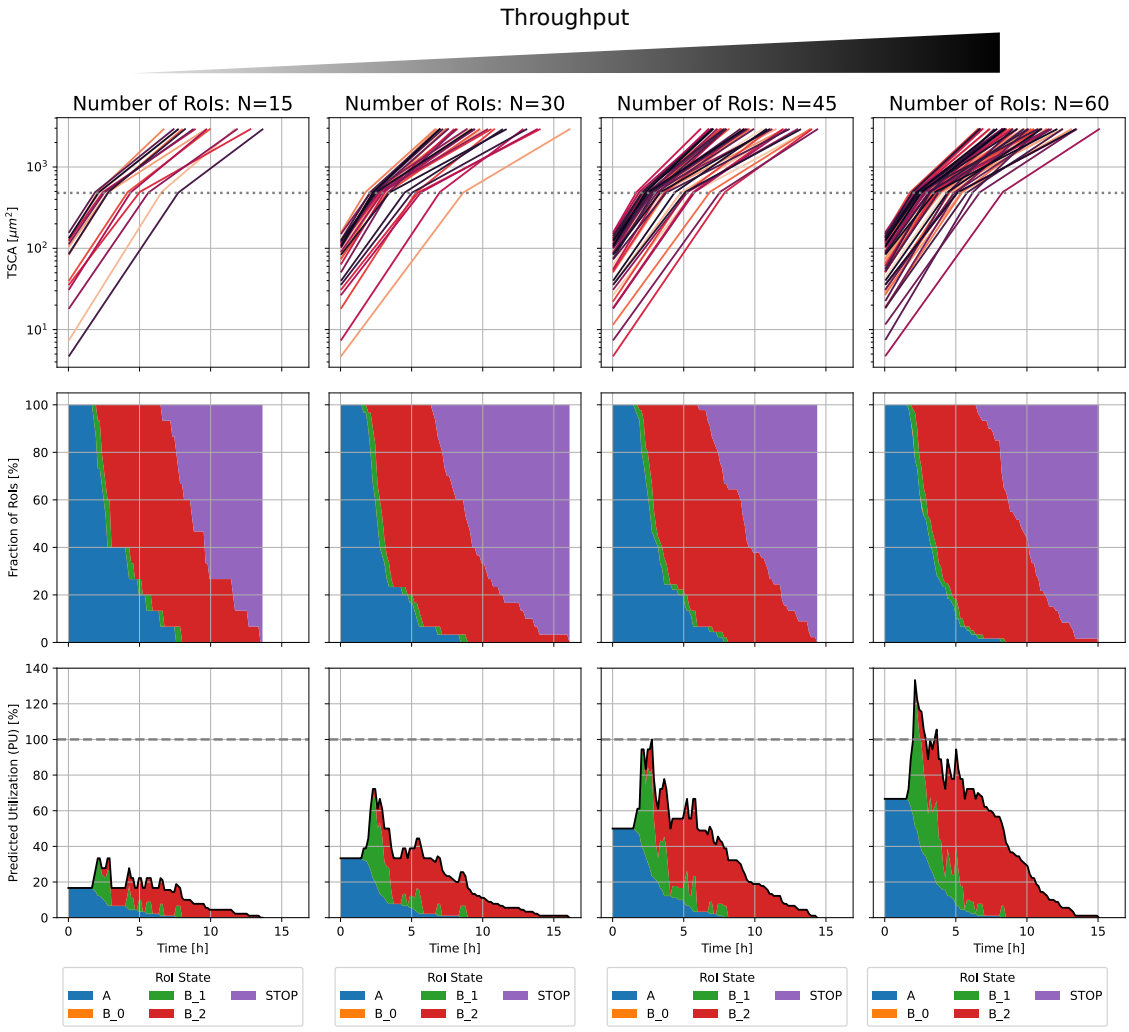

Figure 7.10: **High-throughput simulation of optogenetic influence screening.** Every column shows the metrics (rows) measured using different numbers of RoIs ($N = 15, 30, 45, 96$). The first row shows the TSCA development of the individual RoIs in log space such that their growth rate corresponds to their slope. The gray dotted line denotes the event of 10% filling of the RoI triggering the light-based stress induction. The second row shows the temporal active state distribution of the state-machines. The last row shows the PU (black line) and its decomposition to the contribution of different states (same coloring as in the second row). The dashed gray line indicates 100% utilization.

The first row shows the simulated TSCA development, with the TSCA threshold that triggers the excitation pulse indicated by a dashed gray line. As soon as the cell population's TSCA exceeds this threshold, the state-machine triggers the excitation pulse, leading to a reduction of the growth rate as implemented by the underlying cell population model, causing the "kink" in the temporal TSCA development. As the underlying cell population development triggers these excitation pulses, they happen at different points in time.

This effect is also observed in the temporal state distribution charts shown in the second row. All the state-machines start in state A, and only as the TSCA exceeds the trigger, they move forward through states B_0, B_1 and B_2, where the excitation pulse happens on the transition $B_0 \rightarrow B_1$. Across the different high-throughput scenarios, these state charts appear very similar, with only their granularity changing when more RoIs are included, leading to finer changes in the distribution.

Most interesting is the utilization of the microscope (last row). The scenario with $N = 15$ shows a maximal PU of 30%, much lower than the 100% of the corresponding basic experiment. This low utilization is achieved by the event-driven RoI state-machines that only sample the development of the cell population with high frequency (low imaging interval) in the immediate aftermath of the excitation pulse. As the excitation pulse is triggered by a threshold of TSCA and the cell populations have different initial cell counts and growth rates, these bursts of imaging occur at different points of time throughout the experiment. Thus, the high utilization of the microscope distributes across the experiment time and never accumulates to a 100% utilization of the microscope, making the benefit of asynchronous execution apparent.

Raising the number of RoIs to 30 and 45 shows an increase in PU, especially showing the high utilization caused by state-machines in state B_1 (green), the state used for performing imaging with low intervals (30 seconds). However, every state-machine resides in this state for only 20 minutes and then switches back to a longer imaging interval of 3 minutes (state B_2), requesting fewer imaging resources. These "bursts" in imaging cause spikes in the PU diagram, leading to a very uneven distribution of microscopy utilization. For $N = 60$ RoIs, the PU spikes above 100%, leading to an over-utilization of the microscopy. Nevertheless, with a throughput of 45 RoIs, the event-driven experiment achieves three times the throughput compared to the baseline experiment while sampling the point of interest after the excitation pulse with the same imaging frequency.

7.7 Discussion and Conclusion

In this chapter, we have developed the JediOS platform that utilizes a new divide-and-conquer perspective on high-throughput live-cell imaging events to design event-driven imaging applications and evaluated them in in-silico simulation studies. This divide-and-conquer perspective enables the flexible design of imaging applications for each RoI and cell population. After their design, multiple imaging applications can be combined into a high-throughput live-cell experiment using JediOS. This divide-and-conquer and operating system perspective unlocks completely new opportunities for smart and event-driven imaging applications. Combining this with cell population models leads to a comprehensive in-silico simulation platform with events based on real-time (simulated) biological readouts. These simulations unlock many insights into the experiment design in minutes of compute time before its actual execution on microscopy hardware, which may take hours or days. Especially investigating the utilization of the microscope and choosing an appropriate number of concurrently observed RoIs plays a key role for live-cell imaging design. Such simulations enable us to verify the

design of the state-machines and their composition into a high-throughput experiment, identifying bugs and misconceptions early in the development process. This has the potential to reduce the failure rate of real experiments significantly. Moreover, learning to design imaging applications using EDIL and state-machines becomes easier as the simulation can serve as an educational playground to learn the good and bad practices of designing state-machines and high-throughput live-cell imaging experiments.

Based on all the advantages that come with the current JediOS platform, we would like to give a more detailed view of possible limitations and future improvements.

7.7.1 Asynchronous Execution

During the simulation studies, we found that the effect of asynchronous state-machine executions enables us to observe cell populations at higher throughput while maintaining the same imaging rates at specific points of interest. These beneficial effects of the asynchronous execution highly depend on the cell population development, the selection of reasonable cell population models, their model parameters, and the state-machine designs used in the experiment. The effect of asynchronous becomes most beneficial for wide distributions of initial cell counts and growth rates, leading to a temporally asynchronous development of the cell populations. The JediOS simulation platform enables the simulation, study and test of a wide range of scenarios to determine a suitable number of RoIs using precise models for cell population behavior and parameter distributions, for example, those derived directly from the latest experiment. To incorporate the stochasticity of the inoculation procedure and cell population behavior, multiple simulations can be used in a Monte Carlo scheme to determine the likelihood of potential over-utilization of imaging resources.

7.7.2 Continuous integration of software-defined experiment designs

By designing imaging applications as state-machines using the EDIL, the imaging experiment is fully defined in software. This allows the use of best practices in software development, such as unit testing and continuous integration, to continuously test the experiment execution every time the software is updated. This brings two major advantages: (1) The design of imaging applications is thoroughly tested in-silico before implementation in the lab. Thus, misconceptions or design flaws are uncovered early in the development process, especially since such a simulation takes only minutes compared to several days of work required for a complete experiment setup in the lab. (2) Entire experiments with many imaging applications running concurrently can be simulated and verified while effects such as over-utilization can be thoroughly tested and monitored. These simulations support users in developing new imaging application designs and choosing an appropriate throughput. Moreover, such tests continuously test and verify the JediOS platform. Thus, changes during development that break existing experiment designs are immediately uncovered. This prevents updating the JediOS platform and identifying issues in the lab at the start of the experiment, or even more critically, during the experiment, where an implementation error may cause the entire experiment to terminate. Such errors, for example triggered by an event that occurs during overnight execution, may render the full experiment unusable. Such experiment failures are reduced by the JediOS platform by using the same code during the simulation as during the experiment.

7.7.3 Advanced scheduling strategies

The scheduling strategy applied in JediOS is very basic: The scheduler serves the imaging request with the highest delay or idles if there are no imaging requests. In practice, this is not optimal, for example, in a scenario where five imaging applications send an imaging request at the same time. Currently, this leads to an increasing delay as the scheduler handles their requests one by one. This is particularly unfavorable if the scheduler has been idling before and waited for the burst of imaging requests to be due. Thus, instead of trying to execute an imaging request exactly when it is due (and collecting delays in bad scenarios), the scheduler should be able to look ahead and bring specific image requests forward in time, for example, to minimize the maximal deviation from the requested imaging time. This would relax situations where state-machines are synchronized and request images at the same point in time. Thus, instead of a due time, the imaging requests should come with a due window that allows the scheduler to optimize their execution within certain temporal boundaries.

During our simulation studies, we have assumed that stage movements are constant in time. However, a larger movement of the microscopy stage takes longer and usually increases the refocus period, i.e., the time the microscope needs to regain focus on the sample after a movement. In principle, the microscope needs to visit all the RoIs in a loop and be at different locations at a certain point of time or within a specific time window. In particular, longer moves increase the travel time. This problem is very similar to the well-known Traveling Salesman Problem (TSP) [172]. Even closer to our scheduling case is the TSP with time windows [173]. However, it is also well-known that solving the TSP is NP-hard. Thus, finding solutions for the TSP to improve the scheduling of the imaging applications might be too computationally demanding for a real-time scenario. However, constraining the optimization to a limited time window, for example, the next 10 RoIs requesting images, might be computationally feasible.

Despite finding an optimal scheduling solution, the scheduler may also help in cases of over-utilization. For example, there may be RoIs that are more important than others, maybe due to their far-progressed experiment or specific biologically interesting events that have been observed. For now, the scheduler treats all state-machines with equal priority, and in case of over-utilization, the sampling frequency for all RoIs will decrease. A potential workaround is the introduction of priorities for the RoIs: Thus, the scheduler should prioritize state-machines with a high priority. In the event of over-utilization, the scheduler can still fulfill the imaging requests of the most important RoIs, while low-priority RoIs receive longer delays. Appropriate choices for priorities are determined by simulating complete experiments to verify that they yield higher-quality biological observations.

In general, the scheduler is a separate module of the JediOS platform and can be exchanged or optimized in future versions of the platform. Even multiple maintained schedulers are possible, allowing the user to decide, per experiment, which scheduling strategy is most suitable. This decision can then be verified in the in-silico simulation.

7.7.4 Mixing state-machine designs

For now, all our simulation studies have been performed using a single type of state-machine and executing them in a high-throughput experiment. However, in principle, completely different state-machine designs can be used within a single experiment as their executions are all treated as independent imaging applications. For example, we parameterized the excitation pulse parameters for

state-machine SM3 to screen their effect on the development of the cell populations. Moreover, we can also mix completely different designs. For example, we can execute multiple instances of the SM1 state-machine to observe a control group without any influence in the same high-throughput experiment. Choosing a moderate imaging interval allows us to limit the consumption of imaging resources by this control group, leaving more resources available to investigate the specific behavior of influenced cell populations. We showcase the mixing of state-machine designs in the next chapter when developing a high-throughput experiment to screen optogenetic effects on cell development in a real experiment (Section 8.4.2).

This mixing of state-machine designs underscores the flexibility in designing imaging experiments using the JediOS platform, while also highlighting the importance of in-silico simulation. The increased complexity of these experiments makes it inevitable to simulate them and verify their correct execution and usage of microscopy resources.

7.7.5 From simulation to the real experiment

The JediOS platform is designed with the idea of using the same code that has been verified in the in-silico simulation to run the real experiment. The main difference in the execution is the low-level implementation of the imaging routine on the microscopy platform (Figure 5.3): Instead of triggering a simulated cell model, they need to perform the imaging on the microscopy platform, and these recorded images have to be analyzed in real-time in order to extract the biological information that is used for events of the state-machines. The implementation of these two aspects is presented in detail in the next chapter.

Chapter 8

Smart live-cell experimentation using JediOS at the example of high-throughput photosensitizer stress induction

Contents

8.1 Motivation **112**
8.2 Implementing the microscopy platform **113**
8.3 Real-time image analysis pipeline (RT-IAP) **115**
8.4 Smart MLCI experiment application: High-throughput photosensitizer screening **117**
8.5 Smart MLCI experimentation workflow leveraging real-time information and event-driven imaging **120**
8.6 Discussion and Outlook **131**
8.7 Summary **134**

The development of JediOS in the previous chapter demonstrated the immense potential in event-driven imaging applications that react to the growth of every cell population in high-throughput MLCI experiments. The goal of this chapter is to extend these event-driven experiments into complete "smart" experiments that are executed on the actual microscopy platform and unlock nouveau insights into single-cell behavior. The "smartness" of the MLCI experiments is defined by the decision made during the experiment using the JediOS platform but also by the workflow to conduct and setup the experiment, for example, for selecting suitable RoIs and cell populations on the microfluidic high-throughput chip. Moreover, a smart MLCI experiment should formulate and document all decisions in software, making them easy to reproduce, share and extend.

In this chapter, we combine the achievements of the DMC and JediOS to develop smart MLCI experiments. Therefore, we first develop the software to control the microscopy platform and implement the RT-IAP to quantify the cell population development in real-time. Both developments are

integrated into the JediOS platform (Figure 5.3). We showcase the development and conduction of a smart MLCI experiment at the example of a high-throughput screening for an optogenetic stress induction using so-called photosensitizers. The smart experiment contains the development of a suitable state-machine to trigger and quantify the influence, selecting suitable cell populations in RoIs, and running the state-machines with different excitation parameters to conduct a high-throughput screening and reduce the number of required experiments. Our experiment demonstrates that the JediOS platform enables the rapid acceleration of optogenetic screening procedures, streamlines experiment design in software, and captures high-quality live-cell measurements.

The exemplary smart experiment has been developed in collaboration with Thomas Drepper, and Seung-Hyun Paik contributing their expertise in PS and strain development, and Matthias Pesch preparing the living cells and the screening experiment.

8.1 Motivation

In the previous chapter, we have presented JediOS that introduced event-driven imaging for high-throughput experiments. We have demonstrated that this platform enables new experiment designs and the investigation of novel phenomena in living cells, marking a paradigm shift from pure observation to influencing the cells or even implementing closed-loop control. These new opportunities come with an increase in throughput, accelerating insight generation.

However, we have demonstrated all these advantages exclusively within in-silico simulation studies with virtual microscopy hardware and simulation models that provide perfect measurements of the investigated cell population models. Executing such experiments in practice on real microscopy hardware with real living cells presents a set of new challenges. We need to gain full software-based control over the microscopy hardware to carry out the imaging requests of JediOS. Moreover, such a smart experiment needs to integrate into the existing experiment routines (Section 1.4.2) and cover all its steps from the RoI selection on the microfluidic chip that contain suitable cell populations to the execution of the experiment.

At the forefront is the complexity of executing such smart experiments. Our goal is to make event-driven high-throughput experiments usable to a broad range of users. First steps have been made to reduce the complexity by the operating system perspective, leading to a divide-and-conquer approach of programming the imaging applications in the EDIL. This high usability and simplicity is also needed in the complete smart experiment routine that executes such experiments on the microscopy hardware using JediOS.

The reproducibility of such experiments is another key goal: users should be able to quickly repeat the same experiment multiple times to collect more data or confirm and reproduce insights. Moreover, all the (smart) decisions that are performed during the experiment or its setup should be documented. This is especially important in view of the current reproducibility crisis in life science [174].

Implementing such smart experiments, we would like to increase the efficiency and effectivity of MLCI. While we have already shown that JediOS improves both by increasing the cell population samples in high-throughput experiments and sampling points of interest with higher imaging rates, this only transfers to practice when we develop a convenient method for conducting these experiments. Last but not least, the autonomous execution of such experiments requires high robustness

in the software implementation, as errors in execution can lead to the loss of all experiment data, wasted time in the lab, and unnecessary occupation of expensive microscopy setups.

With all these requirements in mind, we develop a comprehensive smart MLCI workflow that is completely software-based and covers all steps from the initial RoI selection to the execution of the experiment and the extraction and visualization of real-time insights. To achieve this, we will first introduce the basic software interface to the microscopy hardware and then also implement the RT-IAP to complement the JediOS platform. Then, we will develop an exemplary sMLCI workflow at the application scenario of an optogenetic influence screening and highlight the execution of the in-silico tested state-machines and the tremendous quality of the recorded data.

In the remainder of this chapter, we:

1. complement the JediOS platform with the development of the hardware interface for the real microscopy platform and the RT-IAP,

2. introduce the application study of the high-throughput PS screening ideally suited to demonstrate the benefits of designing smart experiments in JediOS

3. develop a five-step sMLCI experiment workflow, tailor it to the application study, and showcase its ease of conduction and the obtained real-time insights.

While this chapter focuses on the smart conduction of the MLCI experiment and recording high-quality data, the next chapter deals with the in-depth data analysis to reveal new insights into the optogenetic influence.

8.2 Implementing the microscopy platform

In Figure 5.3, we have introduced the multi-platform concept of JediOS, including the in-silico simulation platform and a real microscopy platform. The in-silico platform has been implemented in Section 7.5. Thus, we now aim to implement a real microscopy platform that interfaces with the microscopy setup to capture actual camera images and perform real stage movements.

To perform imaging and stage movement, we require full software-based access to the microscopy platform. This includes three primary actions that need to be carried out during the experiment:

- stage movement: move the stage to 3D-coordinates (x,y,z) in order to image different RoI positions.

- phase-contrast imaging: record a phase-contrast image at the current stage position. The phase-contrast imaging procedure involves activating the phase-contrast illumination, capturing an image with a specific exposure duration, and then deactivating the phase-contrast illumination.

- fluorescence imaging: record a fluorescence image at the current stage position using a specific excitation wavelength. First, the corresponding filter cube is rotated into the light path. Then, the image is recorded while the fluorescence light source is activated with a specific excitation intensity and duration.

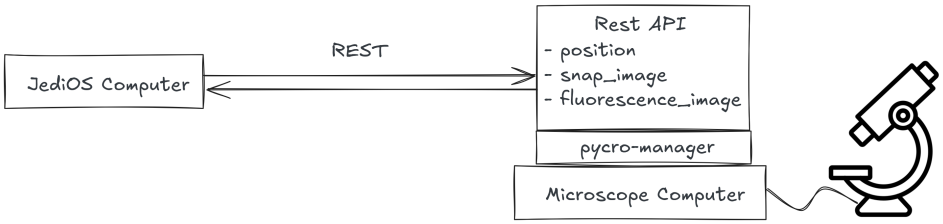

Figure 8.1: **REST API for network based microscopy control.** The microscope computer (right) is wired directly to the microscope setup and hosts the REST API with the three major endpoints. The REST API implementation utilizes Pycro-Manager to interface with the microscopy platform's hardware. The imaging software is performing REST API requests to control the microscope but runs on a different computer (left) that is connected via a local network to the microscope computer.

The implementation of these actions is highly specific to the underlying microscopy platform and requires access to the microscope's drivers and other hardware components. Thus, we utilized Micro-Manager [160] and Pycro-Manager [161] to interface with the microscopy hardware. Micro-manager is an open-source framework for microscopy control, containing a wizard to configure the microscope setup, as well as drivers for many devices. For our microscopy setup, these devices include the NikonTI microscope with its stage, the Andor sCMOS Neo 5.5 camera, the LUMENCOR SOLA SE II Nikon fluorescence light source, and a phase-contrast illumination light source. Once the setup is configured using the Micro-manager wizard, the configuration is exported. We then use Pycro-Manager to load this configuration and access the individual devices and their functionality in Python. The Python interface allows us to implement the three actions from above for our microscopy platform. To separate this very microscope specific implementation from the remainder of the software platform and make it executable on a remote computer located at the microscope setup, we develop a REST API. This REST API is accessible via the local network and implements three endpoints corresponding to the three required actions:

- `position`: sets or gets the 3D-stage position
- `snap_image`: records and returns a phase-contrast image
- `snap_fluorescence_image`: records and returns a fluorescence image

The REST API decouples the specific implementation for the microscope and camera setup from the general functionality. Moreover, the REST API is accessible via network and allows us to separate the microscopy computer from the computer running JediOS platform (Figure 8.1). We implement the REST API using FastAPI (`https://fastapi.tiangolo.com/`), which also provides an automatically generated user interface for convenient manual interaction with the implemented endpoints.

The additional abstraction layer of the REST API also benefits the transfer to other microscope setups. This only requires the implementation of REST API endpoints for the new microscopy platform, assuming that the microscopes have similar imaging capabilities. No changes to the imaging applications are needed.

8.3 Real-time image analysis pipeline (RT-IAP)

The images recorded using the microscopy platform contain the microfluidic structures of the RoI and the developing cell population. However, to make decisions during the experiment based on cell population readouts such as the CC or TSCA, these readouts need to be extracted from the raw recorded images in a fully automated and real-time capable process. In this context, we define real-time processing as "faster than acquisition". In our live-cell microscopy setup, we typically record one image every two seconds. Therefore, we require an efficient and fully automated analysis pipeline that can match this acquisition rate.

In Chapter 6, we have already introduced the DMC for real-time semantic understanding of the microfluidic structures and the precise extraction of the RoI content. We extend this approach by the segmentation of the extracted RoI image and the filtering and extraction of single-cell information using the *acia* library presented in Chapter 4. Figure 8.2A shows the three steps of the proposed Real-time Image Analysis Pipeline. First, the RoI is identified in the raw microscopy image utilizing the semantic understanding provided by the DMC. The resulting cropped RoI image is then segmented using `Omnipose` [30], the identified cell masks are filtered, and biological quantities such as TSCA and CC are extracted using Automated single-cell image analysis (`acia`) (Chapter 4). The execution of this pipeline is intended to happen concurrently with the live-cell imaging experiment. Thus, it needs to run asynchronously to JediOS such that images are continuously recorded while others are concurrently processed and analyzed by the pipeline. Therefore, we decided to implement the image analysis pipeline using dora-rs - a framework optimized for image-based AI robotic applications (`https://dora-rs.ai/`).

The dora-rs framework is a general robotics and AI framework that allows the organization of complex dataflows containing multiple processing steps. Therefore, it allows us to define nodes that receive, process, and send messages. In our case, these messages contain the recorded microscopy images, the segmentation mask, or filtered cell information. Figure 8.2B shows the implementation of the RT-IAP in dora-rs nodes and the incoming and outgoing messages of each node. In addition, we implement the ECN interfacing between the JediOS experiment platform and the real-time analysis pipeline. Thus, the analysis results received by the ECN are redirected as events to the corresponding state-machines identified by the RoI IDs. During the execution of the RT-IAP, dora-rs ensures the efficient transfer and distribution of messages to the respective nodes, as well as the asynchronous data processing within the nodes.

The implementation of the RT-IAP in dora-rs has two key advantages: (1) The implementation of each node is modular with clearly defined message interfaces such that the underlying implementation can be easily exchanged without changing other components of the RT-IAP. For example, using a different segmentation model requires its implementation in a dora-rs node and its integration into the dora-rs dataflow. (2) Every node is executed in a separate process, leading to an asynchronous processing of messages. Thus, while the segmentation node processes an image, the DMC node simultaneously processes the next recorded raw microscopy image. This asynchronous processing is a key to reliable and fast real-time processing. To meet the real-time requirements, the overall processing time can exceed 2 seconds as long as the individual steps of the RT-IAP are performed in less than 2 seconds.

To verify the real-time capability of our implementation, we measured the duration of the individual operations of the RT-IAP. Figure 8.3 shows the asynchronous image processing within the

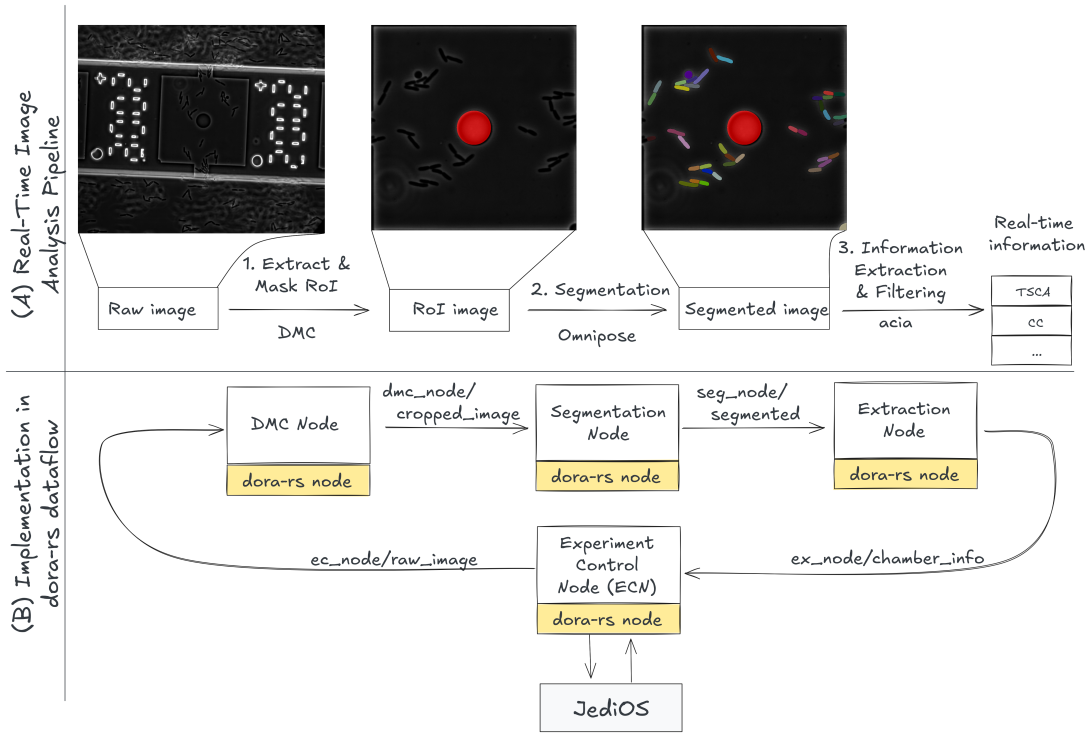

Figure 8.2: **Design and implementation of the real-time image analysis pipeline.** (A) The three sequential steps in the real-time analysis pipeline: (1.) Based on the raw microscopy image, the DMC is used to extract the RoI and mask internal microfluidic structures (red circle), (2.) the cells are segmented (colored masks), and (3.) single-cell information is extracted. (B) Their implementation in the dora-rs framework using nodes (yellow) and the flow of published and received named messages indicated by arrows. The Experiment Control Node (ECN) serves as the interface to JediOS handling raw microscopy images and real-time live-cell information.

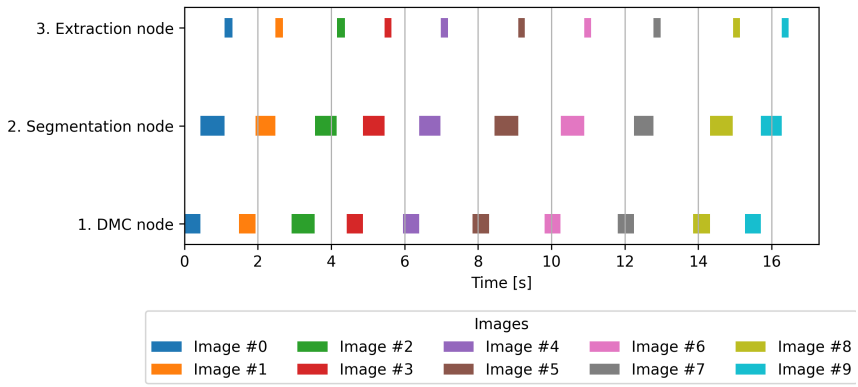

Figure 8.3: **Images flowing through the three nodes of the real-time image analysis pipeline.** Ten exemplary images that transition through the three nodes of the real-time analysis pipeline: (1.) The DMC node that performs the RoI extraction, the (2.) segmentation node that detects and masks individual cells, and (3.) the extraction node that filters cell detections and computes CC and TSCA measures. For each processed image, the timespan spent in each node is depicted as a bar, and each image is assigned a unique color.

RT-IAP in the dora-rs framework. The first image (blue) is first processed by the DMC node, then by the segmentation node, and finally by the data extraction node. The segmentation node has the longest processing time. At the same time, newly recorded images enter the RT-IAP in the DMC node and are processed asynchronously. Figure 8.4 shows the distribution of processing times for each node measured for 1558 images recorded by the microscope and sent through the dora-rs pipeline. All processing times have been measured on a computer equipped with an AMD EPYC 7282 16-Core Processor, and deep learning workloads (DMC, segmentation) have been executed on an NVIDIA A100 (40GB) GPU. The distributions show that no processing step exceeds 2 seconds. Thus, the implemented pipeline is suitable for real-time analysis of recorded microscopy images and the total processing time for each image is consistently below 2 seconds (Figure 8.4D).

8.4 Smart MLCI experiment application: High-throughput photosensitizer screening

We define the smart MLCI experiment as the combination of technical and biological knowledge to effectively and efficiently investigate living cells and their response to influences. For designing such an experiment, it is crucial to understand the biological application scenario and tailor the smart experiment to the specific requirements and needs of the research question. Therefore, we introduce our application case - the screening of photosensitizer influence on the living cells [28, 175]. We briefly present the PS as an optogenetic tool to induce intracellular stress, which impacts the development of living cells. Then, we design imaging applications to effectively screen the effect of PS under various light pulses, with the aim of evaluating their suitability for cybergenetic control applications.

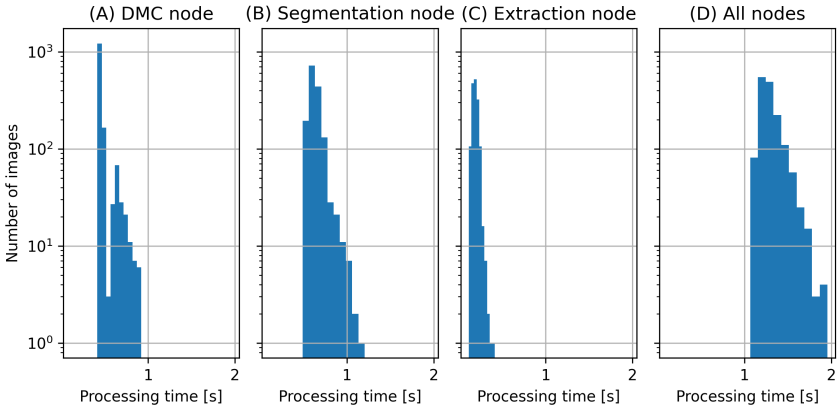

Figure 8.4: **Processing time of the individual real-time pipeline steps.** The processing time distribution measured on 1558 images for the DMC node (A), Segmentation node (B), extraction node (C) and the total processing time (D).

8.4.1 Genetically-encoded photosensitizers: An optogenetic tool for light-based intracellular stress induction.

The term genetically-encoded photosensitizer is an umbrella term for a variety of photoreceptors and light absorbing molecules which release large quantities of Reactive oxygen species (ROS) upon activation with a specific light wavelength [28]. In the following, the Pp1FbFP photosensitizer expressed in a *C. glutamicum* WT strain will serve as the exemplary PS for light-based intracellular stress induction.

The procedure for light-based stress induction is schematically visualized in Figure 8.5. First, a specific expression plasmid containing the Pp1FbFP photosensitizer gene is introduced into the microbial strain. Upon induction with Isopropyl-β-D-1-thiogalactopyranosid to the medium, the cells start the expression of the PSs (1-2). The PSs are then activated using an excitation pulse with a specific light wavelength. This excitation pulse is performed through the objective of the microscope, utilizing its fluorescence imaging capacity, and the excited surface area is roughly constrained to the field of view of the microscopy objective. The energy of the excitation pulse denoted as excitation pulse energy is defined by the excitation power (brightness of the excitation light source) and the excitation duration. The excitation and activation of the PSs leads to the emission of a fluorescence signal recorded by the microscope's camera. The application of an appropriate filter cube ensures that the correct excitation and emission wavelengths are used and recorded - a standard procedure in fluorescence microscopy. Upon activation of the PSs, they release ROS, causing oxidative stress inside the living cells (4) that should lead to intracellular cell damage and influence the cell development [176].

8.4.2 Screening photosensitizer sensitivity to dynamic light activation

The goal is to design an sMLCI experiment that systematically screens different excitation pulse energies and their effect on the development of the cell population and individual cells. Moreover, we want to investigate the heterogeneity of the response on the single-cell level and are specifically

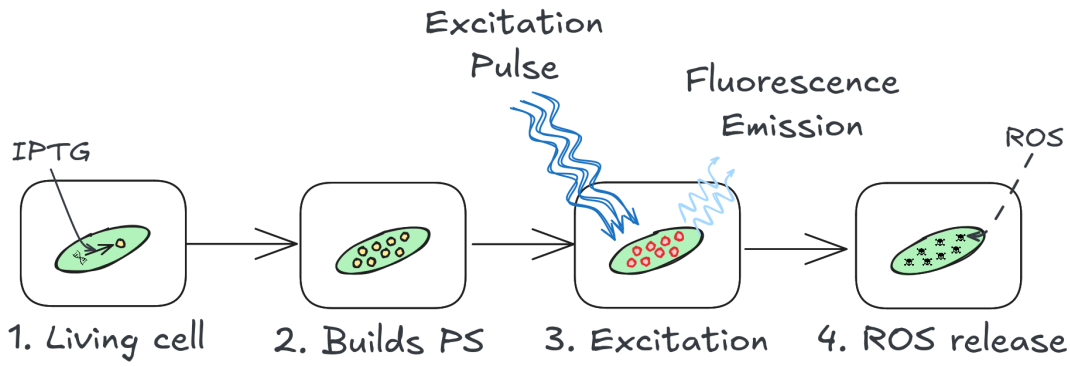

Figure 8.5: **Light-based intracellular stress induction using photosensitzers.** (1.-2.) The living *C. glutamicum* cell produces PSs after induction with Isopropyl-β-D-1-thiogalactopyranosid (IPTG). (3.) The PSs are activated using an EP with a specific light wavelength and energy. The activation of the PSs leads to the emission of a fluorescence signal that is recorded by the microscope's camera and (4.) to the release of ROS causing oxidative stress inside the cell. The response to this stress induction is observed through changes in cell development, such as cell growth, cell division, or cell death.

interested in the cell response in the immediate aftermath of such excitation pulses. Thus, we derive three key requirements for the design of event-driven imaging application:

- the development of the cell populations should be recorded with low imaging intervals to enable single-cell measurements based on cell tracking,
- the excitation pulse should happen at a specific size of the cell population,
- the immediate response of the cells should be recorded at a higher sampling rate (lower imaging interval).

Based on these requirements, we designed the state-machine for the imaging application of a single cell population. The SM3 design introduced in Section 7.3.3 already meets these requirements and has demonstrated to unlock high throughput experimentation, making it ideally suited for screening. For this experiment, we perform a slight adaptation to further reduce the required imaging resources per cell population. The resulting state-machine SM3* is illustrated in Figure 8.6. The adaptation consists of splitting the immediate aftermath observation after the EP into two parts represented by states B_1 and B_2: In B_1, SM3* observes the development of the cell population for 10 minutes with a very short imaging interval of 30 seconds and then moves to state B_2 for 10 minutes observing with an imaging interval of 1 minute.

The goal of the high-throughput screening experiment is to execute multiple SM3* state-machines concurrently, but with different excitation intensity and duration parameters, to apply different excitation pulse energys (EPEs) to the cell populations. Moreover, we would like to add a positive and negative control group to verify the impact of strong and no-stress induction. The cell populations in the positive control group are excited with high light energy, performed with high excitation intensity and duration. The negative control group is not excited with any light pulse. Thus, we use

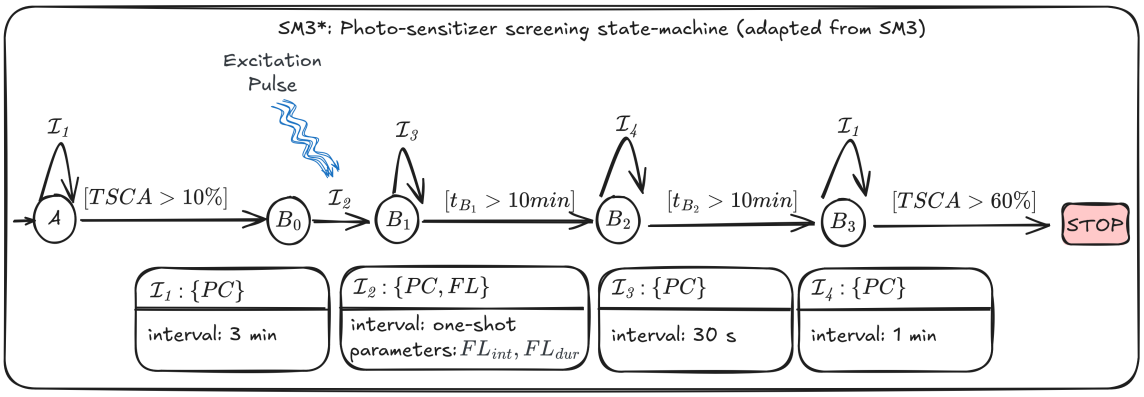

Figure 8.6: **State-machine SM3⋆ design for high-throughput excitation pulse screening.** The SM3⋆ is an adaption of the SM3 introduced in Section 7.3.3 and splits the post-excitation pulse observation into two states with two different imaging intervals of 30 seconds and 1 minute (B_1 and B_2). The EP is performed with a specific excitation pulse intensity (FL_{int}) and duration (FL_{dur}) which are parameters of the state-machine. After the observation period with lower imaging intervals, SM3⋆ returns to the initial imaging interval (state B_3) and stops the imaging application as soon as 60% of the RoI area is covered with cells.

SM1 to observe cell development and verify that the cell populations exhibit the expected growth behavior.

In total, we need to run three different types of imaging applications: the negative control group (SM1), the single-shot SM3⋆ with different parameters, and the positive control group (SM3⋆).

8.4.3 Preventing light scattering and cross-illumination

The light application through the microscope's objective leads to a local influence on each RoI position; however, it may also scatter and potentially illuminate cell populations in the direct neighborhood. We reduce the chance of affecting neighboring cell populations by utilizing the microfluidic chip map. The neighboring cell populations of each observed cell population should not be used in the screening experiment. Thus, we propose to leave "gaps" on the microfluidic chip map to make sure that all EPs influence exactly one monitored cell population.

8.5 Smart MLCI experimentation workflow leveraging real-time information and event-driven imaging

To conveniently conduct such event-driven imaging experiments with hundreds of imaging applications, we now develop a sMLCI workflow for the optogenetic excitation pulse screening experiment. Therefore, we implement all steps from selecting appropriate cell populations to executing the experiment in software and utilize real-time understanding whenever possible to make smart decisions during the experiment.

8.5.1 The five steps of smart MLCI experimentation

To implement sMLCI experiments in practice, we propose a general experimentation workflow composed of five steps:

1. **Map creation:** Create a map containing the microscopy stage coordinates of all RoI positions on the microfluidic chip (see Chapter 6). Using the DMC mapping procedure, this step takes roughly five minutes.
2. **Initial screening:** Perform an initial screening of all RoI positions on the microfluidic chip.
3. **Suitable RoI selection:** During the screening, real-time information is gathered, processed, and visualized in a dashboard. Rule-based criteria are applied to identify suitable cell populations across the many RoI positions.
4. **The event-driven experiment:** The state-machines are associated with suitable RoIs and JediOS is used to conduct the event-driven imaging experiment.
5. **Live-view of the experiment:** In order to supervise the experiment's progress, a live-view dashboard gives insights into the cell population development and the technical information about the experiment's progress.

For each step, we implement a Jupyter Notebook, providing a good combination of documentation and visualization while providing enough flexibility to adapt and customize the procedure to the underlying MLCI experiment. In the following, we present each step in more detail and highlight the adaptions made for the PS screening example.

8.5.2 Step 1 - Map creation

To create a map of the microfluidic chip that contains all the RoI positions in microscopy stage coordinates, we use a DMC and follow the procedure presented in Section 6.5.2. We implement the procedure in a Jupyter Notebook where we first select the underlying microfluidic blueprint design, record three RoI positions, and then estimate the map. Figure 8.7 shows the transformation of RoI positions from the blueprint coordinate system (left) to the microscopy coordinate system (right). Each cross denotes the position of a RoI (1280 in total), and the three manually approached RoI positions are highlighted in orange, green, and red in both coordinate systems. The mapping procedure itself takes roughly five minutes. The resulting map is then exported and builds the foundation for automatically approaching and recording images of the developing cell population within a smart experiment.

8.5.3 Step 2 - Initial screening

The goal of the initial screening procedure is to gain an overview of the developing cell populations in each RoI of the microfluidic high-throughput chip. To achieve this, we implement a simple imaging routine that records phase-contrast images and utilizes the real-time pipeline to analyze the cell population development.

The implemented imaging routine utilizes the computed map to approach the RoI positions one by one, records a phase-contrast image using the REST API, and performs real-time analysis using

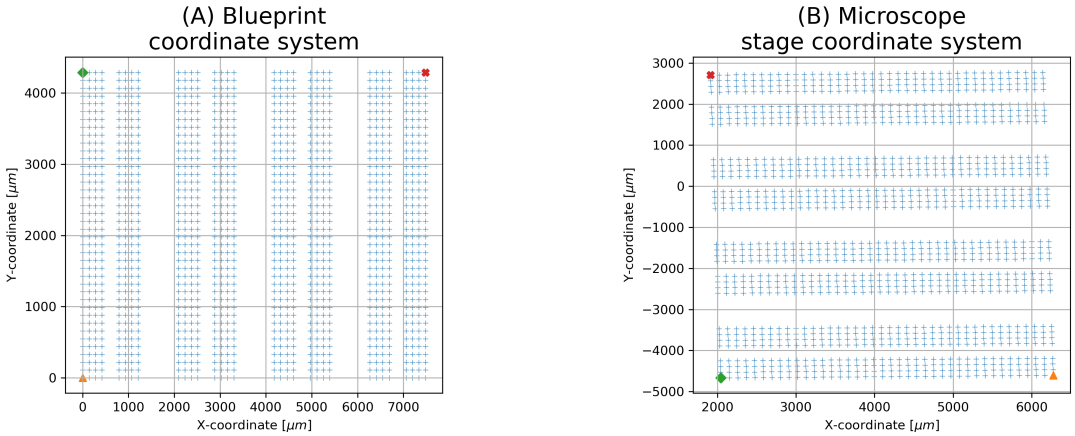

Figure 8.7: **High-throughput microfluidic chip map creation.** (A) shows a map of all RoI positions in the microfluidic blueprint coordinate system. (B) shows the corresponding map of all RoI positions in the microscope state coordinate system. The three manually approached RoI positions are highlighted as an orange triangle, green diamond, and red square. Based on these three RoI positions, the DMC software computes an affine transformation from the blueprint coordinate system to the microscope stage coordinate system.

RT-IAP. The results of the real-time insights contain segmentation masks and information about the CC and TSCA. The imaging of the RoI positions is continuously repeated until the user stops the initial screening procedure.

In general, the initial screening procedure can be executed on all RoI positions of the map (here: 1280) or on a specific subset. In our experiment example, we utilize only a subset of the available RoIs (Figure 8.8). There are two primary reasons for this: First, a lower number of RoIs allows for faster recording of multiple images of each RoI and reduces manual setup time to connect and inoculate all microfluidic channels. Second, in the PS screening, we apply EPs to trigger a stress response of cells within a population. While the EP is applied through the microscope's objective and constrained by the field of view, there exists the danger of cross-illuminating close by RoIs. Thus, we propose leaving gaps between observed cell populations. In practice, we achieve this by applying a checkerboard pattern to the microfluidic chip map. Figure 8.8 shows the resulting microfluidic chip map, where monitored RoI positions during the initial screening are denoted as "Observed". In fact, during the initial screening procedure, we used only one-eighth (160 out of 1280 RoIs) of the microfluidic chip's capacity.

8.5.4 Step 3 - Initial screening dashboard and suitable RoI selection

During the initial screening procedure of the smart MLCI experiment, images of the cell populations in the RoIs are continuously recorded, and the real-time pipeline analyzes and quantifies their development. To gain instant insights into their development, we develop a dashboard that combines information about the chip layout with the biological readouts. Figure 8.9 displays the dashboard, which visualizes four key readouts projected onto the microfluidic chip layout (A-D) or as distributions (E-H). These readouts include the TSCA, IGR, maximal cell size, and filling of each RoI as a

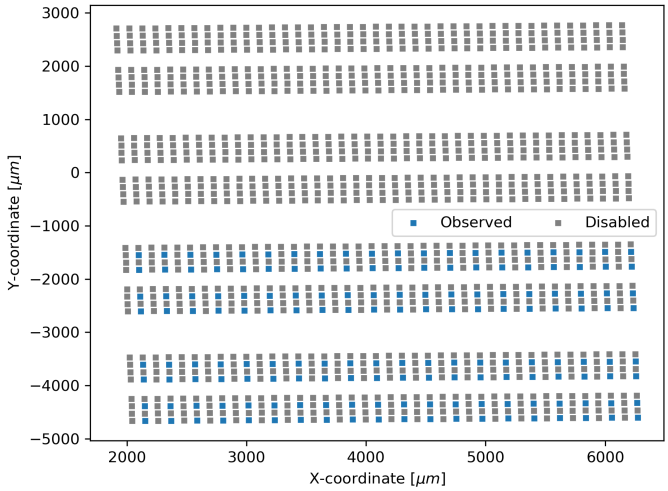

Figure 8.8: **Checkerboard pattern on the microfluidic chip map.** Every square denotes a RoI position. Only a subset of RoI positions is actively monitored (blue square) during the initial screening period. The upper two channels (RoI positions above the legend) are not used to reduce the number of necessary microfluidic pumps. Between every observed chamber (blue square), there is at least one disabled chamber (gray square) that is not observed. This checkerboard pattern is a precaution to prevent the influence of excitation pulses on neighboring chambers.

percentage of its overall area. These four readouts provide a strong foundation to identify suitable RoIs using objective criteria.

Interestingly, the top row of RoIs within the microfluidic map shows missing RoI positions. These are the positions where real-time analysis is not able to extract data, usually due to structural damage to the RoI that prevents automated RoI extraction by the DMC pre-processing pipeline. Moreover, we can observe in Figure 8.9 A, E, D, and H that the cell populations in the RoIs differ in their size. While some populations already contain many cells and exhibit a large TSCA, others are smaller, and still others are barely existent, with a size close to 1 μm^2. Extremely small population sizes indicate empty RoIs where probably just impurities are detected. The IGR (Figure 8.9B, F) gives an insight into the instantaneous growth of the cell population. It is based on the average IGR since the start of the initial screening procedure and, therefore, indicates whether the cell population shows an expected growth rate behavior or deviates from that. Deviations from the expected growth rate either indicate abnormal biological behavior or impurities. These impurities may be detected as cells by the segmentation but do not grow over time, leading to a lower than expected IGR. Moreover, the maximum cell size in Figure 8.9C serves as a biological and technical indicator. Extremely large cells may be an indicator of morphological changes that are either interesting or discouraging for observation or hint at problems with under-segmentation, where multiple cells are detected as one large cell.

Projecting these measures onto the microfluidic chip layout enables the detection of differences in the individual microfluidic channels supplied by independent microfluidic pumps. Failures of the pump or blockage of the medium channel should result in a systemically lower IGR in all chambers of the microfluidic channel. This allows us to detect and correct technical issues already during the

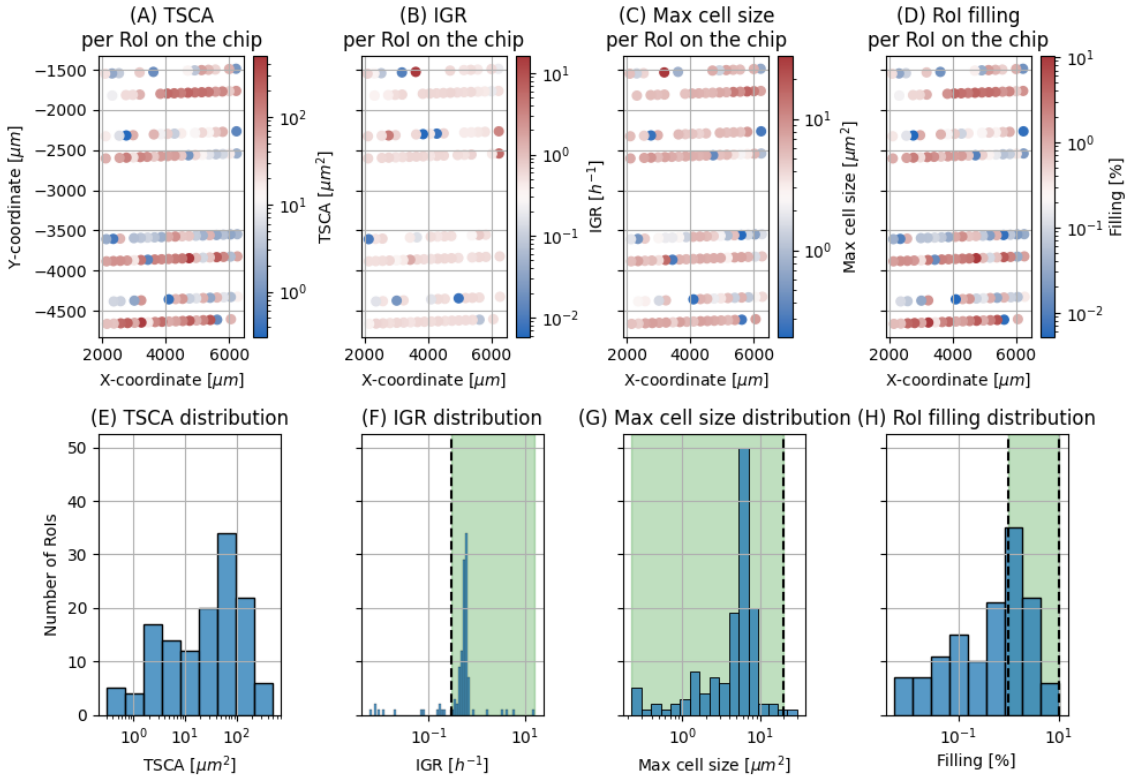

Figure 8.9: **Real-time dashboard during the initial screening procedure (t=47min 23s).** The dashboard shows four real-time measurements of TSCA, IGR, maximum cell size, and relative RoI filling, projected on the microfluidic chip map (A-D, each circle denotes a RoI position) and as a distribution (E-H). The dashed black lines indicate filtering boundaries set by the user, and the green areas in the distribution show RoI measurements within the filter boundaries. The dashboard is updated every 10 seconds and is shown here after 47 minutes of initial screening. Missing circles in A-D indicate that the RT-IAP could not process the recorded images, most likely due to damaged reference markers.

initial screening period.

Apart from the visualization, the real-time information is used to impose a set of rules or constraints that allow the experiment to focus on biologically valuable cell populations. There exist general rules that are valid for all MLCI experiments, as well as quite specific rules that arise from the specific experiment setup and research question. In general, RoIs containing damaged structures, impurities, no cells at all, or cells that do not develop have limited biological use and should be discarded. These cases can be detected by a complete failure of the real-time pipeline (structural damage is so severe that the DMC cannot identify the RoI), too small cell populations (empty RoI), too large maximal cell size (impurities), or low IGR (no development).

Added to these general rules, we can develop rules that are tailored to the specific experiment. For example, some experiments may focus on the initial composition of a cell population, starting with a single cell to investigate heterogeneity in isogenic cell populations [165]. Then the CC measure (not shown in the dashboard) can be used to select specific RoIs with a single cell. Another approach is to only observe cell populations within a certain IGR range, as the expected growth rate of the cell populations under the specific cultivation conditions is often known before the experiment. Abnormal behavior may then indicate technical issues or damage to the cells.

Additionally, we measure the standard deviation of the IGR measurements to quantify their fluctuation (not shown). A high standard deviation is most likely caused by deteriorated image quality or inconsistencies in real-time segmentation, for example, due to impurities or focus problems.

Based on these considerations, we developed a specific set of rules for selecting RoIs for the PS screening experiment shown in Equations (8.1) to (8.4). We term RoIs suitable if their real-time information fulfills all these criteria. In this experiment, we select RoIs where the cell population covers 1% to 10% of the RoI area, the average IGR is larger than 0.3 h^{-1}, the standard deviation of the IGR is below 0.4, and the maximal cell size is below 20 μm^2:

$$1\% < TSCA < 10\% \tag{8.1}$$

$$\overline{IGR} > 0.3\ h^{-1} \tag{8.2}$$

$$\sigma(IGR) < 0.4 \tag{8.3}$$

$$\text{max cell size} < 20\ \mu m^2 \tag{8.4}$$

Due to the implementation within a Jupyter Notebook, these rules can be altered or complemented during the experiment's initial screening period. For example, relaxing the rules leads to a larger number of RoIs termed suitable for the experiment. Figure 8.10 shows the final selection of RoIs suitable for the PS screening experiment.

8.5.5 Step 4 - Conducting the event-driven screening experiment

The set of suitable RoIs is imported into the fourth Jupyter Notebook containing the experiment logic. This notebook imports the map of suitable RoIs. Moreover, the notebook defines all the state-machine designs and imaging configurations used during the experiment. We have already defined the three groups of imaging applications in Section 8.4.2. Once these state-machines are defined, they are assigned to specific RoI positions on the microfluidic chip to build the imaging applications. In our example, we would like to execute 6 replicates of the SM1 that serve as a negative control

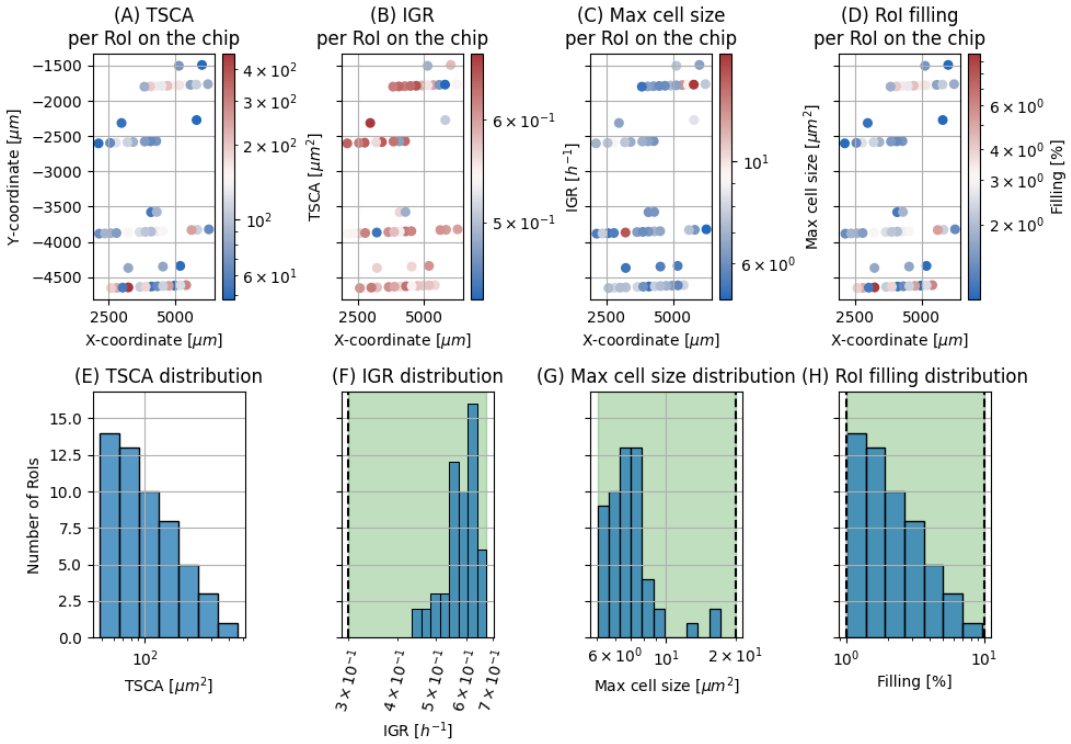

Figure 8.10: **Dashboard of the selected RoI candidates (t=47min 23s).** The same dashboard visualization as before with the RoI information projected on the microfluidic chip layout (A-D) and distributions (E-H). Specific RoI candidates have been selected such that their metrics are within the criteria boundaries (green) area. The dashboard provides a precise overview of where the selected RoIs reside on the microfluidic chip (A-D) and how their metrics are distributed (E-H).

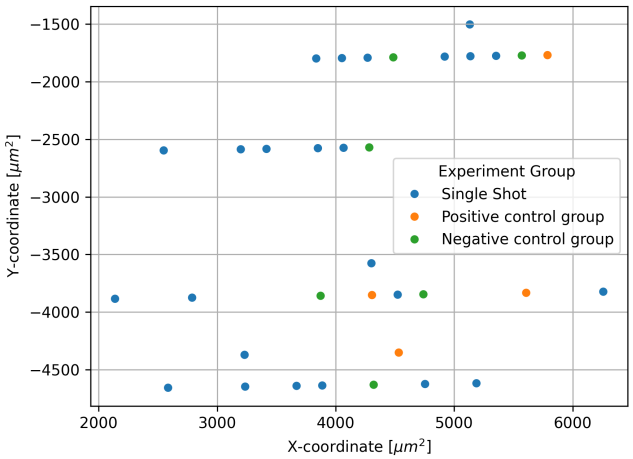

Figure 8.11: **Assignment of imaging applications to RoI positions on the microfluidic chip map.** In this experiment, we have three groups of imaging applications: the negative control group (SM1), the positive control group (SM3⋆), and the regular group denoted as "Single Shot," which executes SM3⋆ with specific parameters for the excitation pulse. These imaging applications are assigned to RoI positions on the microfluidic chip.

group (no EP), 4 replicates of the SM3⋆ per EPE, and 4 replicates of SM3⋆ that serve as a positive control group (very high EPE).

Table 8.1 shows a table of the different parameters used for SM3⋆ and the total number of imaging applications that are executed in this experiment. In total, this experiment consists of 34 imaging application, each monitoring or influencing the cell population in their associated RoI on the microfluidic chip. Figure 8.11 shows the random assignment of negative control group (SM1), positive control group (SM3⋆), and regular replicates denoted as "Single Shot" (SM3⋆) to RoI positions on the microfluidic chip map.

The set of imaging applications is passed to the scheduler of the JediOS platform. The scheduler starts the execution and coordinates the imaging procedure, data storing, and processing in the real-time analysis pipeline. The resulting real-time information is received by the scheduler and redirected to the individual imaging applications, potentially triggering events in the underlying state-machines.

8.5.6 Step 5 - Live-view of the experiment

After starting the experiment loop, the scheduler of JediOS handles the concurrent execution of all imaging applications, requiring no further user interaction. To supervise the experiment's execution and gain insights into the current development of the cell population, we can again benefit from real-time image processing and visualize the development in dashboards. As the visualized information ranges from general to very specific for this sMLCI experiment, we implement these dashboards in a fifth Jupyter notebook to make them accessible while also being easy to customize, adapt, and extend.

Table 8.1: **Screening parameters and number of monitored RoIs.** We screen two parameter sets for intensities and durations with a replicate number of 4, resulting in 24 parameterized imaging applications. Together with 4 replicates of the positive control group and 6 replicates of the negative control group, a total of 34 imaging applications are used.

Experiment parameter	**Value or value set**
Excitation intensities	$\{0.05, 0.1\}$
Excitation durations	$\{0.1, 0.3, 0.5\}$ s
Number of replicates per parameter set	4
Number of parameterized imaging applications	$2 \cdot 3 \cdot 4 = 24$
Positive control group imaging applications	4
Negative control group imaging applications	6
Total number of imaging applications	$24 + 4 + 6 = 34$

At first, we utilize the same dashboard as during the initial screening to gain real-time insights into the biological development of the cell populations projected onto the microfluidic chip or as distributions. Moreover, we develop a second dashboard, shown in Figure 8.12, that provides an overview of the temporal development of all cell populations across different experiment groups and EP parameters. Each row (A-H) denotes a set of replicate cell populations that belong to a specific experiment group and have a specific set of EP parameters. Thus, these cell populations are monitored and influenced by the same state-machine. The lines in each chart show the temporal TSCA development of a cell population, while the black vertical bars denote the recording of an image. The dashed gray line shows the 10% RoI filling area threshold. This real-time dashboard is updated in real-time as the experiment progresses, providing a detailed overview of the experiment's temporal progression.

Specifically, we can observe the asynchronous trigger of the EP changing the imaging rate (and thus the density of vertical black bars) and impacting the cell population development. (A) shows the negative control group without any EP and a constant imaging rate (3 minutes) leading to equidistant image sampling (black bars). (B) shows the positive control group that uses SM3* and a strong EP to verify that the PS are expressed and their strong activation leads to cell death. In the temporal development of (C-H), we can nicely observe the event-driven imaging in action. As soon as the TSCA exceeds 10 % RoI filling (gray dashed line), the EP is performed, and the imaging rate is increased, leading to a better sampling of the immediate cell response. For long-term observation, the imaging applications reduce the imaging rate again.

The vertical bars denoting the imaging sampling time points also reveal gaps, for example, in C (RoI ID: 0228, red) or F (RoI ID: 3210, red). These gaps are a result of failed real-time analysis, most likely due to deteriorated image quality or missing marker detections. Nevertheless, for the majority of the recorded images, the real-time analysis works robustly across the full experiment time.

As expected, the EPs and imaging bursts happen at different points of time in the experiment and, therefore, exploit the effect of asynchronous execution (Section 7.5). To better investigate the impact on the utilization of the microscope, we developed a third dashboard shown in Figure 8.13. It

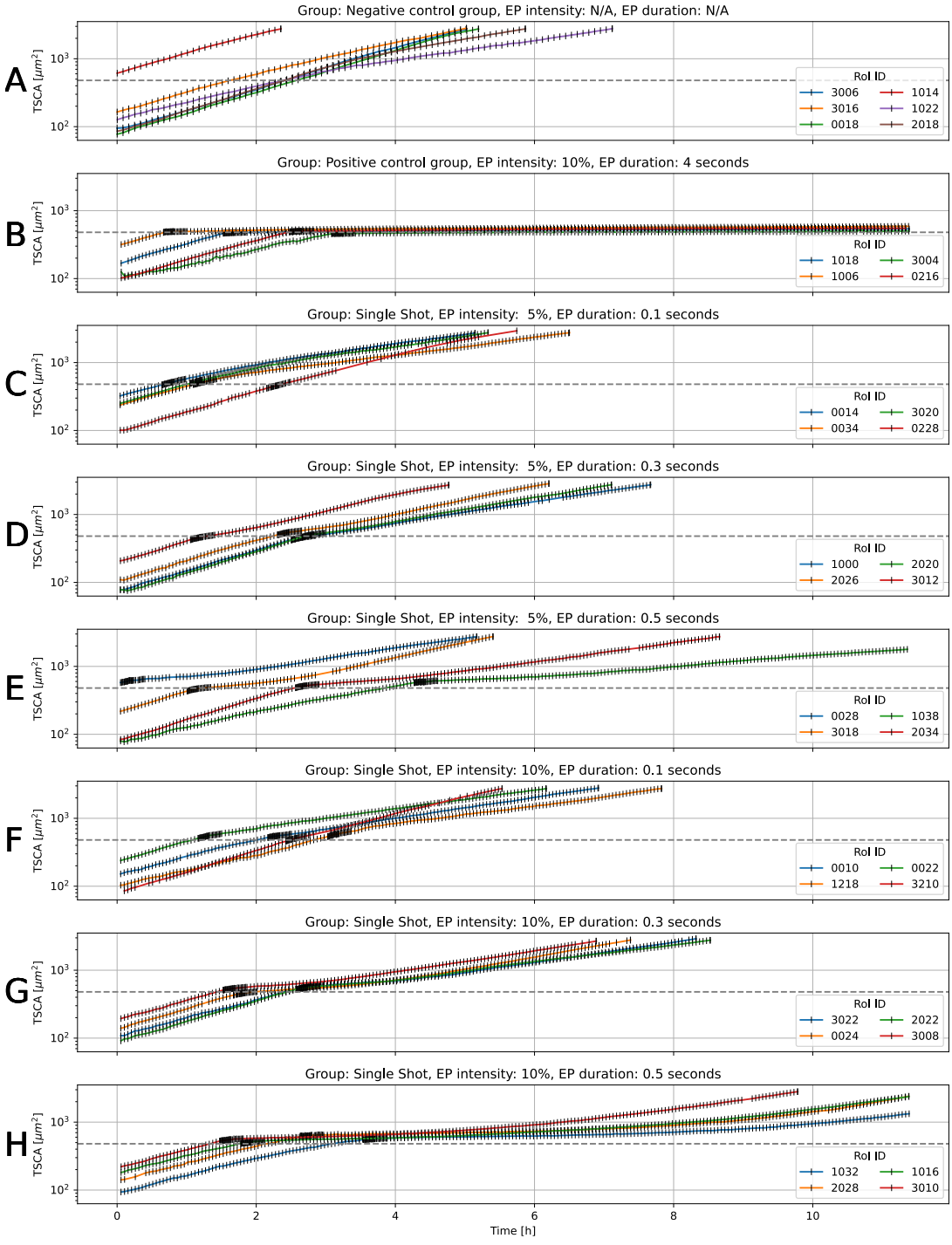

Figure 8.12: **Live dashboard of the temporal development of all cell populations in the PS screening experiment.** (A-H) show the temporal TSCA development of all cell populations, where each letter denotes a group of cell populations with the same excitation parameters applied. The temporal development of each replicate of such a group is shown in a time-series colored by their RoI ID. (A) shows the TSCA development of the six cell populations of the negative control group recorded using state-machine SM1. The vertical bars denote the time points of imaging. (B-H) show the experiment groups recorded with SM3⋆ applying a specific set of EP parameters when exceeding a cell population threshold (dashed gray line). The vertical black bars denote the recording time points of an image and TSCA real-time readouts. The high density of vertical bars denotes the short-term imaging with a higher rate that is triggered after the EP (see SM3⋆).

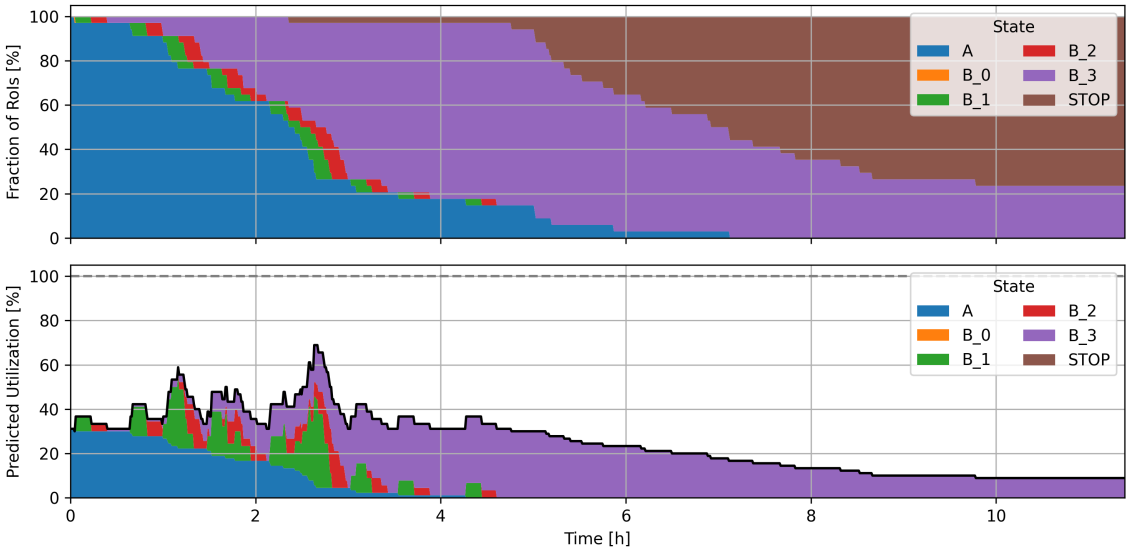

Figure 8.13: **Live dashboard showing the "task manager" of the JediOS platform during the experiment.** The two charts show the temporal distribution of RoIs across the different states of the state-machines (top) and the predicted utilization of the microscopy hardware (bottom). The contribution of the imaging routines used in the different states are color coded. Within different states images are recorded at different intervals. Initially, images are recorded every 3 minutes (A). When the cell area threshold is reached (B_0), the excitation pulse is executed and the imaging interval of 30 seconds is used for 10 minutes (B_1), then increased to one minute for 10 minutes (B_2), and finally adjusted back to the initial interval of 3 minutes (B_3). We term the PU visualization the "task manager" of the experiment as it shows the microscopy utilization, spikes due to imaging rate changes, and reveals potential over-utilization problems during the conduction of the experiment.

is very similar to the visualizations show in our simulation studies (Section 7.5) but now is updated in real-time. As it gives primarily technical insights into the execution of the experiment by JediOS, we term this visualization the "task manager".

Figure 8.13A shows the temporal distribution of active state-machine states for all running imaging applications. Specifically interesting is the trigger of the EP happening on the transition from state B_0 to B_1. As the state-machines are in B_0 only for a very short time, this state is not visible in the chart. Thus, the transition from A to B_1 shown in blue and green means that the state-machine has performed the EP and now increased the imaging rate. This higher imaging rates also observed in Figure 8.13B that shows the temporal development of the Predicted Utilization of the microscope. State-machines entering state B_1 lead to a spike in microscopy utilization due to the increase in the imaging rate. However, due to the asynchronous execution, the microscope's maximal utilization stays below 70 % throughout the full experiment. This is very similar to our simulation study with SM3 with 30 concurrently observed RoIs and shows that the in-silico simulations give a good understanding of the microscopy utilization (Section 7.6.3).

Towards the end of the experiment, more and more cell populations exceed the 60% TSCA threshold such that their imaging applications terminate leading to a strong decrease in total microscope

utilization.

In combination, all three dashboards give the experimenter a comprehensive overview of the execution of the experiment and allow to visualize the progression of the event-driven imaging applications. In particular, the user can compare the temporal development of the cell populations with the expected development and can react to abnormalities. Due to the implementation in Jupyter Notebooks these dashboards can also be adapted and customized on the fly, for example, to introduce another important metric or further investigate a specific phenomena while the experiment is conducted.

The temporal development of the cell populations and the influence of the EP of course raise the question whether we can quantify and compare the influence across various EPEs. The detailed biological analysis of this nouveau PS screening experiment is performed offline and presented in the next chapter (Chapter 9).

8.6 Discussion and Outlook

In this chapter, we have completed the implementation of the JediOS platform with an interface to the real microscopy platform and the RT-IAP. Moreover, we developed the five-step workflow for smart MLCI experiments and presented the conduction of such a smart experiment at the example of a high-throughput PS screening. Herein, we automated the mapping, RoI selection, and applied parameterized imaging applications across the RoIs of the microfluidic chip. With this exemplary experiment, we demonstrated the usability and benefit of smart MLCI experiments in practice, showed the tailoring of state-machines to the biological research hypothesis, and underlined how the workflow simplifies such experiments in the future providing high accessibility and reproducibility.

While the simulation studies in Chapter 7 demonstrated significant advantages in throughput and the ability to interact with developing cell populations, these benefits can only be fully tested in practice by having complete software control over the microscopy platform. Thus, we developed the REST API for the three basic imaging actions: movement, phase-contrast, and fluorescence imaging using Micro-Manager and Pycro-Manager for the hardware interfacing. The encapsulation of this functionality in the REST API allows for sending commands via the network and disentangling the JediOS experiment platform from the microscopy computer. Thus, deep-learning functionality can be executed on a remote server with powerful computing hardware while the imaging is offloaded to the computer next to the microscope.

With the implementation of the RT-IAP, we have shown that our previous development of the DMC platform (Chapter 6) in combination with segmentation and cell information extraction (Chapter 4) is able to extract the cell population development in real-time. The modular pipeline serves as the foundation for processing images, extracting data in real-time, and utilizing measures of cell population development in events for imaging applications. As the RT-IAP is a separate and independent module of JediOS, it can also be used stand-alone, for example, in the initial screening procedure.

With the developed five-step workflow for sMLCI, we laid the foundation for developing smart experiments while using high-throughput and event-driven imaging capabilities. We have demonstrated that the workflow encompasses all the necessary steps during the execution of an MLCI experiment, from initial mapping to real-time insights into the running experiments. Moreover, the workflow allows the execution of the very same imaging applications tested during simulation studies.

Moreover, the workflow offers high degrees of automation while allowing the user to intervene at any point. Their implementation in Jupyter Notebooks allows the combination of extensive documentation, visualization, and customization for future experiments. Casting such an sMLCI experiment in software at the same time documents all the decisions during the experiment. With this smart sMLCI platforms, future experiments become convenient to use and exploit the high-throughput of microfluidic chips. Moreover, the repetition of the experiment or reproduction of generated data essentially involves the simple re-execution of the notebook. Thus, we also view the platform as a contribution to the fight against the current reproducibility crisis in life science [174].

With this sMLCI workflow, we have achieved a tremendous increase in efficiency regarding both in terms of the amount of valuable biological information and the manual labor time of the experimenter. While designing the state-machines requires additional time for experiment preparation, once they are designed, they can be executed repeatedly across different experiments. Thus, the amount of manual labor spent during these experiments primarily involves the preparation of the microscopy platform and the inoculation of the microbial cells. Afterwards, steps 1-3 require minimal interaction, and the final experiment (step 4) is executed fully autonomously. Thus, the manual labor time spent at the microscopy platform is roughly reduced to 30 minutes by this workflow.

Moreover, during the conduct of the PS experiment, we observed that the smart selection of RoI positions yields high-quality time-series data. In particular, every selected RoI provided valuable experimental data. Moreover, the parameterization of the imaging application and the successful triggering of EPs, as well as the adaptation of the imaging applications, leads to an efficient PS screening within a single experiment. However, the increase in efficiency is difficult to quantify, as previous imaging routines did not even allow triggering the EPs based on cell development.

8.6.1 Limitations

Along with these advantages, smart MLCI experiments also introduce a few new limitations and pitfalls, which we would like to highlight briefly.

We have introduced the smart RoI selection as a systematic tool to automatically identify the most suitable cell populations to be observed during the experiment. While this is necessary to study specific phenomena under well-controlled conditions, it bears the potential for introducing a bias into the observation. We systematically restrict the set of selected cell populations. Thus, formulating these rules for the suitability of RoIs needs to consider whether it influences the research hypothesis and pre-determines the outcome of the experiment. Interestingly, this is even an issue in non-"smart" experiments. Herein, the selection of RoIs is performed manually by the user. Thus, the advantage of our system is that all the rules and selections are formalized and documented in the Jupyter Notebook, while users may perform selections based on their "gut feeling". Moreover, it is still possible to uniformly sample the RoI positions to not introduce any bias into their selection or define control groups that are not selected based on suitability. However, this may lead to many empty RoIs and a reduction in valuable biological information. Thus, there is always a trade-off between smart selection, which improves the data quality, and smart selection, which introduces a bias that affects the research hypothesis.

Another critical bottleneck is the quality of the real-time processing and extracted information. During the sMLCI experiment, we make numerous decisions based on real-time information, ranging from RoI selection to event-driven decisions in imaging applications. Therefore, high image and cell

segmentation quality are inevitable. Lower quality, for example, caused by focus loss, may lead to incorrect real-time information or misleading analysis of the images. These errors then directly influence decisions during the smart experiment, potentially leading to erroneous decisions. We have observed during our exemplary study that the methods we have developed are robust enough to perform real-time analysis. However, further developments and benchmarks, such as those presented in Chapter 3, are needed to verify and improve the robustness of all RT-IAP pipeline steps, especially for applications involving new cell morphologies or imaging modalities. New datasets and training of deep learning-based segmentation models should be sufficient to extend high-quality segmentation information to these new imaging modalities and cell morphologies. The modular implementation in the dora-rs pipeline then allows the implementation of these models in nodes and makes them available in the RT-IAP.

Conversely, real-time information may also be used to counteract the deterioration in imaging quality immediately. Suppose the DMC approach suddenly cannot identify any markers at the RoI position or the extracted cell segmentation information changes unexpectedly. In that case, this is likely caused by deteriorated image quality (or another severe event). Noticing this in real-time allows for the employment of automated measures to regain focus or to identify the user who needs to perform a manual refocusing intervention, thereby preventing the loss of valuable information collected during the experiment.

8.6.2 Outlook

The integration of the real microscopy platform into JediOS and the workflow for sMLCI experimentation bears immense potential and flexibility for upcoming high-throughput studies in MLCI experiments.

The smart selection of suitable RoI positions has been primarily used in this experiment to improve the recorded data quality, for example, rejecting empty RoIs or making sure they show regular growth behavior. In the research domain of co-cultures, however, this technique can also be used to investigate specific biological scenarios. In these co-cultures, multiple microbial strains are cultivated in one RoI to investigate their mutual interaction [26]. Labeling them with fluorescence markers allows us to distinguish cells belonging to the individual strains based on fluorescence information (Section 4.3.2). One parameter of interest that we cannot control is the initial ratio of the cell strains in each RoI [26]. For example, with two strains, initial ratios of 1:2, 1:5, or 4:3 may be of specific interest. However, the inoculation procedure leads to a random filling of the RoIs and no control on these ratios. The initial screening procedure and smart selection of RoIs allows us to search the full microfluidic chip for RoIs containing suitable ratios and select them for observation in the experiment. Thus, the smart decision at the beginning of the experiment can be crucial for effectively gaining insights into specific research questions, such as the influence of the initial ratios on the co-culture development.

Another key advantage of sMLCI experimentation is the flexible assignment and execution of individual event-driven imaging applications to each RoI position on the microfluidic chip. We have demonstrated that JediOS is capable of executing imaging applications with various parameters while also allowing for the mixing of completely different underlying state-machine designs. Thus, in one experiment, multiple state-machine designs can be executed, either to evaluate which design yields the most biologically valuable insights or by combining and investigating multiple biological

research questions within a single experiment. The smartness of assigning imaging applications to the RoIs has ensured that all selected cell populations yield valuable results, resulting in a massive increase in efficiency. In addition, we could also distribute replicates across different microfluidic channels so that in the case of a pump failure, channel blockage, or local structural chip failure, it only affects a subset of the replicates, and the experiment still yields usable recorded data.

Combining multiple experiments into one screening bears the opportunity to accelerate the experiment throughput and knowledge generation. For example, using different media across the microfluidic channels or different microbial strains allows for the utilization of various imaging applications to observe cell development without exceeding the microscope's imaging resources. The state-machine designs can then be "smartly" allocated to the corresponding RoI positions with the corresponding medium supply or microbial strain. Combining multiple experiments is especially useful if the cell populations show different growth rates depending on their medium or strain, potentially requiring different imaging rates to maximize the throughput of a single experiment.

The impact of this new smart MLCI experiment workflow, however, depends on the broad adaptation across the live-cell imaging community. This adaptation is primarily depending on the ease of transferring experiments to other microscopy platforms. We have designed JediOS explicitly with the idea in mind to execute the experiment on different platforms, resulting in an in-silico platform and the real microscopy platform presented in this chapter. Therefore, the only hardware-specific implementation in the software platform is the image requests that interface with the underlying microscopy platform or in-silico simulation to record phase-contrast or fluorescence images. We encapsulated the interfacing with the microscopy hardware in a REST API. Thus, in the simplest case of another microscopy platform having similar functionality but different devices, a custom implementation of the REST API is sufficient to execute the experiment on the microscopy platform. If new functionality is needed, a custom implementation of the corresponding imaging requests allows the integration of these functionalities into the experimentation platform. This again nicely aligns with our operating system analogy, such that once an imaging application or smart experiment is developed, it can be executed on different imaging hardware, with the operating system handling the low-level hardware interfacing.

8.7 Summary

In this chapter, we have combined the JediOS platform with a real microscopy platform and a real-time image analysis pipeline, allowing the transfer of event-driven experiments designed and tested on the in-silico platform to the real microscopy platform. We have verified this approach at the example of screening the effects of PSs and developed a smart experiment workflow consisting of five steps ranging from mapping the high-throughput microfluidic chip, selecting the most suitable RoI positions for imaging, and the assignment of imaging applications and their execution in the event-driven experiment. All these steps are implemented in Python Notebooks, providing high customization opportunities, rich visualizations, and interactivity, such as pausing experiment execution. Moreover, the real-time dashboards provide a detailed view of the experiment's progress in both spatial and temporal dimensions, as well as insights into the biological development of the samples. In summary, the JediOS platform unlocks smart, high-throughput experimentation and provides a powerful platform for designing flexible imaging applications and recording high-quality datasets while making automated and rule-based decisions. Through this workflow, smart experi-

ments in MLCI become much more approachable to the standard user and the implementation in Jupyter Notebooks, with their ease of execution, customization, and sharability, has the potential to accelerate research in this live-cell imaging.

Chapter 9

Data-driven analysis of the high-throughput photosensitizer stress induction screening

Contents

9.1 Measuring single-cell excitation fluorescence 138

9.2 Quantifying stress-induction effects on the cell population level . . . 143

9.3 Quantifying stress-induction impact at the single-cell level 147

9.4 Discussion & Outlook 155

9.5 Summary 156

In the previous chapter, we developed and conducted an sMLCI experiment, demonstrating its capabilities in a high-throughput screening of PS influence. We have focused on a smart sMLCI workflow and a state-machine design that allows us to precisely record the development of cells after the EP with two screening parameters: the intensity and duration.

In this chapter, we introduce a detailed statistical analysis of the performed high-throughput screening. In particular, we focus on understanding the effect of PS excitation and quantifying its impact on microbial growth at both the population and single-cell levels. Using these detailed insights extracted from the recorded images, we show, for the first time, that the energy of the EP is linked to the growth behavior of cell populations and individual cells. This finding indicates that dynamic excitation energy enables precise control of the amount of intracellular-induced stress, thereby controlling the severity of damage to individual cells. Gaining these insights within a single experiment underlines the efficiency of high-throughput MLCI screenings using JediOS and makes PSs a promising candidate for precise cybergenetic closed-loop growth control.

The results in this chapter have been obtained within a collaboration with Thomas Drepper, and Seung-Hyun Paik contributing their expertise in PS and strain development, and Matthias Pesch preparing the living cells and the screening experiment.

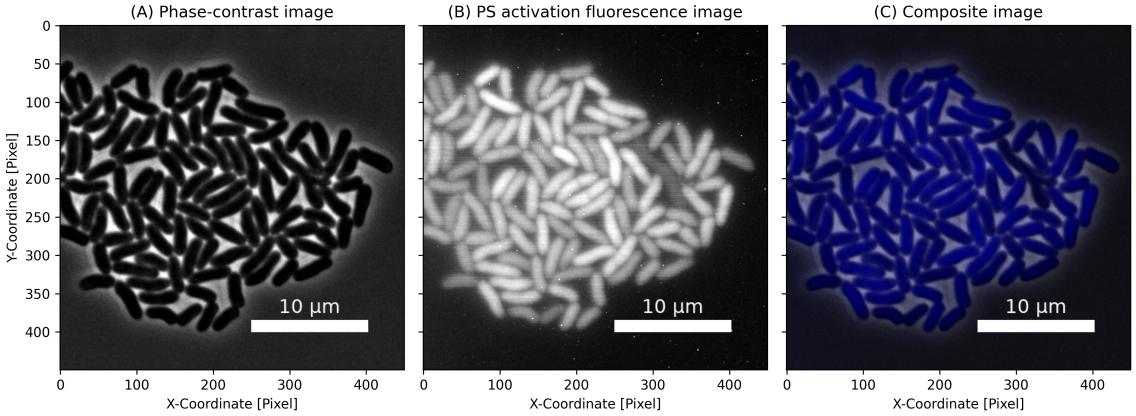

Figure 9.1: **Photosensitizer excitation and fluorescence emission imaging.** (A) A phase-contrast image is recorded to determine the spatial information of the living cells. (B) While the photosensitizers are activated with a specific excitation pulse, their fluorescence emission is measured, and (C) both recorded images are combined into a composite image to align the fluorescence readouts with the cell structures.

9.1 Measuring single-cell excitation fluorescence

Exciting the PSs with an EP activates them and leads to the emission of fluorescence light. This fluorescence emission is recorded by the camera during the excitation using standard fluorescence imaging routines. Hereby, the excitation light wavelength is blocked by a specific filter cube such that primarily the PS emission light is measured. Figure 9.1 shows the phase-contrast image, the fluorescence emission image, and a composite image where the emission light is indicated with a blue color. Based on the cell morphology extracted from the phase-contrast image, the corresponding fluorescence readouts are computed for each living cell in the cell population. The first goal is to investigate two fundamental questions for the PS excitation:

1. Do higher EPEs lead to higher single-cell fluorescence emission?
2. How heterogeneous is the fluorescence signal across cells within a cell population? Do some cells show high fluorescence signals while others show low fluorescence signals despite an excitation with the same EPE?

Thus, in the following, we investigate the EPE and the corresponding single-cell fluorescence readouts.

9.1.1 Calibrating the excitation pulse energy

The PSs are activated by a (short) EP leading to the release of ROS. The two main parameters of this pulse are the excitation intensity and the excitation duration. The excitation intensity of the utilized light source determines the power of the excitation light pulse. Thus, the excitation power and the excitation duration are the two parameters that determine the energy of the excitation

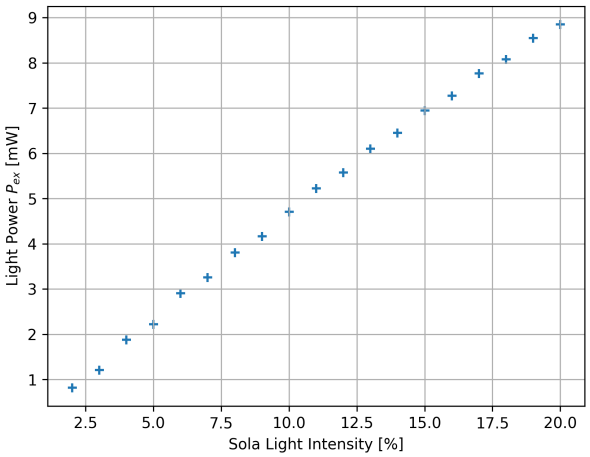

Figure 9.2: **Excitation light source power calibration.** The light power (P_{ex}) is measured for different intensity parameters of the SOLA fluorescence lamp. The PD300-MS Microscope Slide Power Sensor (Ophir Spiricon Europe GmbH) was used to measure the light power at the location of the microfluidic chip. The filter cube (FluBoV2), appropriate for the Pp1FbFP excitation, is used providing an excitation wavelength of 445 ± 10 nm and emission wavelength of 494 ± 10 nm.

pulse. This excitation pulse energy (E_{ex}) is proportional to the product of excitation power (P_{ex}) and duration (t_{ex}):

$$E_{ex} \sim P_{ex} \cdot t_{ex}. \tag{9.1}$$

The excitation duration (t_{ex}) is the time that the light source is active and illuminates the living cells. This parameter is defined for each imaging application and applied using Pycro-Manager. The excitation power (P_{ex}) is indirectly controlled by the light intensity of the fluorescence light source, a parameter that can be defined in Pycro-Manager in integer percent steps. For calibrating the excitation power emitted by the LUMENCOR SOLA SE II (SOLA) light source at different intensity parameters, we used the PD300-MS Microscope Slide Power Sensor (Ophir Spiricon Europe GmbH) placed at the location of the microfluidic chip to measure the applied excitation light power with the corresponding filter cube configuration in place. Figure 9.2 shows the measured light power for various light intensities. These measurements serve as a lookup table to compute the excitation power P_{ex} for a specific light intensity parameter.

During the PS screening experiment, we executed SM3* with several excitation intensity and duration parameter combinations. Table 9.1 shows their corresponding EPE in ascending order.

9.1.2 Excitation pulse energy and average cell fluorescence

To address the first question, whether higher light energies lead to more fluorescence emission, we measured the average fluorescence intensity for a cell in the cell population excited with a specific EPE using an acia-workflow (Chapter 4). Performing this computation for all utilized EPEs and cell populations in our experiment yields Figure 9.3, allowing us to investigate the correlation between EPE (x-axis) and cell fluorescence (y-axis). The cell fluorescence is computed as the average cell flu-

Table 9.1: **Excitation screening parameters and EPEs**. The seven combinations of excitation intensity and duration parameters applied during the PS screening experiment and their resulting EPE. The table entries are sorted in ascending order by their EPE.

Excitation intensity [%]	Excitation duration [s]	EPE [mWs]
5	0.1	0.22
10	0.1	0.47
5	0.3	0.67
5	0.5	1.11
10	0.3	1.41
10	0.5	2.35
10	4.0	18.84

orescence pixel intensity (ACFI) of cells averaged over all cells in each cell population (Figure 9.1B). Thus, every data point denotes the average ACFI of a single cell population measured during the EP. The parameters controlled by the state-machine, excitation duration, and excitation intensity are marked with colors and marker styles, respectively.

We confirm that increasing the EPE generally leads to an increase of the ACFI, indicating a stronger excitation of the PSs within the cells of the population and, thus, having the potential to cause a higher ROS-based stress induction. Interestingly, there appears to be a saturation of fluorescence emission, especially shown by the positive control group that has been excited with specifically high EPEs (4 seconds, 10% intensity, green). This saturation effect is expected, as at any point in time, there is a finite amount of PS inside each cell, and upon the excitation of all these PS, we do not expect any further increase in ACFI.

Clearly, the saturation effect is not well sampled within this experiment as the saturation occurs somewhere between the positive control group and the second-highest EPE used in this experiment. However, in a future experiment exactly this domain can be sampled to precisely quantify the correlation between EPE and ACFI. Moreover, the improved repeatability of the sMLCI experiment makes adapting and executing these experiments very straightforward.

9.1.3 Heterogeneity in single-cell fluorescence response

Interestingly, the ACFI in Figure 9.3 shows a spread across the replicate cell populations despite the excitation with the same EPE. This brings us to our second question of investigating the heterogeneity in the response of individual cells. We investigate this phenomenon by measuring the fluorescence intensity on the single-cell level: The average single-cell fluorescence intensity (ASCFI) denotes the average fluorescence intensity for each single cell. Figure 9.4 shows the distribution of ASCFI for each cell population and RoI measured during the EP. Each histogram shows the ASCFI distribution across all cells within a single cell population. Each column shows the fluorescence distribution histogram for one specific EPE while each row corresponds to one of the four replicates, all excited with the same EPE. The last row shows the combined ASCFI distribution of all replicates (fifth row). The average cell fluorescence intensity (ACFI) of a cell population is marked by a dashed black line and corresponds with the ACFI measurements shown in Figure 9.3.

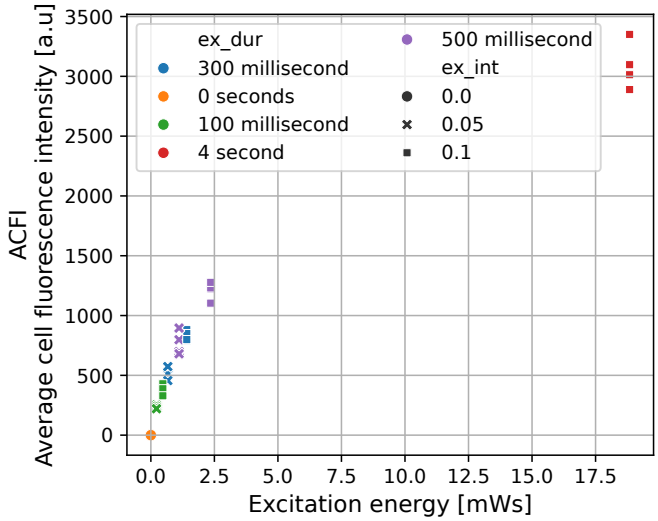

Figure 9.3: **Average cell fluorescence response based on the EPE.** We measured the average cell fluorescence for the cell populations excited with specific EPEs. These EPEs are defined by the duration (ex_dur) and intensity (ex_int) of the excitation pulse of the SOLA light source denoted with marker color and style, respectively. The four replicates with the same excitation duration and intensity lead to four data points per EPE.

To quantify the spread of ACFI in each distribution, we measure the Coefficient of Variation (CV), that is the standard deviation (σ) normalized by the average ASCFI ($\bar{x}$):

$$CV = \frac{\sigma}{\bar{x}} \cdot 100. \tag{9.2}$$

The histograms show that there is indeed a variation in single-cell fluorescence intensities within each cell population despite their homogeneous excitation. As observed before, the fluorescence intensities increase with higher EPEs. Interestingly, the CV remains similar across the replicates but also across different EPEs. Thus, we find that individual cells in the population show variations and heterogeneity in their fluorescence intensities. Interestingly, the amount of variation is relatively consistent across replicates and EPEs, specifically including the positive control group with a very high EPE.

Based on these measurements, we conclude:

1. Higher EPEs lead to higher ACFI and ASCFI readouts, therefore, indicating stronger intracellular excitation of the PSs.

2. The single-cell response (ASCFI) shows variations indicating single-cell heterogeneity. The amount of variation quantified by the CV is quite consistent across replicate cell populations and different EPEs.

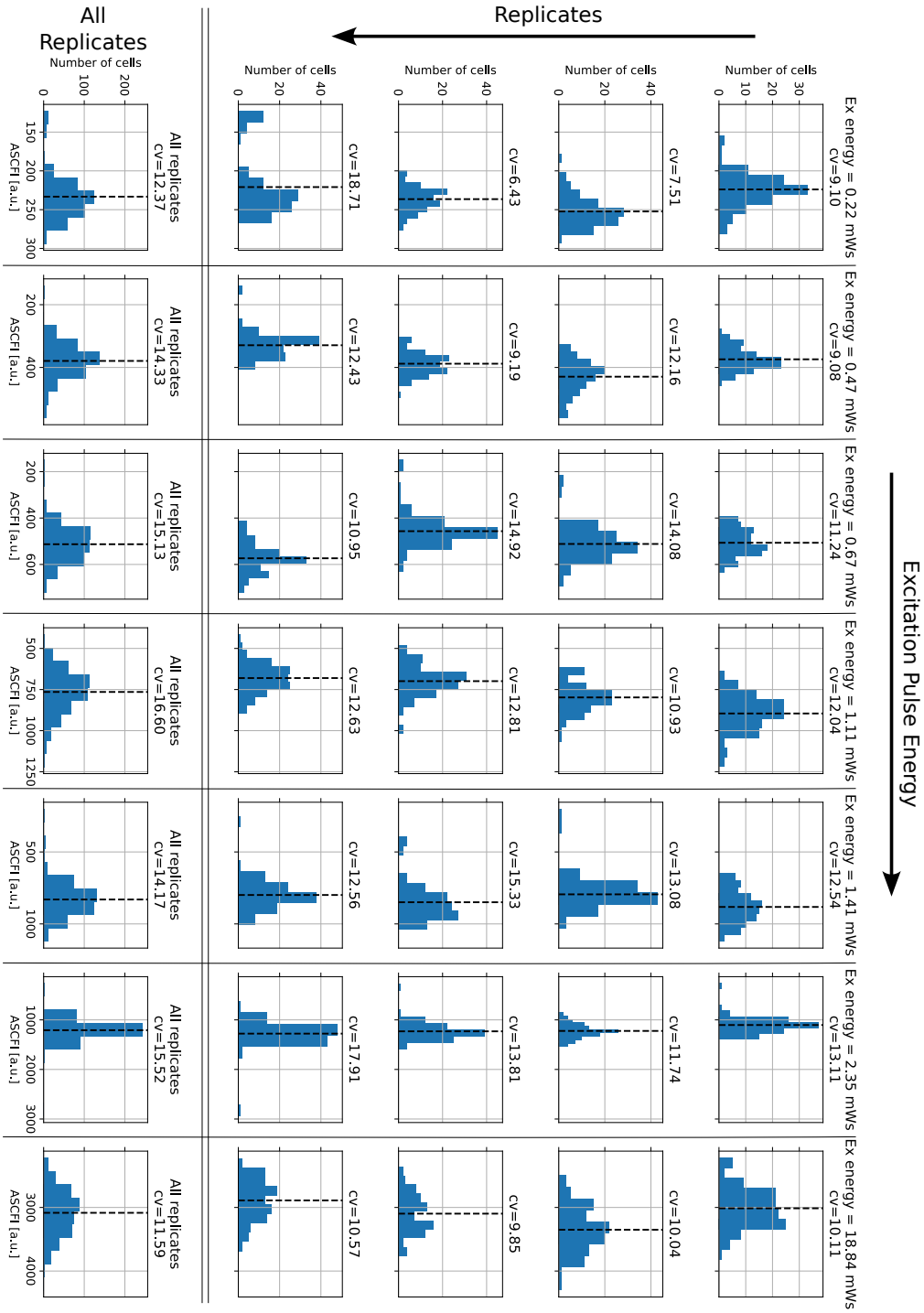

Figure 9.4: **Heterogeneity in single-cell fluorescence intensity.** Cell populations have been excited with various EPEs (columns, increasing to the right) and across four replicates (rows), while the last row shows the combined distributions of all four replicates. Each histogram shows the distribution of ASCFI across a single population. The average cell fluorescence intensity per population ACFI is marked by the black dashed line. For each histogram, the CV is computed. Histograms in the same column share the same x-axis.

9.2 Quantifying stress-induction effects on the cell population level

Having verified the technical excitation of the PSs using short EPs and the readout of the single-cell fluorescence response, we can now investigate the impact of the PS-mediated ROS release on the temporal cell development. Specifically, we address the question of whether higher EPEs and higher ASCFI lead to higher intracellular induced stress levels and, therefore, have a stronger negative impact on cell development. We quantify this by measuring the effect of various EPEs on the temporal population development.

9.2.1 Temporal cell population development

Figure 9.5 shows the time-series of population developments recorded during the PS screening experiment. All the replicate populations of a single EPE are shown in one row, and the EPEs are shown in ascending order. The 10% population size threshold (dashed gray line) triggers the EP at different points in time. After the EP, the imaging rate is increased to precisely sample and measure the response of cell development, as indicated by a higher temporal density of measurements. The EP for each cell population is performed at the beginning of the sampling burst.

As expected, the negative control group (A) shows almost exponential growth corresponding to a straight line in the log space. However, most cell populations exhibit a slowdown in growth later during the experiment. This slowdown is caused by cells leaving the open RoI chamber during the experiment. Furthermore, the positive control group (H) shows almost no increase in TSCA after the EP, confirming the presence of intracellular PS and the light-based ROS formation in our setup that has the potential to kill all cells within a cell population.

These two control groups represent the extreme cases of the PS excitation. Specifically interesting is the EPE range in between the negative and positive control groups (B-G) sorted in ascending order by their EPE. With increasing EPEs, we observe a stronger and stronger ”bump” in the TSCA development. The bump occurs immediately after the EP and indicates a decrease or a complete stop in the growth of the cell population size. However, this decrease is usually not permanent, and the cell populations retain growth after a certain amount of time, even for higher EPEs (G).

These temporal TSCA developments give a good first impression on PS screening and seem to support the hypothesis that higher EPEs induce more oxidative stress within the cells, causing a stronger delay in their development. In the following, we would like to quantify this delay in growth and, therefore, develop measures to quantify the ”bump” in the temporal TSCA development.

9.2.2 Quantifying stress-induction impact on the temporal cell population development

Based on these initial findings on the time-series data, we raise three major questions:

1. Do higher EPEs lead to a stronger impact on the temporal development of cell populations?

2. Does a dynamic range in the PS excitation exist where changes in the EPE lead to a changes in the response of the cell population development?

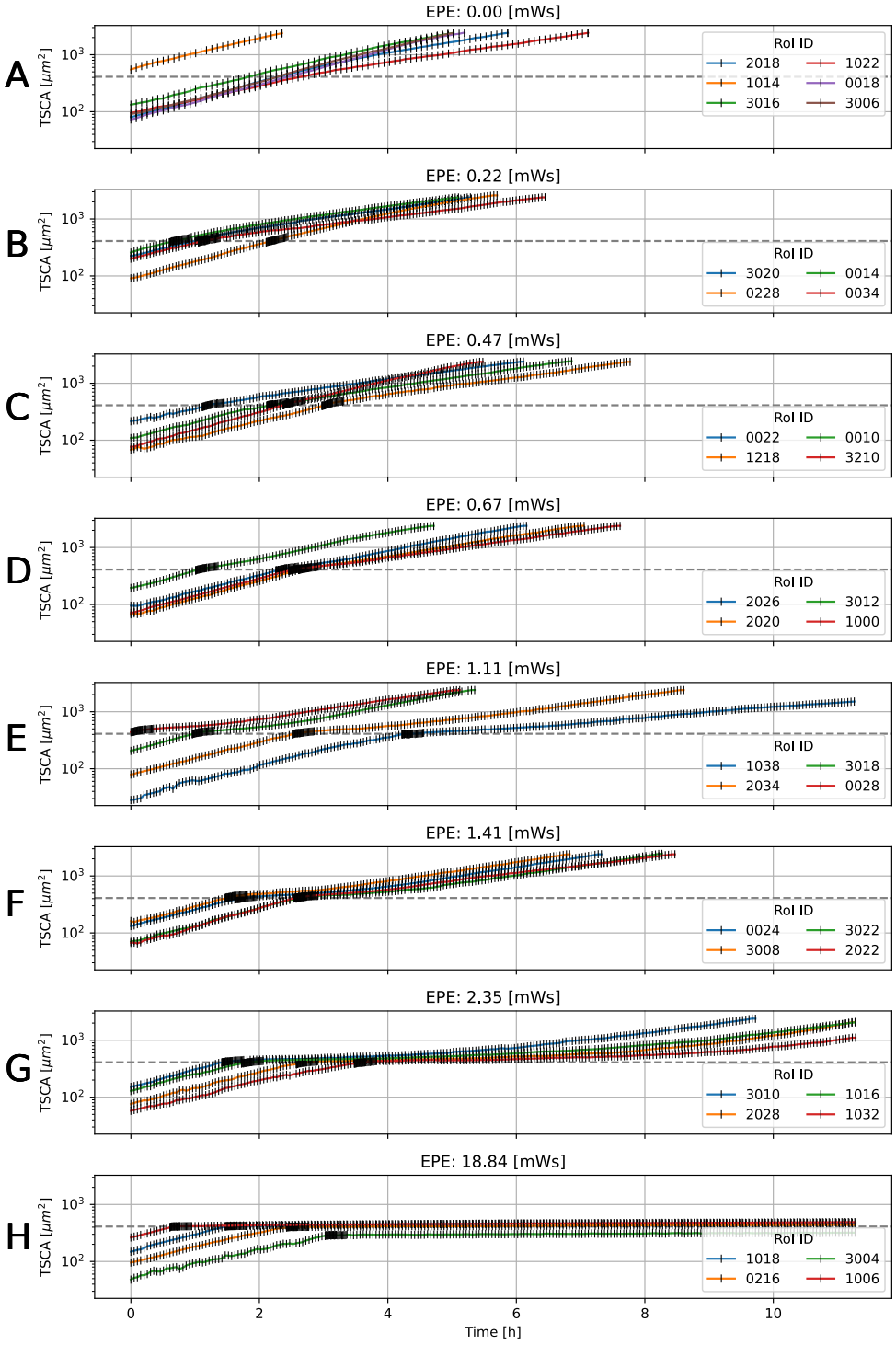

Figure 9.5: **Temporal cell population development with stress induction.** The development of all cell populations in this PS screening experiment, including the control group (A, no excitation), the various EPEs in ascending order (B-G), and the positive control group with high EPE (H). The TSCA development of each population is colored according to their RoI ID, and each image recording is denoted by a vertical bar. In particular, after the excitation, higher marker density indicates more frequent imaging of the cell population implemented by SM3⋆.

3. Is the fluorescence intensity a measure for the amount of induced single-cell stress so that higher fluorescence intensities are linked to stronger delays in the cell population growth?

In the following, we develop measures to quantify the EP influence on the temporal cell population development, discuss the advantages and disadvantages of the measures, and address the raised questions.

9.2.2.1 Developing cell population impact measures

Figure 9.6 shows the three measures that we introduce to quantify the impact of the EP on the temporal development of a cell population based on the TSCA time series. To reduce noise in the measurements, we apply a Gaussian filter using SciPy [144] and refer to this as the "filtered TSCA". The three measures are illustrated in Figure 9.6 and defined as follows:

1. **Population doubling time (PDT):** The Population doubling time is the duration between the EP and the doubling of the TSCA of the excited cell population.

2. **Population growth factor (PGF):** The Population growth factor measures the relative TSCA gain in a certain time window Δt after the excitation pulse at time $t^\star$:

$$GFT(t^\star, \Delta t) = \frac{A(t^\star + \Delta t)}{A(t^\star)}, \tag{9.3}$$

   where $A(t)$ denotes the most recent TSCA measurement after time t.

3. **Predicted population development delay (PPDD):** The PPDD estimates the temporal delay in TSCA development caused by the EP compared to a non-influenced cell population development. Thus, we estimate the exponential growth rate based on the TSCA development prior to the EP and extrapolate the "expected" TSCA development using this growth rate. The PPDD is then defined as the difference in time between extrapolated TSCA development reaching the final TSCA ($\hat{t}_{end}$) and the timepoint ($t_{end} > \hat{t}_{end}$) the observed cell population reaches the TSCA limit (see Figure 9.6 C):

$$PPDD = t_{end} - \hat{t}_{end}. \tag{9.4}$$

9.2.2.2 Evaluating stress-induction impact on the cell population level

We apply the three measures to all the recorded time series of the PS screening experiment (Figure 9.5) to investigate the three questions raised above. The resulting measures are shown in Figure 9.7 where each data point quantifies the EP impact on a single cell population, and the measures are resolved by the applied EPE (left column), and the measured ACFI (right column).

To address the first question whether higher EPEs lead to a stronger impact on cell population development, we take a look at the left column of Figure 9.7. Indeed, increasing EPEs lead to an increase in PDT and PPDD and a decrease in PGF, all indicating a slower development of the cell population in the aftermath of the EP.

The evaluation also shows the dynamic range of the PS excitation addressing the second question. Increasing the EPE gradually leads to a progressively negative impact on cell population development. This effect saturates with higher EPEs, for example, in the positive control group, as all cell

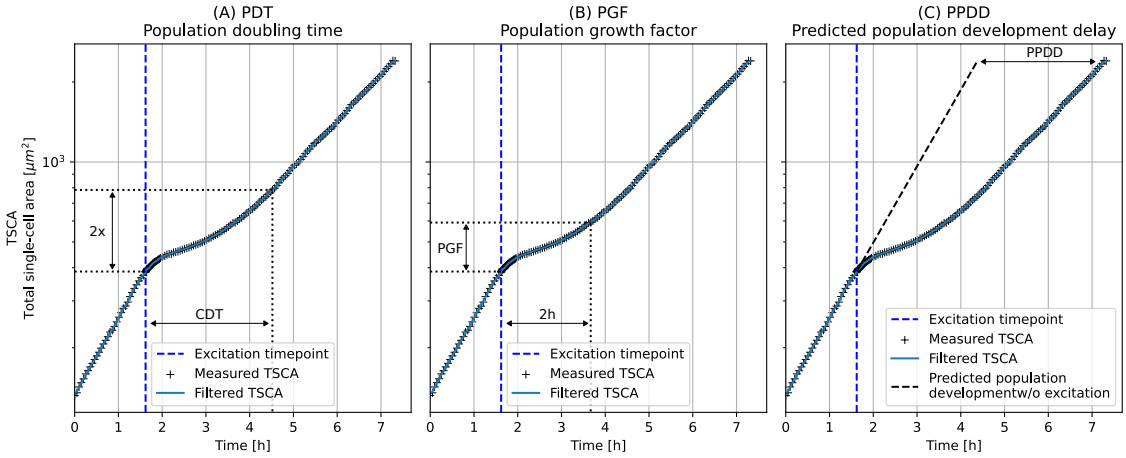

Figure 9.6: **Three population based measurements to quantify temporal influence of stress-induction.** (A) The Population doubling time measures the time it takes the cell population to double its TSCA after the excitation pulse. (B) the Population growth factor measures the relative growth of the cell population in a given time window after the excitation pulse, e.g. 2 hours. (C) The Predicted population development delay measures the difference in time to reach the final TSCA of the measured development to an exponential development after the excitation pulse. The exponential growth rate is estimated based on the growth rate before the excitation pulse.

development is terminated. Thus, precisely exploring the dynamic range requires experiments with further EPEs.

In order to answer the third question, we turn to the second column of Figure 9.7. Herein, we investigated whether the emitted fluorescence recorded during the EP is a suitable indicator of the amount of induced oxidative stress. This is the case if higher ACFI readouts are linked to a slower cell population development. For all three measures, increasing ACFI generally leads to a rise in PDT and PPDD, and a decrease in PGF, thus slowing the cell population development. Therefore, the ACFI seems to be a good indicator of the amount of stress applied to a cell population.

Despite the suitability of the measures to quantify the increasing impact of rising EPEs, they come with several shortcomings. For the PDT, low EPEs lead to low (1-2 hours) and higher EPEs lead to a higher PDT (up to 6 hours) quantifying the time it takes the cell population to double its TSCA. However, for the positive control group with the highest EPE in the experiment, the PDT cannot be measured as the cell development stops and no doubling of the TSCA is observed.

In contrast, the PGF and PPDD are also measurable for the positive control group. However, the PGF only measures the TSCA development within a specific (short) time window. Therefore, the values for the largest two EPEs are very similar although the temporal cell development showed that the cell populations exposed to the second highest EPE (2.35 mWs) regain growth after several hours (Figure 9.5G) while the cell populations in the positive control group showed no further TSCA over the complete observation period Figure 9.5H). Thus, similar PGF values only indicate a similar growth behavior within the temporal window.

The PPDD does not have this disadvantage. However, the PPDD of a dead cell population will increase linearly with the observation period. Moreover, the quality of the PPDD highly depends

on the quality of the extrapolated TSCA development. These extrapolations are based on the assumption of exponential cell population development, which is only a rough approximation as shown by observing the negative control group (Section 9.2.1A).

Notable is the variation in the measures, specifically observed in the left column, due to the replicates having the same EPE value. This is partly explained by different PS concentrations present within the cells of the cell population. The ACFI captures this distribution of PS concentration. Especially the PGF shows large variations for lower EPEs. This is unexpected as we would expect the cell populations to behave similarly, especially for the control group. These variations can be partly explained by high levels of noise that are common in single-cell data due to the heterogeneity in cell development [177, 20]. However, there is also a technical explanation: cell populations exposed to lower EPEs (or no EP at all) will develop faster. In the open RoI designs, the faster development leads to a higher chance of cells leaving the RoI and missing in the TSCA quantification. In particular, this depends on the initial random placement of the cells within the RoIs. Cell populations located at the open ends of the RoI suffer more dramatically from cell loss, impacting all three measures and causing high variations across replicate measurements.

9.3 Quantifying stress-induction impact at the single-cell level

The population-based impact quantification has revealed that the phototoxicity of PS has a dynamic range and can be used to induce precise amounts of intracellular oxidative stress, causing various degrees of slowdown in cell population growth, ranging from a short-term slowdown in growth to a complete stop of cell development. However, these measures do not quantify the heterogeneity in the single-cell stress response: some cells may die from PS excitation while others survive and continue to grow. Considering the presence of single-cell heterogeneity [177, 20], we raise the question of how this slow down in TSCA growth works at the single-cell level. Do all cells slow down their development at the same rate? Or are there differences in the cell response, and some cells are more robust to the stress-induction while others suffer more severely and may even die?

A first indication of heterogeneous cell response to the EP has already been presented in Figure 9.4. Here, cells within a cell population showed a variation in the ASCFI despite the excitation with the same EPE, most likely due to differences in single-cell concentrations of PS. This raises the question of whether these differences in ASCFI result in different induced stress levels and, therefore, a different impact on the temporal cell development.

To address these questions about the development of individual cells, we focus on the single-cell level. Therefore, we utilize the acia-workflow developed in Section 4.3.3 to track each cell in the PS screening experiment, analyze single-cell development, and highlight both the potential and limitations of this approach.

9.3.1 Challenges in single-cell tracking in PS experiments

In Chapter 3, we have shown that tracking individual cells highly depends on the choice of a suitable imaging interval. These findings have been considered in the design of the experiment constraining the number of EPEs that can are concurrently investigated within a single experiment leading to a maximum imaging interval of 3 minutes. Moreover, we used the short imaging interval of 30 seconds after the excitation to gain a higher temporal resolution into the single-cell response immediately

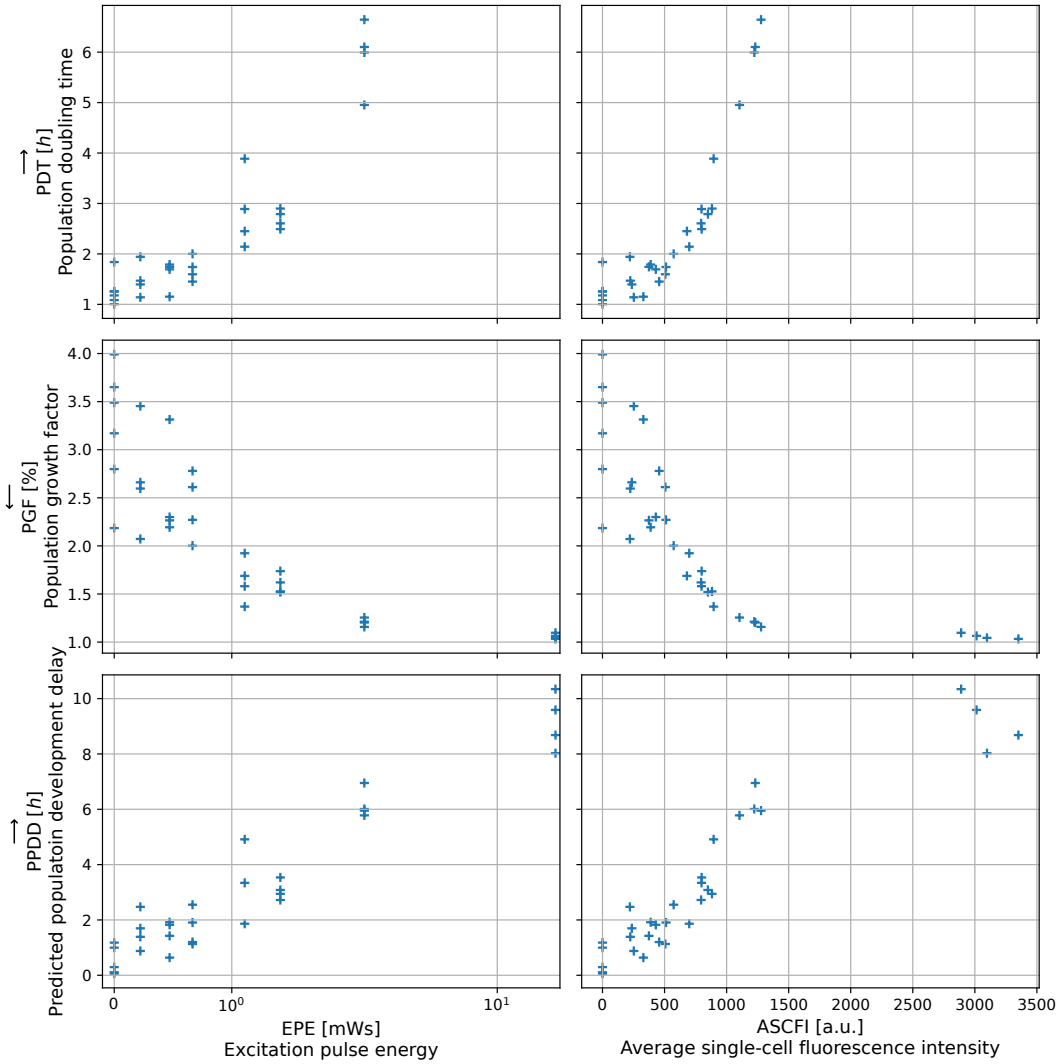

Figure 9.7: **Quantification of the cell population response to stress induction based on the PDT, PGF and PPDD measures.** The three quantities are shown with respect to the EPE (left column) and average cell fluorescence intensity (right column). Every data point is estimated from the cell population development (TSCA) of a cell population in a RoI. In the left column, the four replicates per EPE result in four data points with the same EPE x-axis value. For the three quantities, we have the following impact correlation: higher PDT, higher PPDD, and lower PGF indicate slower cell population development. The arrows on the y-axis indicate the direction of strong impact and slower cell population growth.

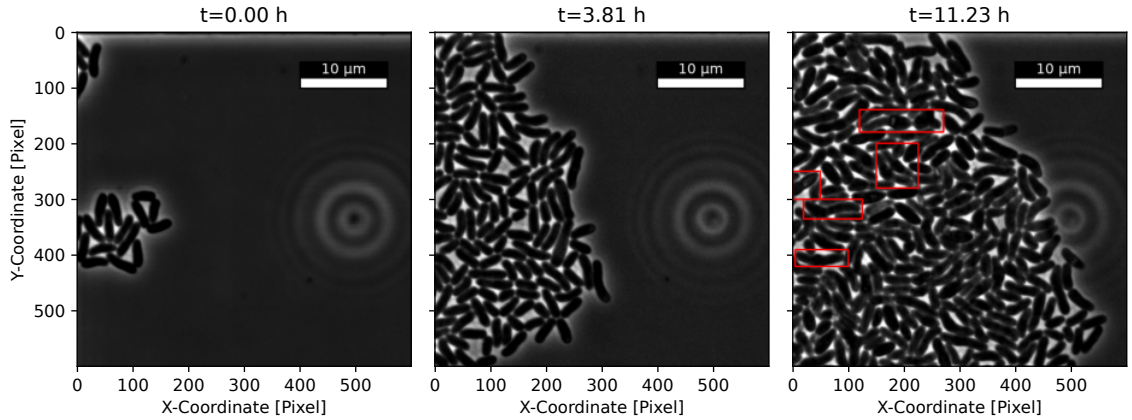

Figure 9.8: **Abnormal cell morphology caused by stress-induction.** Three snapshots of the temporal cell population development excited with an EPE of 2.35 mWs at $t = 3.81\ h$. Exemplary cells with abnormal morphology are marked with red rectangles. The center image shows the time point of the EP.

after the EP.

However, during the conduction of the PS screening, other effects influenced the tracking quality that we had not previously considered: Using higher EPEs leads to the development of abnormal cell morphologies. Figure 9.8 shows the development of an exemplary cell population, where in the aftermath of the EP cells show abnormal morphological behavior (marked in red). Interestingly, these changes in morphology are not observed in the positive control group where no more cell development is observed after the EP (Figure 9.9).

These changes in morphology impact the image analysis: the change in cell appearance in the phase-contrast image exacerbates the challenge of distinguishing individual cells or determining whether they have disappeared due to cell membrane failure or division. This leads to a much more challenging segmentation task, causing an increase in over- and under-segmentation. For population-based measures, over- and under-segmentation are not as severe because cell measurements are combined into a single population measure. However, when tracking individual cells, over- and under-segmentation can break the tracking links, decreasing the reliability of the tracking data.

Upon further investigation of the morphological abnormalities, we found that these typically only appear at higher EPEs and only hours after the EP. Thus, to evaluate this experiment, we utilize tracking data within a limited time horizon to mitigate the challenges of cell morphology changes and deteriorating segmentation and tracking quality.

9.3.2 Measuring single-cell stress-induction response

Tracking each individual cell and extracting its temporal area development over its life cycle from birth to division yields very similar time series compared to the temporal development of cell populations. Thus, in principle, the same three measures used for quantifying the impact on the cell population can be applied to quantify the impact of the EP on each individual cell. Accordingly,

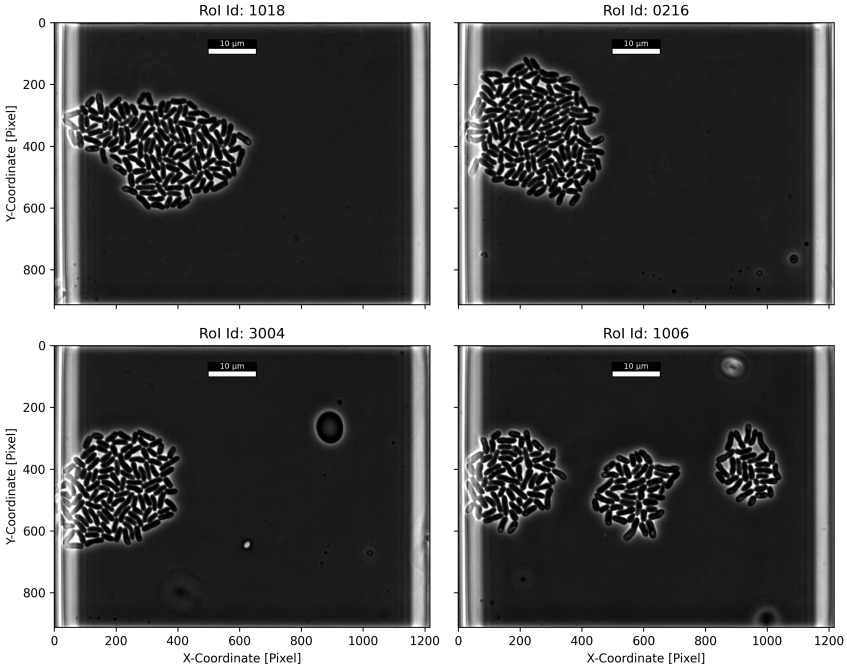

Figure 9.9: **Final images of the four positive control group cell populations.** The positive control group populations have been excited with an EPE of 18.84 mWs. The four images show that cell population at the end of the experiment at $t = 11.3\ h$.

we refer to these measures as Cell doubling time (CDT), Cell growth factor (CGF), and Predicted cell development delay (PCDD). Figure 9.10 shows the three measures now applied to an exemplary temporal single-cell development. This specific cell was alive when the EP was triggered (dashed blue line), then the imaging rate increased (indicated by higher cell area measurement sampling frequency) to precisely record the temporal behavior of the cell in the immediate aftermath of the EP. In contrast to population-level measures, the single-cell area time series contains more noise. Therefore, we apply a Savitzky-Golay filter [178].

9.3.3 Evaluating temporal single-cell stress-induction response

Figure 9.11 shows the three measures applied to all cells in the experiment that experience the EP. The first difference to population measures is the amount of data points. Each observed cell population contains roughly a 100 cells (10% of the RoI area) at the time point of the EP, and each of these cells is a potential data point. However, similar to the population measures, the single-cell time series needs to satisfy certain conditions to yield a data point. For example, for measuring the CDT, the cell area needs to double, which often does not happen as the cell divides beforehand. Similarly, for the CGF, the cell needs to be observed at least 2 hours after the EP (and not divide beforehand). For the PCDD, sufficient observations before and after the EP are required to estimate the growth rate and measure the delay. Thus, all three measures are based on a different number of cell observations denoted by N_{cells}.

The amount of data points requires not only visualization as a point cloud (first row) but encour-

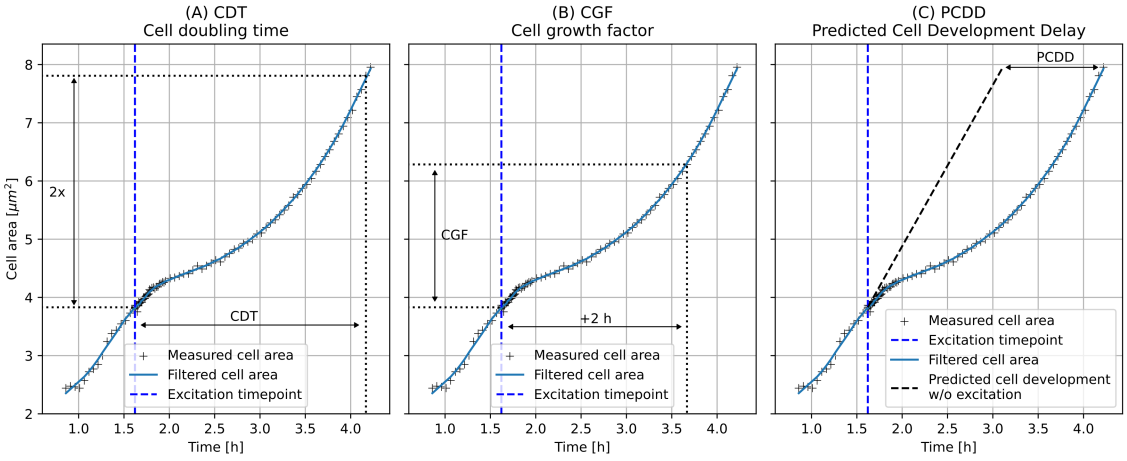

Figure 9.10: **Three single-cell based measurements CDT, CGF, PCDD to quantify temporal influence of stress-induction.** These measures are the same as in Figure 9.6 but are now performed on a single-cell level instead of a population level. (A) The CDT measures the time it takes the individual cell to double its area after the excitation pulse. (B) the CGF measures the relative growth of the cell in a given time window after the excitation pulse, e.g., 2 hours. (C) The PCDD measures the difference in time to reach the final cell size of the measured development in comparison to linear development after the excitation pulse. The linear growth rate is estimated based on the growth rate before the excitation pulse.

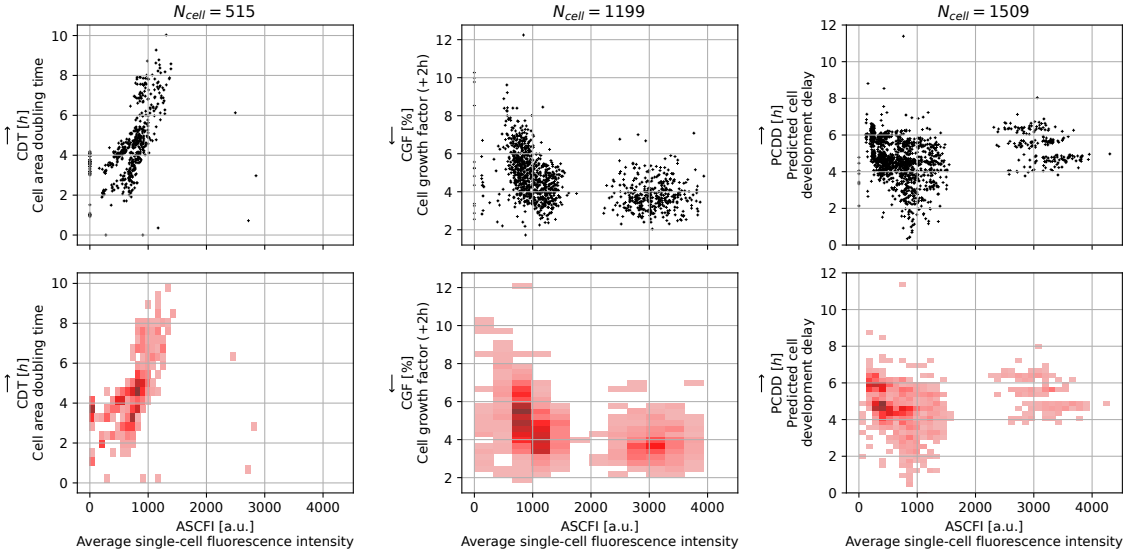

Figure 9.11: **Three measures CDT, CGF, and PCDD for single-cell stress impact quantification.** Each column shows one of the three measures with arrows pointing in the direction of slow cell development. The first row shows the data points as a point cloud where each data point denotes the measure of a single-cell development. The second row shows a 2D histogram of the point cloud. The darker red color indicates a higher density of data points. The gap in the point clouds (center and righ) is due to the large EPE of the positive control group leading to a high ASCFI.

ages the visualization as a 2D histogram (second row). For the CDT, we can confirm that higher ASCFI leads to slower cell growth indicated by higher CDT values. However, the CDT measure is only computed for cells that experience the EP early in their life cycle and, thus, still double their size. This potentially introduces a bias as only a specific subset of cells is taken into account during the analysis.

In contrast, the CGF and PCDD can be computed for a larger amount of cells. While the CGF shows a decrease with higher ASCFI, the PCDD does not show a correlation but exhibits high variations in the measurements. This may be caused by the high amounts of noise in the single-cell area measurements causing bad approximations of the cell growth rate and low-quality predictions for the cell development without the EP.

In summary, CDT and CGF indicate a correlation between ASCFI and the temporal cell development. However, all three measures show very high variations in the cell response. Thus, we conclude that observing the single-cell cycle gives a first indication about a link but the observation period is too short to sufficiently quantify the impact of oxidative stress to the individual cell.

9.3.4 Identifying single-cell heterogeneity in the lineage tree

While the single-cell life cycle may not be sufficient to quantify the stress-induced impact on the development of a single cell, we can extend the temporal observation window by investigating the development of each cell's offspring, that is, all the cells that descend from this excited mother cell. Performing such an offspring-based analysis requires single-cell tracking, yielding a lineage tree that shows cell development and division. Figure 9.12 shows such a lineage tree for an exemplary cell population. Each node represents a cell detection, and each edge indicates that the cell life cycle continues or a cell division occurs. The size of the nodes is proportional to the size of the detected cell in the microscopy image.

The lineage tree reveals a regular division behavior before the EP and a substantial change in the cell development and division behavior after the EP. Initially, the cells exhibit regular growth, as indicated by continuous increases in node size and frequent cell divisions. After the EP (dashed blue line), this behavior changes, and we have identified three exemplary cell behaviors, which we term the "surrenderer", "survivor", and "robust" cell. The "surrenderer" cell (left red rectangle) is a cell that divides just after the excitation, but its two descendants do not show any further increase in cell area or cell division. Thus, the descendants are either dead or so severely damaged that normal cell development is no longer possible. The "survivor" cell (center red rectangle) is a cell that does not grow for a long time after the EP but eventually resumes cell growth and cell division again. This cell suffers heavily from stress induction but can eventually recover and grow again. The "robust" cell (right red rectangle) is a group of cells that also show a delay in the cell division behavior after the EP but then starts to divide frequently again. Thus, in the lineage, they can be identified by a larger offspring, that is, more cells that originate from the robust mother cell.

These three cell behaviors are hand-picked examples that highlight the heterogeneity in cell behavior contained in the lineage tree information. To quantitatively investigate the cause of this heterogeneity in cell behavior and its correlation with the single-cell ASCFI, we need to develop measures based on the lineage tree that measure the offspring development.

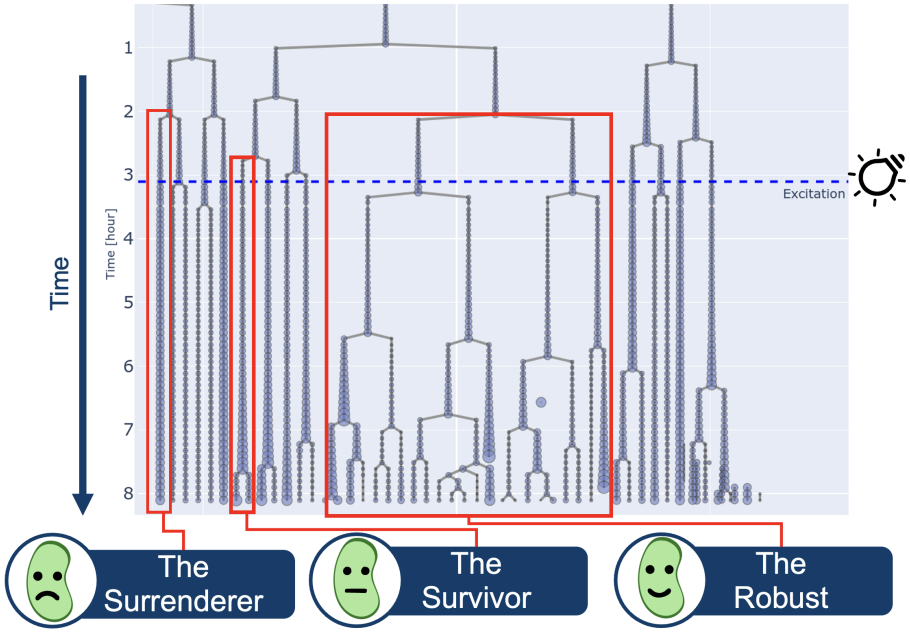

Figure 9.12: **Lineage tree of a stress-induced cell population.** Each node represents a cell detection at a specific time point, where the size of the node is proportional to the size of the cell detection. Edges link cell detections and branches indicate cell divisions. The dashed blue line indicates the excitation pulse. Three exemplary cell behaviors are highlighted (red rectangles) and termed ”surrenderer”, ”survivor”, and ”robust” according to their stopped cell development, long recovery time but eventual cell division, and rather robust cell division despite the stress induction. The analyzed cell population has been excited with an EPE of 1.11 mWs.

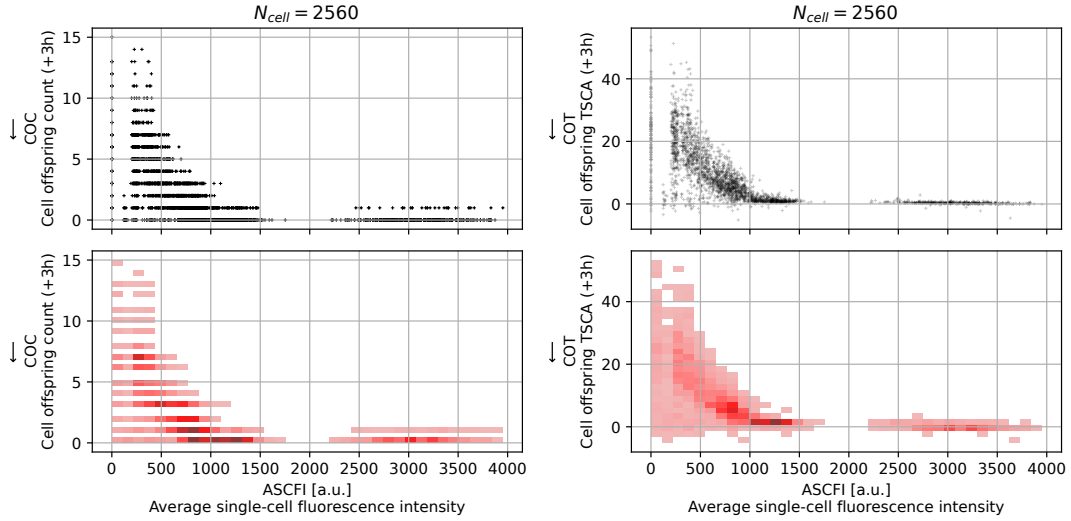

Figure 9.13: **Lineage-based measures for single-cell offspring quantification.** COC (left) and COT (right) measures evaluated for the PS screening experiment and visualized as individual data points (first row) or the corresponding 2D histograms. Each excited cell during the experiment produces one data point. The arrow on the y-axis points in the direction of less cell offspring development, indicating a stronger PS impact.

9.3.5 Combining lineage and population measures for quantitative single-cell offspring quantification

Based on the insights gained from investigating lineage trees, we propose a combination of population and single-cell measures. We combine the lineage information, containing all cell divisions and heritage, with population-based measurements to quantify the impact of the EP on the offspring of the excited cell. Using the lineage, we compute these measures based on the offspring sub-populations: For every living cell during the EP, we determine the offspring sub-population based on the lineage tree and count the number of offspring cells or their total TSCA at a certain point of time after the EP and term these measures Cell offspring count (COC) and Cell offspring TSCA (TSCA), respectively.

Figure 9.13 shows the result for the COC (left) and COT (right) measures three hours after the EP. In particular, we combine the information from all cell populations and extract one data point per excited cell in the experiment, resulting in $N_{cell} = 2560$ data points. Again, we use a point cloud (first row) and a 2D histogram (second row) to visualize their distribution.

The first noticeable difference between both measures is their granularity. The COC measure is based on counting the number of offspring cells. Thus, it is a discrete integer measure. In contrast, the COT measure is based on the sum of all offspring cell areas and, thus, a continuous measure. Both measures show two clusters, where the right cluster with high ASCFI values represents the measure for cells within the positive control group. Due to the large gap in the applied EPEs, there is no measurement point between the two clusters.

The left cluster shows the measurements from cell populations excited with different EPEs. Indeed, the quantification of the cell offspring reveals that the higher the ASCFI, the more likely

is a decrease in the offspring development quantified by lower COC and COT measures. From this, we can deduce that higher ASCFI indicate higher single-cell stress levels that correlate with lower offspring development in the immediate aftermath of the EP (here 3 hours). Due to its finer granularity, the COT shows a smoother and more precise link between single-cell fluorescence and offspring development and is recommended for future use to quantify the impact on single-cell development.

While both measures show the general trend, they also exhibit high variation, especially for small fluorescence readouts. These high variations are caused by the same technical issue as before: Both measures are based on the assumption that tracking the cells and computing the lineage allows us to observe the development of the complete offspring population of each excited cell. However, this experiment has been performed with open RoI designs such that cells occasionally leave the RoI. While this is noted in the lineage tree by a track end, the complete offspring population of cells can no longer be observed. This causes an underestimation of the COC and COT measures and is one reason for the lower values in both measures for low single-cell fluorescence readouts, resulting in a large spread. Thus, future experiments utilizing COC and COT measures during analysis should employ closed RoI designs, ensuring that all cells within the offspring populations remain within the RoI and the full offspring population development is observed.

9.4 Discussion & Outlook

In this chapter, we have utilized the recorded high-quality screening data to perform a detailed data analysis, aiming to better understand the optogenetic application of PS, link the light energy, the induced intracellular stress level, and the impact on cell development. We achieved this by leveraging the information depth of MLCI experiments and analyzing cell development at various levels of granularity. Therefore, we developed several measures at the population and single-cell levels to perform, to our knowledge, the first quantitative analysis of EPE-dependent impact of PS impact on cell growth with single-cell resolution. We found that increasing EPEs lead to an increased excitation of PSs, leading to higher stress levels for the individual cells which is reflected in reduced development after the EP. We demonstrated that combining single-cell fluorescence readouts with single-cell tracking and observation of the cell's offspring development can reveal these individual cell responses. Thus, high-throughput screening of PSs in MLCI is an ideal tool combination for investigating the response of cells to various stress levels and decoding single-cell heterogeneity.

Across the analyses, we have highlighted the advantages and disadvantages of the several measures that we have developed. The single-cell fluorescence response turned out to be a precise indicator of the induced stress level. Moreover, the cell response of the PS excitation has a dynamic range: varying the PS energy allows to induce different stress levels and, thus, precisely dose the impact on the cell development. In particular, the combination of structural lineage information and spatio-temporal sub-population development provides a very detailed view of the response of individual cells, specifically with respect to their single-cell fluorescence signal, which indicates their induced stress level.

Notably, all these results have been extracted from a single experiment. This highlights the efficiency and data quality achieved by the smart experimentation workflow introduced in the chapter before. However, conducting further PS screening experiments is inevitable to continue exploring the dynamic range of the PS and gain a better view of the cell response across a larger range

of EPEs. With independently prepared and conducted experiments, we also face the question of the impact of so-called batch effects: The batch effect is known in life science to cause variation in repeated experiments due to the extensive amount of variables - such as medium composition, cell pre-culture, and inoculation - and makes a precise replication of life science experiments very challenging [179]. However, the detailed view and high-throughput capabilities in MLCI and the smart experimentation workflow provide the opportunity to better understand batch effects and their influences. For example, when exploring the dynamic range of the PS by sampling the EPE space in multiple experiments, we suggest always choosing EPEs that overlap with those from previous experiments. This overlap of EPEs allows to investigate whether the living cells in this experiment respond similarly across multiple experiments, providing a good indicator where batch effects arise and whether they need to be considered.

9.5 Summary

In this chapter, we have analyzed the high-throughput screening of PS to investigate its impact on the development of stress-induced cells, from the population to the single-cell level. In this process, all the technologies developed throughout this thesis have been applied in the attempt to quantify the amount of induced intracellular stress and the corresponding response of the cells: The tracking benchmark lead to the appropriate choice of the imaging interval, the development of analysis and visualization tools is the foundation to extract and generate insights on population and single-cell level, and the real-time insights enabling smart and autonomous experimentation served as the foundation to acquire the screening data at high quality.

To this end, we find that a single experiment is sufficient to give a detailed insight into the dynamic effects of PS. We found that EPs with the same EPE yield different excitations of the PS inside the individual cells shown by variations in the fluorescence emission signal. We developed measures to quantify the impact of stress induction on cell population development and found that the single-cell fluorescence signal indeed serves as an indicator of the amount of induced stress and subsequent cell development. Interestingly, the PS showed a dynamic range, i.e., an energy band where the light excitation influences the cell development but does not kill the cells. We found that this effect is confirmed by a view on the single-cell level, however, the short observation period, high noise levels and other influences - such as cell size, cell state, and others - lead to no clear link between fluorescence readout and impact on single-cell behavior. Combining both population-level readouts and single-cell information by investigating the development of the cell offspring population overcomes these limitations and precisely shows that cells emitting higher fluorescence signals, indicating higher stress levels, have reduced offspring development.

In conclusion, we find that smart and autonomous experimentation using the JediOS platform is ideally suited for screening PS impacts at the population or single-cell level. While all the results presented in this chapter are obtained from a one-day screening experiment, many more smart MLCI experiments can be carried out in the future to improve our understanding of the causes and consequences of PS-based stress induction and prepare them for utilization in closed-loop cell growth control.

Chapter 10

Discussion and Outlook

Contents

10.1 Data Benchmarking & Analysis 157
10.2 Smart MLCI experimentation 158
10.3 Towards cybergenetic growth control using photosensitizers 159
10.4 Screening additional cell influencing factors 161
10.5 The costs of MLCI experimentation 161
10.6 Distributed experimentation in multi-microscope setups 165
10.7 Beyond bacteria towards eukaryotic cells 166
10.8 Autonomous experiment design for high-throughput screenings . . . 166

While every individual chapter already contained a discussion of the achievements and their implications, we use this final discussion chapter to zoom out of the individual projects and discuss the achievements and implications in the broader context of live-cell imaging. Moreover, we will provide an outlook on the impact of the developed technology, its importance for life science research, and the continuous effort to make MLCI experiments smarter and more autonomous.

10.1 Data Benchmarking & Analysis

In the first part of this thesis, we have provided the data foundation and established a powerful high-throughput live-cell image analysis platform to gain insights into single-cell behavior.

Providing the data foundation is crucial for benchmarking SOTA methods in the field and for understanding the solved and existing challenges in extracting information from MLCI experiments. Moreover, our large-scale dataset immediately contributed to the further development of data-driven tracking methods [33]. By developing the experiment-aware metrics, we have established a tighter connection between the experiment setup and the resulting challenges in the benchmark. Thus, we raised awareness that many challenges can be overcome by designing the experiment more effectively in the first place. With our contribution, we facilitate the interdisciplinary and comprehensive perspective on such live-cell experiments.

The development of the single-cell analysis workflows targeted on making MLCI experiments suitable to generate everyday insights at high-throughput. The key decision here was the agile

development of the software: During the software development, it was continuously deployed and used by colleagues in the lab. Their feedback and experience contributed to the implementation of the easy-to-use workflows, underlined by the more than 800 million cells analyzed by the lab members using various workflows during the time of this PhD project.

Moreover, the established example workflows highlight the complexity but also the opportunity to connect all steps of data processing and research hypothesis-driven data analysis. Combining powerful deep learning-based analysis tools with the extensive ecosystem for processing and visualization data into a single Python workflow unlocks insights into live-cell development for a wide range of users. In particular, master students have been able to adapt the workflows in a very short time to analyze experiments and extract quantitative information at population and single-cell levels. These acia-workflows represent a solid toolset that equips the high-throughput capabilities of MLCI with the necessary data analysis methods to fully exploit its high-throughput potential. Thus, gaining robust insights into single-cell behavior, especially for investigating single-cell heterogeneity, cell interactions in co-cultures, or simple growth behavior, has now become an everyday application and lead to the automated analysis of more than one billion cells over the four year period of this thesis.

10.2 Smart MLCI experimentation

Inspired by the immense potential in post-experiment data analysis, the second part of this thesis focused on establishing a smart MLCI experimentation platform that makes decisions based on real-time information during the experiment. While these so-called event-driven imaging systems have already been used before, our combination with an in-silico simulation platform, the operating system perspective, and the establishment of a smart experimentation workflow unlocks completely new live-cell experiments at high-throughput. These achievements represent a paradigm shift from passively observing cell development to actively influencing living cells and provoking their responses.

In the process of developing sMLCI, we had to establish important technological developments, including the DMC and JediOS. The development of the DMC enabled us to gain real-time insights, regardless of the underlying microfluidic structures used to cultivate cell populations. Moreover, the concept of mapping all RoI positions makes it possible to efficiently exploit the high-throughput capabilities in practice and speed up the experiment setup. Thus, the DMC represents the technological cornerstone for robust real-time experimentation in MLCI and has the potential to also influence other microfluidic applications, such as organ-on-chip devices [180].

Recognizing the potential of real-time information processing, we developed an operating system perspective on high-throughput MLCI and a divide-and-conquer approach for implementing imaging applications within each RoI. Together with scheduling imaging acquisition, this builds the foundation of JediOS. Across various scenarios, we demonstrated the simplicity and flexibility of the EDIL to program the logic for these event-driven experiments. Moreover, the complete in-silico simulation of high-throughput experiments laid the foundation for understanding all aspects of the experiment without touching a microscope. Especially the opportunity to understand the interdependence of imaging rate, throughput, and state-machine design allows us to design, improve, and verify the implementation of the experiment without any real execution. This is particularly important for handling the increased complexity of event-driven experiments and ensuring an effective and error-free execution of the experiments. Moreover, we have demonstrated in simulations that event-driven

imaging applications have the potential to further increase throughput due to the asynchronous occurrence of events and the differences in cell population development. While the cell population models used in the simulation are basic for now, their extension may lead to a closed-loop improvement: every real experiment enhances the understanding of the underlying cellular behavior. It enables the development of better models, ultimately leading to a more realistic simulation.

Based on these technological advantages that enable real-time insights and decisions, the last two chapters of this thesis are dedicated to presenting real experiments and highlighting the new potential for gaining insights. Therefore, we have developed a high-throughput screening of so-called PSs that are used to induce intracellular stress using light pulses. With the development of a smart experimentation workflow, we have shown that all experiment decisions are now performed and documented in Jupyter notebooks. This software implementation comes with several advantages: (1) experiments are easier to reproduce as the workflow can just be re-excuted and applies the same real-time decisions based on the biological development; (2) the workflow can be easily adapted to other experiments or research hypotheses even during the running experiment; (3) it becomes much easier to share and extend these experiment workflows to develop the experiments of the future. Especially the smart selection of RoI positions has shown to be crucial to recording high-quality data as it selects the cell populations that are then observed or influenced during the experiment.

In the detailed data analysis of the screening experiment, we apply the acia-workflow framework to investigate the cell behavior at different energies of the EP. We are the first to show that the PSs have a dynamic range that allows to provoke a single-cell response ranging from slight growth reduction to complete cell death. Moreover, we demonstrated that the fluorescence response is a suitable indicator of the induced stress level and is directly linked to the development of both population and single-cell offspring. Gaining such insights within a single MLCI experiment is unprecedented and was unthinkable before. Thus, sMLCI is an important contribution to screening the usability of optogenetic tools for their potential in closed-loop cybergenetic applications.

The tool character of this work lays the foundation of applying sMLCI in the future to precisely investigate single-cell influences. While the developed tools have been applied to specific examples, their general applicability is quite broad, especially due to their machine-learning foundation. For example, we have primarily focused on prokaryotic cells, specifically bacteria, as they allow for very short experiments benefiting the continuous development of the tools. However, MLCI is also applicable to eukaryotic cells [139] such that retraining of the deep learning approaches has the potential to establish similar workflows and smart experiments in that domain. Nevertheless, eukaryotic cells present their own challenges, including phototoxicity, the need for extensive cultivation, longer experiments, and risk of contamination with bacteria, which necessitate further development of ”smart” procedures (Section 10.7).

10.3 Towards cybergenetic growth control using photosensitizers

During the PS screening experiment, we primarily focused on investigating the impact of the EPE parameter on cell development. We designed the Optogenetic single-cell screening state-machine (SM3)* design to screen different EPEs precisely when 10% of the RoI was filled with cells. Now that we quantified the effect of a single EP, this raises the question of how the cells respond to

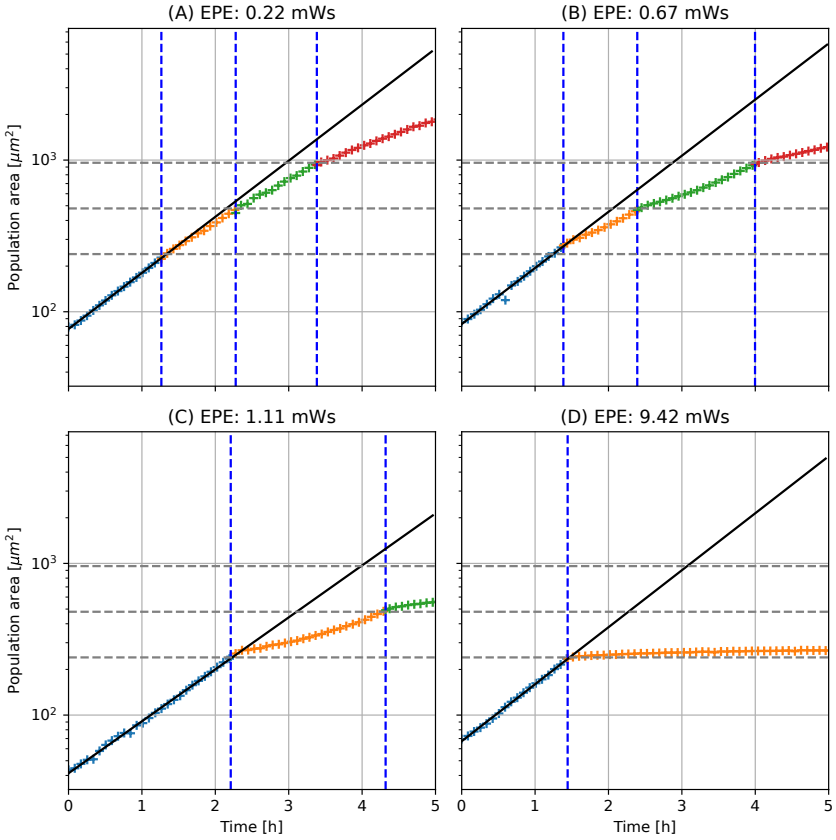

Figure 10.1: **Smart experiment with triple EP stress-induction.** Triple EP application to four cell populations excited with rising EPEs (A-D). The cell population development is measured in TSCA, and the EP is triggered at thresholds of 5 %, 10 %, and 20 % TSCA of the total RoI area (gray dashed line). EP executions are marked as dashed blue lines. The time series are divided into four sequences: before the EP (light blue), after the first EP (orange), after the second EP (green), after the third EP (red). The straight black line indicates the exponential growth of the cell population without any EP extrapolated based on the growth in the first sequence.

multiple EPs and whether the impact of the EP is additive in its energy such that two EPs lead to the same impact as one EP as long as their total EPE is the same.

The EDIL allows the design of advanced imaging applications that effectively investigate these research questions using the JediOS platform. Figure 10.1 shows the preliminary evaluation of four cell populations excited multiple times, each with a specific EPE. We observe that each EP leads to a stronger deviation from the exponential cell population development (black line), indicating that every EP induces additional intracellular stress reducing cell growth.

The immense flexibility in programming the PS experiment using EDIL on the JediOS platform allows for precise investigation of these research questions, designing experiments with multiple excitation pulses executed at different population sizes, and allows to render a very detailed picture of the dynamic PS influence. While the analysis of these experiments is quite fast and can be performed overnight, it shifts the challenge to determining which set of parameters to explore next,

discussed further in Section 10.8.

The dynamic range of the PSs makes them ideally suited for closed-loop control applications. Lugange et al. [37] have already demonstrated optogenetic closed-loop control of protein secretion with a binary influence, that is, either excitation or no excitation. The fluorescence readout provided by the PS activation may improve the control strategy here: Instead of using a binary influence, the precise EPE can be computed to bring the controlled parameter, e.g. cell stress or growth, into a specific target stress level or growth rate. The fluorescence readout confirms the precise influence on individual cells, which now opens up new opportunities in the field of cybergenetics [159]. MLCI, with its combination of high-throughput and immense detail, fosters our understanding of single-cell variability and is ideally suited to further explore these control methods.

For example, the SM4 design can be used with this PS setup to control the growth rate of a cell population precisely. The main limitation of the PSs is that the stress induction only reduces the cell growth. Thus, only cell growth rates lower than the default growth rate can be approached. PS-based growth control application is further complicated by the change in morphology and seemingly structural damage that the stress-induction causes inside the cells. Therefore, these effects need to be further studied and screened before applying these in closed-loop cybergenetic applications.

10.4 Screening additional cell influencing factors

While we have shown the high-throughput screening capabilities for optogenetic light influence, it can be extended to other influences such as dynamic adaption of the medium [27], global illumination [123] or gassing [35]. However, these influences differ in granularity: light exposure through the microscopy objective only affects the cell population within a single RoI, leading to fine-grained control on the RoI level. However, the supplied medium, for example, is controlled per microfluidic channel and affects 10s to 100s of RoIs located in the same microfluidic channel (medium granularity). Global illumination and gassing then affect all the RoIs on a microfluidic chip (coarse granularity).

To incorporate the granularity of different influences, we propose extending the state-machine model into a hierarchical structure in the future (Figure 10.2). Thus, state-machines that are jointly affected are grouped together into a RoI group that is controlled by an additional state-machine at a higher hierarchical level. Thus, specific events are triggered at this higher hierarchical level, and the conditions for these triggers can then incorporate information about multiple RoIs and feature commands, such as ”change the medium if 10 RoIs of this group have more than 10 cells”. The downside of such a hierarchical approach is the tremendous increase in complexity: decisions are performed at different hierarchical levels and in different state-machines. However, the in-silico simulation of the JediOS platform allows for mastering the complexity by simulating hierarchical decisions prior to the experiment.

10.5 The costs of MLCI experimentation

Our advancements and contributions in analysis tools and smart experimentation software have demonstrated that sMLCI is capable of extracting unique insights while increasing accessibility with acia-workflow and JediOS. Moreover, these software contributions are open-source and available to

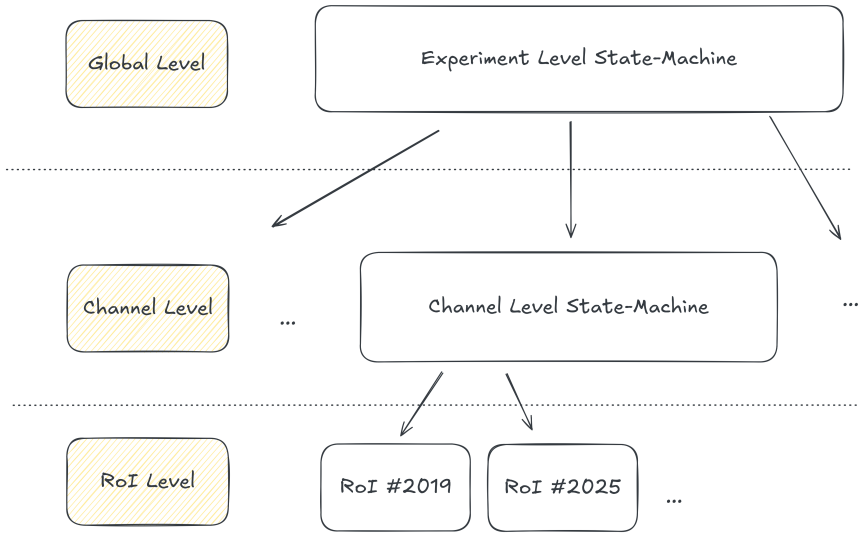

Figure 10.2: **Hierarchical state-machines for granularity based influence.** State-machine hierarchy to manage decisions at different granularity levels. Decisions on the global level affect all state-machines. Decisions on the channel level affect all state-machines whose RoIs are located in the same microfluidic channel. Decisions on the RoI level only affect a single cell population and have been presented in this thesis.

everyone, drastically reducing the entry barrier into sMLCI. However, one main barrier remains: all these single-cell insights are recorded on professional microscopy hardware. The optics, the 3D movable stage with precise focusing, and the device body easily lead to costs exceeding $100K$ €. Thus, these expensive and large setups are only available to a small number of well-funded labs and research institutions, making sMLCI a niche application that is difficult to establish for a broad user base.

This situation contradicts our vision that sMLCI and our developed tools open the broad application of MLCI to investigate single-cell behavior for various biotechnological and biomedical applications. Thus, for opening sMLCI to a broader research community, we need more affordable imaging setups. Consequently, we raise the question: How affordable can we make an sMLCI setup?

In recent years, the open microscopy community has presented numerous approaches for building open-hardware microscopy setups [181, 182, 183, 184]. These are usually assembled using widely available parts and are significantly more affordable and customizable than off-the-shelf setups. Auspicious is the development of ultra-low-cost microscopy devices. For example, the Open Flexure Microscope (OFM) [181] is a microscope constructed with low-cost electrical components and a largely 3D-printed microscope body. Thus, building such a microscope usually costs less than 300€, components can be easily repaired, and it provides a 3D movable stage.

Thus, we decided to build such an OFM, mounted the microfluidic chip and conducted an exemplary experiment cultivation *C. glutamicum* in BHI medium. Due to the modularity of the JediOS platform and the available open-source interfaces to the OFM platform, we simply implemented the OFM as a new microscopy platform in JediOS.

Figure 10.3 shows the microscope and the conduction of a high-throughput MLCI experiment.

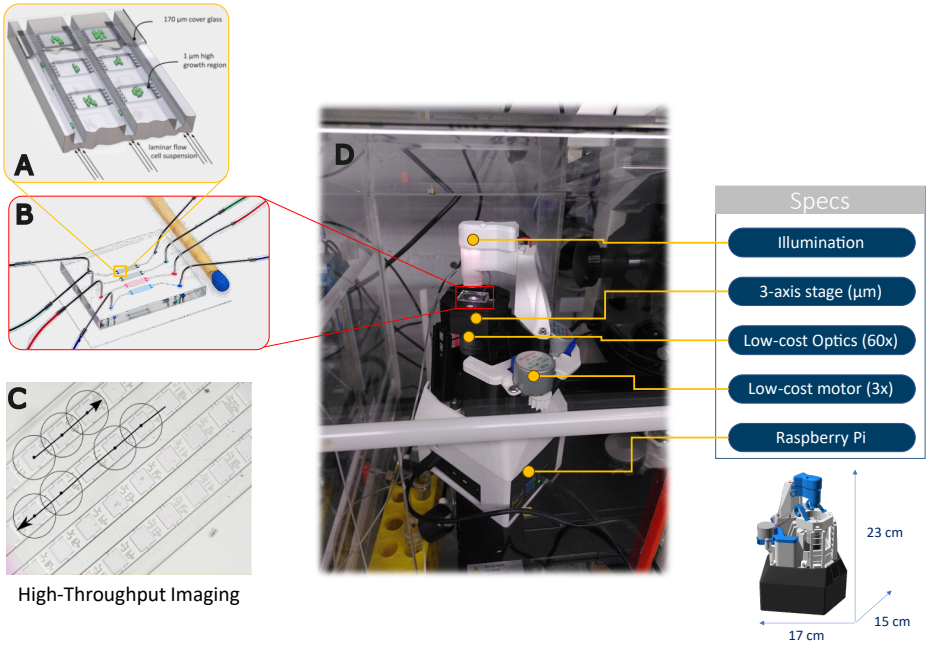

Figure 10.3: **Open Flexure Microscope for microfluidic live-cell imaging.** The RoIs (A) on the microfluidic chip (B) are mounted on the 3D printed stage of the OFM (D), and high-throughput imaging is performed (C) where multiple RoIs are imaged in a loop and the circles denote the field of view of the objective. The 3D-printed microscope is placed inside the cultivation chamber of its larger microscopy counterparts to guarantee the same temperature conditions.

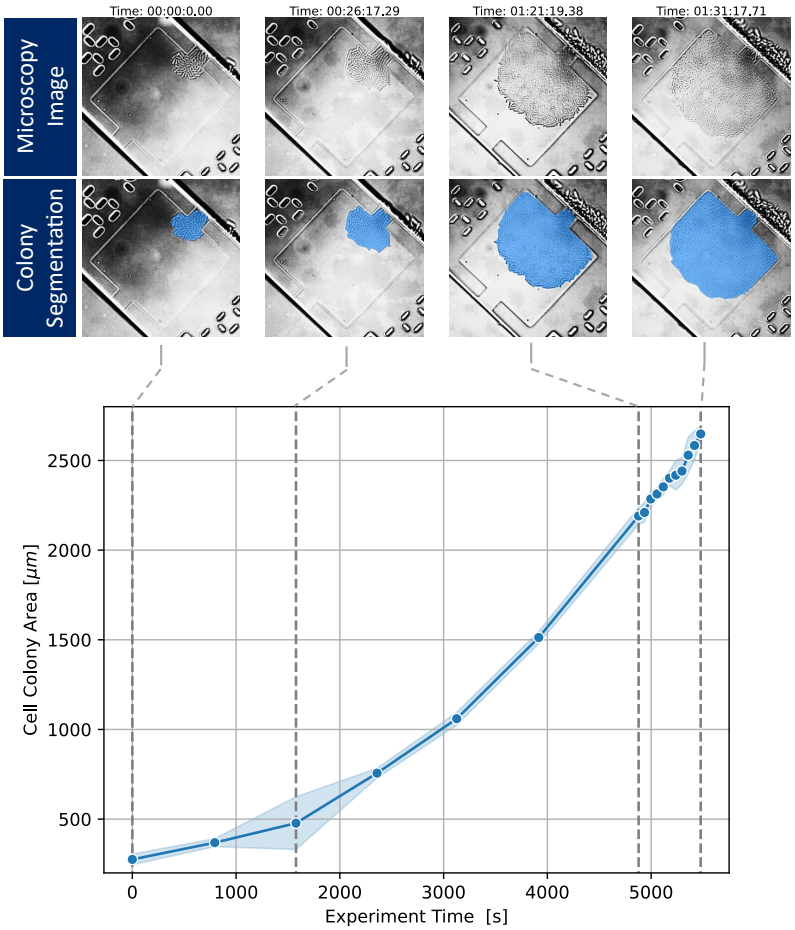

Figure 10.4: **Recorded and analyzed images of cultivation on the ultra-low-cost platform.** The microscope images are segmented (top), and the TCA development is extracted (bottom). Due to focus issues, a z-stack of images is recorded and all the images are analyzed resulting in the average TCA (blue line) and one standard deviation (blue shade).

For the sake of simplicity, we placed it into the heated chamber of our usual Nikon TI microscope to have comparable heating conditions. Figure 10.4 shows the TCA development of a cell population cultivated in this experiment. The low-cost motors and 3D-printed platform lead to challenges with the focusing. Thus, at every RoI position we have recorded a z-stack covering a wider focus range. All the images of a z-stack are analyzed and their mean and standard deviation is shown. Thus, the temporal development of a cell population can also be measured with such an ultra-low-cost microscope.

While the OFM platform suffers from challenges with focus and is much slower in its movements due to the cheap motor hardware and body components, it can be built almost everywhere around the earth, comes with all the essential functionality to execute sMLCI experiments, is much smaller and cheaper, and thus especially suitable to enter the domain of MLCI and execute simple experiments such as quantifying growth rates. Using such ultra-low-cost microscopy platforms has the potential to reduce the costs of the imaging setup by a factor of 100-1000 and make sMLCI available worldwide

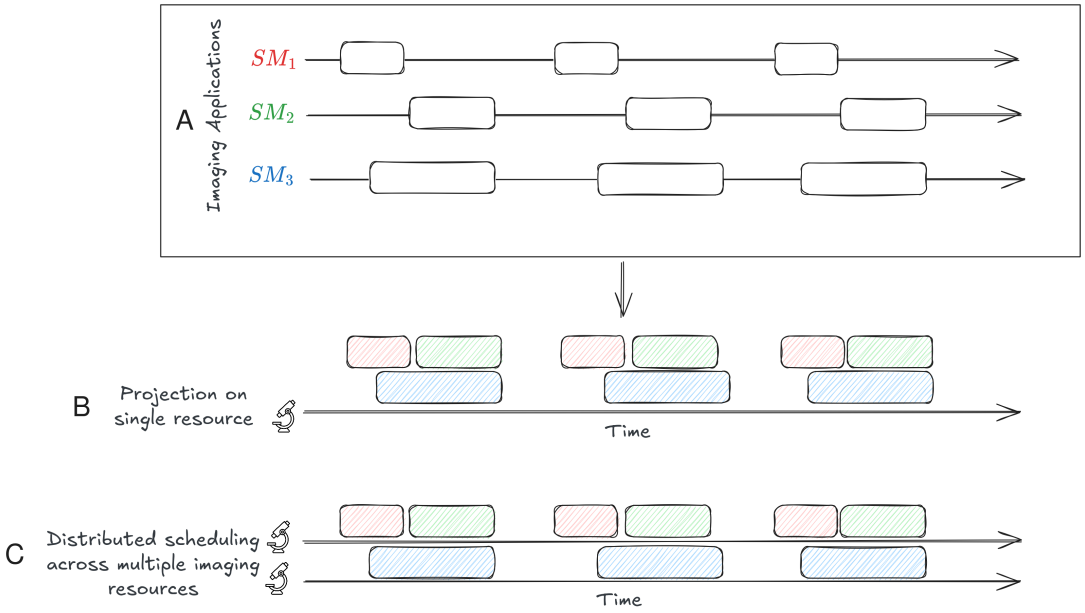

Figure 10.5: **Distributed sMLCI experimentation with multi-microscopy setups.** Some experiments may require a higher amount of imaging resources than are available on a single microscope, as denoted by the three imaging applications (A) requesting imaging resources simultaneously (B). Distributing their scheduling across two microscope setups allows them to find a valid scheduling and conduct the experiment as planned (C).

with open hardware and software.

10.6 Distributed experimentation in multi-microscope setups

During our high-throughput simulation studies, we have shown that the flexible imaging application programming, for example, adapting the imaging rate, can exceed the capacities of the microscope setup, causing delays and a drop in imaging rate. The typical first reaction to this effect is to either reduce the imaging rate requirements or decrease the number of concurrently observed cell populations to minimize the utilization of the microscope. However, the operating system perspective opens up another interesting opportunity: In modern computers CPUs usually contain multiple compute cores such that they can perform multiple compute operations concurrently. Analogously, we can also think of a JediOS setup with multiple microscopes. Instead of restricting throughput, we distribute the experiment across multiple microscopes to increase imaging capabilities. We call this a ”distributed experiment”, and similar to the multi-core term for CPUs, we call such a setup a ”multi-microscope” setup (Figure 10.5).

Interestingly, we can even combine different types of microscope setups to measure specific information in a single experiment. For example, we can use two 3D-printed microscopes to determine basic growth rates and then utilize a fluorescence microscope to perform fluorescence measures. Such a distributed experiment does not require only small changes to the JediOS platform. Essentially, we need to associate imaging applications not only with the RoI ID but also with the target microscope

and then asynchronously run multiple schedulers that deal with the scheduling on the individual microscopes. A suitable distribution of imaging applications across the multi-microscope setup is then performed in the sMLCI workflow and supervised by the user.

This vision of distributed experiments breaks the barrier of limited microscopy resources and opens up the opportunity to uncover the hidden secrets of living cells within large-scale microscopy farms combining 10s to 100s of devices across smart live-cell experiments.

10.7 Beyond bacteria towards eukaryotic cells

Throughout this thesis, we have developed general analysis tools and the sMLCI platform at the example of cultivating and investigating bacterial cells. There are primarily two reasons for choosing bacterial cells: their cultivation is comparatively simple, and they grow extremely fast, leading to shorter experiments. Especially, the rapid growth of bacterial cells with division times as short as 20 minutes [185] allows for conducting many experiments, which has proven very important in testing the JediOS platform and sMLCI experiments in practice.

However, with a mature platform that has been proven robust in real-world experimentation, sMLCI is also applicable to eukaryotic cells. For example, Schmitz et al. [139] have performed the cultivation of Chinese hamster ovary (CHO) cells in microfluidic chip systems. However, cultivating eukaryotic cells comes with its own challenges, including increased phototoxicity, multi-day experiments, and the risk of contamination, where bacteria enter the microfluidic device and contaminate the cultivated cells. Due to the rapid growth of bacteria, such contamination typically leads to the failure of the experiment [139].

However, especially in these longer experiments, real-time information and making smart decisions along the experiment become even more important. For example, contamination could be automatically detected, most likely occurring at the in and out-lets of the microfluidic chips. We can envision a light-based sterilization procedure where areas of contamination are exposed to high doses of ultraviolet light to kill all cells in these areas and protect the non-contaminated cell populations. Thus, such smart procedures may have an even stronger impact in the domain of longer and more expensive experiments, serving as a gateway to unlocking experiments with unprecedented data quality and insights.

10.8 Autonomous experiment design for high-throughput screenings

The high-throughput screening of PSs using various EPEs has demonstrated the immense opportunities of exploiting high-throughput in sMLCI and quickly gaining insights into single-cell responses. However, this screening has primarily investigated one parameter, the EPE. By extending the investigation to multiple pulses, their time interval, or the size of the cell population, the screening needs to explore a multi-dimensional parameter domain. While such experiments are possible to design in JediOS, sampling such a multi-dimensional domain of experiment parameters uniformly leads to a tremendous amount of experiments and becomes much more time-consuming. Thus, in addition to conducting smart experiments making real-time decisions, autonomous experiment design will become increasingly important in the future. During autonomous experiment design a

software recommends a set of suitable experiment parameters to investigate based on a model. The experiment is conducted, and the model is update with the results. This loop is repeated to explore the experiment parameter space efficiently. Similar concepts have been shown to efficiently and autonomously optimize the cultivation conditions of specific cells [186] or explore the protein fitness landscape [187]. With the contributions in this thesis, we have laid the foundation for performing such autonomous experiments in sMLCI in the future while benefiting from the high density of single-cell information and the flexibility in experiment design.

Chapter 11

Conclusion

Microbial organisms play essential roles in human health, the Earth's ecosystem, and sustainable biotechnological production. Their behavior is remarkably diverse and dynamic, shaped by heterogeneity, environment conditions, and cell-to-cell interactions. To understand and predict how these organisms and their communities function in complex environments, it is crucial to monitor microbial development under precisely controlled yet dynamic conditions and at single-cell resolution. MLCI has emerged as a powerful technology to meet these needs, offering high-throughput capabilities, precise dynamic control over environmental conditions, such as medium, gassing, or optogenetic stimulation, and single-cell resolution on cell development in space and time.

However, the full potential of MLCI has been held back by ongoing challenges in extracting valuable insights from large volumes of recorded live-cell imaging data. This thesis presented tools that overcome these limitations of MLCI, performing a paradigm shift in analysis and smart experimentation having the potential to better understand the complex interactions of living cells in space and time.

In the first part of this thesis, we demonstrated that data-driven image analysis, powered by large-scale datasets, is the cornerstone for robustly gaining single-cell insights. Our fully automated image analysis workflows enabled overnight high-throughput single-cell quantification combined with custom data analysis and visualization. These capabilities fill a critical gap, serving as the digital analysis backbone accelerating insight generation based on MLCI experiments. Together, these developments enable researchers to effectively harness single-cell data to address complex questions about microbial communities and to decode spatio-temporal behaviors in dynamically changing environments. This is underpinned by the more than one billion cells that have been analyzed over the period of this thesis.

In the second part, we introduced new opportunities for event-driven experimentation at high-throughput. The development of the DMC concept and its tight integration with the image analysis pipeline made information available in real-time during the experiment. Developing the JediOS unlocked adaptive, event-driven experimental designs flexibly programmed using the EDIL. Simulating such experiments in our simulation framework showed that these event-driven experiments not only enhance the temporal resolution and throughput of screenings but also enable the precise triggering of optogenetic responses. Integrating JediOS into a smart experimentation workflow leads to a high degree of experimentation automation, autonomous decisions and events during the experiment and

simplifies repetition and adaptation of experiment designs. We demonstrated the capabilities of these sMLCI workflows at the example of a high-throughput screening of photosensitizer excitation using excitation pulses to evaluate its suitability for closed-loop cell growth control. We are the first to quantify the dynamic range of stress-induction using various EPEs, the induction of stress on the single-cell level, and to link stress-induction to the single-cell development. Moreover, all these insights have been generated within a single experiment taking a single day - demonstrating the efficiency and depth of insights unlocked by sMLCI.

Altogether, this work provides a comprehensive platform that forms the foundation for event-driven MLCI experimentation unlocking accelerated life science discovery. Fusing the live-cell imaging setups and deep learning-based analysis into one digital platform enhances both the usability and accessibility of the technology, allowing overnight analysis of millions of individual cells. In particular, the smart experimentation paradigm opens new frontiers for controlling and studying cellular behavior at high-throughput. Moreover, the ability to generate vast datasets, makes sMLCI a promising technology for building predictive deep learning models based on systematic screenings of the cell behavior.

As a result, the joint development of MLCI, data-driven image analysis, and smart experimentation has the potential to accelerate life-cell imaging research, empowering single-cell observations in complex environments and dynamic influences at an unprecedented scale and resolution.

List of Figures

1.1 Schematic conduction of a high-throughput MLCI experiment. 3
1.2 Typical MLCI image analysis pipeline. 4
1.3 High-throughput microfluidic cultivation chips - From design to manufacturing. . . . 7
1.4 Structure of this thesis 11

2.1 ObiWan-Microbi platform overview 17
2.2 Overview of *AnnUI* functionality. 19
2.3 The *ObiWan-Microbi* microservice architecture. 21
2.4 Application of DL segmentation methods. 22

3.1 Data acquisition in Microfluidic Live-Cell Imaging 27
3.2 Temporal imbalance of the five microbial time-lapse recordings of the benchmark dataset. 32
3.3 Subsampling and cell count limiting of time-lapse sequences. 33
3.4 The *EATM* based on the *DIV* metric measured across different imaging intervals and cell count limits. 36

4.1 Acia-workflows overview. 41
4.2 Data extraction with acia-workflows. 45
4.3 Three variants to quantify colony growth rates. 47
4.4 Co-culture cultivation with fluorescence-based strain labeling. 50
4.5 Single-cell insights under oxygen shifts. 51
4.6 Scaling MLCI analysis to multiple replicates. 53

5.1 Screenshots of the developed interactive visualizations. 58
5.2 Event-driven imaging for smart microscopy experiments. 62
5.3 Schematic visualization of the just event-driven imaging operating system (JediOS) . 63

6.1 The digital microfluidic chip. 67
6.2 CAD design of the swiss army knife chip. 69
6.3 Evaluation of the YOLO model configurations and marker detection quality. 72
6.4 Evaluation of the YOLO model configurations for detecting microfluidic reference markers. 73
6.5 Full RoI masking pipeline speed. 75
6.6 Masked RoIs on the SAK. 76

6.7 Map computation procedure. 79
6.8 Map RoI position errors 80

7.1 The high-throughput imaging system - a multi-tasking perspective. 86
7.2 Basic imaging state-machine (SM1) and legend. 88
7.3 Adaptive imaging interval state-machine (SM2). 90
7.4 Optogenetic single-cell screening state-machine (SM3). 92
7.5 Closed-loop optogenetic growth control state-machine (SM4). 93
7.6 The scheduled execution of a single state-machine. 94
7.7 Scheduled execution of multiple state-machines on a single microscope. 95
7.8 High-throughput simulation of standard imaging experiment (SM1). 100
7.9 High-throughput simulation of adaptive imaging rates (SM2). 103
7.10 High-throughput simulation of optogenetic influence screening. 105

8.1 REST API for network based microscopy control. 114
8.2 Design and implementation of the real-time image analysis pipeline. 116
8.3 Images flowing through the three nodes of the real-time image analysis pipeline. . . . 117
8.4 Processing time of the individual real-time pipeline steps. 118
8.5 Light-based intracellular stress induction using photosensitzers. 119
8.6 State-machine SM3* design for high-throughput excitation pulse screening. 120
8.7 High-throughput microfluidic chip map creation. 122
8.8 Checkerboard pattern on the microfluidic chip map. 123
8.9 Real-time dashboard during the initial screening procedure (t=47min 23s). 124
8.10 Dashboard of the selected RoI candidates (t=47min 23s). 126
8.11 Assignment of imaging applications to RoI positions on the microfluidic chip map. . 127
8.12 Live dashboard of the temporal development of all cell populations in the PS screening experiment. 129
8.13 Live dashboard showing the ”task manager” of the JediOS platform during the experiment. 130

9.1 Photosensitizer excitation and fluorescence emission imaging. 138
9.2 Excitation light source power calibration. 139
9.3 Average cell fluorescence response based on the excitation pulse energy. 141
9.4 Heterogeneity in single-cell fluorescence intensity. 142
9.5 Temporal cell population development with stress induction 144
9.6 Three population based measurements to quantify temporal influence of stress-induction. 146
9.7 Quantification of the cell population response to stress induction based on the PDT, PGF and PPDD measures. 148
9.8 Abnormal cell morphology caused by stress-induction. 149
9.9 Final images of the four positive control group cell populations. 150
9.10 Three single-cell based measurements CDT, CGF, PCDD to quantify temporal influence of stress-induction. 151
9.11 Three measures CDT, CGF, and PCDD for single-cell stress impact quantification. . 151
9.12 Lineage tree of a stress-induced cell population. 153
9.13 Lineage-based measures for single-cell offspring quantification. 154

10.1 Smart experiment with triple EP stress-induction. 160
10.2 Hierarchical state-machines for granularity based influence. 162
10.3 Open Flexure Microscope for microfluidic live-cell imaging. 163
10.4 Recorded and analyzed images of cultivation on the ultra-low-cost platform. 164
10.5 Distributed sMLCI experimentation with multi-microscopy setups. 165

List of Tables

3.1 Amount of manual segmentation and tracking annotation corrections. 31
3.2 Tracking benchmark dataset split statistics. 31
3.3 Robustness metric evaluations. 36

8.1 Screening parameters and number of monitored RoIs. 128

9.1 Excitation screening parameters and excitation pulse energies. 140

Bibliography

[1] Ron Sender, Shai Fuchs, and Ron Milo. Revised Estimates for the Number of Human and Bacteria Cells in the Body. *PLOS Biology*, 14(8):e1002533, 2016. doi: 10.1371/journal.pbio.1002533.

[2] Curtis Huttenhower, Dirk Gevers, Rob Knight, Sahar Abubucker, Jonathan H. Badger, Asif T. Chinwalla, Heather H. Creasy, Ashlee M. Earl, Michael G. FitzGerald, Robert S. Fulton, et al. Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207–214, 2012. doi: 10.1038/nature11234.

[3] Tenzin Choden and Nathaniel Aviv Cohen. The gut microbiome and the immune system. *Exploration of Medicine*, 3(3):219–233, 2022. doi: 10.37349/emed.2022.00087.

[4] Xue-Er Zhang, Pai Zheng, Sheng-Zhen Ye, Xiao Ma, E. Liu, Yao-Bin Pang, Qing-Ying He, Yu-Xiao Zhang, Wen-Quan Li, Jin-Hao Zeng, and Jing Guo. Microbiome: Role in Inflammatory Skin Diseases. *Journal of Inflammation Research*, 17:1057–1082, 2024. doi: 10.2147/JIR.S441100.

[5] Tiffany C. Scharschmidt and Julia A. Segre. Skin microbiome and dermatologic disorders. *The Journal of Clinical Investigation*, 135(3), 2025. doi: 10.1172/JCI184315.

[6] Wing Ho Man, Wouter A. A. de Steenhuijsen Piters, and Debby Bogaert. The microbiota of the respiratory tract: gatekeeper to respiratory health. *Nature Reviews Microbiology*, 15(5):259–270, 2017. doi: 10.1038/nrmicro.2017.14.

[7] Huanrong Li, Yuqin Zang, Chen Wang, Huiyang Li, Aiping Fan, Cha Han, and Fengxia Xue. The Interaction Between Microorganisms, Metabolites, and Immune System in the Female Genital Tract Microenvironment. *Frontiers in Cellular and Infection Microbiology*, 10, 2020. doi: 10.3389/fcimb.2020.609488.

[8] Paola Villa, C Cipolla, Silvia D'Ippolito, Inbal Dona Amar, M Shachor, F Ingravalle, Franco Scaldaferri, P Puca, Nicoletta Di Simone, Giovanni Scambia, et al. The interplay between immune system and microbiota in gynecological diseases: a narrative review. *European review for medical and pharmacological sciences*, 24(10):5676–5690, 2020. doi: 10.26355/eurrev_202005_21359.

[9] Anna Grenda, Ewelina Iwan, Barbara Kuźnar-Kamińska, Arkadiusz Bomba, Katarzyna Bielińska, Pawel Krawczyk, Izabela Chmielewska, Malgorzata Frak, Michal Szczyrek, Anna

Rolska-Kopińska, Tomasz Jankowski, Robert Kieszko, and Janusz Milanowski. Gut microbial predictors of first-line immunotherapy efficacy in advanced NSCLC patients. *Scientific Reports*, 15(1):6139, 2025. doi: 10.1038/s41598-025-89406-1.

[10] M. Vouga and G. Greub. Emerging bacterial pathogens: the past and beyond. *Clinical Microbiology and Infection*, 22(1):12–21, 2016. doi: 10.1016/j.cmi.2015.10.010.

[11] David Shaw, Raquel Abad, Zahin Amin-Chowdhury, Adriana Bautista, Desiree Bennett, Karen Broughton, Bin Cao, Carlo Casanova, Eun Hwa Choi, Yiu-Wai Chu, et al. Trends in invasive bacterial diseases during the first 2 years of the COVID-19 pandemic: analyses of prospective surveillance data from 30 countries and territories in the IRIS Consortium. *The Lancet Digital Health*, 5(9):e582–e593, 2023. doi: 10.1016/S2589-7500(23)00108-5.

[12] Peter J. Turnbaugh, Micah Hamady, Tanya Yatsunenko, Brandi L. Cantarel, Alexis Duncan, Ruth E. Ley, Mitchell L. Sogin, William J. Jones, Bruce A. Roe, Jason P. Affourtit, Michael Egholm, Bernard Henrissat, Andrew C. Heath, Rob Knight, and Jeffrey I. Gordon. A core gut microbiome in obese and lean twins. *Nature*, 457(7228):480–484, 2009. doi: 10.1038/nature07540.

[13] Junjie Qin, Ruiqiang Li, Jeroen Raes, Manimozhiyan Arumugam, Kristoffer Solvsten Burgdorf, Chaysavanh Manichanh, Trine Nielsen, Nicolas Pons, Florence Levenez, Takuji Yamada, et al. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464(7285):59–65, 2010. doi: 10.1038/nature08821.

[14] David N. Fredricks, Tina L. Fiedler, and Jeanne M. Marrazzo. Molecular Identification of Bacteria Associated with Bacterial Vaginosis. *New England Journal of Medicine*, 353(18):1899–1911, 2005. doi: 10.1056/NEJMoa043802.

[15] Zhongbiao Fang, Jingting Lyu, Jianhua Li, Chaonan Li, Yuxuan Zhang, Yikai Guo, Ying Wang, Yanjun Zhang, and Keda Chen. Application of bioreactor technology for cell culture-based viral vaccine production: Present status and future prospects. *Frontiers in Bioengineering and Biotechnology*, 10, 2022. doi: 10.3389/fbioe.2022.921755.

[16] Volker F Wendisch. Microbial production of amino acids and derived chemicals: Synthetic biology approaches to strain development. *Current Opinion in Biotechnology*, 30:51–58, 2014. doi: 10.1016/j.copbio.2014.05.004.

[17] Rajwinder Kaur, Loveena Pathak, and Pratibha Vyas. Biobased polymers of plant and microbial origin and their applications - a review. *Biotechnology for Sustainable Materials*, 1(1):13, 2024. doi: 10.1186/s44316-024-00014-x.

[18] Marianna Arvaniti and Panagiotis N Skandamis. Defining bacterial heterogeneity and dormancy with the parallel use of single-cell and population level approaches. *Current Opinion in Food Science*, 44:100808, 2022. doi: 10.1016/j.cofs.2022.100808.

[19] Alexander Grünberger, Christopher Probst, Stefan Helfrich, Arun Nanda, Birgit Stute, Wolfgang Wiechert, Eric von Lieres, Katharina Nöh, Julia Frunzke, and Dietrich Kohlheyer. Spatiotemporal microbial single-cell analysis using a high-throughput microfluidics cultivation platform. *Cytometry Part A*, 87(12):1101–1115, 2015. doi: 10.1002/cyto.a.22779.

[20] Mojca Mattiazzi Usaj, Clarence Hue Lok Yeung, Helena Friesen, Charles Boone, and Brenda J. Andrews. Single-cell image analysis to explore cell-to-cell heterogeneity in isogenic populations. *Cell Systems*, 12(6):608–621, 2021. doi: 10.1016/j.cels.2021.05.010.

[21] Sarah Täuber, Julian Schmitz, Luisa Blöbaum, Niklas Fante, Heiko Steinhoff, and Alexander Grünberger. How to Perform a Microfluidic Cultivation Experiment—A Guideline to Success. *Biosensors*, 11(12):485, 2021. doi: 10.3390/bios11120485.

[22] Paige Allard, Fotini Papazotos, and Laurent Potvin-Trottier. Microfluidics for long-term single-cell time-lapse microscopy: Advances and applications. *Frontiers in Bioengineering and Biotechnology*, 10, 2022.

[23] Sarah Täuber and Alexander Grünberger. Microfluidic single-cell scale-down systems: introduction, application, and future challenges. *Current Opinion in Biotechnology*, 81:102915, 2023. doi: 10.1016/j.copbio.2023.102915.

[24] Huabing Yin and Damian Marshall. Microfluidics for single cell analysis. *Current Opinion in Biotechnology*, 23(1):110–119, 2012. doi: 10.1016/j.copbio.2011.11.002.

[25] Markus Leygeber, Dorina Lindemann, Christian Carsten Sachs, Eugen Kaganovitch, Wolfgang Wiechert, Katharina Nöh, and Dietrich Kohlheyer. Analyzing Microbial Population Heterogeneity—Expanding the Toolbox of Microfluidic Single-Cell Cultivations. *Journal of Molecular Biology*, 431(23):4569–4588, 2019. doi: 10.1016/j.jmb.2019.04.025.

[26] Simone Schito, Rico Zuchowski, Daniel Bergen, Daniel Strohmeier, Bastian Wollenhaupt, Philipp Menke, Johannes Seiffarth, Katharina Nöh, Dietrich Kohlheyer, Michael Bott, Wolfgang Wiechert, Meike Baumgart, and Stephan Noack. Communities of Niche-optimized Strains (CoNoS) – Design and creation of stable, genome-reduced co-cultures. *Metabolic Engineering*, 73:91–103, 2022. doi: 10.1016/j.ymben.2022.06.004.

[27] Sarah Täuber, Corinna Golze, Phuong Ho, Eric von Lieres, and Alexander Grünberger. dMSCC: a microfluidic platform for microbial single-cell cultivation of Corynebacterium glutamicum under dynamic environmental medium conditions. *Lab on a Chip*, 20(23):4442–4455, 2020. doi: 10.1039/D0LC00711K.

[28] Stephan Endres, Marcus Wingen, Joaquim Torra, Rubén Ruiz-González, Tino Polen, Gabriela Bosio, Nora Lisa Bitzenhofer, Fabienne Hilgers, Thomas Gensch, Santi Nonell, Karl-Erich Jaeger, and Thomas Drepper. An optogenetic toolbox of LOV-based photosensitizers for light-driven killing of bacteria. *Scientific Reports*, 8(1):15021, 2018. doi: 10.1038/s41598-018-33291-4.

[29] Stefan Helfrich, Eugen Pfeifer, Christina Krämer, Christian Carsten Sachs, Wolfgang Wiechert, Dietrich Kohlheyer, Katharina Nöh, and Julia Frunzke. Live cell imaging of SOS and prophage dynamics in isogenic bacterial populations. *Molecular Microbiology*, 98(4):636–650, 2015. doi: 10.1111/mmi.13147.

[30] Kevin J. Cutler, Carsen Stringer, Teresa W. Lo, Luca Rappez, Nicholas Stroustrup, S. Brook Peterson, Paul A. Wiggins, and Joseph D. Mougous. Omnipose: a high-precision

morphology-independent solution for bacterial cell segmentation. *Nature Methods*, pages 1–11, 2022. doi: 10.1038/s41592-022-01639-4.

[31] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, 2021. doi: 10.1038/s41592-020-01018-x.

[32] Eric Upschulte, Stefan Harmeling, Katrin Amunts, and Timo Dickscheid. Uncertainty-Aware Contour Proposal Networks for Cell Segmentation in Multi-Modality High-Resolution Microscopy Images. In *Proceedings of The Cell Segmentation Challenge in Multi-modality High-Resolution Microscopy Images*, pages 1–12, 2023.

[33] Benjamin Gallusser and Martin Weigert. Trackastra: Transformer-based cell tracking for live-cell microscopy. In *In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds) Computer Vision – ECCV 2024. ECCV 2024. Lecture Notes in Computer Science*, volume 15134, 2024. doi: 10.1007/978-3-031-73116-7_27.

[34] Teresa W Lo, Kevin J Cutler, H James Choi, and Paul A Wiggins. Omnisegger: A time-lapse image analysis pipeline for bacterial cells. *PLOS Computational Biology*, 21(5):e1013088, 2025. doi: 10.1371/journal.pcbi.1013088.

[35] Keitaro Kasahara, Johannes Seiffarth, Birgit Stute, Eric von Lieres, Thomas Drepper, Katharina Nöh, and Dietrich Kohlheyer. Unveiling microbial single-cell growth dynamics under rapid periodic oxygen oscillations. *Lab on a Chip*, 25(9):2234–2246, 2025. doi: 10.1039/D5LC00065C.

[36] Keitaro Kasahara, Markus Leygeber, Johannes Seiffarth, Karina Ruzaeva, Thomas Drepper, Katharina Nöh, and Dietrich Kohlheyer. Enabling oxygen-controlled microfluidic cultures for spatiotemporal microbial single-cell analysis. *Frontiers in Microbiology*, 14, 2023.

[37] Jean-Baptiste Lugagne, Caroline M. Blassick, and Mary J. Dunlop. Deep model predictive control of gene expression in thousands of single cells. *Nature Communications*, 15(1):2148, 2024. doi: 10.1038/s41467-024-46361-1.

[38] Sant Kumar, Marc Rullan, and Mustafa Khammash. Rapid prototyping and design of cybergenetic single-cell controllers. *Nature Communications*, 12(1):5651, 2021. doi: 10.1038/s41467-021-25754-6.

[39] Mustafa H. Khammash. Cybergenetics: Theory and Applications of Genetic Control Systems. *Proceedings of the IEEE*, 110(5):631–658, 2022. doi: 10.1109/JPROC.2022.3170599.

[40] Nico Scherf and Jan Huisken. The smart and gentle microscope. *Nature Biotechnology*, 33(8):815–818, 2015. doi: 10.1038/nbt.3310.

[41] Lionel Chiron, Matthias Le Bec, Céline Cordier, Sylvain Pouzet, Dimitrije Milunov, Alvaro Banderas, Jean-Marc Di Meglio, Benoit Sorre, and Pascal Hersen. CyberSco.Py an open-source software for event-based, conditional microscopy. *Scientific Reports*, 12(1):11579, 2022. doi: 10.1038/s41598-022-15207-5.

[42] Zachary R. Fox, Steven Fletcher, Achille Fraisse, Chetan Aditya, Sebastián Sosa-Carrillo, Julienne Petit, Sébastien Gilles, François Bertaux, Jakob Ruess, and Gregory Batt. Enabling reactive microscopy with MicroMator. *Nature Communications*, 13(1):2199, 2022. doi: 10.1038/s41467-022-29888-z.

[43] Alexander Grünberger. *Single cell analysis of microbial production strains in microfluidic bioreactors*. PhD thesis, RWTH Aachen, 2015.

[44] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9(7):676–682, 2012. doi: 10.1038/nmeth.2019.

[45] Chi-Li Chiu, Nathan Clack, and the napari community. napari: a Python Multi-Dimensional Image Viewer Platform for the Research Community. *Microscopy and Microanalysis*, 28(S1): 1576–1577, 2022. doi: 10.1017/S1431927622006328.

[46] Stella Stylianidou, Connor Brennan, Silas B. Nissen, Nathan J. Kuwada, and Paul A. Wiggins. SuperSegger: robust image segmentation, analysis and lineage tracking of bacterial cells. *Molecular Microbiology*, 102(4):690–700, 2016. doi: 10.1111/mmi.13486.

[47] Owen M. O'Connor, Razan N. Alnahhas, Jean-Baptiste Lugagne, and Mary J. Dunlop. DeLTA 2.0: A deep learning pipeline for quantifying single-cell spatial and temporal dynamics. *PLOS Computational Biology*, 18(1):e1009797, 2022. doi: 10.1371/journal.pcbi.1009797.

[48] David R. Stirling, Madison J. Swain-Bowden, Alice M. Lucas, Anne E. Carpenter, Beth A. Cimini, and Allen Goodman. CellProfiler 4: improvements in speed, utility and usability. *BMC Bioinformatics*, 22(1):433, 2021. doi: 10.1186/s12859-021-04344-9.

[49] Martin Maška, Vladimír Ulman, Pablo Delgado-Rodriguez, Estibaliz Gómez-de Mariscal, Tereza Nečasová, Fidel A. Guerrero Peña, Tsang Ing Ren, Elliot M. Meyerowitz, Tim Scherr, Katharina Löffler, Ralf Mikut, Tianqi Guo, Yin Wang, Jan P. Allebach, Rina Bao, Noor M. Al-Shakarji, Gani Rahmon, Imad Eddine Toubal, Kannappan Palaniappan, Filip Lux, Petr Matula, Ko Sugawara, Klas E. G. Magnusson, Layton Aho, Andrew R. Cohen, Assaf Arbelle, Tal Ben-Haim, Tammy Riklin Raviv, Fabian Isensee, Paul F. Jäger, Klaus H. Maier-Hein, Yanming Zhu, Cristina Ederra, Ainhoa Urbiola, Erik Meijering, Alexandre Cunha, Arrate Muñoz-Barrutia, Michal Kozubek, and Carlos Ortiz-de Solórzano. The Cell Tracking Challenge: 10 years of objective benchmarking. *Nature Methods*, 20(7):1010–1020, 2023. doi: 10.1038/s41592-023-01879-y.

[50] Brenda Andrews, Jae-Byum Chang, Lucy Collinson, Dong Li, Emma Lundberg, Julia Mahamid, Suliana Manley, Musa Mhlanga, Akihiko Nakano, Johannes Schöneberg, David Van Valen, Ting 'C-ting' Wu, and Assaf Zaritsky. Imaging cell biology. *Nature Cell Biology*, 24(8):1180–1185, 2022. doi: 10.1038/s41556-022-00960-6.

[51] Michael Boutros, Florian Heigwer, and Christina Laufer. Microscopy-Based High-Content Screening. *Cell*, 163(6):1314–1325, 2015. doi: 10.1016/j.cell.2015.11.007.

[52] Erik Meijering, Oleh Dzyubachyk, and Ihor Smal. Chapter nine - Methods for Cell and Particle Tracking. In *Methods in Enzymology*, volume 504, pages 183–200. 2012. doi: 10.1016/B978-0-12-391857-4.00009-4.

[53] Alexander Grünberger, Nicole Paczia, Christopher Probst, Georg Schendzielorz, Lothar Eggeling, Stephan Noack, Wolfgang Wiechert, and Dietrich Kohlheyer. A disposable picoliter bioreactor for cultivation and investigation of industrially relevant bacteria on the single cell level. *Lab on a Chip*, 12(11):2060–2068, 2012. doi: 10.1039/C2LC40156H.

[54] Dorina Lindemann, Christoph Westerwalbesloh, Dietrich Kohlheyer, Alexander Grünberger, and Eric von Lieres. Microbial single-cell growth response at defined carbon limiting conditions. *RSC Advances*, 9(25):14040–14050, 2019. doi: 10.1039/C9RA02454A.

[55] Eugen Kaganovitch, Xenia Steurer, Deniz Dogan, Christopher Probst, Wolfgang Wiechert, and Dietrich Kohlheyer. Microbial single-cell analysis in picoliter-sized batch cultivation chambers. *New Biotechnology*, 47:50–59, 2018. doi: 10.1016/j.nbt.2018.01.009.

[56] Stephan Binder, Georg Schendzielorz, Norma Stäbler, Karin Krumbach, Kristina Hoffmann, Michael Bott, and Lothar Eggeling. A high-throughput approach to identify genomic variants of bacterial metabolite producers at the single-cell level. *Genome Biology*, 13(5):R40, 2012. doi: 10.1186/gb-2012-13-5-r40.

[57] Julia Otten, Niklas Tenhaef, Roman P. Jansen, Johannes Döbber, Lisa Jungbluth, Stephan Noack, Marco Oldiges, Wolfgang Wiechert, and Martina Pohl. A FRET-based biosensor for the quantification of glucose in culture supernatants of mL scale microbial cultivations. *Microbial Cell Factories*, 18(1):143, 2019. doi: 10.1186/s12934-019-1193-y.

[58] James C. W. Locke and Michael B. Elowitz. Using movies to analyse gene circuit dynamics in single cells. *Nature Reviews Microbiology*, 7(5):383–392, 2009. doi: 10.1038/nrmicro2056.

[59] Alina Burmeister and Alexander Grünberger. Microfluidic cultivation and analysis tools for interaction studies of microbial co-cultures. *Current Opinion in Biotechnology*, 62:106–115, 2020. doi: 10.1016/j.copbio.2019.09.001.

[60] Hannah Jeckel and Knut Drescher. Advances and opportunities in image analysis of bacterial cells and communities. *FEMS Microbiology Reviews*, 45(4), 2021. doi: 10.1093/femsre/fuaa062.

[61] Christian Dusny. Microfluidic Single-Cell Analytics. In *Bahnemann, J., Grünberger, A. (eds) Microfluidics in Biotechnology*, pages 159–189, 2022. doi: 10.1007/10_2020_134.

[62] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

[63] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid Task Cascade for Instance Segmentation. *arXiv:1901.07518 [cs]*, 2019. doi: 10.48550/arXiv.1901.07518.

[64] Axel Theorell, Johannes Seiffarth, Alexander Grünberger, and Katharina Nöh. When a single lineage is not enough: Uncertainty-Aware Tracking for spatio-temporal live-cell image analysis. *Bioinformatics*, 35(7):1221–1228, 2019. doi: 10.1093/bioinformatics/bty776.

[65] Katharina Löffler and Ralf Mikut. EmbedTrack—Simultaneous Cell Segmentation and Tracking Through Learning Offsets and Clustering Bandwidths. *IEEE Access*, 10:77147–77157, 2022. doi: 10.1109/ACCESS.2022.3192880.

[66] Romain F. Laine, Ignacio Arganda-Carreras, Ricardo Henriques, and Guillaume Jacquemet. Avoiding a replication crisis in deep-learning-based bioimage analysis. *Nature Methods*, 18(10): 1136–1144, 2021. doi: 10.1038/s41592-021-01284-3.

[67] Juan C. Caicedo, Allen Goodman, Kyle W. Karhohs, Beth A. Cimini, Jeanelle Ackerman, Marzieh Haghighi, CherKeng Heng, Tim Becker, Minh Doan, Claire McQuin, Mohammad Rohban, Shantanu Singh, and Anne E. Carpenter. Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl. *Nature Methods*, 16(12):1247–1253, 2019. doi: 10.1038/s41592-019-0612-7.

[68] Devin P. Sullivan, Casper F. Winsnes, Lovisa Åkesson, Martin Hjelmare, Mikaela Wiking, Rutger Schutten, Linzi Campbell, Hjalti Leifsson, Scott Rhodes, Andie Nordgren, Kevin Smith, Bernard Revaz, Bergur Finnbogason, Attila Szantner, and Emma Lundberg. Deep learning is combined with massive-scale citizen science to improve large-scale image classification. *Nature Biotechnology*, 36(9):820–828, 2018. doi: 10.1038/nbt.4225.

[69] Noah F. Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J. McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, et al. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature Biotechnology*, 40(4):555–565, 2022. doi: 10.1038/s41587-021-01094-0.

[70] Matthias G. Haberl, Christopher Churas, Lucas Tindall, Daniela Boassa, Sébastien Phan, Eric A. Bushong, Matthew Madany, Raffi Akay, Thomas J. Deerinck, Steven T. Peltier, and Mark H. Ellisman. CDeep3M—Plug-and-Play cloud-based deep learning for image segmentation. *Nature Methods*, 15(9):677–680, 2018. doi: 10.1038/s41592-018-0106-z.

[71] Wei Ouyang, Florian Mueller, Martin Hjelmare, Emma Lundberg, and Christophe Zimmer. ImJoy: an open-source computational platform for the deep learning era. *Nature Methods*, 16 (12):1199–1200, 2019. doi: 10.1038/s41592-019-0627-0.

[72] Lucas von Chamier, Romain F. Laine, Johanna Jukkala, Christoph Spahn, Daniel Krentzel, Elias Nehme, Martina Lerche, Sara Hernández-Pérez, Pieta K. Mattila, Eleni Karinou, Séamus Holden, Ahmet Can Solak, Alexander Krull, Tim-Oliver Buchholz, Martin L. Jones, Loïc A. Royer, Christophe Leterrier, Yoav Shechtman, Florian Jug, Mike Heilemann, Guillaume Jacquemet, and Ricardo Henriques. Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nature Communications*, 12(1):2276, 2021. doi: 10.1038/s41467-021-22518-0.

[73] Christoph Spahn, Estibaliz Gómez-de Mariscal, Romain F. Laine, Pedro M. Pereira, Lucas von Chamier, Mia Conduit, Mariana G. Pinho, Guillaume Jacquemet, Séamus Holden, Mike

Heilemann, and Ricardo Henriques. DeepBacs for multi-task bacterial image analysis using open-source deep learning approaches. *Communications Biology*, 5(1):1–18, 2022. doi: 10.1038/s42003-022-03634-z.

[74] Cristian Versari, Szymon Stoma, Kirill Batmanov, Artémis Llamosi, Filip Mroz, Adam Kaczmarek, Matt Deyell, Cédric Lhoussaine, Pascal Hersen, and Gregory Batt. Long-term tracking of budding yeast cells in brightfield microscopy: CellStar and the Evaluation Platform. *Journal of The Royal Society Interface*, 14(127):20160705, 2017. doi: 10.1098/rsif.2016.0705.

[75] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, et al. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, 2016. doi: 10.1038/sdata.2016.18.

[76] Christian Carsten Sachs, Karina Ruzaeva, Johannes Seiffarth, Wolfgang Wiechert, Benjamin Berkels, and Katharina Nöh. Cellsium: versatile cell simulator for microcolony ground truth generation. *Bioinformatics Advances*, 2(1):vbac053, 2022. doi: 10.1093/bioadv/vbac053.

[77] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, 2022.

[78] Nicholas Sofroniew, Talley Lambert, Kira Evans, Juan Nunez-Iglesias, Grzegorz Bokota, Philip Winston, Gonzalo Peña-Castellanos, Kevin Yamauchi, Matthias Bussonnier, Draga Doncila Pop, Ahmet Can Solak, Ziyang Liu, Pam Wadhwa, Alister Burt, Genevieve Buckley, Andrew Sweet, Lukasz Migas, Volker Hilsenstein, Lorenzo Gaifas, Jordão Bragantini, Jaime Rodríguez-Guerra, Hector Muñoz, Jeremy Freeman, Peter Boone, Alan Lowe, Christoph Gohlke, Loic Royer, Andrea PIERRÉ, Hagai Har-Gil, and Abigail McGovern. napari: a multi-dimensional image viewer for Python. *zenodo*, 2022. doi: 10.5281/zenodo.3555620.

[79] Tim Scherr, Johannes Seiffarth, Bastian Wollenhaupt, Oliver Neumann, Marcel P. Schilling, Dietrich Kohlheyer, Hanno Scharr, Katharina Nöh, and Ralf Mikut. microbeSEG: A deep learning software tool with OMERO data management for efficient and accurate cell segmentation. *PLOS One*, 17(11):e0277601, 2022. doi: 10.1371/journal.pone.0277601.

[80] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. doi: 10.1177/0278364913491297.

[81] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268, 2009. doi: 10.1109/ICCV.2009.5459260.

[82] Divya Sivaraman, Payal Biswas, Lakshmi N. Cella, Marylynn V. Yates, and Wilfred Chen. Detecting RNA viruses in living mammalian cells by fluorescence microscopy. *Trends in Biotechnology*, 29(7):307–313, 2011. doi: 10.1016/j.tibtech.2011.02.006.

[83] Alyson M. Hockenberry, Gabriele Micali, Gabriella Takács, Jessica Weng, Wolf-Dietrich Hardt, and Martin Ackermann. Microbiota-derived metabolites inhibit *Salmonella* virulent subpopulation development by acting on single-cell behaviors. *Proceedings of the National Academy of Sciences*, 118(31):e2103027118, 2021. doi: 10.1073/pnas.2103027118.

[84] Simon van Vliet, Alma Dal Co, Annina R. Winkler, Stefanie Spriewald, Bärbel Stecher, and Martin Ackermann. Spatially correlated gene expression in bacterial groups: the role of lineage history, spatial gradients, and cell-cell interactions. *Cell systems*, 6(4):496–507.e6, 2018. doi: 10.1016/j.cels.2018.03.009.

[85] Luisa Blöbaum, Luca Torello Pianale, Lisbeth Olsson, and Alexander Grünberger. Quantifying microbial robustness in dynamic environments using microfluidic single-cell cultivation. *Microbial Cell Factories*, 23(1):44, 2024. doi: 10.1186/s12934-024-02318-z.

[86] R. A. Hoebe, C. H. Van Oven, T. W. J. Gadella, P. B. Dhonukshe, C. J. F. Van Noorden, and E. M. M. Manders. Controlled light-exposure microscopy reduces photobleaching and phototoxicity in fluorescence live-cell imaging. *Nature Biotechnology*, 25(2):249–253, 2007. doi: 10.1038/nbt1278.

[87] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *arXiv:1703.06870 [cs]*, 2018.

[88] Yudong Liu, Yongtao Wang, Siwei Wang, TingTing Liang, Qijie Zhao, Zhi Tang, and Haibin Ling. Cbnet: A novel composite backbone network architecture for object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11653–11660, 2020.

[89] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution, 2020.

[90] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA: Exploring the Limits of Masked Visual Representation Learning at Scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[91] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking. *Int. J. Comput. Vision*, 129(11):3069–3087, 2021. doi: 10.1007/s11263-021-01513-4.

[92] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-Object Tracking by Associating Every Detection Box, 2022.

[93] Yu-Hsiang Wang, Jun-Wei Hsieh, Ping-Yang Chen, Ming-Ching Chang, Hung-Hin So, and Xin Li. SMILEtrack: SiMIlarity LEarning for Occlusion-Aware Multiple Object Tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5740–5748, 2024. doi: 10.1609/aaai.v38i6.28386.

[94] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023.

[95] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[96] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[97] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 25278–25294, 2022.

[98] Tim Scherr, Katharina Löffler, Moritz Böhland, and Ralf Mikut. Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy. *PLOS One*, 15(12):e0243219, 2020. doi: 10.1371/journal.pone.0243219.

[99] Vladimír Ulman, Martin Maška, Klas E. G. Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, et al. An objective comparison of cell-tracking algorithms. *Nature Methods*, 14(12):1141–1152, 2017. doi: 10.1038/nmeth.4473.

[100] Christoffer Edlund, Timothy R. Jackson, Nabeel Khalid, Nicola Bevan, Timothy Dale, Andreas Dengel, Sheraz Ahmed, Johan Trygg, and Rickard Sjögren. LIVECell—A large-scale dataset for label-free live cell segmentation. *Nature Methods*, 18(9):1038–1045, 2021. doi: 10.1038/s41592-021-01249-6.

[101] Morgan Sarah Schwartz, Erick Moen, Geneva Miller, Tom Dougherty, Enrico Borba, Rachel Ding, William Graf, Edward Pao, and David Van Valen. Caliban: Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning. *bioRxiv*, 2024. doi: 10.1101/803205.

[102] Khuloud Jaqaman, Dinah Loerke, Marcel Mettlen, Hirotaka Kuwata, Sergio Grinstein, Sandra L. Schmid, and Gaudenz Danuser. Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods*, 5(8):695–702, 2008. doi: 10.1038/nmeth.1237.

[103] Katharina Löffler, Tim Scherr, and Ralf Mikut. A graph-based cell tracking algorithm with few manually tunable parameters and automated segmentation error correction. *PLOS One*, 16(9):e0249257, 2021. doi: 10.1371/journal.pone.0249257.

[104] Samreen Anjum and Danna Gurari. CTMC: Cell Tracking with Mitosis Detection Dataset Challenge. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4228–4237, 2020. doi: 10.1109/CVPRW50498.2020.00499.

[105] Tal Ben-Haim and Tammy Riklin Raviv. Graph Neural Network for Cell Tracking in Microscopy Videos, 2022.

[106] Keni Bernardin and Rainer Stiefelhagen. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. doi: 10.1155/2008/246309.

[107] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A Higher Order Metric for Evaluating Multi-object Tracking. *International Journal of Computer Vision*, 129(2):548–578, 2021. doi: 10.1007/s11263-020-01375-2.

[108] Pavel Matula, Martin Maška, Dmitry V. Sorokin, Petr Matula, Carlos Ortiz-de Solórzano, and Michal Kozubek. Cell Tracking Accuracy Measurement Based on Comparison of Acyclic Oriented Graphs. *PLOS One*, 10(12):e0144959, 2015. doi: 10.1371/journal.pone.0144959.

[109] Johannes Seiffarth and Katharina Nöh. PyUAT: Open-source Python framework for efficient and scalable cell tracking. *PLOS One*, 20:e0337110, 2025. doi: 10.1371/journal.pone.0337110.

[110] Johannes Seiffarth, Tim Scherr, Bastian Wollenhaupt, Oliver Neumann, Hanno Scharr, Dietrich Kohlheyer, Ralf Mikut, and Katharina Nöh. ObiWan-Microbi: OMERO-based integrated workflow for annotating microbes in the cloud. *SoftwareX*, 26, 2024. doi: 10.1016/j.softx.2024.101638.

[111] Maria Alieva, Amber K. L. Wezenaar, Ellen J. Wehrens, and Anne C. Rios. Bridging live-cell imaging and next-generation cancer treatment. *Nature Reviews Cancer*, 23(11):731–745, 2023. doi: 10.1038/s41568-023-00610-5.

[112] Yoshitaka Shirasaki, Mai Yamagishi, Nobutake Suzuki, Kazushi Izawa, Asahi Nakahara, Jun Mizuno, Shuichi Shoji, Toshio Heike, Yoshie Harada, Ryuta Nishikomori, and Osamu Ohara. Real-time single-cell imaging of protein secretion. *Scientific Reports*, 4(1):4736, 2014. doi: 10.1038/srep04736.

[113] Marc P. Raphael, Joseph A. Christodoulides, James B. Delehanty, James P. Long, and Jeff M. Byers. Quantitative Imaging of Protein Secretions from Single Cells in Real Time. *Biophysical Journal*, 105(3):602–608, 2013. doi: 10.1016/j.bpj.2013.06.022.

[114] Carina Weissmann and Roland Brandt. Mechanisms of neurodegenerative diseases: Insights from live cell imaging. *Journal of Neuroscience Research*, 86(3):504–511, 2008. doi: 10.1002/jnr.21448.

[115] Edward M. Campbell and Thomas J. Hope. Live cell imaging of the HIV-1 life cycle. *Trends in Microbiology*, 16(12):580–587, 2008. doi: 10.1016/j.tim.2008.09.006.

[116] Marcus K. Preedy, Michael R. H. White, and Vinay Tergaonkar. Cellular heterogeneity in TNF/TNFR1 signalling: live cell imaging of cell fate decisions in single cells. *Cell Death & Disease*, 15(3):1–12, 2024. doi: 10.1038/s41419-024-06559-z.

[117] Yihong Huang, Zidong Zhou, Tianqi Liu, Shengnan Tang, and Xuegang Xin. Exploring heterogeneous cell population dynamics in different microenvironments by novel analytical strategy based on images. *npj Systems Biology and Applications*, 10(1):1–10, 2024. doi: 10.1038/s41540-024-00459-w.

[118] Knut Drescher, Carey D. Nadell, Howard A. Stone, Ned S. Wingreen, and Bonnie L. Bassler. Solutions to the Public Goods Dilemma in Bacterial Biofilms. *Current Biology*, 24(1):50–55, 2014. doi: 10.1016/j.cub.2013.10.030.

[119] Raimo Hartmann, Praveen K. Singh, Philip Pearce, Rachel Mok, Boya Song, Francisco Díaz-Pascual, Jörn Dunkel, and Knut Drescher. Emergence of three-dimensional order and structure in growing biofilms. *Nature Physics*, 15(3):251–256, 2019. doi: 10.1038/s41567-018-0356-9.

[120] Matteo Cornaglia, Thomas Lehnert, and Martin A. M. Gijs. Microfluidic systems for high-throughput and high-content screening using the nematode Caenorhabditis elegans. *Lab on a Chip*, 17(22):3736–3759, 2017. doi: 10.1039/C7LC00509A.

[121] Vera Ortseifen, Martina Viefhues, Lutz Wobbe, and Alexander Grünberger. Microfluidics for Biotechnology: Bridging Gaps to Foster Microfluidic Applications. *Frontiers in Bioengineering and Biotechnology*, 8, 2020. doi: 10.3389/fbioe.2020.589074.

[122] Douglas B. Weibel, Willow R. DiLuzio, and George M. Whitesides. Microfabrication meets microbiology. *Nature Reviews Microbiology*, 5(3):209–218, 2007. doi: 10.1038/nrmicro1616.

[123] Lennart Witting, Johannes Seiffarth, Birgit Stute, Tim Schulze, Jan Matthis Hofer, Katharina Nöh, Marion Eisenhut, Andreas P. M. Weber, Eric von Lieres, and Dietrich Kohlheyer. A microfluidic system for the cultivation of cyanobacteria with precise light intensity and CO2 control: enabling growth data acquisition at single-cell resolution. *Lab on a Chip*, 25(3): 319–329, 2025. doi: 10.1039/D4LC00567H.

[124] Eun Seon Chung, Prathitha Kar, Maliwan Kamkaew, Ariel Amir, and Bree B. Aldridge. Single-cell imaging of the Mycobacterium tuberculosis cell cycle reveals linear and heterogenous growth. *Nature Microbiology*, 9(12):3332–3344, 2024. doi: 10.1038/s41564-024-01846-z.

[125] Johannes Seiffarth, Luisa Blöbaum, Richard D. Paul, Nils Friederich, Angelo Jovin Yamachui Sitcheu, Ralf Mikut, Hanno Scharr, Alexander Grünberger, and Katharina Nöh. Tracking one-in-a-million: Large-scale benchmark for microbial single-cell tracking with experiment-aware robustness metrics. In *Computer Vision – ECCV 2024 Workshops*, pages 318–334, 2025.

[126] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I. Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny, Chong Zhang, Ullrich Koethe, Fred A. Hamprecht, and Anna Kreshuk. ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, 16(12):1226–1232, 2019. doi: 10.1038/s41592-019-0582-9.

[127] Dmitry Ershov, Minh-Son Phan, Joanna W. Pylvänäinen, Stéphane U. Rigaud, Laure Le Blanc, Arthur Charles-Orszag, James R. W. Conway, Romain F. Laine, Nathan H. Roy, Daria Bonazzi, Guillaume Duménil, Guillaume Jacquemet, and Jean-Yves Tinevez. TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nature Methods*, 19(7):829–832, 2022. doi: 10.1038/s41592-022-01507-1.

[128] Adrien Ducret, Ellen M. Quardokus, and Yves V. Brun. MicrobeJ, a tool for high throughput bacterial cell detection and quantitative analysis. *Nature Microbiology*, 1(7):1–7, 2016. doi: 10.1038/nmicrobiol.2016.77.

[129] Wei Ouyang, Fynn Beuttenmueller, Estibaliz Gómez-de Mariscal, Constantin Pape, Tom Burke, Carlos Garcia-López-de Haro, Craig Russell, Lucía Moya-Sans, Cristina De-La-Torre-Gutiérrez, Deborah Schmidt, et al. Bioimage model zoo: a community-driven resource for accessible deep learning in bioimage analysis. *BioRxiv*, 2022. doi: 10.1101/2022.06.07.495102.

[130] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, et al. Jupyter Notebooks – a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90. 2016. doi: 10.3233/978-1-61499-649-1-87.

[131] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2.

[132] Chris Allan, Jean-Marie Burel, Josh Moore, Colin Blackburn, Melissa Linkert, Scott Loynton, Donald MacDonald, William J. Moore, Carlos Neves, Andrew Patterson, Michael Porter, Aleksandra Tarkowska, Brian Loranger, Jerome Avondo, Ingvar Lagerstedt, Luca Lianas, Simone Leo, Katherine Hands, Ron T. Hay, Ardan Patwardhan, Christoph Best, Gerard J. Kleywegt, Gianluigi Zanetti, and Jason R. Swedlow. OMERO: flexible, model-driven data management for experimental biology. *Nature Methods*, 9(3):245–253, 2012. doi: 10.1038/nmeth.1896.

[133] Johannes Seiffarth and Katharina Nöh. PyUAT: Open-source Python framework for efficient and scalable cell tracking. *arXiv*, 2025. doi: 10.48550/arXiv.2503.21914.

[134] The pandas development team. pandas-dev/pandas: Pandas. *zenodo*, 2024. doi: 10.5281/zenodo.13819579.

[135] John D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[136] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021.

[137] Johannes Seiffarth, Luisa Blöbaum, Alexander Grünberger, and Katharina Nöh. Customizable and Interactive Visualizations for Investigating Spatio-Temporal Single-Cell Information. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, 2024. doi: 10.1109/ISBI56570.2024.10635297.

[138] Alexander Grünberger, Jan van Ooyen, Nicole Paczia, Peter Rohe, Georg Schiendzielorz, Lothar Eggeling, Wolfgang Wiechert, Dietrich Kohlheyer, and Stephan Noack. Beyond growth rate 0.6: Corynebacterium glutamicum cultivated in highly diluted environments. *Biotechnology and Bioengineering*, 110(1):220–228, 2013. doi: https://doi.org/10.1002/bit.24616.

[139] Julian Schmitz, Boris Yermakov, and Alexander Grünberger. Protocol for microfluidic single-cell cultivation and live-cell imaging of Chinese hamster ovary suspension cell lines. *STAR Protocols*, 5(2):103106, 2024. doi: 10.1016/j.xpro.2024.103106.

[140] David G. Kendall. On the role of variable generation time in the development of a stochastic birth process. *Biometrika*, 35(3/4):316, 1948. doi: 10.2307/2332354.

[141] Christian A. Yates, Matthew J. Ford, and Richard L. Mort. A multi-stage representation of cell proliferation as a markov process. *Bulletin of Mathematical Biology*, 79(12):2905–2928, 2017. doi: 10.1007/s11538-017-0356-4.

[142] Richard D. Paul, Johannes Seiffarth, Hanno Scharr, and Katharina Nöh. Robust Approximate Characterization of Single-Cell Heterogeneity in Microbial Growth. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, Athens, Greece, 2024. IEEE. doi: 10.1109/ISBI56570.2024.10635267.

[143] Yaïr Hein and Farshid Jafarpour. Competition between transient oscillations and early stochasticity in exponentially growing populations. *Physical Review Research*, 6(3):033320, 2024. doi: 10.1103/PhysRevResearch.6.033320.

[144] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[145] Roger Y. Tsien. The green fluorescent protein. *Annual Review of Biochemistry*, 67(1):509–544, 1998. doi: 10.1146/annurev.biochem.67.1.509.

[146] Loïc A. Royer. Omega — harnessing the power of large language models for bioimage analysis. *Nature Methods*, 21(8):1371–1373, 2024. doi: 10.1038/s41592-024-02310-w.

[147] James R. Heath, Antoni Ribas, and Paul S. Mischel. Single-cell analysis tools for drug discovery and development. *Nature Reviews Drug Discovery*, 15(3):204–216, 2016. doi: 10.1038/nrd.2015.16.

[148] Sarah Täuber, Eric von Lieres, and Alexander Grünberger. Dynamic Environmental Control in Microfluidic Single-Cell Cultivations: From Concepts to Applications. *Small*, 16(16):1906670, 2020. doi: 10.1002/smll.201906670.

[149] Stefan Helfrich, Charaf E. Azzouzi, Christopher Probst, Johannes Seiffarth, A. Grünberger, W. Wiechert, D. Kohlheyer, and K. Nöh. Vizardous: interactive analysis of microbial populations with single cell resolution. *Bioinformatics*, 2015. doi: 10.1093/bioinformatics/btv468.

[150] Irepan Salvador-Martínez, Marco Grillo, Michalis Averof, and Maximilian J Telford. CeLaVi: an interactive cell lineage visualization tool. *Nucleic Acids Research*, 49(W1):W80–W85, 2021. doi: 10.1093/nar/gkab325.

[151] Changbeom Shim, Wooil Kim, Tran Thien Dat Nguyen, Du Yong Kim, Yu Suk Choi, and Yon Dohn Chung. CellTrackVis: interactive browser-based visualization for analyzing cell trajectories and lineages. *BMC Bioinformatics*, 24(1):124, 2023. doi: 10.1186/s12859-023-05218-y.

[152] Plotly Technologies Inc. Collaborative data science, 2015. URL `https://plot.ly`.

[153] Shammamah Hossain. Visualization of Bioinformatics Data with Dash Bio. *Proceedings of the 18th Python in Science Conference*, pages 126–133, 2019. doi: 10.25080/Majora-7ddc1dd1-012.

[154] Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, M. Pacer, Yuvi Panda, Fernando Perez, Benjamin Ragan-Kelley, and Carol Willing. Binder 2.0 - Reproducible, interactive, sharable environments for science at scale. *Proceedings of the 17th Python in Science Conference*, pages 113–120, 2018. doi: 10.25080/Majora-4af1f417-011.

[155] Wes McKinney. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a.

[156] Aric A Hagberg, Daniel A Schult, and Pieter J Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

[157] Beate Neumann, Thomas Walter, Jean-Karim Hériché, Jutta Bulkescher, Holger Erfle, Christian Conrad, Phill Rogers, Ina Poser, Michael Held, Urban Liebel, Cihan Cetin, Frank Sieckmann, Gregoire Pau, Rolf Kabbe, Annelie Wünsche, Venkata Satagopam, Michael H. A. Schmitz, Catherine Chapuis, Daniel W. Gerlich, Reinhard Schneider, Roland Eils, Wolfgang Huber, Jan-Michael Peters, Anthony A. Hyman, Richard Durbin, Rainer Pepperkok, and Jan Ellenberg. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature*, 464(7289):721–727, 2010. doi: 10.1038/nature08869.

[158] Johannes Seiffarth et al. Data for - Tracking one in a million: Performance of automated tracking on a large-scale microbial data set (1.0.0). *Zenodo*, October 2022. doi: 10.5281/zenodo.7260137. URL `https://doi.org/10.5281/zenodo.7260137`.

[159] Joaquín Gutiérrez Mena, Sant Kumar, and Mustafa Khammash. Dynamic cybergenetic control of bacterial co-culture composition via optogenetic feedback. *Nature Communications*, 13(1):4808, 2022. doi: 10.1038/s41467-022-32392-z.

[160] Arthur D. Edelstein, Mark A. Tsuchida, Nenad Amodaj, Henry Pinkard, Ronald D. Vale, and Nico Stuurman. Advanced methods of microscope control using μmanager software. *Journal of Biological Methods*, 1(2):1, 2014. doi: 10.14440/jbm.2014.36.

[161] Henry Pinkard, Nico Stuurman, Ivan E. Ivanov, Nicholas M. Anthony, Wei Ouyang, Bin Li, Bin Yang, Mark A. Tsuchida, Bryant Chhun, Grace Zhang, Ryan Mei, Michael Anderson, Douglas P. Shepherd, Ian Hunt-Isaak, Raymond L. Dunn, Wiebke Jahr, Saul Kato, Loïc A. Royer, Jay R. Thiagarajah, Kevin W. Eliceiri, Emma Lundberg, Shalin B. Mehta, and Laura Waller. Pycro-Manager: open-source software for customized and reproducible microscope control. *Nature Methods*, 18(3):226–228, 2021. doi: 10.1038/s41592-021-01087-6.

[162] Marc Rullan, Dirk Benzinger, Gregor W. Schmidt, Andreas Milias-Argeitis, and Mustafa Khammash. An Optogenetic Platform for Real-Time, Single-Cell Interrogation of Stochastic Transcriptional Regulation. *Molecular Cell*, 70(4):745–756.e6, 2018. doi: 10.1016/j.molcel.2018.04.012.

[163] Sonali Gupta, Tyler D. Ross, Marcella M. Gomez, Job L. Grant, Philip A. Romero, and Ophelia S. Venturelli. Investigating the dynamics of microbial consortia in spatially structured environments. *Nature Communications*, 11(1):2418, 2020. doi: 10.1038/s41467-020-16200-0.

[164] Felix J. H. Hol and Cees Dekker. Zooming in to see the bigger picture: Microfluidic and nanofabrication tools to study bacteria. *Science*, 346(6208):1251821, 2014. doi: 10.1126/science.1251821.

[165] Jean Ollion, Marina Elez, and Lydia Robert. High-throughput detection and tracking of cells and intracellular spots in mother machine experiments. *Nature Protocols*, 14(11):3144–3161, 2019. doi: 10.1038/s41596-019-0216-9.

[166] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment Anything in Images and Videos. *arXiv:2408.00714*, 2024. doi: 10.48550/arXiv.2408.00714.

[167] Muhammad Hussain. YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision. *arXiv:2407.02988 [cs]*, 2024. doi: 10.48550/arXiv.2407.02988.

[168] Rafael Padilla, Sergio L. Netto, and Eduardo A. B. da Silva. A Survey on Performance Metrics for Object-Detection Algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, 2020. doi: 10.1109/IWSSIP48289.2020.9145130.

[169] Chao Ma, Yansong Peng, Hongtong Li, and Weiqiang Chen. Organ-on-a-Chip: A New Paradigm for Drug Development. *Trends in Pharmacological Sciences*, 42(2):119–133, 2021. doi: 10.1016/j.tips.2020.11.009.

[170] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation*. Pearson, Bosten San Francisco Munich, 3. ed edition, 2007. ISBN 978-0-321-45536-9.

[171] K. Ramamritham and J.A. Stankovic. Scheduling algorithms and operating systems support for real-time systems. *Proceedings of the IEEE*, 82(1):55–67, 1994. doi: 10.1109/5.259426.

[172] Gilbert Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, 1992. doi: 10.1016/0377-2217(92)90138-Y.

[173] Yvan Dumas, Jacques Desrosiers, Eric Gelinas, and Marius M. Solomon. An Optimal Algorithm for the Traveling Salesman Problem with Time Windows. *Operations Research*, 43(2): 367–371, 1995. doi: 10.1287/opre.43.2.367.

[174] Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016. doi: 10.1038/533452a.

[175] Thomas Drepper, Thorsten Eggert, Franco Circolone, Achim Heck, Ulrich Krauß, Jan-Karl Guterl, Marion Wendorff, Aba Losi, Wolfgang Gärtner, and Karl-Erich Jaeger. Reporter proteins for in vivo fluorescence without oxygen. *Nature Biotechnology*, 25(4):443–445, 2007. doi: 10.1038/nbt1293.

[176] Yueshu Li, Yuanyi Guo, Fangyuan Niu, Hui Gao, Qing Wang, and Meijuan Xu. Regulation of oxidative stress response and antioxidant modification in Corynebacterium glutamicum. *World Journal of Microbiology and Biotechnology*, 40(9):267, 2024. doi: 10.1007/s11274-024-04066-z.

[177] Maureen A. Walling and Jason R. E. Shepard. Cellular heterogeneity and live cell arrays. *Chemical Society Reviews*, 40(7):4049–4076, 2011. doi: 10.1039/C0CS00212G.

[178] Abraham. Savitzky and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964. doi: 10.1021/ac6021 4a047.

[179] Laura Marie Helleckes, Michael Osthege, Wolfgang Wiechert, Eric von Lieres, and Marco Oldiges. Bayesian calibration, process modeling and uncertainty quantification in biotechnology. *PLOS Computational Biology*, 18(3):e1009223, 2022. doi: 10.1371/journal.pcbi.1009223.

[180] Sangeeta N. Bhatia and Donald E. Ingber. Microfluidic organs-on-chips. *Nature Biotechnology*, 32(8):760–772, 2014. doi: 10.1038/nbt.2989.

[181] Joel T. Collins, Joe Knapper, Julian Stirling, Joram Mduda, Catherine Mkindi, Valeriana Mayagaya, Grace A. Mwakajinga, Paul T. Nyakyi, Valerian L. Sanga, Dave Carbery, Leah White, Sara Dale, Zhen Jieh Lim, Jeremy J. Baumberg, Pietro Cicuta, Samuel McDermott, Boyko Vodenicharski, and Richard Bowman. Robotic microscopy for everyone: the OpenFlexure microscope. *Biomedical Optics Express*, 11(5):2447–2460, 2020. doi: 10.1364/BOE.385729.

[182] Ando C. Zehrer, Ana Martin-Villalba, Benedict Diederich, and Helge Ewers. An open-source, high resolution, automated fluorescence microscope. *eLife*, 12, 2024. doi: 10.7554/eLife.8982 6.2.

[183] Tobias Wenzel. Open hardware: From DIY trend to global transformation in access to laboratory equipment. *PLOS Biology*, 21(1):e3001931, 2023. doi: 10.1371/journal.pbio.3001931.

[184] Mohammad Nour Alsamsam, Aurimas Kopūstas, Meda Jurevičiūtė, and Marijonas Tutkus. The miEye: Bench-top super-resolution microscope with cost-effective equipment. *HardwareX*, 12, 2022. doi: 10.1016/j.ohx.2022.e00368.

[185] Katharina Smaluch, Bastian Wollenhaupt, Heiko Steinhoff, Dietrich Kohlheyer, Alexander Grünberger, and Christian Dusny. Assessing the growth kinetics and stoichiometry of Escherichia coli at the single-cell level. *Engineering in Life Sciences*, 23(1):e2100157, 2023. doi: 10.1002/elsc.202100157.

[186] Keiji Fushimi, Yusuke Nakai, Akiko Nishi, Ryo Suzuki, Masahiro Ikegami, Risa Nimura, Taichi Tomono, Ryota Hidese, Hisashi Yasueda, Yusuke Tagawa, and Tomohisa Hasunuma. Development of the autonomous lab system to support biotechnology research. *Scientific Reports*, 15(1):6648, 2025. doi: 10.1038/s41598-025-89069-y.

[187] Jacob T. Rapp, Bennett J. Bremer, and Philip A. Romero. Self-driving laboratories to autonomously navigate the protein fitness landscape. *Nature chemical engineering*, 1(1):97–107, 2024. doi: 10.1038/s44286-023-00002-4.

Curriculum Vitae

22.10.2025	**Doctoral Examination** for Dr. rer. nat.
2021–2025	**Research Assistant and Doctoral Candidate** Institute of Bio- and Geosciences: Biotechnology (IBG-1) Forschungszentrum Jülich, Germany
2014–2021	**Studies of Computer Science** RWTH Aachen University, Germany
April 2021	**Master of Science** RWTH Aachen University, Germany
August 2017	**Bachelor of Science** RWTH Aachen University, Germany
2014	**General Higher Education Entrance Qualification (Abitur)** Städtisches Gymnasium Kreuztal Kreuztal, Germany