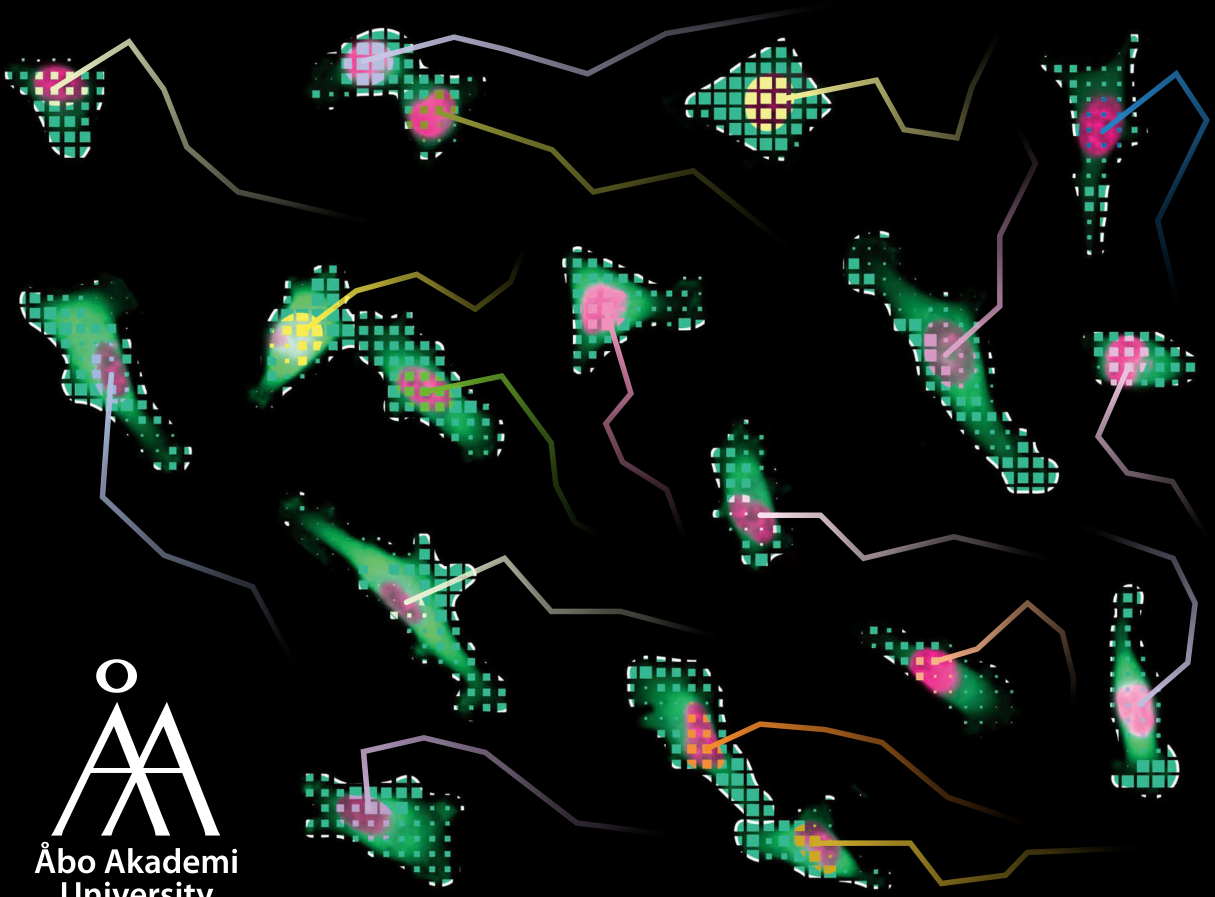
Joanna Pylvänäinen

Bioimage Analysis for Life Scientists: Tools for Live Cell Imaging

Cell Biology



Åbo Akademi
University

Joanna Pylvänäinen

Born 1984, in Tampere, Finland

Previous studies and degrees

- L.Sc degree in Cell Biology, Åbo Akademi University, 2020
- Vocational degree in Media Studies, Turku Vocational Institute, 2018
- M.Sc degree in Biomedical Imaging, University of Turku, 2013
- M.Sc degree in Biochemistry, University of Turku, 2010
- B.Sc degree in Biochemistry, University of Turku, 2010

picture by Solveig Eriksson

Bioimage Analysis for Life Scientists: Tools for Live Cell Imaging

Joanna Pylvänäinen

Cell Biology
Faculty of Science and Engineering
Turku Doctoral Network in Molecular Biosciences
Åbo Akademi University
Turku, Finland, 2024

From the Faculty of Science and Engineering, Cell Biology, Turku Doctoral Network in Molecular Biosciences, InFLAMES Research Center, Åbo Akademi University.

Supervised by

Associate Professor Guillaume Jacquemet, PhD
Faculty of Science and Engineering, Cell Biology, Åbo Akademi University, Turku, Finland

Co-supervised by

Associate Professor Diana Toivola, PhD
Faculty of Science and Engineering, Cell Biology, Åbo Akademi University, Turku, Finland

Research Manager Pasi Kankaanpää, PhD
Turku BioImaging, University of Turku and Åbo Akademi University, Turku, Finland

Reviewed by

Associate Professor Pekka Ruusuvuori, PhD
Institute of Biomedicine, University of Turku, Turku, Finland

and

Lassi Paavolainen, PhD
Institute for Molecular Medicine Finland, University of Helsinki, Finland

Opponent

Professor Carolina Wählby, PhD
Science for Life Laboratory, Centre for Image Analysis, Uppsala University, Uppsala, Sweden

ISBN 978-952-12-4371-4 (Print)
ISBN 978-952-12-4372-1 (Digital)
Painosalama Oy, Turku, Finland, 2024

To my parents for their endless support

Contents

Abstract

Live imaging is essential in visualizing biological processes such as normal tissue development, wound healing, and cancer — processes too small for the bare human eye to observe. Optical microscopy has enabled the magnification of these processes, and the integration of sensitive digital cameras has enabled the acquisition of images for subsequent observation and analysis. For these reasons, microscopy has become an indispensable tool in studying cells.

However, extracting meaningful information from live imaging poses several challenges. Living cells are fragile and should be imaged in controlled environments and using low doses of light, often leading to the generation of noisy images. Low signal-to-noise ratios often hinder accurate object detection and tracking, while sample drifting complicates video analysis. Although several tools to improve the analysis of live cell imaging exist, many of them remain unreachable for life scientists as their usage requires programming skills or they lack proper documentation and user-friendly interfaces. These hinder their usability and reproducibility.

To address these issues, we have developed user-friendly live cell image analysis tools for biologists. First, Fast4DReg, a Fiji plugin developed to swiftly correct drift in 4D images, enhances the quality of live imaging. Second, DL4MicEverywhere allows life scientists to implement deep learning on various computational platforms to improve and segment live cell imaging data. Third, TrackMate v7 is a sophisticated tracking software integrating cutting-edge segmentation algorithms into tracking pipelines, facilitating robust and precise cell tracking. To ensure the usability of these tools, we have written extensive documentation and step-by-step guides complemented with openly available test datasets. We incorporate these tools to enable quantitative analysis of the interaction between pancreatic cancer cells and endothelium during metastasis and in a study of cancer cell drug resistance.

In summary, our user-friendly image analysis tools offer efficient and accessible solutions for processing and analyzing live cell imaging data, thus benefiting researchers across various fields and contributing to our understanding of cell behavior and disease processes.

Sammanfattning (Abstract in Swedish)

Cellmigration är en väsentlig process för hälsa och sjukdom och är involverad i processer som utveckling, sårläkning och cancermetastas. Avbildning av levande celler spelar en avgörande roll vid observation av dessa processer och när den kombineras med moderna verktyg för bildanalys blir visualisering och kvantifiering av cellens liv möjligt. Att extrahera meningsfulla data från avbildning av levande material innebär dock flera utmaningar. Levande celler är känsliga och bör avbildas i kontrollerade förhållanden och med låg ljusintensitet, vilket leder till generering av brusiga och oskarpa bilder. Ofta försvårar den låga signal-till-brusförhållanden exakt objektdetektion och cellspårning, medan provdrift komplicerar videoanalysen. Trots att det finns flera verktyg för att förbättra analysen av levande cellavbildning, förblir många av dem otillgängliga för forskare eftersom användning kräver programmeringsfärdigheter eller de saknar grundlig dokumentation och användarvänliga gränssnitt. Detta begränsar deras användbarhet och reproducerbarhet.

För att lösa dessa problem har vi utvecklat tre användarvänliga verktyg för bildanalys av levande celler. Bildanalysprogrammet Fast4DReg är ett Fiji-tillägg utvecklad för att snabbt korrigera drift i 4D-bilder och för att förbättra kvaliteten på avbildning av levande celler. DL4MicEverywhere tillåter livsvetenskapsmän att implementera djupinlärning för att förbättra och segmentera avbildningsdata från levande celler på olika beräknings-plattformar. TrackMate v7 är en sofistikerad spårningsprogramvara som integrerar toppmoderna segmenteringsalgoritmer i spårningspipelines, vilket underlättar robust och exakt cellspårning. För att förbättra användbarheten av dessa verktyg har vi skrivit omfattande dokumentation, steg-för-steg guider kompletterade med fritt tillgängliga testdatasatser. Vi har utnyttjat dessa verktyg för att möjliggöra kvalitativ och kvantitativ analys av interaktionen mellan bukspottkörtelcancerceller och endotelceller under metastas, och i studier av läkemedelsresistens.

Sammanfattningsvis erbjuder våra användarvänliga bildanalysverktyg för driftborttagning och cellspårning effektiva och tillgängliga lösningar för bearbetning och analys av avbildningsdata från levande celler, vilket gynnar forskare inom olika områden och bidrar till vår förståelse av cellbeteende och sjukdomsprocesser.

Abstrakti (Abstract in Finnish)

Elävien solujen kuvantamisella on keskeinen rooli solubiologiassa. Sen avulla voidaan tutkia terveyden ja sairauden kannalta tärkeitä prosesseja, kuten yksilön kehitystä, haavan paranemista sekä syöpää. Kun elävien solujen kuvantamiseen tarkoitetut mikroskooppitekniikat yhdistetään moderneihin kuva-analyysityökaluihin, tulee elämän kvantitatiivinen mittaaminen mahdolliseksi.

Merkityksellisen tiedon tuottaminen mikroskoopeilla kuvatuista soluvideoista aiheuttaa kuitenkin monia haasteita. Elävät solut ovat hauraita ja niitä tulee kuvantaa kontrolloiduissa ympäristöissä käyttäen vähäistä valon määrää. Tämä aiheuttaa usein videoihin kohinaa, joka puolestaan vaikeuttaa kohteiden havaitsemista taustasta ja näin ollen niiden seurantaa. Lisäksi mikroskoopin tai näytteen liikkumisen aiheuttama heilunta vaikeuttaa videon kohteiden seuraamista. Vaikka soluvideoiden analysoimiseksi on jo olemassa useita työkaluja, monet niistä eivät ole tutkijoiden saavutettavissa, sillä niiden käyttö vaatii usein ohjelmointitaitoja tai niissä ei ole asianmukaisia dokumentaatioita ja käyttäjäystävällisiä käyttöliittymiä.

Näiden ongelmien ratkaisemiseksi olemme kehittäneet käyttäjäystävällisiä työkaluja luonnontieteilijöille soluvideoiden käsittelyyn ja analysointiin. Näistä ensimmäinen, Fast4DReg, on Fiji-liitännäinen, joka on kehitetty nopeasti korjaamaan heiluntaa 4D-kuvissa, vakauttaen näin kuvaa. Toinen työkalu, DL4MicEverywhere, mahdollistaa syväoppimisen käyttämisen soluvideoiden laadun parantamiseksi ja niissä olevien kohteiden tunnistamiseksi. Kolmas työkalu on TrackMate v7, jonka avulla voidaan mitata elävien solujen liikettä. Tämä työkalu pystyy integroimaan moderneja segmentointi- ja analyysialgoritmeja osaksi soluvideoiden mittaamista, mahdollistaen tarkan ja toistettavan solujen seurannan. Työkalujen käytettävyyttä on pyritty parantamaan kirjoittamalla kattava dokumentaatio ja jakamalla testidataa avoimesti. Lisäksi tässä väitöskirjatutkimuksessa sovellettiin kehitettyjä työkaluja solubiologisessa tutkimuksessa kahden esimerkin kautta. Näistä ensimmäisessä tutkittiin haimasyöpäsolujen ja endoteelisolujen vuorovaikutusta pahanlaatuisen syövän etenemisen aikana, ja toisessa selvitettiin, miten tietyt mutaatiot vaikuttavat syöpäsolujen lääkeresistenssiin.

Tutkimuksien tulokset osoittavat, että kehitetyt käyttäjäystävälliset työkalut soveltuvat soluvideoiden laadun parantamiseen ja solujen seurantaan ja ne tarjoavat tehokkaita ja saavutettavia ratkaisuja soluvideoiden käsittelyyn ja analysointiin. Nämä työkalut auttavat eri alojen tutkijoita ymmärtämään solujen käyttäytymistä ja prosesseja, erityisesti syöpätutkimuksessa.

List of Original Publications

This thesis comprises two original publications and one manuscript which is now accepted for publication, indicated with Roman numerals.

I. **Pylvänäinen, J.W.,** Laine, R. F., Saraiva, B. M. S., Ghimire, S., Follain, G., Henriques, R., & Jacquemet, G., (2023). Fast4DReg - fast registration of 4D microscopy datasets. *Journal of Cell Science,* doi: 10.1242/jcs.260728

II. Ershov, D*, Phan, M-S*, **Pylvänäinen, J.W.*,** Rigaud, S.U.*, Le Blanc, L, Charles-Orszag, A, Conway, J.R.W., Laine, RF, Roy, NH, Bonazzi, D, Duménil, G, Jacquemet, G & Tinevez, J-Y., (2022). TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nature Methods,* doi: 10.1038/s41592-022-01507-1

III. Hidalgo I., **Pylvänäinen, J.W.,** Ferreira M.G., Russell C.T., Arganda-Carreras I., AI4Life Consortium, Jacquemet G., Henriques, R., Gómez-de-Mariscal, E.G., (2023). DL4MicEverywhere: Deep learning for microscopy made flexible, shareable, and reproducible. *Accepted for publication in Nature Methods. Preprint available at bioRxiv;* doi: 10.1101/2023.11.19.567606

**Equal contribution*

Publications not included in this thesis

i. **Pylvänäinen, J.W.,** Mairinoja L.M., Tuomisto, T, Solis, J, Toivola, DM, Kankaanpää, P (2023). Supporting Microscopy Learning with Ocul-AR, a Virtual and Augmented Reality-Powered Mobile Application. *CEUR Workshop Proceedings*. doi: https://ceur-ws.org/Vol-3393/TELL23_paper_7227_3.pdf

ii. **Pylvänäinen, J.W.,** Gómez-de-Mariscal, E., Henriques, R., & Jacquemet, G. (2023). Live-cell imaging in the deep learning era. *Current Opinion in Cell Biology*, doi: 10.1016/j.ceb.2023.102271

iii. Hyder, C., Lazaro, G.*, **Pylvänäinen, J.W.*,** Roberts, M., Qvarnström, S., & Eriksson, J. (2014). Nestin regulates prostate cancer cell invasion by influencing the localisation and functions of FAK and integrins. *Journal of Cell Science*, doi: 10.1242/jcs.125062

iv. Gómez-de-Mariscal, E.*, Del Rosario M.*, **Pylvänäinen, J.W.**, Jacquemet, G., Henriques, R. (2023). Harnessing Artificial Intelligence to Reduce Phototoxicity in Live Imaging. *Journal of Cell Science,* doi: 10.1242/jcs.261545

v. Wright, G.D*, Thompson, K.A*, Reis, Y.*, Bischof, J., Hockberger, P.E., Itano M.S., Yen L., Bialy N., Brown C.M., Chaabane L., Chew T-L., Chitty, A.I., Cordelières, F.P., De Niz, M. , Ellenberg, J., Engelbrecht L., Fabian-Morales E., Fazeli E., Fernandez-Rodriguez, J., Ferrando-May, E., Fletcher G., Galloway G.J., Guerrero A., Matos Guimarães J., Jacobs C.A., Jayasinghe S., Kable E., Kitten G.T., Komoto S., Xiaoxiao M., Araújo Marques J., Millis B.A., Miranda K., O'Toole P.J., Pollak C.N, Prats C., **Pylvänäinen J.W.**, Rahmoon M.A., Reiche M.A, Riches J.D., Rossi A.H., Salamero J., Taiye A.S., Thiriet C., dos Santos Vasconcelos A., Olatunji, S. Y., Antje Keppler A. (2023). Charting a course for success: International recommendations for Imaging Scientist Careers in core facilities*. International recommendations by Global BioImaging*. doi: 10.5281/zenodo.10200758

**Equal contribution*

Author Contribution

I. In the Fast4DReg publication, I developed an algorithm to correct drifty 4D data and integrated it as a plugin into the Fiji ecosystem. I built a user-friendly interface and maintain the tool. I also created several datasets to benchmark the tool. Additionally, I wrote extensive documentation on using the tool to make it more accessible for users. I participated in the experiment design, data analysis, and manuscript writing.

II. I designed experiments, acquired and analyzed datasets to showcase the usability of the TrackMate v7 and participated in the beta testing and troubleshooting. Additionally, I participated in the manuscript writing and wrote extensive step-by-step user documentation to ease usage.

III. My primary role in this project was in the alpha testing phase. I tested DL4MicEverywhere on two operating systems while bringing insight into the needs of biologists. I created one dataset to test the tool and provided feedback on developing the tool to be more user-friendly. Additionally, I participated in the writing of the manuscript and documentation.

Abbreviations

AI	artificial intelligence
CAFI	content-aware frame interpolation
CCM	cross-correlation map
CTC	cell tracking challenge
DFCAN	Fourier channel attention network
DIC	differential interference contrast
DL	deep learning
DoG	difference of Gaussians
GAN	generative adversarial network
GPU	graphics processing unit
GUI	graphical user interface
IoU	intersection over union
LAP	linear assignment problem
LED	light-emitting diodes
LoG	Laplacian of Gaussian
LUT	look-up table
ML	machine learning
mSSIM	mean structural similarity index measure
NA	numerical aperture
NRMSE	normalized root mean squared error
OMERO	Open Microscopy Environment Remote Objects
PCC	Pearson's correlation coefficient
PDAC	pancreatic ductal adenocarcinoma
PMT	photomultiplier tube
PSNR	peak signal-to-noise ratio
Quarep-Limi	quality assessment and reproducibility for instruments and images in light microscopy
RAM	random access memory
ReLU	a rectified linear unit
RGB	red-green-blue
ROI	region of interest
ROS	reactive oxygen species
SFSRM	single-frame deep-learning super-resolution microscopy
SMLM	single-molecule localization microscopy
SNR	signal-to-noise ratio
SSIM	structural similarity index measure
STED	stimulated emission depletion microscopy
UV	ultraviolet

Introduction

Live imaging is essential in visualizing processes such as normal tissue development, wound healing and cancer — processes too small for the bare human eye to observe. Optical microscopy has enabled the magnification of these processes, and the integration of sensitive digital cameras has enabled the acquisition of images for subsequent observation and analysis. For these reasons, microscopy has become an indispensable tool in studying cells.

Cells are fragile units of life. During microscopy experiments, they must be kept in controlled environments that retain their viability. Imaging cells using brightfield light, it is possible to observe their behavior for long periods of time. However, observation of cellular proteins and processes typically requires the usage of fluorescent markers and therefore the usage of intense illumination using monochromatic or near-monochromatic light. To study cells' movement undisturbed under such conditions, the light dosage and exposure times should be kept at a minimum. Consequently, low dosages of light and short acquisition times may result in noisy images, limiting object detection and complicating downstream analysis. Furthermore, even modern instruments can cause the acquired image data to drift, posing additional challenges in following cell movements. Many of these issues can be solved post-acquisition using software specialized for bioimage analysis. Therefore, bioimage analysis has become integral to life sciences research, providing tools to extract meaningful information from image data.

The literature review of this thesis focuses on introducing imaging methods and requirements for live imaging, as well as how to process and analyze the data acquired from live cell experiments. This thesis resulted in three user-friendly and accessible image analysis tools we developed to facilitate live cell imaging data processing and analysis. These tools are Fast4DReg (I), a fast and reproducible drift correction and channel alignment tool for 4D image datasets, TrackMate (II), a user-friendly tracking and track analysis tool, and DL4MicEverywhere (III), a non-code environment for training and deployment of DL models on multiple computational platforms. These tools are presented in the original publications in this thesis, and my contribution is further highlighted in the aims, methods, results, and discussion sections. Additionally, two unpublished projects (unpublished 1 and unpublished 2) are included in this thesis to showcase how I incorporated these tools in a user-friendly way to extract quantitative information from live cell imaging experiments.

These three software tools are available under open-source licenses, come with comprehensive documentation, and provide intuitive, user-friendly interfaces. They contribute to accelerating research in life sciences, ultimately saving more time for scientific discoveries.

Review of the Literature

1 Live cell imaging in biological research

Understanding the behavior of living cells has interested researchers for decades. Due to the small size of cells, they are not visible to the human eye and require a microscope for magnification. The first microscopes date back to the 16th century. Antonie van Leeuwenhoek was the first person to use microscopes for scientific discoveries. Among his discoveries, he was the first person to describe red blood cells. The first analysis of living cells using video microscopy goes back to the 1950s when Michael Abercrombie showed the behavior of normal and cancer cells in cell culture (https://wellcomecollection.org/works/z6h7jzv4, 14.3.2024). While microscopy as a field is not novel, advances in technology continually give rise to new techniques, especially enabling gentle long-term imaging of living samples with higher resolution using methods such as lattice sheet microscopy (Chen et al., 2014).

Although the selection of available techniques for cell imaging is vast (Lemon and McDole, 2020), in this first chapter of my thesis, I focus on introducing microscopy in the context of imaging living biological samples. I focus on giving an overview of the microscopy techniques I have used in this thesis and introducing challenges that live cell imaging introduces.

1.1 Basic concepts of optical microscopy

Optical microscopy seeks to enhance human vision by enlarging objects that would otherwise remain below the resolution of the human eye. A microscope consists of one or more aligned lenses that magnify and resolve light as it passes through. The magnifying power of a microscope depends on several factors, including the magnifying power of the objective lens and adjacent lenses on the light path, including the eyepiece, and the distance between the lens and the object (Pawley, 2006).

The resolution of the microscope refers to its ability to distinguish between two closely spaced point objects surrounded by diffraction rings, called the Airy disk, as separate objects. This disk is generated when light waves unite and interfere at the focal point, producing a diffraction pattern of circular rings surrounding a brighter central disk. This diffraction of light is what limits the separation of the objects from each other. The extent and magnitude of the diffraction pattern are affected by both the wavelength of light (λ), the refractive material used to manufacture the objective lens, and the objective lens's numerical aperture (NA). Therefore, there is a finite limit beyond which it is impossible to resolve individual points in the field of interest, known as the diffraction limit. This limit was first described by Ernst Abbe in 1873 (Abbe, 1873) and is often referred to as the Abbe diffraction limit. He reported that the smallest resolvable distance between two point-objects captured using an optical microscope can never be smaller than half the wavelength of the imaging light. The diffraction limit is a central concept in optical microscopy and affects both bright-field and fluorescence microscopy (Figure 1).

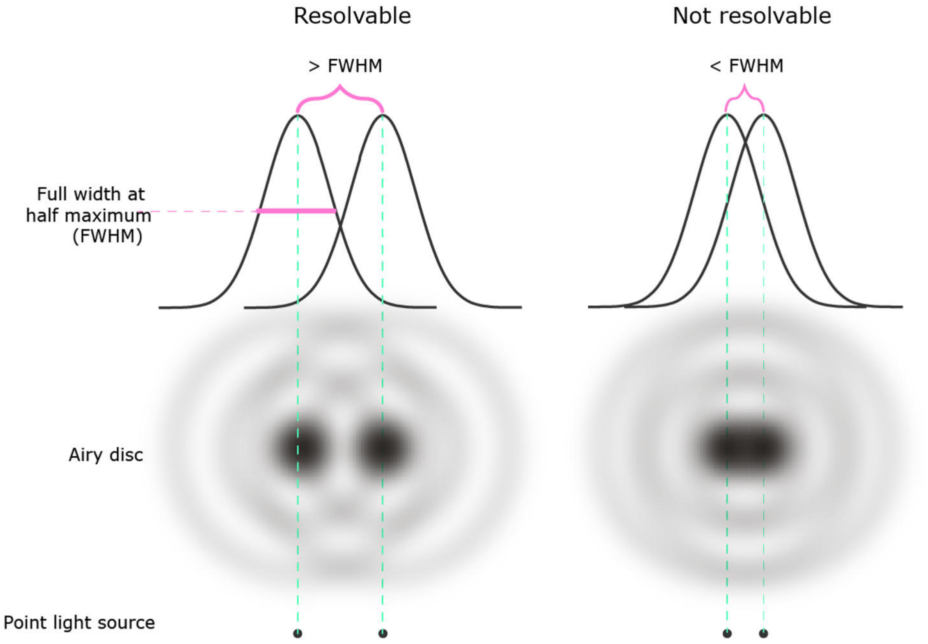


Figure 1: Resolution in microscopy. The resolution of the microscope refers to its ability to distinguish between two closely spaced point objects from each other. The separation ability is limited by the diffraction of light, resulting in an airy disc pattern generated when light waves unite and interfere at the focal point, producing a diffraction pattern of circular rings that surround a brighter central disk. The image of the diffraction pattern can be represented as an intensity distribution. If the point light sources are further from each other than the length of the full width at half maximum (FWHM) of the intensity distribution, the points are still resolvable.

Mathematically, the diffraction limit is defined by Abbe's equation (Figure 2) where Abbe's diffraction limit (d) describes that the resolution of an optical microscope is limited to roughly half the wavelength (λ) of the light being used.

$$d = \frac{\lambda}{2NA}$$

d = distance
λ = wavelength
NA = numerical aperture

Figure 2: Abbe's equation. Abbe's equation is used to calculate the theoretical resolution limit of an optical microscope. d is the minimum resolvable distance between two objects (resolution), λ is the wavelength of light used for imaging and NA is the numerical aperture of the microscope objective.

In live cell imaging, the selection of the appropriate magnification resolving power is essential as light can be toxic for cells and may alter their behavior, therefore affecting the imaged phenomenon. This is discussed in more detail in chapter 1.2.2.

1.1.1 Microscopy techniques and methods for live cell imaging

The number of existing microscopy techniques is vast and keeps growing, and choosing the correct technique for a particular experiment can remain challenging. When selecting a microscopy technique, it is essential to ask the question: *"What is my biological question, and what must I measure to answer it?*" (Senft et al., 2023). When selecting an instrument for live-cell imaging, the first step is to know the structure of interest and how it is labeled, if at all, and if it should be imaged in 2D or 3D. Next, the desired resolution is defined, keeping in mind that the selected microscope must be compatible with live cell imaging (Lemon and McDole, 2020). Additionally, the microscope's acquisition speed plays an important role as some biological phenomena can occur very fast and might be missed by a microscope with a slow acquisition speed (Kiepas et al., 2020; Jonkman et al., 2020).

To visualize the structures of interest, living cells can be labeled using fluorescence probes, or they can stably express fluorescent proteins. This allows visualization of substructures of cells using fluorescence light and allows, for example, studies in organelle trafficking inside cells (Miihkinen et al., 2021) and observation of dynamic processes, such as signaling pathways (Conway et al., 2023).

1.1.2 Widefield Microscopy

In widefield microscopy, the entire sample on the microscope stage is exposed to a light source and the whole image is captured at once. The resulting image can be viewed through the microscope eyepiece or acquired through a digital camera. The sample illumination can occur from below (inverted microscope) or above (upright microscope) the sample. Commonly for live cell imaging, the sample must be illuminated from below as the sample dish usually contains liquid media.

The light source in a widefield microscope can be a transmitted or fluorescent light. In a transmitted light microscope, a white unfiltered light, usually from a lamp or light-emitting diodes (LEDs), is directed to the sample through the microscope optics. Köhler alignment plays an important role when aligning the microscope light path. Correct Köhler alignment ensures that the incoming light is uniform and focused correctly on the entire field of view (Keller, 2007). Contrast-enhancing techniques, such as differential interference contrast (DIC) (Salmon and Tran, 1998) or Phase contrast imaging (Zernike, 1942), have been developed to enhance the contrast, as when imaging transparent or low-contrast specimens, such as living cells, the contrast in the resulting images may be insufficient for data extraction.

When imaging fluorescently labeled cells, the microscope needs to be equipped with filter cubes that allow only the optimal wavelength to excite the fluorophore. Dichroic mirrors further direct the excitation light to the sample, and as this light interacts with the fluorophores in the sample, the fluorophore is excited. Eventually, the fluorophore is released from the excited state, emitting light of a lower energy and, therefore, a longer wavelength, which can then be visualized or captured using a digital camera. The difference between the emission and excitation peaks called the Stoke's shift, is caused by the energy loss as heat or vibration during the excitation process (Sanderson, 2020).

1.1.3 Confocal Microscopy

Compared to widefield microscopy, confocal microscopy allows more detailed fluorescent image acquisition and is, therefore, broadly used to resolve the detailed structure of specific objects within the cell, such as filopodia tips (Popović et al., 2023). Compared to widefield microscopy, where the entire sample signal is simultaneously collected, in confocal microscopy, the sample is excited point-by-point. Additionally, the confocal microscope takes advantage of a pinhole placed at the conjugate focal plane of the objective. This pinhole effectively blocks out-of-focus light from above and below the focal plane from reaching the detector, therefore greatly improving the contrast of the acquired image. The pinhole also enables optical sectioning, thus allowing 3D imaging. As the sample excitation occurs point-by-point, the signal is also produced point-by-point. This signal is collected using photon counters, such as a photomultiplier tube, which can further enhance the detected signal (Wolf, 2007).

Spinning disk confocal microscopy is commonly used to achieve faster and more delicate sample illumination. In this technique, a disk containing thousands of micro-lenses and a disk with thousands of small pinholes arranged in a spiral pattern is placed on the light path (Gräf et al., 2005). The excitation light travels through both disks, and the emission light is directed to the camera-based detector after passing again the pinhole disk. This setup allows high-speed and gentle scanning of the sample and makes this system especially suitable for imaging dynamic processes in living cells (Stehbens et al., 2012).

1.1.4 Super-resolution Microscopy

In conventional microscopy techniques such as widefield and confocal microscopy, described above, the resolution of the resulting image is limited to the diffraction of light used for exciting the sample and the NA of the used objective. Using ultra-violet (UV) light, it is possible to distinguish objects that are maximally 250 nm apart. There have been numerous successful efforts to break the diffraction limit, such as stimulated emission depletion microscopy (STED) by Stefan Hell (Hell and Wichmann, 1994), photoactivated localization microscopy (PALM) (Betzig et al., 2006), and stochastic optical reconstruction microscopy (STORM) (Rust et al., 2006) by Erik Betzig and William E. Moerner. They shared a Nobel Prize in Chemistry in 2014 for their work in super-resolution microscopy.

While super-resolution imaging has revolutionized our ability to observe tiny cellular structures with unprecedented detail, it is important to acknowledge certain limitations when applying super-resolution techniques to live cell imaging. The use of intense illumination in super-resolution techniques can lead to phototoxicity and photobleaching, potentially affecting cell viability and sample integrity over time. Balancing high spatial resolution with temporal resolution remains a challenge, as some methods may require prolonged acquisition times, thus limiting their ability to capture fast dynamic processes (Jacquemet et al., 2020).

In our work, we have used a super-resolution technique called structured illumination microscopy (SIM) (Neil et al., 1997). SIM is a wide-field based microscopy technique commonly used for live-cell imaging as it can double the resolution compared to conventional widefield techniques (Gustafsson, 2000) and allow fast acquisition (Huang et al., 2018). In SIM, a movable diffraction grating is inserted into the excitation beam path. The resulting striped images interfere with each other at the focal plane of the objective, creating a striped pattern called the Moiré pattern. To reconstruct the final super-resolved image, several of the striped raw images must be collected, each acquired at a different orientation of the structured illumination, which are then combined by a computer algorithm to reconstruct a super-resolved image (Schermelleh et al., 2019; Jacquemet et al., 2020).

1.2 Requirements and limitations for live cell imaging

Creating a stable environment for live cell imaging is crucial to ensure that cells preserve their natural behavior. The additional stress imposed by imaging experiments has the potential to alter cellular characteristics. Unstable imaging conditions, coupled with excessively intense light, can compromise the accuracy of observations, leading scientists to observe and report the cellular responses to the unideal environment or light-induced cell damage (Magidson and Khodjakov, 2013). In this chapter, I highlight the importance of proper experimental planning for successful live cell imaging.

1.2.1 Microscope requirements and sample preparation

When examined outside of their natural environment, cells are fragile and easily affected by external conditions. To mitigate external factors that might affect the cell's health, live cell imaging experiments are performed in a microscope set-up that includes a special live cell imaging chamber that controls the temperature, pH, CO_2 levels, and humidity during the imaging experiment (Jensen, 2013). Fortunately, such chambers are common in modern microscopes, allowing long-term live-cell imaging.

Sample preparation is one of the most critical steps in any fluorescence microscopy experiment. For live cell imaging, cells should be placed in a special imaging chamber, usually with a glass bottom to match the NA of the microscope objective and kept in their growth medium to ensure cell health. Cell growth medium often contains a colorful pH indicator, such as phenol red, which might cause excitation and emission of light to be absorbed by the medium, reducing the signal or causing autofluorescence. Ideally, a clear medium should be used for fluorescence imaging of living cells (Waters, 2007).

One of the biggest challenges in live cell imaging is labeling structures and molecules of interest and imaging them without compromising cellular processes or health. Recombinant fluorescent proteins conjugated to the protein of interest can be directly expressed in an organism or a cell. A variation of organelle-specific dyes can also be added to the imaging medium before imaging. In our lab, we commonly use sir-DNA dye (Lukinavičius et al., 2015) or Lifeact (Riedl et al., 2008) to visualize cells.

1.2.2 Phototoxicity

In optical light and fluorescence microscopy, the sample is illuminated by light to visualize and acquire images. High fluorescence light exposure can induce reactive oxygen species (ROS) formation in the cell, which in excess can lead to cellular changes (Sies and Jones, 2020), whereas low exposure to light usually causes minimal and reversible effects on the cells (Wäldchen et al., 2015). In more severe cases, the impact of light can cause cells to be permanently changed or even lead to cell death (Laissue et al., 2017).

While ongoing efforts in hardware development persist to achieve milder illumination and enhance detector sensitivity to mitigate phototoxicity, (Castello et al., 2019; Schmidt et al., 2011) modern microscopes still have their limitations. This leads to necessary optimization and trade-offs in spatial and temporal resolution, illumination power, and image quality. Image quality and spatial and temporal resolution are co-dependent. Optimization of these factors is often coupled with increased exposure to light, which usually comes at a cost to sample health (Laissue et al., 2017). No single factor can be changed without affecting the other; therefore, these factors form the pyramid of frustration (Figure 3).

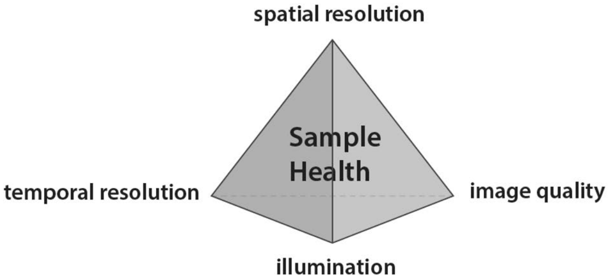


Figure 3: The pyramid of frustration. The integrity of the sample is of the highest priority when conducting a live cell imaging experiment and therefore placed in the middle of the pyramid. Image quality, spatial and temporal resolution are co-dependent and directly affect the sample's health. Enhancing resolution and image quality often leads to heightened light exposure, which, in consequence, compromises the well-being of the sample. On the other hand, efforts to retain the sample health during acquisition typically lead to decreased resolution and noisy images. Therefore, careful balancing between these factors is essential when conducting live cell imaging experiments. Modified from (Lemon and McDole, 2020).

To acquire meaningful data, sample health holds the highest priority in live cell imaging. If the effect of fluorescence microscopy on sample health is not minimized, the images might be falsely interpreted, leading to mistaken conclusions. In situations where using low illumination power is the only way to image the interested process, the image quality can be significantly improved by using image post-processing methods, which are further discussed in the chapters 2.2.2 and 2.3.2.

1.2.3 Photobleaching

Photobleaching is a process where fluorophores irreversibly lose their fluorescent capacities due to excessive light exposure over prolonged excitation. In their excited state, fluorophores can react with oxygen and become degraded (Laissue et al., 2017). As many live cell image quantifications rely on the detection of fluorescence intensities, avoiding photobleaching enables improved visualization, processing, and analysis of the acquired image data.

To overcome the problem of photobleaching, instrument environment, image acquisition, and sample preparation can be adjusted. Selecting a suitable fluorophore, employing anti-fading agents, utilizing lower laser excitation power, enhancing detector sensitivity, and extending the time interval for live image capturing all contribute to reducing photobleaching during imaging (Waters, 2007). Alternatively, photobleaching can be minimized by using gentler illumination techniques, such as light sheet microscopy (Wu et al., 2011). In situations where photobleaching cannot be completely eradicated, post-acquisition image processing plays an important role. These are further discussed in the chapter 2.2.1.

1.2.4 Sample stabilization

In addition to the above-discussed challenges in live cell imaging, imaging is prone to drifting: the inability of a microscope to maintain the selected focal plane over an extended period. Drifting can be caused, for example, by temperature changes leading to thermal expansion of the microscope mechanical components or moving of the sample. Drifting primarily affects the focus but can also lead to lateral drifting. To minimize drift, it is essential to allow the microscope to achieve thermal equilibrium before starting time-lapse experiments and use auto-focus systems to maintain the focus.

Drift can be corrected in several ways during image acquisition. During image acquisition, microscopes can be equipped with an autofocus system that uses infrared light that is reflected on the glass-sample interface and captured by a detector (e.g., Leica's Adaptive Focus Control or Nikon's Perfect Focus System). Lateral drift can be compensated in real-time by tracking algorithms that follow the sample and correct the stage position accordingly (von Wangenheim et al., 2017; Fox et al., 2022). However, drifting is rarely completely eliminated at the acquisition stage, especially when acquiring multiple positions for an extended period of time or when the sample itself moves, thus, post-processing of the image data is required before visualization and downstream analyses. Methods for post-acquisition drift removal are discussed in the chapter 3.1.

1.2.5 Local image data infrastructure

Live cell imaging data is often large in size and can easily exceed the petabyte (1 PB = 1000 TB) scale at a single institution level (Ouyang and Zimmer, 2017). Therefore, it is crucial to establish an organized data infrastructure to handle this data. Such infrastructures facilitate seamless image data transfer from the microscope computers, where it is initially stored during acquisition, to users for efficient analysis, enabling streamlined image data utilization for scientific discoveries.

The image data transfer from the microscope computer to users can be handled for example using institutional network drives or dedicated image handling servers such as Open Microscopy Environment Remote Objects (OMERO) (Allan et al., 2012) or the Acquifer HIVE (Acquifer) (Muehlboeck et al., 2014). Data management solutions and storage solutions can also provide additional features. For example, the OMERO servers support multiple image data file formats and store the image metadata (Li et al., 2016), which is essential for image data management and further downstream analysis. Additionally, OMERO allows the usage of image analysis software to access images on OMERO and save results back to the server (Burel et al., 2015). With OMERO.figure (https://github.com/ome/omero-figure, 12.03.2024) it is possible to create figure panels for scientific publications that trace back to the original data on the server ensuring efficient data management and traceability. These features greatly reduce the hurdle in image data handling streamlining the efficient use of image data.

Despite the available technologies, storing and sharing this data still remains a bottleneck in many institutions. Data transfer through such networks or servers requires fast network connections and an established infrastructure, which are still lacking in many institutions. Consequently, numerous researchers continue to rely on external hard drives for the transmission of their image data (Andreev and Koo, 2020).

1.2.6 Skills to conduct and report image analysis

Bioimage analysis skills have become an essential part of cell biology. With microscopes capable of generating a large amount of multi-dimensional live cell imaging data, automated analysis has become necessary. Bioimage analysis is an emerging research field that has rapidly progressed over the last couple of decades, introducing multiple new tools especially for multi-dimensional image data, such as live cell imaging data (Pylvänäinen et al., 2023b). This rapid development has led to confusion among life scientists in identifying and utilizing suitable tools, mainly due to a lack of basic image analysis knowledge, incomplete documentation, lack of coding skills or insufficient communication between developers and life scientists (Schlaeppi et al., 2022). The imaging community has reported a growing need for training in image analysis skills (Sivagurunathan et al., 2023).

To keep up with the developing field and to be able to harness image data more efficiently, continuous training in the field is required. Local support in image analysis is already incorporated through many MSc and PhD level courses, but these courses typically only scratch the surface of what is needed for in-depth bioimage analysis. NeuBIAS initiative has successfully breached this gap by organizing training schools for early career investigators, bioimage analysts and core facility personnel. These training schools have played a vital role in enhancing researchers' comprehension of image analysis and disseminating knowledge about novel tools for image analysis, thereby directly influencing their experimental design for live cell imaging studies. (Martins et al., 2021).

Further breaching this gap, the number of online training resources has peaked in past years (Whiting et al., 2022), including for example video series such as the NeuBIAS Academy (https://eubias.org/NEUBIAS/training-schools/neubias-academy-home, 06.03.2024), iBiology (https://www.ibiology.org/?s=image+analysis, 06.03.2024) and online books such as the Introduction to Bioimage Analysis -gitbook (https://bioimagebook.github.io/index.html, 08.03.2024). For example, the NeuBIAS academy has provided multiple training videos for the processing and analysis of live cell image data (https://www.youtube.com/@NEUBIAS/videos, 12.03.2024). These videos, viewed by thousands of people, have encouraged the integration of novel methods into life sciences research. Together with the in-person workshops, online training resources have played a crucial role in the knowledge acquisition of bioimage analysis, consequently reducing the barriers for researchers to experiment with new bioimage analysis tools for their research.

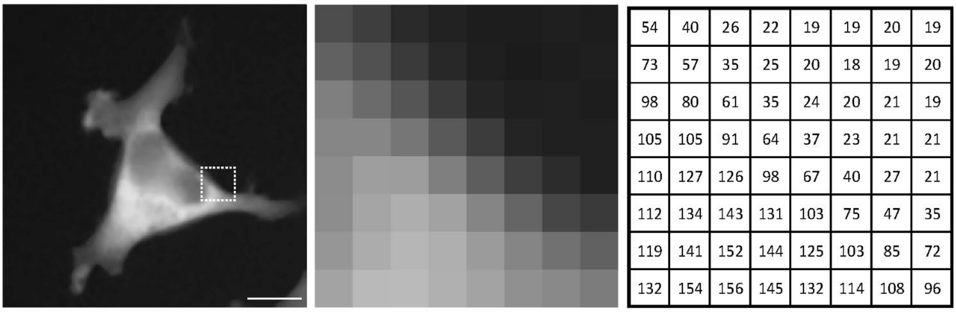
The support from the image analysis community, such as the image.sc forum (https://forum.image.sc/, 08.03.2024), further enhances the acquisition of image analysis skills and proves particularly useful in troubleshooting (Jamali et al., 2021). Based on common issues in the field, community efforts have led to the establishment of international recommendations on how to report image analysis in scientific articles (Schmied et al., 2024) and how to ensure that image analysis is reproducible (Aaron and Chew, 2021). These guidelines aim to guide life scientists to accurately analyze, report and share their work.

2 Microscopy image processing

In optical microscopy, after light has reached the sample, it is directed through the optical system to a detector. Depending on the microscope modality used, the detector can be a camera that records color or grayscale images or a photomultiplier tube (PMT). This detector then translates the image into a digital image, which can then be viewed using a computer screen (Sanderson, 2020). While the acquired images are usually beautiful to look at, they also contain enormous amounts of useful information for life sciences research. In this chapter, I introduce images and how information can be extracted from them using classical and artificial intelligence (AI) boosted image analysis. The primary focus is on elucidating methods and applications specifically tailored for live cell imaging.

2.1 Images are just pixels

At their simplest, digital images are 2D arrays that are composed of picture elements called pixels. Each pixel holds a numeric value that represents its gray value. The gray values are essential in visualizing digital images, while pixel values are crucial for in-depth image analysis (Figure 4).



54	40	26	22	19	19	20	19
73	57	35	25	20	18	19	20
98	80	61	35	24	20	21	19
105	105	91	64	37	23	21	21
110	127	126	98	67	40	27	21
112	134	143	131	103	75	47	35
119	141	152	144	125	103	85	72
132	154	156	145	132	114	108	96

Figure 4: An image is a collection of pixels, where each pixel has a numerical value that represents its gray value. Here, an 8-bit image of an MDA-MB-231 cell is cropped to an 8x8 region, and its corresponding pixel values are visible in the 2D array on the right. Scale bar 10 µm.

Grayscale image pixels can take different values depending on the image type, for example, an 8-bit image, each pixel can take values from 0-255, and a 16-bit image from 0-65535. The image bit depth defines the dynamic scale of the image; the greater the bit depth, the greater the number of tones that can be represented in the images. A greater bit depth might be beneficial in a situation where the signal-to-noise ratio (SNR) is low, though storing such images requires more space. Red-green-blue (RGB) images, on the other hand, are composed of three images: red, green, and blue overlayed.

Therefore, each pixel in the RBG image has three values, one for each layer. In this thesis, I focus only on grayscale images.

The display of grayscale images can be enhanced using look-up-tables (LUTs) that provide a way to alter the appearance of an image without changing the pixel values. Choosing an appropriate LUT can improve the understandability of the image contents, although, for single-channel images, grayscale color schemes are preferred for unbiased interpretation of the image (Schmied et al., 2024).

2.2 Classical image processing methods and applications

Image processing refers to the modification or enhancement of digital images using various operations such as filtering, segmentation and geometric transformations. Classical image processing operations can be applied to microscope images to remove bleaching from live cell videos, to reduce noise and to improve object segmentation. Image processing enhances the image for downstream analysis, where meaningful information is extracted from the image (Gonzalez and Woods, 2018).

In the simplest analyses, the image intensity of the whole image frame is measured, or the objects of interest are manually counted. To extract information about the morphology or intensity of the objects in the image, they are first segmented using for example thresholding followed by connected components labeling. These methods translate the image to a computer-readable mask, highlighting regions of interest (ROI) for downstream analysis (Peng, 2008). Image modification and enhancement play a key role in improving the accurate segmentation of objects.

Due to many different biological specimens and microscopy methods, captured images vary enormously. As technology advances, so does the complexity and sophistication of image processing methods, offering new solutions to improve image analysis tasks. For these reasons, it is not possible to find one correct way to process images; rather, look for the least wrong approach.

2.2.1 Bleach correction

Bleach correction is a process where the light-induced photobleaching is corrected computationally post-acquisition. Usually, bleach correction algorithms compare the average intensity of the entire image at each point in time to define the amount of bleaching, followed by matching the intensity of the following frames to the intensity of the first frame. More advanced methods use the information acquired from the image histogram to compute a similar histogram for each time point, taking the first frame as the reference (Miura, 2020).

The performance of intensity ratio-based bleach correction algorithms relies on the estimated intensity of the background. Variations of the background may cause unsuccessful bleach correction. The histogram matching method assumes a stable distribution of fluorescence throughout the video. Changes in the signal distributions or cell shapes typically cause changes in the image histogram, leading to unsuccessful bleach correction. To choose the best bleach correction method, these limitations and the observed biological phenomenon should be considered. Successful bleach correction can improve sample visualization, processing, and downstream analysis of the acquired live cell imaging data (Miura, 2020).

2.2.2 Noise and denoising

Live-cell imaging is always a compromise between image quality and sample health (Figure 3) and retaining the sample health usually comes at the cost of image quality. Live cell imaging using fluorescent light requires low illumination power and fast capturing, often leading to the acquisition of noisy images.

In microscopy, Gaussian and Poisson noise can be considered as the two major sources of noise (Rasal et al., 2022). These noises are random in nature and equally plague the entire image. Gaussian noise is mainly caused by electronic thermal vibration inside the detector and arises at the stage of quantifying the number of photons detected for each pixel. Therefore, Gaussian noise is also called the read noise. On the other hand, Poisson noise is caused by the photons arriving randomly at the microscope detector; therefore, it is also called photon noise. To reduce image noise, it is necessary to increase the illumination power to ensure that an adequate quantity of photons reaches the detector. However, the increase in illumination power can negatively impact the sample's health (Laissue et al., 2017).

Image denoising methods offer a powerful way to recover high-quality images and facilitate downstream processes such as image segmentation. In successful noise reduction, certain aspects should be considered: areas expressing similar signals should be smooth, edges should be protected without blurring, and textures should be preserved. In addition, noise reduction should not introduce new artifacts (Fan et al., 2019).

Filtering the image in the spatial domain, where the image is presented in its coordinate space (each pixel or sample corresponds to a specific position in the image), is a classical method to remove noise. In filtering, a kernel of a specific size is moved across the original image, calculating a new gray value for each pixel based on the surrounding pixels within the kernel (Li et al., 2010). The new pixel values are calculated differently based on the selected filter type. Linear filters, transform the pixel values based on a defined mathematical kernel. For example, the mean filter averages the pixel values in the defined neighborhood of each pixel, resulting in a smoothing or blurring effect. It helps reduce noise and highlight larger-scale features and is commonly used to remove Gaussian or Poisson noise (Gonzalez and Woods, 2018). Non-linear filters, such as the median filter, can help with the detection of edges that have a steep increase or decrease in the signal (Anoraganingrum, 1999). The median filter moves the kernel across the image and calculates the new gray value for each pixel by first sorting all the pixel values within the kernel to numerical order and replacing the pixel being considered with the middle (= median) value. In images with high impulses of noise, the median filter gives better denoising compared to the mean filter, as it is not significantly affected by unrepresentative pixel values in the neighborhood. Furthermore, since the new median value corresponds to an actual value from the target pixel environment, not a newly calculated value, the median filter is much better at preserving sharp edges and texture than the mean filter.

Image denoising can also be performed using image transforms. These methods first transform the given noisy image to another domain, such as the frequency domain, where the image is represented in terms of its frequency components, and then apply a denoising procedure on the transformed image before returning it to the spatial domain. Among these, the wavelet domain methods such as block-matching 3D (BM3D) (Dabov et al., 2007) have been widely used (Chen et al., 2013). Although deep learning (DL) methods have gained popularity in denoising in recent years (discussed in chapter 2.3.2) some newer wavelet transform methods can still outperform these when working, especially with microscopy images (Silberberg and Grecco, 2024).

2.2.3 Image segmentation

The human brain is excellent at separating objects from their background, while computers struggle with this task, especially in the presence of an irregular background. Therefore, before a computer can automatically measure anything, it first needs to separate the image into foreground, including the objects of interest and background, a process called image segmentation. Using segmentation methods, it is possible to detect objects such as cells or nuclei from their background.

The easiest way to segment an image is by using histogram-based thresholding, where a global threshold is applied to the whole image. This threshold identifies pixels that are above or below a selected threshold value, giving a binary image as the output (Figure 5). Global thresholding assumes the presence of two classes of pixels in the image: the foreground above the selected threshold and the background pixels below the selected threshold.

Manual selection of the threshold easily introduces bias; therefore, the usage of segmentation algorithms is recommended. One of the most applied thresholding algorithms is the Otsu method (Otsu, 1979), which uses the grayscale histogram of an image to find an optimal threshold value that separates two classes with minimum intensity variance within each class.

Uneven illumination of a sample is a frequent problem in fluorescence microscopy, which can be caused, for example, by misaligned microscopy optics or stage or issues in the sample mounting. Uneven illumination leads to varied background intensities in the image and further problems in segmentation. In such situations, a local thresholding, which generates a threshold for each pixel value relative to the local background, can be used.

The output of the segmentation is called a mask, which can be used to assess various features of the entire foreground quantitatively. To comprehensively analyze each object in the image, it is necessary to assign a unique label to each object (Figure 5). This labeled image can be easily generated directly from the image mask using connected component labeling, which groups connected foreground pixels into distinct objects. Alternatively, different regions with specific features can be classified using semantic segmentation (Figure 5).

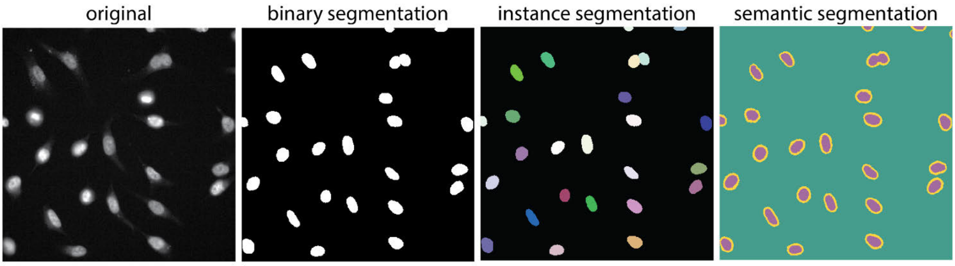


Figure 5: Types of image segmentation. Images can be segmented using thresholding, which results in a binary mask image. To analyze individual objects in an image, a unique identifier needs to be assigned for each object, an operation called instance segmentation. Different regions of the image can be classified using semantic segmentation. Here, the green color is classified as the background; the objects are purple, and the object edges are yellow.

However, in some cases, objects in the original image may be too close to each other, causing the thresholding process to combine them into one single object. In such situations, it's crucial to refine the mask further using binary operations before connected component labeling. These operations include tasks like reducing or expanding the masked area, which can help separate objects or remove small objects in the background. One popular method to separate touching nuclei masks involves using a watershed algorithm (Figure

6). This algorithm is particularly effective at separating closely touching objects, provided valleys (or ridges) with different binary pixel values exist between them.

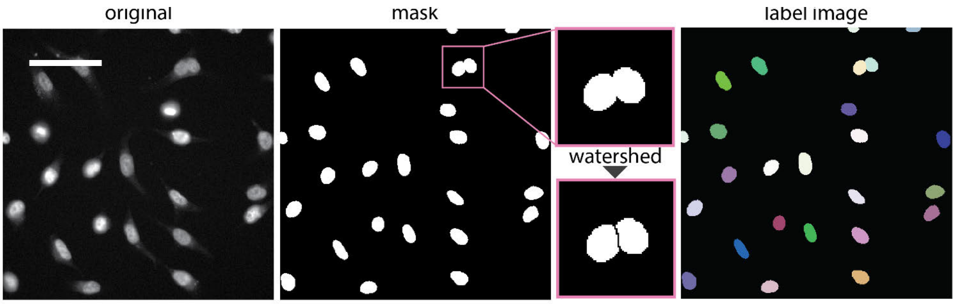


Figure 6: Image segmentation. In image segmentation, foreground, including the objects of interest and background, are defined. A histogram-based thresholding (Otsu) has been applied to the original image to create an image mask containing the objects of interest. A binary operation watershed was applied to the mask to separate touching objects. Connected components labeling was applied to the watershed image to create a label image. The resulting label image can be used for measuring object features such as object area or roundness. Scale bar 50 µm.

2.3 Deep learning boosted image processing

In the past years, the amount and complexity of image data have dramatically increased, leading to a need for more complex and computationally expensive processing and analysis tools. Artificial intelligence (AI), machine learning (ML) and especially their subgroup deep learning (DL) have provided a powerful solution for tackling these issues. AI is a field of computer science that focuses on developing intelligent machines that can perform tasks that typically require human intelligence. ML is a subset of AI that focuses on the development of algorithms that can learn patterns and insights from data and then use this knowledge to make predictions or decisions. In cell biology, ML algorithms, such as the random forest classifier, are used to perform pixel-based classifications based on extracted image features and for interpretation of classification and feature importance (Berg et al., 2019; Arganda-Carreras et al., 2017). DL is a subset of ML that uses multi-layered neural networks, called DL networks, to simulate the complex decision-making power of the human brain. In cell biology, DL methods are commonly used for tasks such as image segmentation, object detection, feature extraction, denoising, and image restoration (Moen et al., 2019; Pylvänäinen et al., 2023b). DL methods have been shown to outperform traditional image processing and analysis algorithms frequently (Ronneberger et al., 2015; Krizhevsky et al., 2012).

In this chapter, I discuss the usage of DL methods, highlighting tools and DL networks I have found most usable for my research in the context of live cell imaging.

2.3.1 Deep learning algorithms and models

A standard DL workflow involves training a multi-layer artificial neural network, commonly referred to as a deep neural network (DNN) or DL network for short. This network generates a DL model capable of executing a particular task. The DL network has hidden layers that allow the network to learn features from the input data. In more detail, a DL network takes raw data as input at the lowest layer and uses the first layer to learn the simplest features of the data. This information is then passed to the next layer and used to learn more complex features. This is repeated throughout all the layers of the DL network to learn the most complex features. The UNET DL network architecture was one of the first proposed networks designed for biomedical image segmentation tasks (Ronneberger et al., 2015). It has played a crucial role in advancing the application of DL for segmentation tasks, and its architecture's versatility has led to its widespread adoption in various segmentation applications across different domains. Figure 7 explains the U-Net DL network architecture in more detail.

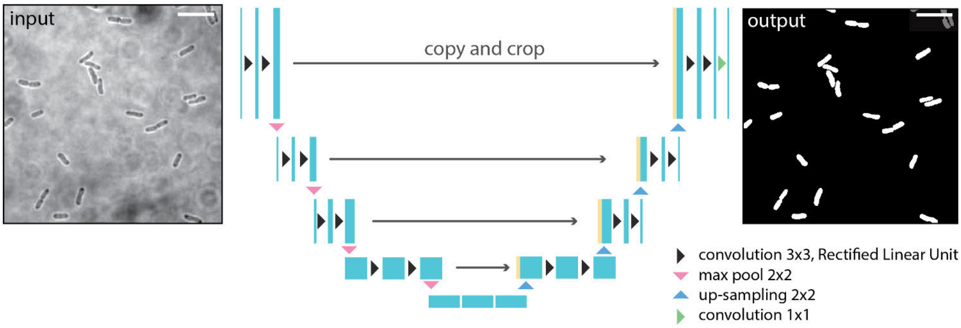


Figure 7: U-Net DL architecture for semantic segmentation. U-Net consists of contracting and expansive paths. The contracting path consists of applications of two 3x3 convolutions, each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation. At each down-sampling step, the number of feature channels is doubled. The expansive path consists of an up-sampling of the feature map followed by a 2x2 up-convolution, copying the corresponding cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. At each up-sampling step, the number of feature channels is halved. As the final step, a 1x1 convolution is applied to map each 64-component feature vector to the desired number of classes. In total, the network has 23 convolutional layers. Scale bar 10 µm. The input and output images are from (Spahn et al., 2022).

To effectively train a DL model, DL networks require specific types of training data tailored to the task that the network is expected to perform. Proper training using appropriate input data is of the highest importance, as it directly influences the DL model's performance. In image processing, training is typically approached through three main methods: supervised, unsupervised, or self-supervised training (LeCun et al., 2015). During supervised DL training (Figure 8A), the network is exposed to paired source and target images. Using this prior information, the network learns how to transform the source image into the target image. In contrast, in unsupervised DL training (Figure 8B), unpaired source and target images are introduced to the network. The network discovers patterns, structures, or representations within the data to learn how to transform the source image to its target. In the self-supervised approach (Figure 8C), the network learns inherent image features to perform the desired image transformation task without the need for training data. One big advantage of using DL in image analysis is its ability to extract features not visible to the human eye and, therefore, minimize human bias (Hinton and Salakhutdinov, 2006).

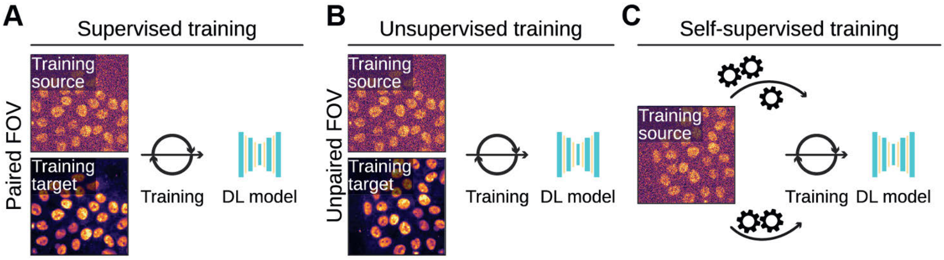


Figure 8: Deep learning algorithms. Training of a DL model can be done using a supervised, unsupervised or self-supervised approach. In the supervised DL approach (A), paired source and target images are introduced to the network whereas in unsupervised training (B) unpaired source and target images are introduced. Using this prior information, the DL network learns how to transform the source image into the target image. In the self-supervised approach (C), networks learn inherent patterns in the images to perform the desired task. Image modified from (Pylvänäinen et al., 2023b).

Training DL models is computationally demanding and time and data expensive. Additionally, it is difficult to estimate a sufficient number of training rounds (epochs) and images. To optimize the training process, it is common to use pre-trained models as a starting point and improve them using own image data via transfer learning (Pan and Yang, 2010). In transfer learning, the used model is improved by introducing additional images, image augmentation, and training for additional epochs to generate an improved model for a desired task (von Chamier et al., 2021).

There are numerous pre-trained DL models to conduct different DL tasks openly available in model zoos such as the Bioimage model zoo (Ouyang et al., 2022). One of the most popular pre-trained segmentation models for nuclei segmentation is the StarDist versatile nuclei model trained with a subset of 670 nuclei images from diverse modalities of the DSB 2018 nuclei segmentation challenge dataset (Schmidt and Weigert, 2022; Caicedo et al., 2019). I have used this model as such and as the basis for generating my custom StarDist models in several DL tasks in this thesis.

There are multiple initiatives to help life scientists train their models and incorporate pre-trained models into their research without having to learn programming. ZeroCostDL4Mic platform (von Chamier et al., 2021) has democratized DL training by providing an extensive collection of DL networks through user-friendly Google Colab notebooks, which can utilize the graphics processing units (GPUs) provided by Google for faster model training. The Cellpose 2.0 (Pachitariu and Stringer, 2022) allows interactive human-in-the-loop training of Cellpose models. These trained models can then be directly utilized as part of a complete image analysis workflow, for example, in Fiji using the DeepImageJ plugin (Gómez-de-Mariscal et al., 2021) or in QuPath (Pettersen et al., 2022).

2.3.2 Deep learning to remove noise

Live cell imaging data encounters various challenges, typically arising from the necessity to find a balance between image quality and sample health, leading to the acquisition of images with low SNR (Icha et al., 2017). There has been substantial progress in how SNR can be improved with the help of denoising DL networks (Figure 9A).

Denoising DL networks can significantly improve image quality and allow the usage of lower illumination power for imaging. Supervised denoising networks such as CARE (Weigert et al., 2018) and 3D-RCAN (Chen et al., 2021) can effectively improve the SNR of microscopy images. Clever planning of the experiment and training datasets enables efficient restoration of image quality and can even enhance the resolution when trained with an appropriate dataset (Laine et al., 2021a). Usually, the training is performed using a paired dataset of fixed samples imaged with a higher resolution and illumination power and then in expected low SNR conditions (Spahn et al., 2022). Depending on the selected DL network and the instrument set-up, in some cases, the training datasets can be generated simultaneously during the image acquisition (Wagner et al., 2021).

In cases where no paired training data is available, it is also possible to utilize self-supervised DL networks, such as Noise2Void, for DL-enabled denoising. This network learns directly from the input image, and in the case where the noise is random, it can even perform equally well when compared to networks that require paired training datasets (Krull et al., 2019). DL-enabled denoising methods are efficient in denoising image data for morphological or spatial analyses. However, they, nor classical denoising methods, should not be used prior to image intensity analyses, as they may introduce artifacts and distort pixel intensities, potentially leading to erroneous intensity quantification (Laine et al., 2021b).

2.3.3 Deep learning for artificial labeling

In cases where it is necessary to label a structure of interest in a cell, but the fluorescent illumination severely damages the cells, alters their behavior, or in the absence of access to suitable dyes or instrumentation, artificial labeling can help. Artificial labeling in the context of microscopy typically involves the application of generative adversarial networks (GANs) for image-to-image translation, enabling the prediction of cellular structures. Artificial labeling employs DL models, such as pix2pix (Isola et al., 2017), that have been trained, for example, with pairs of brightfield and fluorescent images, to predict the fluorescent staining in unseen brightfield images (Figure 9B) (Christiansen et al., 2018; Ounkomol et al., 2018).

2.3.4 Deep learning for improvement of spatiotemporal resolution

The spatial resolution of light microscopes is usually limited to 250 nm due to the diffraction of light, hindering the possibility of observing the dynamics of small objects. Various super-resolution strategies have tried to break the diffraction limit (Jacquemet et al., 2020), although usually at the expense of the sample's health. Classical image processing methods, such as deconvolution (Soulez et al., 2012) or Wiener filtering, are commonly used to enhance the resolution of fluorescent microscopy images. Additionally, DL networks can be used to improve the spatial resolution of fluorescent microscopy images (Figure 9C). For example, single-frame deep-learning super-resolution microscopy (SFSRM) super-pixelates an image and predicts missing details, reaching up to 30 nm spatial resolution (Chen et al., 2023). The Fourier channel attention network (DFCAN) deploys the Fourier domain for robust SIM reconstruction, even under low SNR imaging conditions and high noise, achieving comparable image quality to SIM (Qiao et al., 2021). Other DL networks can aid the reconstruction of images acquired using super-resolution techniques. Such networks can successfully reconstruct, for example, SIM (Jin et al., 2020) or single-molecule localization microscopy (SMLM) images (Speiser et al., 2021).

DL networks can also be used to improve the temporal resolution of the video (Figure 9D). This is especially useful when long acquisition times are required for imaging a certain process, but the sample health is compromised by light or the sample bleaches easily. Slower imaging also allows cells to recover from possible light-induced cellular damage. A content-aware frame interpolation (CAFI) network can predict missing frames to improve downstream analysis, for example, tracking (Priessner et al., 2021).

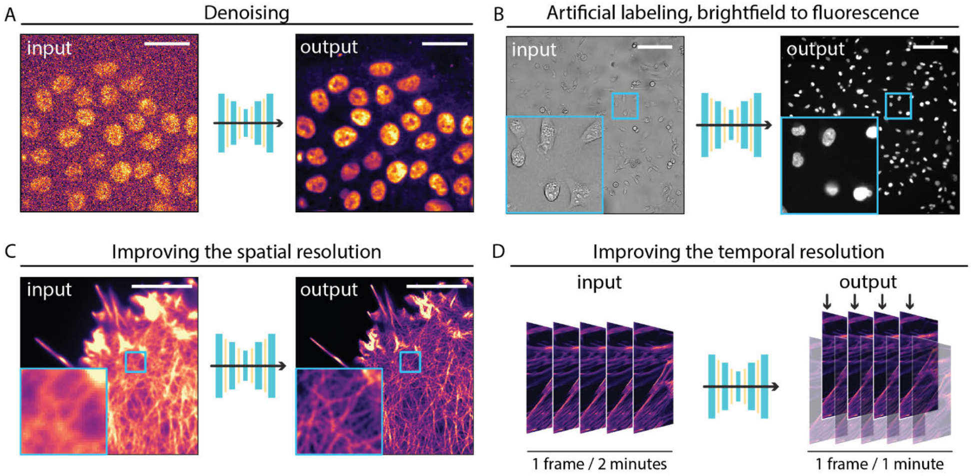


Figure 9: Deep learning for improving live cell imaging data. A) Noisy images of nuclei acquired using a spinning disk confocal microscope, denoised using a CARE 2D model. Scale bar: 50 µm. B) Brightfield microscopy was used to image migrating breast cancer cells, and the nuclei image was digitally generated from the brightfield image using a Pix2pix model. Scale bar: 100 µm. C) Cells labeled with Life-Act were imaged using a widefield microscope. The increased image resolution was achieved using the DFCAN DL network. Scale bar: 5 µm. D) Illustration of how a DL network like CAFI can enrich the temporal resolution of a live cell imaging dataset through smart interpolations. Image adapted from (Pylvänäinen et al., 2023b).

2.3.5 Machine and deep learning for segmentation

In live cell imaging, the goal is often to learn about cell movements and their behavior using tracking. Tracking commonly relies on successful object recognition, which has been greatly aided by ML and DL algorithms (Lucas et al., 2021). Although this chapter mainly describes DL methods for image segmentation, some ML methods and tools deserve to be mentioned in this context.

In ML approaches for microscopy, a user annotates an image by labeling and assigning the annotations to two or more classes. The image features in the annotations are passed on to a powerful algorithm, the classifier. Based on the user annotations, the classifier builds a decision surface in feature space and projects the class assignment back to pixels and objects of the whole image. The user can further optimize the annotations interactively to improve the classifier. ML learning algorithms are available for life scientists, for example, through the Weka trainable segmentation Fiji plugin (Arganda-Carreras et al., 2017) and ilastik (Berg et al., 2019). Both these tools provide non-coding environments for training ML algorithms for pixel classification. Ilastik also extends the pixel classification to enable object classification and segmentation of objects based on their boundary information.

Preparing data for subsequent downstream analysis often involves the segmentation of nuclei or entire cells. In this context, two DL models, namely StarDist and Cellpose, provide excellent segmentation capabilities (Figure 10). StarDist DL network (Figure 10A) is designed to detect Star-convex polygons and, therefore, is well suited for nuclei segmentation (Schmidt et al., 2018). On the other hand, the Cellpose algorithm, commonly used for whole-cell segmentation (Figure 10B), employs a DL network that first detects and segments cells by identifying their boundaries and then assigns labels to each segmented cell based on learned patterns and features (Stringer et al., 2021). Segment anything for microscopy algorithm is one of the newer additions to the DL segmentation networks, allowing training of generalist models for multiple imaging modalities (Archit et al., 2023). All these above-mentioned algorithms output label images, allowing robust analysis using analysis software such as Fiji (Schindelin et al., 2012) or CellProfiler (Stirling et al., 2021) and can be directly integrated into cell tracking tools such as TrackMate (Ershov et al., 2022).

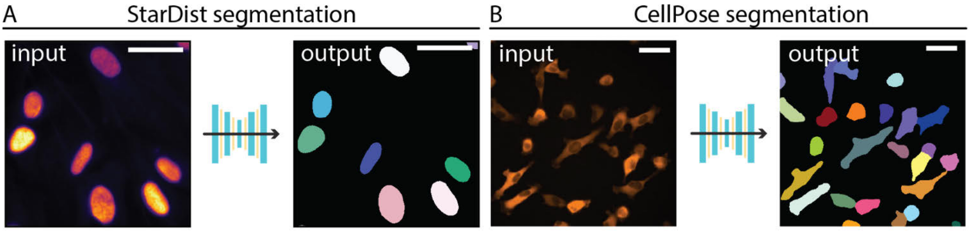


Figure 10: Deep learning enabled image segmentation. A) Cell nuclei were imaged using a widefield microscope and segmented using the Versatile nuclei StarDist model. Scale bar 50 µm. B) Cells expressing a cytoplasmic reporter were imaged using a widefield microscope and segmented using a custom-made Cellpose model based on the Cyto2 model. Scale bar 50 µm.

2.3.6 Quality assessment of deep learning models

Quality control of the trained DL models is essential in any DL pipeline. A common problem when applying DL to image data is that the quality control is done by visual inspection rather than quantitative analysis. In this chapter, I focus on the quality control metrics and tools that I have used in my thesis.

Segmentation is an essential step in multiple image analysis workflows. Commonly used DL networks such as StarDist (Schmidt et al., 2018) and Cellpose (Stringer et al., 2021) usually provide excellent segmentation results that can be further improved by transfer learning. Often, the question is, "*When is my deep learning model good enough?".* In DL, quality control is a two-step process. First, the loss functions over the number of epochs trained are inspected to monitor the training. Second, image quality metrics, between the model predictions and manually annotated ground truth, are typically compared (von Chamier et al., 2021). Ground truth images play a significant role in training supervised DL models and when assessing the trained model's performance. The ground truth images can be generated directly at a microscope or annotated manually. Manual annotation is a slow and laborious process, but fortunately, there are several tools available to help in the generation of manual annotations (Arzt et al., 2022; Schindelin et al., 2012; Bankhead et al., 2017; Gupta et al., 2022).

During DL model training, part of the labeled training data is set aside as validation data. At the end of each epoch, the model performance is compared to this validation data to generate two losses, a training and a validation loss. These are calculated by computing the difference, for example, calculating the MSE values between the predicted and the actual images at the end of each epoch. During training, the goal is to minimize the loss function; a smaller loss function indicates that the model's predictions are closer to the actual target image. Training loss tells how well the model performs on the training data, whereas the validation loss tells how well the model performs on the validation data. If the training loss and validation curves are both decreasing at the end of the training, typically the model should be trained for more epochs, whereas a flattened curve indicates that no more training is needed (Figure 11A). In a case where the validation loss increases while training loss simultaneously decreases (Figure 11B), the network is overfitting, meaning that it remembers the exact patterns from the training data and no longer generalizes well to unseen data. In such a case, the training dataset should be increased, or the training parameters should be adjusted. Overall, a comparison between the development of the validation loss with the training loss can give insights into the model's performance.

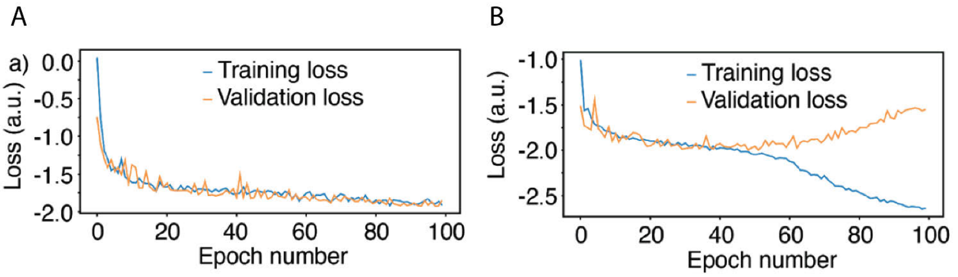


Figure 11: Assessment of training and validation loss during training. During training, a DL model part of the training data is set aside for validation. At the end of each epoch, the model performance is compared to this validation data to generate two scores, a training, and a validation loss. Training loss tells how well the model performs on the training data whereas the validation loss tells how well the model performs on the validation data. A) If both training and validation loss curves are flattened no more training is needed. B) In a case where the validation loss increases while training loss simultaneously decreases, the network is overfitting, meaning that it remembers the exact patterns from the training data and no longer generalizes well to unseen data. In such a case, the training dataset should be increased.

In segmentation models, the overall model performance can be evaluated by comparing the ground truth images to the prediction of the DL model. One of the most used metrics is the Jaccard index (Jaccard, 1901) also called the intersection over union (IoU), which measures the overlap between objects on a scale from 0 (no overlap) to 1 (perfect overlap). This metric can be used to assess the quality of the used DL model to accurately predict structures of interest.

For instance segmentation, it is useful to evaluate the model performance in a level of individual objects, by comparing true and false objects between the ground truth and the prediction (Figure 12). True positive objects (TP) are objects that have an IoU above a given threshold between the ground truth and the segmentation result. False positives (FP) are not present in the ground truth but are in the segmentation result, and finally, the false negatives (FN) are present in the ground truth but missing in the segmentation result. Out of these values, accuracy, precision, and recall can be calculated. The accuracy score displays the ratio of the sum of TP and TN out of all predictions indicating how well the segmentation result performs in separating the foreground and background. The precision score is the count of TP out of all positive predictions. Recall score (or sensitivity) measures the count of TP out of all actual positive values. F1 score is the harmonic mean between precision and recall score (Laine et al., 2021, https://focalplane.biologists.com/2023/04/13/quality-assurance-of-segmentation-results, 06.03.2024).

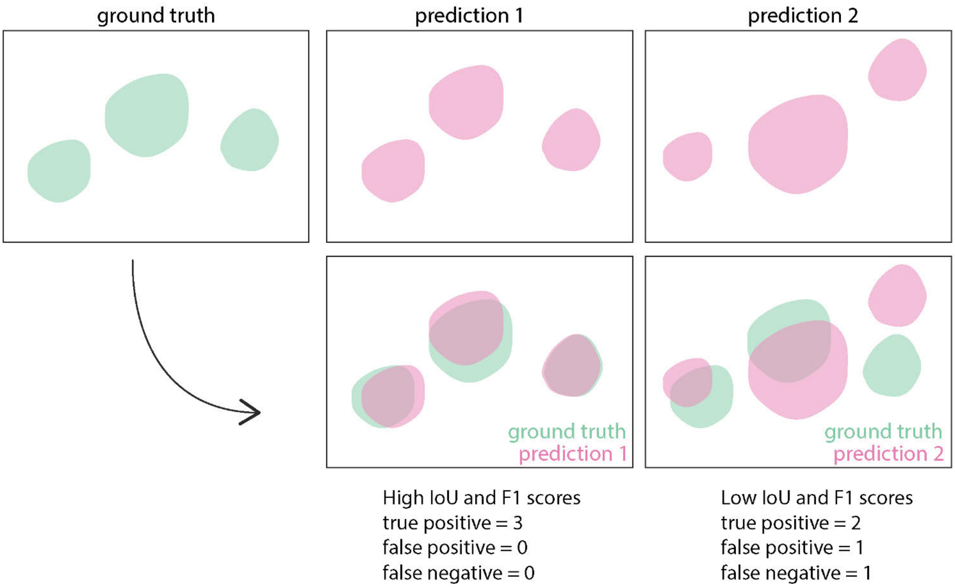


Figure 12: Segmentation quality control. The ground truth image (green) is compared with the output predictions (pink). When the predicted segmentation is close to the ground truth, IoU and F1 scores are high (prediction 1), whereas in cases where the prediction differs from the ground truth, the IoU and F1 scores are low (prediction 2). In the context of instance segmentation, the segmentation performance of a DL model can be assessed by comparing the numbers of overlapping objects above a given IoU and non-overlapping objects between the prediction and the ground truth images. This comparison yields additional insights into the model's segmentation capabilities.

Pix2Pix (Isola et al., 2017) is a type of conditional GAN designed to perform image-to-image translation tasks. To assess the performance of a pix2pix model, unlike in segmentation models, inspection of the loss curves is not useful. Pix2Pix introduces adversarial training, where a generator learns to create realistic-looking images, while a discriminator tries to learn to distinguish between real and generated images. The generator is guided by an adversarial loss, which calculates the difference between real and generated images produced by the generator. The discrimination is guided by a pixel-wise loss, which measures the difference between corresponding pixels in the generated and real images. This game-like competition might lead to the generation of realistic-looking images, which fool the discriminator but don't capture the true characteristics and details of the target image. Instead of comparing these loss functions, it is more useful to incorporate image similarity metrics and visual inspection between the ground truth and the predicted image to ensure the performance of the generated pix2pix model. Most used image similarity metrics are structure similarity (SSIM) (Wang et al., 2004), learned perceptual image patch similarity (lpips) (Zhang et al., 2018), root squared error (RSE), normalized root mean squared error (NRMSE) and peak signal-to-noise ratio (PSNR) (Laine et al., 2021a).

SSIM metric is a commonly used normalized measure for assessing whether two images share the same structures. The similarity of the two images is indicated by how close the SSIM value is to 1. Additionally, the SSIM maps can be used to visually represent the SSIM value. These maps are generated by calculating the SSIM metric for each pixel while considering the surrounding structural similarity in the pixel's neighborhood. RSE map visually represents the square root of the squared difference between the normalized predicted and ground truth images. A smaller RSE value indicates a higher similarity. The NRMSE calculates the average difference between all pixels in the compared images. Higher similarity is shown when the scores are lower. PSNR is a metric that quantifies the difference between the ground truth and prediction in decibels. This is done using the peak pixel values of the prediction and the Mean Squared Error (MSE) between the images. Better agreement is indicated by a higher PSNR score. Lastly, a score that is more aligned with human perception is provided by the lpips (Zhang et al., 2018), a metric used to measure the perceptual similarity between two images. It captures the differences in visual perception as perceived by humans and is trained using natural images.

There are several tools for evaluating DL models. For example, the ZerocostDL4Mic platform (von Chamier et al., 2021) integrates useful quality control metrics as a part of the training workflow. Additionally, the napari plugin called the segmentation game (Haase and Schätz, 2022) allows quantitative comparison of segmentation results against ground truth annotations. It is possible to fine-tune parameters and even compare different segmentation algorithms to find the best possible parameters for a particular dataset.

2.4 Feature extraction

Feature extraction refers to a process where image data is transformed into numerical non-image data. Image features can be divided into three main categories of information: intensity-based features, shape or spatial features, and spatio-temporal features (Nketia et al., 2017). Intensity-based features can be derived directly from the image's pixel values, such as the mean or total intensity of the object or image or object texture. Shape features are extracted from a segmented object and can describe, for example, the object's area, volume, roundness, or information about the object's coordinate location. Spatio-temporal features consider the time dimension and can describe, for example, the overall features, such as the mean speed of the object or local features, such as acceleration. Additionally, when studying images that have two or more channels, features such as colocalization or signal overlap can be measured.

Fiji (Schindelin et al., 2012) is probably the most widely used tool for feature extraction as it comes with a built-in feature extraction function coupled with the possibility to manage and store ROIs. Fiji extends the feature extraction to 3D data by using the 3D ImageJ Suite plugin (Ollion et al., 2013) and provides specialized tools, such as the OrgaMapper (Schmied et al., 2023), for measuring spatial relationships between objects. Additionally, tracking plugins such as TrackMate (Ershov et al., 2022; Tinevez et al., 2017) allows the extraction of spatiotemporal information, which can be directly linked to object intensity or shape features for elegant live cell imaging data analysis.

3 Image processing and analysis for live cell imaging

Live cell imaging provides a means for the visualization recording of central biological processes, such as cell migration and proliferation, which both have a key role in health and disease (Franz et al., 2002). Although live cell imaging methods are powerful, the generated image data can suffer from issues, such as sample drifting or bleaching, making the downstream analysis difficult. Therefore, the images usually need to be processed and analyzed using computational image analysis methods to extract quantitative information from the produced image data. In this chapter, I introduce common image processing and analysis methods, applications, and tools for live cell imaging analysis that are relevant to the work we conduct in our laboratory.

3.1 Image registration and drift correction

3.1.1 Image registration problem

Image registration is the process of transforming two images into one coordinate system. Registration is essential when combining image data from different imaging modalities, allowing their comparison and analysis. Matching the observed areas between modalities has long been a major issue, especially in the medical imaging field, but also becomes relevant in life science research, for example, in correlative light and electron microscopy, a combination of fluorescence microscopy and high-resolution electron microscopy (de Boer et al., 2015). Channel misalignment can also be considered a registration problem. Small shifts in the channel alignment can cause problems, especially in multichannel super-resolution imaging, such as STORM (Øvrebø et al., 2023).

Sample drifting is introduced to live image data from an unstable microscope environment or the sample movements. Drift correction between frames can essentially be considered an image registration problem, but instead of matching images from different modalities, matching now needs to be performed between images at different time points. Drift correction allows alignment and stabilization of sequential images for reliable quantitative downstream analysis. Image drifting, which is impossible to remove during acquisition, must be removed post-processing.

Most of the post-acquisition image registration methods consist of four steps. First, extrinsic or intrinsic features of the image are extracted. Second, detected features in the moving and reference images are matched. Third, a suitable image transformation is defined, and the transformation is applied to the moving image (Zitová and Flusser, 2003). In the following chapters, I give an overview of these four steps in the context of drift correction in live cell imaging.

3.1.2 Image feature detection

For the estimation of drift between frames, two types of information can be extracted from images: extrinsic or intrinsic information. For example, one can place static fiducial markers like fluorescent beads or micropatterns on the sample before imaging. These markers can then function as reference points during the frame alignment process. They can be manually or computationally identified (i.e., segmentation), and the drift between frames can be calculated by measuring the displacement of these markers (Lee et al., 2012).

In live cell videos, frames are usually highly alike; therefore, it is possible to use intrinsic image features to find similarities between frames. Intrinsic image information refers to characteristics inherent to the images themselves, which can be extracted directly without the need for static markers. Intensity distributions or computationally extracted local features, such as corners or blobs, can be considered intrinsic features (Gonzalez and Woods, 2018). The challenges in utilizing intrinsic image information for drift correction live cell imaging arise from the movement of cells during the imaging process. It is useful to use a reference channel labeled with a non-moving object and use the previous frame as a reference rather than the first frame of the video for frame matching.

Image features can also be extracted in the frequency domain. In this technique, images are usually transformed into the frequency domain by using the Fourier transformation, and registration involves manipulating and matching the frequency components to achieve alignment. This approach is beneficial when dealing with images affected by distortions, noise, or variations in intensity or contrast (Tong et al., 2019).

3.1.3 Feature matching

After detecting relevant features in different images, the next step in registration is feature matching. Feature matching involves finding the correspondence between the identified features in the images being registered (Gonzalez and Woods, 2018). Matching ensures that the same biological structures or landmarks are correctly identified in each image, forming the basis for subsequent alignment.

Scale-invariant feature transform (SIFT) combines the feature detection feature matching in one step. It matches frames by extracting local features that are invariant to image scale and rotation. A corresponding feature of the moving image is recognized by individually comparing each feature between the reference and the moving image based on Euclidean distances and finding its nearest neighbor. SIFT is resistant to noise, which makes it usable for live cell imaging (Lowe, 2004). Like SIFT, speeded-up robust features (SURF) is designed for fast detection of key points independent of scale, rotation, and changes in illumination of the image. It is computationally more efficient than SIFT (Bay et al., 2008).

Compared to feature-based methods, correlation-based methods consider the overall structural similarity of the image and, therefore, are not limited to extracting features in local areas (González, 2011). Cross-correlation is a frequently used drift correction technique for live cell image data (Wester et al., 2021; Mlodzianoski et al., 2011; Wang et al., 2014). It measures the similarity between two images by computing the cross-correlation of their intensity values. Cross-correlation is usually applied in the frequency domain and has been used for image registration of 2D (Laine et al., 2019) and 3D image data (McGorty et al., 2013). Phase correlation is a subset of cross-correlation techniques commonly used for image registration (Parslow et al., 2014; Preibisch et al., 2009). It isolates the phase information from the frequency representation of the image and analyzes the phase difference between them to estimate the required transformation. Both cross-correlation and phase-correlation methods are best suited for linear transformations, where the entire image undergoes the same transformation.

Block matching in image registration aligns two images by dividing them into smaller blocks or patches and finding corresponding blocks between the images by searching for the best matching block in the moving image within a neighborhood of the same size in the reference image. This approach is particularly useful when dealing with rigid transformations and local deformations and has been widely used for medical image registrations (Commowick et al., 2012), although applications for live cell image data also exist (Fernandez and Moisy, 2021; Fernandez et al., 2022).

3.1.4 Transformation models

Image registration operations come in different types; though they all perform a similar task, they move one image in relation to the reference image. Different transformation models are used to complete this task based on the differences between images. Linear transformation models include rotation, scaling, translation, and affine registration transformations (Figure 13). Linear transformations consider the image as a whole and cannot correct local geometric differences between images. Elastic or non-rigid transformation models can be used to correct local geometric differences. These transformations can locally warp the moving image to align it with the reference image (Gonzalez and Woods, 2018).

Linear shift is a frequently encountered form of shift where the moving image is shifted solely in the x-, y-, and z-directions. This type of shift can be corrected using translational transformation models. When a rotational shift is introduced into the moving image, a rigid body transformation model is used for the correction. In the case of the moving image having a different scale, the scaling transformation model is used, and when skewing is incorporated into the mix, an affine transformation model can be used (Figure 13).

In live cell imaging, the scale and the geometry of the images usually remain constant, but lateral and axial drifting pose significant challenges, particularly in long-term multi-position imaging experiments. In some cases, rotation can also be observed, for example, when imaging living organoids (Gsell et al., 2023). Linear transformation models are usually sufficient for correcting drift in live cell imaging.

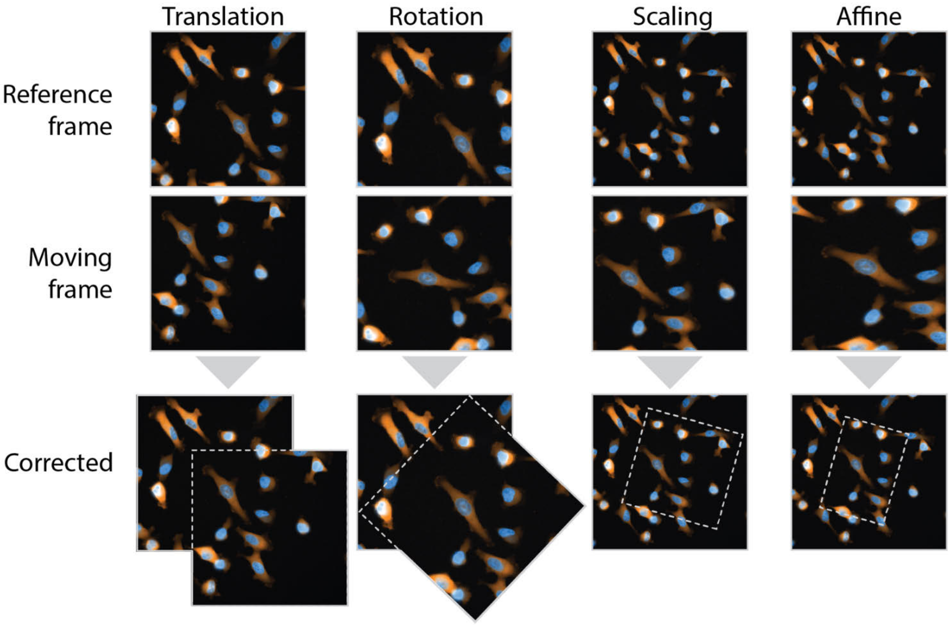


Figure 13: Linear transformation models. Linear shifts between images can be divided into four different types. In the first column, a linear shift in the xy-direction is corrected using a translational transformation model. In the second column, a rigid body transformation model is used to correct image rotation. In the third column images of different sizes are transformed using a scaling transformation model. In the last column, an image with translation, rotation, scaling, and skewing is corrected using an affine transformation model.

In the case of rotation, scaling, affine, or non-rigid transformations, the moving image needs to undergo image interpolation to match the coordinate system of the reference image. The selection of the suitable interpolation method relies on the requirement of interpolation precision. In most cases, the nearest-neighbor or bilinear interpolation provides sufficient results; however, some applications require more precise methods (Zitová and Flusser, 2003).

3.1.5 Deep learning methods for image registration

As many modern image processing methods include the use of DL, image registration is no different. DL registration methods include both supervised and self-supervised approaches. Supervised approaches, for example, FlowNet (Dosovitskiy et al., 2015), usually rely on the use of convolutional neural networks (CNNs) that learn how to transform the moving image to the reference image by being first trained on pairs with known transformation. This way, when new images are introduced to the network, they can be similarly aligned. Although supervised methods can achieve high accuracy in image registration, they rely on available training data which might be challenging or expensive to acquire. Self-supervised methods, for example, VoxelMorph (Balakrishnan et al., 2019), on the other hand, use inherent information within the images and are therefore independent of training data.

DL methods for image registration can also be categorized into feature detection methods and end-to-end methods. Feature detection methods involve finding distinct features in images followed by matching and aligning these features (Kuppala et al., 2020), for example, by using flow estimation (Dosovitskiy et al., 2015). They are robust to changes in image appearance and illumination but only in the presence of distinctive features uniformly distributed in all images. In end-to-end methods, the networks learn the optimal transformation directly from the images and they can usually handle complex and non-linear transformations. These methods are computationally demanding and may require large amounts of training data (Zhao et al., 2020). One example of an end-to-end DL method is Matchnet, which uses image similarity metrics to match two images (Han et al., 2015).

DL methods used for drift correction in live cell imaging usually aim to align consecutive frames, compensating for any translational or rotational shifts. Some specialized networks have been developed to specifically correct drifting in live cell imaging data. For example, *DenoiseReg* incorporates a denoising step to end-to-end unsupervised DL-enabled affine registration of live cell microscopy data (Celikay et al., 2022).

Applying DL-enabled registration to image data usually requires coding experience and thus remains unavailable for most life scientists. The exception to this is the DRMIME (Nan et al., 2020), an unsupervised end-to-end network that uses mutual information as a metric, which was made available through the ZeroCostDL4Mic platform (von Chamier et al., 2021). Although, in some cases, DL-based registration methods outperform traditional registration algorithms, they are still slow in speed (Fischer et al., 2014), difficult to use, and sometimes require large amounts of training data, causing life scientists often to look for more traditional image registration methods.

3.2 Tracking

Tracking is a part of live imaging experiments, allowing us to understand dynamic cellular movements and processes, providing crucial insights into various aspects of cell biology, physiology, and pathology. As cells are the fundamental units of life and are involved in numerous physiological and pathological phenomena, tracking their behavior over time through imaging techniques has become indispensable in modern biological research.

Spot tracking focuses on monitoring movements of a specific point or feature, such as a filopodia tips (Miihkinen et al., 2021), while object tracking involves the monitoring of entire objects, such as cell nuclei (Peuhu et al., 2022). In this chapter, I introduce the basic concepts of object tracking using cells as an example, how the acquired data can be interpreted, and how tracking performance can be assessed.

3.2.1 Object tracking principle

To extract biological insights from time-lapse microscopy videos, it is essential to identify objects, such as cells, and subsequently link their motions to create cell tracks. Once the tracks have been created, it becomes possible to monitor cell movements over time, which offers valuable details about cell velocities, trajectories, and the identification of cell-lineage relationships (Figure 14).

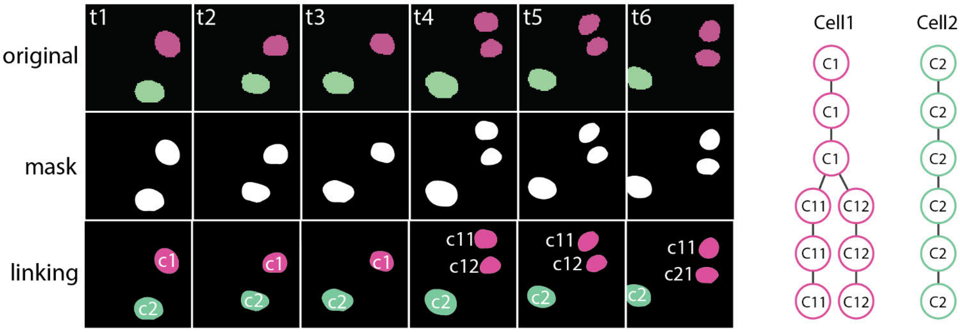


Figure 14: Basic concept of segmentation and tracking. On the top row are six sequential frames from fluorescently labeled nuclei. In the middle row are binary segmentation masks of these nuclei. These masks help to locate nuclei from the background. A linking algorithm has been used on the bottom row to find correspondences between the masks. From the location of the nuclei in consecutive frames, it is possible to determine the trajectory of each nucleus and its velocity and cell division events (for example, cell C1 divides into two daughter cells, C11 and C12). On the right is a graph-based representation of the generated cell lineages found by the tracking algorithm. These graphs can be used to learn, for example, about cell division events. Redrawn based on (Ulman et al., 2017).

3.2.2 Cell proliferation

During cell proliferation, a cell grows and divides into two daughter cells, eventually leading to exponential growth in cell number. This process is essential in development and in maintaining cellular homeostasis. Cell proliferation is controlled by the cell-cycle control system, and when the system malfunctions, excessive cell division can result in cancer (Alberts et al., 2022).

Live cell imaging and image analysis play an important role in cell proliferation studies. In the most basic experiment, the cells are imaged using brightfield microscopy at different time points, and the total area covered by the cells between the time points is quantified, or the movement of the entire cell population is tracked. In many cases, measuring the total cell area in every time frame is sufficient to provide information about cell proliferation. The cell number can also be quantified by labeling the nuclei followed by manual or automated counting frame by frame using suitable image analysis software.

Proliferating cells can also be identified using cell stains that bind to specific cell cycle markers, for example, the fluorescence ubiquitin cell cycle indicator (FUCCI) sensors (Zielke and Edgar, 2015). The FUCCI sensors highlight cells in the G1 phase with red fluorescence, while cells in S/G2/M are highlighted in green. During a short period at the G1/S transition, both probes are present. Therefore, the cells appear yellow (Sakaue-Sawano et al., 2008). The acquired images can then be quantified by measuring the amount of each signal over time or by using specialized FUCCI image analysis pipelines (Ghannoum et al., 2021; Taïeb et al., 2022).

3.2.3 Object detection for tracking

In live cell image data, objects, for example, cells, can be manually detected, involving a human observer manually marking the positions of cells in each frame. Manual detection is time-consuming, so automated detection approaches are commonly used. Although manual annotation is slow, it plays an essential role in the validation of cell tracking performance (Maška et al., 2023). There are many available methods for automated cell detection that can be used to accelerate and standardize the cell detection process.

For optical imaging, especially when working with fluorescent images, objects of interest are usually identified by using detectors. For roundish fluorescent objects of equal size, traditional filters such as Laplacian of Gaussian (LoG) or Difference of Gaussians (DoG) detectors can be used (Tinevez et al., 2017). These detectors use the frequency space of the image to identify local maxima and then further compute coordinates for the detected spots in the original image but without shape information. LoG and DoG detectors perform well when detecting blob-like objects of similar sizes, such as nuclei (Xu et al., 2017) or Myosin-X labeled filopodia tips (Miihkinen et al., 2021), but perform less efficiently when objects have irregular shapes, like whole cells.

To learn about cell morphology or changes in their intensity over time, cells need to be identified from their backgrounds, or segmented in other words. Cells can be tracked by first performing segmentation in all frames, followed by linking these in time. Segmentation is typically done by filtering the image followed by thresholding. The created mask can be improved by applying morphological operations such as closing holes or watersheding and labeling using connected component labeling (Zimmer et al., 2006, Figure 6). However, when segmenting non-fluorescent cells, simple intensity-based thresholding commonly fails and raises the importance of more sophisticated cell segmentation methods such as ML algorithms like random forest classifier (Breiman, 2001) or DL methods, such as Cellpose (Stringer et al., 2021). Segmentation methods and algorithms are discussed in more detail in chapters 2.2.3 and 2.3.5.

In a situation where cells express a high fluorescent signal compared to the background and retain consistent shape, it is possible to perform cell tracking without segmentation using methods such as correlation-based template matching (Perez-Careta et al., 2008). In this method, first, the object's neighborhood is defined, and the pixel whose neighborhood maximizes the correlation with the object's neighborhood in the previous frame or a manually defined reference frame is then considered the corresponding object in the following frame.

3.2.4 Object linking

Regardless of the detection methods, the objects need to be linked over time to track the detected objects to form trajectories. Among the more basic object linking approaches is the nearest neighbor algorithm (Crocker, 1996; Pietzsch et al., 2012), which assigns an object to the one closest object in the same trajectory within a user-defined distance. The nearest neighbor tracker performs quickly as it depends on a K-dimensional-tree technique, ensuring efficient tracking of many particles. However, the nearest neighbor tracker cannot perform frame-to-frame linking. Therefore, any cell division or a missing frame will be considered as the beginning of a new track.

As cell division is an essential part of understanding cell behavior, more sophisticated algorithms are needed. For example, the linear assignment problem (LAP) algorithm (Jaqaman et al., 2008), available in TrackMate, addresses this problem by a two-step track-building process. First, detected cells or spots are linked between frames and track segments are built. These track segments are then investigated in a second step to identify missing detections, splitting, and merging events. This tracker overlooks the Brownian motion, and tracks with similar features can be favored.

Another example of a linking algorithm featured in TrackMate is the Kalman tracker (Figure 15), which is best suited when tracking objects that move at a constant velocity. This tracker first links the two first frames using the LAP tracker and uses the Kalman filter (Kalman, 1960) to predict the most probable position of a particle in the following frame, assuming that it moves with a constant velocity. In case of a missing spot in the defined radius, the Kalman tracker will predict the following frame and re-iterate the search. In addition to bridging gaps, the Kalman tracker has also been extended to detect track splitting and merging, which is essential for studying cell lineage.

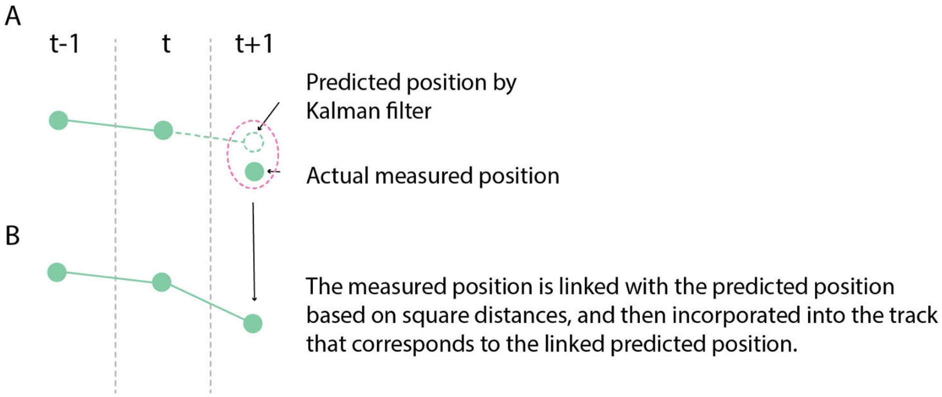


Figure 15: The Kalman tracker. The Kalman tracker relies on the Kalman filter to A) first predict the most probable position of a particle moving at a constant velocity and B) then the predicted positions are linked against the actual spot positions in the frame. Modified from (Tinevez, 2016).

Tracks can also be generated by forming links between spots that overlap between consecutive frames using IoU. In the TrackMate v7, (Ershov et al., 2022) this tracker is implemented so that the area of an object can be up or downscaled before the IoU calculation. If this scaled area overlaps with the scaled object in the consecutive frame by user-defined IoU, a link is created. As a bonus, this tracker can be used to generate 3D labels from image stacks by first flipping the z- and time dimensions, in other words tracking the slices as time. To generate the 3D image, the objects on the tracks are exported and time flipped back to z-slices (Ershov et al., 2022).

The cell contour detection method brings another solution to study cell morphological dynamics. In this method, the cells are first identified by a cell contour evolution method, where the cells are segmented in the first frame, and these contours are then evolved into consecutive frames. Tracking by contour evolution performs the segmentation and tracking steps simultaneously. This method assumes spatiotemporal overlap between objects between frames and is not suitable for fast-moving objects (Chaumont et al., 2012; Dufour et al., 2011).

3.2.5 Available tracking tools

There are numerous cell-tracking tools available that use classical approaches or DL-enabled cell tracking. Here, I introduce a few commonly used tools covering different cell tracking and analysis approaches.

MaMuT (Manual Tracking for Multi-object) (Wolff et al., 2018) is a plugin for Fiji, designed for manual annotation and tracking of objects in multi-dimensional time-lapse images. It provides a user-friendly interface to facilitate the tracking of multiple objects, such as cells, in microscopy data.

TrackMate (Tinevez et al., 2017) is a popular tracking tool in Fiji, which we have used in this thesis. TrackMate allows LoG or DoG detector-based object detection, multiple tracking algorithms, and analysis tools for quantification of cell migration. I contributed to the development of TrackMate (v7) (Ershov et al., 2022), which incorporates various segmentation methods for cell detection and new tracking and analysis features. I further discuss these developments in the results and discussion sections.

In live cell imaging, the resulting image data can be several terabytes in size. Dealing with these large datasets and creating annotations for them is difficult. Mastodon (https://github.com/mastodon-sc/mastodon, 06.03.2024) is a tracking and track-editing framework for large and multi-view images, allowing automated cell or particle tracking, manual curation and correction of tracking results, and manual and semi-automatic tracking. It allows interactive browsing of the data by storing it in small chunks corresponding to a neighborhood. As the Mastodon viewer shows a slice through the image, the required chunks are loaded on demand and cached. The image is also stored as a multi-scale pyramid to speed up interactive zooming.

A recent Python-based cell-tracking tool, Ultrack (Bragantini et al., 2023), simultaneously computes cell tracks and segments using a hierarchy of segmentation hypotheses and selects separate segments by maximizing the overlap between adjacent frames. This tool can track terabyte-scale datasets with high precision, allowing tracking of thousands of objects in 3D image data.

Cell tracking can be enhanced using DL. Most tracking tasks handle segmentation and linking steps as consecutive tasks, and few incorporate these two steps in one workflow. One such tool is the *btrack* Python package (Ulicna et al., 2021). *btrack* approach consists of a U-NET-based cell detection step, where individual cells are segmented from fluorescent live-cell images, which are then cropped to patches. These patches are extracted from the corresponding cell positions in both transmission and fluorescence channel images and used as inputs to a CNN-based cell state classifier to label the cell cycle phase of each segmented object. The segmented and classified objects are then tracked using a Bayesian tracking algorithm to reconstruct individual cell trajectories. With all the acquired information, *btrack* finally assembles parent-children relationships into lineage trees.

Similar to *btrack*, the DeepSea network (Szabó et al., 2023) uses a two-step process where it first segments cells and then applies another DL network that extracts convolutional information from two consecutive frames to localize and detect the same target cell or its daughter cells among the segmented cells in the second frame by generating a binary mask. DeepSea allows the following of multiple cellular phenotypes and cell division cycles across the video and generates lineage trees (Zargari et al., 2023).

To ease the generation of training data by enhancing human interaction during training new tools such as ELEPHANT (Efficient learning using sparse human annotations for nuclear tracking) (Sugawara et al., 2022) are developed. ELEPHANT is implemented as an extension of Mastodon and it implements two algorithms optimized for incremental DL using sparse annotations, one for detecting nuclei in 3D and a second for linking these nuclei across time points in 4D image datasets. Incremental learning allows DL models to be trained step-by-step, starting from sparse annotations that are incrementally enriched by human proofreading.

3.2.6 Evaluation of tracking performance

Although cell tracking algorithms have been available for a long time, they still usually consist of two consecutive tasks: cell detection and linking. The quality of cell detection and segmentation is greatly dependent on the quality of the image data produced. Low SNR, cell division, and long gaps between image frames greatly affect the tracking quality, bringing challenges to reliably tracking the cells. The diversity of available algorithms, life scientists often struggle to decide which algorithm to use in their experiment. An open-source community effort known as the Cell Tracking Challenge (CTC) (http://celltrackingchallenge.net, 11.03.2024) has emerged to address this issue. Its primary goal is to establish an unbiased assessment framework for cell segmentation and tracking algorithms. This initiative offers valuable guidance to biologists in selecting suitable algorithms for their specific datasets while also assisting developers in enhancing their tracking algorithms (Ulman et al., 2017; Maška et al., 2023).

The TrackMate-Helper tool (Ershov et al., 2022) was developed to help evaluate which tracking parameters (detectors and tracking algorithms), output tracking results closest to the ground truth. It provides a tool to first create the segmentation and tracking ground truths on a representative movie. Next, the TrackMate-Helper segments and tracks the same movie using different detection algorithms, linking algorithms, and parameter combinations. These are compared then with the ground truth, and 8 CTC-metrics (Table 1) are computed defining the optimal tracking parameters. The identified optimal settings can then be used on the whole dataset.

Table 1: TrackMate-helper metrics (Ulman et al., 2017).

Metric	Explanation	Scale
Segmentation accuracy measure (SEG)	Evaluates the average amount of overlap between the segmentation and the ground truth.	0-1 (1 is best)
The tracking accuracy measure (TRA)	Normalized weighted distance between the tracking solution selected by the user and the reference tracking ground truth.	0-1 (1 is best)
The detection accuracy (DET)	How accurately each given object has been identified.	0-1 (1 is best)
Complete tracks (CT)	Measures the fraction of ground truth cell tracks that a given method can reconstruct from the frame they appear into the frame they disappear.	0-1 (1 is best)
Track fractions (TF)	Averages the fractions of the longest continuously matching algorithm-generated track concerning the reference track.	0-1 (1 is best)
Cell cycle accuracy (CCA)	Measures how accurately an algorithm reconstructs the length of the cell cycles.	0-1 (1 is best)
Branching correctness (BC)	Measures how efficient a selected tracking method is at detecting division events.	0-1 (1 is best)
Execution time (TIM)	Time of algorithm execution	Seconds. Lower is better.

3.3 Cell tracking analysis

Phenotypic and behavioral information extracted from live cell imaging experiments is growing due to the easier availability of high throughput imaging systems and the development of automated cell tracking algorithms and tools. In many cases, some analysis tools are integrated with object detection or tracking software (Ershov et al., 2022; Hu et al., 2021; Al-Zaben et al., 2019; Aragaki et al., 2022; Ulicna et al., 2021), while newer tools are built to incorporate outputs of tracking software to complement the tracking software analysis (Wiggins et al., 2023; Wortel et al., 2021). Here I present a few tools I familiarized with during my thesis work, that can be used to understand how cells behave.

CellTrackerR (Wortel et al., 2021) is an R package published as a Shiny app called the MotilityLab (http://www.motilitylab.net/startpage.php, 08.03.2024). It can be used to manage tracking data, perform quality control, extract and visualize migration statistics and clustering tracks, and simulate cell migration of both 2D and 3D cell tracks. Tracking data provided by TrackMate data can be exported as a MotilityLab spreadsheet, which can be directly uploaded to the MotilityLab app for further analysis and visualization. In our work, we use it, for example, for visualization of cell track direction and changes in signal intensities over time (Ershov et al., 2022).

To further complement the TrackMate (Ershov et al., 2022) analysis, a CellPhe (Wiggins et al., 2023) toolbox can be used. This toolbox can use the output of TrackMate or Phasefocus (Phasefocus Limited, Sheffield, UK) and recognize patterns to characterize changes in the cells' appearance and behavior over time. The common downside in high throughput analysis is the uncertainty of segmentation and tracking quality, and error might add to the analysis. CellPhe uses decision trees to detect segmentation and tracking errors so that they can be removed from the analysis. CellPhe also interpolates possible missing tracks in cases where the cell might leave the field of view to prevent abrupt data.

Furthermore, TrackMate data can be analyzed using a recent Python-based tool, CellTracksColab (Jacquemet, 2023), which operates in Google Colab with a user-friendly GUI for life scientists without coding experience. CellTracksColab is optimized to handle large datasets and allows track visualization, cell population analysis, statistical assessments, and exploring tracking data through dimensionality reduction techniques. CellTracksColab was originally developed to use TrackMate data as input but was later extended so that it can use any csv files as input as long as certain column requirements are filled.

CellPLasticityAnalysisTOol (cellPLATO) (Shannon et al., 2023) is a Python-based software that can measure and classify cell behavior. It uses cell morphology and motility parameter clustering to group cells into behavioral subtypes and generates a phenotypic fingerprint for each experimental condition. Finally, representative cells from each subtype can be graphically displayed. CellPLATO can use outputs from *btrack* (Ulicna et al., 2021), TrackMate (Ershov et al., 2022), or Usiigaci (Tsai et al., 2019) as input.

3.3.1 Lineage tracing

While cells migrate, they face different fates. Lineage tracing allows following these fates at the level of individual cells to further understand complex biological processes. Live cell imaging data, when combined with cell tracking, allows for the reconstruction of cell lineages, offering insights into the entire cell division history over several generations. In development, lineage tracing can provide information on how cells form distinct cell types found in various tissues and organs (Wolf et al., 2021). Cell lineages are usually visualized in the form of lineage trees, where the relationship of different generation cells can be understood (Figure 14) (Sandler et al., 2015).

3.4 Choosing a suitable image analysis tool

There is a high number of image analysis software available and choosing the best for your purpose is difficult. The choice of the bioimage analysis software is closely related to the purpose of a given research project (Cimini and Eliceiri, 2023), the institute's availability of tools, for example, in the case of commercial software, and the skillset of the researcher (Haase et al., 2022).

Commercial software such as Aivia (Leica Microsystems GmbH) and Imaris (Oxford Instruments) are commonly offered by imaging facilities, but open-source software with intuitive user interfaces such as ImageJ/Fiji (Schindelin et al., 2012), CellProfiler (Stirling et al., 2021), Icy (Chaumont et al., 2012) or TissUUmaps (Pielawski et al., 2023) are also popular. Additionally, the Python image processing community has grown rapidly due to the availability of scientific Python software packages (van der Walt et al., 2014; Virtanen et al., 2020; Harris et al., 2020). To improve the usability of these packages, a napari project (Ahlers et al., 2023) has implemented a user-friendly graphical user interface (GUI) that incorporates scientific Python packages and includes an n-dimensional image viewer and plugin development tools. Napari has also implemented a napari-ImageJ, based on the PyImageJ project (Rueden et al., 2022), which provides an accessible solution to access the ImageJ ecosystem via napari (Selzer et al., 2023). Similarly, TissUUmaps, a software specialized for the analysis of spatial transcriptomics data, can be run through napari through the napari-tissuumaps plugin (https://github.com/TissUUmaps/napari-tissuumaps, 06.03.2024)

The usage of modern image analysis software and management of image analysis workflows require computational skills, which can be a struggle for an individual life scientist to master. Therefore, the developers of image analysis software must make significant efforts to develop user-friendliness and documentation of the software and build intuitive user interfaces to aid in the selection of suitable image analysis tools (Carpenter et al., 2012).

3.5 Image analysis workflows

Most of the time, standalone components cannot solve complex bioimage analysis problems on their own. For this reason, bioimage analysis workflows need to be assembled. A bioimage analysis workflow can be described as a sequence of computational tools arranged in a particular order to process and analyze bioimages and calculate parameters that are relevant to the biological system being investigated (Paul-Gilloteaux et al., 2021).

Most modern image analysis software, such as Fiji (Schindelin et al., 2012), Icy (Chaumont et al., 2012), and CellProfiler (Stirling et al., 2021) allow the combining of image processing, analysis, and even data visualization tools into a customizable image analysis pipeline. Fiji uses its recordable macro language for easy workflow construction, while a recent Fiji plugin called Modular Image Analysis (MIA) (Cross et al.) aims to provide a code-free graphical environment in which complex automated analysis workflows can be constructed and distributed. Worth mentioning is also the CLIJ (OpenCL - ImageJ bridge) Fiji plugin (Haase et al., 2020) that allows users with no coding skills to build GPU-accelerated workflows to speed up their image processing and segmentation workflows. These workflows can then be exported as Fiji macro language or as human-readable language.

Python programming has increased usage for bioimage analysis due to its numerous available scientific libraries for image processing, ML, DL, analysis, and data visualization. Nowadays, Python scripts are often bundled in web-based interactive computational environments, such as Jupyter or Google Colaboratory notebooks, providing user-friendly installation of dependencies and bioimage analysis in different operating systems (Levet et al., 2021). Similarly to CLIJ above, napari provides an assistant tool (Haase et al., 2023), which allows easy construction of image analysis workflows and their direct export to a Jupyter notebook. Omega (Royer, 2023) is a new and exciting tool to help life scientists build image analysis workflows. It is an interactive conversational agent based on large language models. Upon user request, Omega can perform image processing tasks, analyze images, or even provide step-by-step advice and instructions on how to build an image analysis workflow. Omega can create on-demand user interface widgets to create automated image analysis workflows. Omega is integrated as a plugin to napari (Ahlers et al., 2023), allowing users to instruct complex tasks without requiring prior programming knowledge.

Despite the exciting advancements in bioimage analysis and workflows, researchers still face challenges when it comes to collaboration, data sharing, and the implementation of diverse analysis approaches (Soltwedel and Haase, 2023). Traditionally, image analysis workflows have been run on local computers, where software installation and hardware requirements can become a headache, especially when working with large datasets. Local workflows are also difficult in collaborative projects and might not be reproducible due to varying software versions and platform dependencies (Ouyang et al., 2023).

Therefore, the availability of web-based image processing and analysis tools, such as DeepCell Kiosk (Bannon et al., 2021) and ZeroCostDL4Mic (von Chamier et al., 2021), has been increasing and provides improved user experience, accessibility, and efficiency of biological research, allowing more time spent on biomedical discoveries (Ouyang et al., 2023). The software can be packed into a container to further enable the reproducibility and portability of bioimage analysis workflows. Containerization involves embedding a piece of software and all its dependencies and specific configurations in a container image file. Using this container image, it is possible to run the software consistently across different computing environments, including Google Colaboratory, one's own computational resources, or possibly available high-performance computing systems (Hidalgo-Cenalmor et al., 2023; Paul-Gilloteaux et al., 2021).

Aims of the Study

Live cell imaging plays a crucial role when observing biological processes, such as normal development, wound healing, and cancer metastasis. Live cell imaging data is prone to several issues originating from unstable instruments, and the cells' sensitivity to light, thus complicating the downstream analysis. Bioimage analysis tools have become an integral part of the improvement and analysis of live cell imaging data, providing tools for extracting meaningful information in life science research.

In this thesis, I aim to solve some of the issues in live cell imaging through the development of image processing and analysis tools specifically intended to improve and analyze live cell imaging data. I also aim to highlight the usability and modularity of these tools by applying them to two case examples where I incorporate the developed tools in image analysis pipelines to extract quantitative information from live cell imaging experiments.

The specific aims of my thesis were as follows:

Aim 1: To develop user-friendly image analysis tools for live cell imaging. To achieve this aim, I contributed to the development of:

- Fast4DReg: a tool for fast and reliable video registration tool to correct drifting in 4D image data **(work I)**
- TrackMate v7: a sophisticated cell tracking tool that can quantify morphological features during cell migration **(work II)**
- DL4MicEverywhere: a tool that provides flexible training and deployment of DL models across diverse computational environments **(work III)**

Aim 2: To incorporate image analysis tools in life science research. To achieve this aim, I contributed to creating image analysis pipelines to analyze live-cell imaging data using the tools developed in Aim 1. Using these pipelines, I aimed to:

- study how circulating pancreatic cancer cells adhere to endothelial cells during cancer metastasis **(unpublished work 1)**.
- investigate how specific mutations affect cancer cell behavior and drug resistance **(unpublished work 2)**.

Experimental Procedures

1 Summary of dataset, tools, and methods

Experimental procedures used in this thesis publications I-III are summarized in Table 2. Tools developed in this thesis have been applied to two ongoing unpublished works, unpublished 1 and 2, abbreviated u1 and u2, respectively. The original publications and the following chapter have detailed descriptions of my contributions to methods.

Table 2: An overview of the experimental procedures used in this thesis.

Experiment name	Study
Algorithm development	I, II*, III*
Dataset generation	I, II, III
Deep learning enabled segmentation	II, III, u1, u2
Deep learning enabled virtual staining	III, u1, u2
Documentation	I, II, III
Live cell image data processing and analysis	I, II, u1, u2
Live cell imaging	I, II, III, u1*, u2*
Microfluidics to study extravasation	u1*
Object tracking	II, u1, u2
Quality assessment	I, II, u1, u2

*experiment performed by collaborators

Image analysis software and tools used in this thesis. Several different image analysis software and tools were created and used in this thesis. These are summarized in Table 3.

Table 3: An overview of the image analysis software and tools used in this thesis.

Image analysis software and tools	Reference	Study
DL4MicEverywhere	(Hidalgo-Cenalmor et al., 2023)	III
Fast4DReg	(Pylvänäinen et al., 2023c)	I, u1
Fiji	(Schindelin et al., 2012)	I, II, u1, u2
LOCI	https://imagej.net/orgs/loci	II, u1, u2
StackReg	(Thevenaz et al., 1998)	u2
TrackMate	(Ershov et al., 2022; Tinevez et al., 2017)	II, u1, u2
ZeroCostDL4Mic	(von Chamier et al., 2021)	II, u1, u2
Drift correction quality control notebook	(Pylvänäinen et al., 2022)	I
Drift generator	(Pylvänäinen et al., 2022)	I

Datasets used in this thesis. In this thesis, we used several openly available datasets as well as generated new datasets to fit our purposes. These datasets are summarized in Table 4. Datasets generated for this thesis are described in more detail later.

Table 4: An overview of datasets used in this thesis. u1 = unpublished project 1, u2 = unpublished project 2.

ID	Dataset name	Reference of dataset	Study
D1	Datasets with synthetic drift*	Zenodo: (Pylvänäinen et al., 2022)	I
D2	DeepSTORM dataset	(Leterrier and Laine, 2020)	III
D3	ERK dataset (BF)*	Zenodo: (Pylvänäinen and Jacquemet, 2023)	III
D4	ERK dataset (fluorescent)*	Zenodo: (Tinevez and Pylvänäinen, 2021) Zenodo: (Pylvänäinen et al., 2023a)3/27/2024 2:33:00 PM	II
D5	Focal adhesion dataset	Zenodo: (Jacquemet et al., 2022a)	II
D6	FUCCI dataset	unpublished	u2
D7	Glioblastoma dataset	http://celltrackingchallenge.net/ Zenodo: (Jacquemet et al., 2022b)	II
D8	LIVECell dataset	(Edlund et al., 2021)	III
D9	MCF10 DCIS.com spheroid dataset	Zenodo: (Tinevez et al., 2021b)	II
D10	Microfluidics dataset	unpublished	u1
D11	Mouse hematopoietic stem cells	http://celltrackingchallenge.net/	II
D12	Neisseria meningitidis dataset	Zenodo: (Le Blanc et al., 2021)	II
D13	Noisy synthetic drift datasets *	Zenodo: (Pylvänäinen et al., 2022)	I
D14	T-cell dataset	Zenodo: (Tinevez et al., 2021a)	II
D15	The calibration slide dataset	Zenodo: (Pylvänäinen et al., 2022)	I
D16	The DCIS.com filopodia dataset	Zenodo: (Pylvänäinen et al., 2022)	I
D17	The filopodia dataset	Zenodo: (Pylvänäinen et al., 2022)	I
D18	The HUVEC monolayer dataset	unpublished	u1
D19	The mouse lung dataset	Zenodo: (Pylvänäinen et al., 2022)	I

*datasets generated by me

Deep learning models used in this thesis. Both pre-trained and custom-made deep learning (DL) models were used in this thesis (Table 5). All custom-made DL models used in this work were trained using the ZerocostDL4Mic platform (von Chamier et al., 2021).

Table 5: An overview of DL models used in this thesis. u1 = unpublished work 1, u2 = unpublished work 2

ID	DL models	Training dataset	Reference	Study
M1	Cellpose Cyto model	-	(Stringer et al., 2021)	II
M2	Cellpose Cyto2 model	-	(Stringer et al., 2021)	II, u1
M3	Pix2Pix custom model 1 (BF -> nuclei) *	D10	unpublished	u1
M4	Pix2pix custom model 2 (BF ->PECAM) *	D10	unpublished	u1
M5	StarDist BF cancer cells	D10	unpublished	u1
M6	StarDist custom model (U2OS) *	D2	(Ershov et al., 2022)	II
M7	StarDist fluorescent endothelial nuclei *	D10	unpublished	u1
M8	StarDist MCF10DCIS.com	D9	(von Chamier et al., 2021)	II
M9	StarDist T-cell model	D14	(Fazeli et al., 2020)	II
M10	StarDist Versatile nuclei	-	(Schmidt and Weigert, 2022)	II, III, u1, u2
M11	Pix2pix custom model 3 (BF -> nuclei) *	D3	(Pylvänäinen et al., 2023b)	III
M12	StarDist custom model (FUCCI) *	D6	unpublished	u2
M13	Pix2pix custom model 4 (FUCCI) *	D6	unpublished	u2
M14	Stardist custom model (FUCCI – round cells) *	D6	unpublished	u2
M15	DeepSTORM model	D2	(von Chamier et al., 2021)	III
M16	Cellpose for LIVECell	D8		III

*models generated by me

In this thesis, we also used machine learning algorithms trained using ilastik (Berg et al., 2019) or Trainable Weka Segmentation (Arganda-Carreras et al., 2017) in Fiji (Schindelin et al., 2012).

Table 6: An overview of ML models used in this thesis.

ID	Machine learning models	Training tool and dataset	Reference	Study
M17	Neisseria meningitidis model*	Ilastik, D12	(Ershov et al., 2022)	II
M18	Focal adhesions model*	Weka (Fiji), D5	(Ershov et al., 2022)	II

*models generated by collaborators

2 Benchmarking the performance of Fast4DReg (I)

The inspiration to develop a new tool for fast 3D drift correction of live cell videos arose from the need of the lab to analyze live cell imaging data, which suffered from drifting and made downstream analysis difficult. Our lab had previously used other tools, such as Correct 3D Drift (Parslow et al., 2014) or Fijiyama (Fernandez and Moisy, 2021), which did not provide sufficient drift correction for our data (Study I: Fig. 4A-C) and were slow in performance (Study I: Fig. 4D). For 2D time-lapse data, we successfully used the drift correction tool from the NanoJ toolbox (Laine et al., 2019). We wanted to extend this algorithm to 1) allow drift correction in 3D time-lapse data, 2) allow alignment of 3D images with misaligned channels, 3) allow application of estimated drift parameters to another dataset, and 4) allow batch processing. All these needed to be packaged in a user-friendly pipeline and operate fast. Here, I focus on how we benchmarked the drift correction performance and speed against other similar tools.

2.1 Development of the Fast4DReg algorithm

The development of the Fast4DReg algorithm was inspired by the NanoJ tools, which allowed drift correction of 2D images using cross-correlation (Laine et al., 2019). Briefly, as described before (Sun, 2002), the cross-correlation between two images is calculated by first performing a discrete Hartley transform on the reference and the moving image, followed by a multiplication of one of the transformed images by the complex conjugate of the other. The result of this multiplication is then inversely transformed back to real space, generating a cross-correlation map (CCM). A bicubic spline interpolation is then used to upscale the CCM and achieve subpixel precision. The upscaled CCM is normalized by calculating the Pearson's correlation coefficient between the two images shifted according to the minimum and maximum values of the upscaled CCM. Finally, the linear shift between the reference and the moving image is then calculated by taking the global maximum peak of the normalized up-scaled CCM (Laine et al., 2019). To allow 3D images as an input, we added a step that created z- or y-intensity projections to create a 2D representation of each frame and applied the above-described algorithm to those. In the case of aligning channel alignment, we first convert the channels into time frames. Fast4DReg is written using a combination of an ImageJ macro and Java and is distributed via an ImageJ update site.

To allow smooth operation for all life scientists, we designed Fast4DReg for Fiji (Schindelin et al., 2012) and added a user-friendly UI that allows the import of various image file formats using Bio-Formats (Linkert et al., 2010), and allows batch processing. In the case of large datasets and a computer with limited resources, the user can use a random-access memory (RAM) saving mode, which saves the output images in the same bit depth as the source image instead of an upscaled 32-bit image.

As the output, Fast4DReg creates a new folder with the experiment number (inserted in the UI) and the date of the experiment. This folder contains the corrected files, drift tables, drift plots, and a settings file, which can be applied to another dataset to correct similar drift.

2.2 Generation of benchmarking datasets

2.2.1 Datasets with synthetic drift (D1)

3D image of a single fixed pancreatic ductal adenocarcinoma cell (AsPC1, American Type Culture Collection, CRL-1682) expressing Lifeact–mScarletI migrating inside the vasculature of a zebrafish embryo grown in Roswell Park Memorial Institute medium (Thermo Fisher Scientific, 11875093) supplemented with 10% fetal bovine serum (FBS) (Biowest, S1860). The image was acquired using a 3i CSU-W1 spinning disk confocal microscope equipped with a 40× water immersion objective (NA 1.15) and a Hamamatsu sCMOS Orca Flash camera and controlled by the Slidebook 6 software (Intelligent Imaging Innovations, Inc.). The synthetic drift dataset was created by multiplying this image 25 times while adding a known amount of x-, y- and z-drift (Table 7) between each frame, by using the Drift generator script (Pylvänäinen et al., 2022). After the drift was simulated, the image background was made homogeneous via pixel intensity subtraction and by adding specified noise using Fiji ('add specified noise' function).

Table 7: Drift table indicating the known amount of drift added between frames. Values are in pixels.

Frame	**X-drift**	**Y-drift**	**Z-drift**
1	-0.1846	-0.842	0.7736
2	1.1534	0.69	-1.0418
3	0.3099	1.261	-0.5265
4	1.3837	1.961	-2.517
5	2.8785	1	-3.3751
6	3.221	-0.072	-4.0643
7	4.83	-2.012	-2.4789
8	6.2691	-4.293	-4.9687
9	7.1845	-7.21	-5.6736
10	8.8191	-11.375	-7.2127
11	12.1096	-13.948	-6.6842
12	13.6142	-18.579	-7.6793
13	15.5335	-25.244	-7.4236
14	18.8507	-32.933	-7.7894
15	21.5718	-37.042	-9.3144
16	25.3066	-45.681	-8.9694
17	28.4077	-52.989	-9.1761
18	32.8398	-60.376	-9.9103
19	34.7591	-71.216	-10.8287
20	40.0535	-79.337	-10.9413
21	42.8704	-90.376	-11.5039
22	47.4757	-100.918	-13.4116
23	51.8525	-112.391	-14.2919
24	56.8755	-124.903	-14.3824
25	62.7437	-137.609	-14.8706

2.2.2 Noisy synthetic drift datasets (D13)

To test how susceptible to noise Fast4DReg is, we generated a series of noisy images. This dataset consists of 12 images of Dataset with synthetic drift with 12 levels of added Gaussian noise with standard deviations of 0, 5000, 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000 and 60000. The noise was added using the “add noise function” in Fiji (Schindelin et al., 2012) and yielded images with SNR of 30.053, 5.586, 2.964, 2.111, 1.686, 1.478, 1.327, 1.227, 1.196, 1.127, 1.119 and 1.070, respectively. The SNR was calculated by dividing the mean cell signal by the mean background signal.

2.2.3 Other datasets

Four other datasets were used in this project: The HUVEC monolayer dataset (D18), the mouse lung dataset (D19), the calibration slide dataset (D15), the filopodia dataset (D17), and the DCIS.com filopodia dataset (D16), described in detail in work I. These datasets were raw files directly from a microscope, and no post-processing was done before drift correction experiments. These datasets were used to demonstrate 1) the speed of Fast4DReg, 2) how the drift table can be applied to another dataset, and 3) how misaligned channels can be re-aligned.

2.3 Benchmarking the Fast4DReg Performance

The xy- and z-drift in the synthetic dataset was corrected using Fast4DReg, Correct 3D Drift (Parslow et al., 2014), and Fijiyama (Fernandez and Moisy, 2021), using the parameters that provided the best possible drift correction performance (used settings are described in I, ST1). After drift correction, the images were first cropped to be the same size (352×275 pixels, 69 z-slices, 25 frames) using Fiji. To quantitatively assess the drift-correction performance of Fast4DReg against Correct 3D Drift, we first visually assessed the data using time projections of slice 51 (middle of the cell) and a standard deviation projection of the same slice to visualize the standard deviation of the pixel intensities through time. To quantify the correction performance, we measured the mean signal of the standard deviation image of each z-slice and created boxplots using PlotsOfData (Postma and Goedhart, 2019). To further quantify the correction performance, four image-similarity metrics between frames (the reference frame was the first frame) of a selected z-slice (z-slice 51) were used: Pearson's correlation coefficient (PCC), mean structural similarity index (mSSIM) (Wang et al., 2004), peak signal-to-noise ratio (PSNR), and normalized root mean squared error (NRMSE) (Table 8). These metrics were calculated using a custom-made Google Colaboratory notebook (modified from Laine et al., 2021). This notebook is available on Zenodo (https://zenodo.org/record/7514913, 06.03.2024).

Table 8: Image similarity metrics used to assess drift correction quality.

Metric	Explanation	Scale
PCC	Measures the linear correlation between two images.	0-1 (1 is best)
mSSIM	Evaluates the similarity of two images based on their contrast, luminance, and structural content	0-1 (1 is best)
PSNR	Compares the peak signal amplitudes of two images and is typically expressed in decibels.	Higher is better.
NRMSE	Measures the average difference between the pixel intensity in two images.	Lower is better.

The xy- and z-drift in all the noisy synthetic datasets images were corrected using Fast4DReg, using the parameters that provided the best possible drift correction performance (used settings are described in Study I, ST1). The generated drift tables were then used to correct the original large drift dataset (Fig. 3A, B). Corrected images were then cropped to be the same size (192×192 pixels, 69 z-slices, 25 frames) using Fiji. The drift-correction performance was then quantified by measuring image-similarity metrics between frames (Table 8) (the reference frame was the first frame) of a selected z-slice in the middle of the cell (z-slice 51). These metrics were calculated using a custom-made Google Colaboratory notebook (modified from Laine et al., 2021). This notebook is available on Zenodo (https://zenodo.org/record/7514913, 06.03.2024).

The HUVEC monolayer dataset (D18) was used to benchmark the speed of Fast4DReg against Correct 3D Drift and Fijiyama. Two computers were used to compare the execution times of all compared methods: computer 1 (operating system, Windows; processor, AMD Ryzen 7 5800X 8-Core; graphics card, GeForce GTX 3080; RAM, 32 GB; Fiji version 1.53q) and computer 2 [operating system, macOS; processor, M1 chip (8-core CPU, 8-core GPU); RAM: 16 GB; Fiji (version 1.53q). Times of three replicates using each tool and both computers, in the regular and RAM saving mode (only for Fast4DReg), were compared.

3 Analysis and visualization of ERK activity in migrating cells over time using the ERK dataset (II)

3.1 Cell lines and imaging

MDA-MB-231 and U2OS cells were engineered to stably express clover-ERK-KTR according to the manufacturer's protocol (Kudo et al., 2018; Regot et al., 2014). These cells were seeded on fibronectin-coated (1 µg /ml) Ibidi 8-well slides (Ibidi) 1 day before imaging. Four hours before imaging, the medium was supplemented with 250 nM sir-DNA (Cytoskeleton) and 25 mM HEPES (Sigma). Cells were then imaged live (37 °C, 5% CO2) using a Nikon Eclipse Ti2-E microscope (Nikon) equipped with an sCMOS Orca Flash4.0 camera (Hamamatsu) and controlled by the NIS-Elements software (Nikon, v 5.11.01). MDA-MD-231 cells were imaged using a 20× Nikon CFI Plan Apo Lambda objective (NA 0.75), either 1 frame per minute for 2 hours or 1 frame every 5 minutes for 17 hours. U2OS cells were imaged using a 10× Nikon CFI Plan-Fluor objective (NA 0.3) every 5 minutes for 3 hours. In these experiments, a camera binning of 2 × 2 was used.

3.2 Tracking of MDA-MB-231 cells

MDA-MB-231 cells were tracked using the "Versatile nuclei" StarDist model (M10) (Schmidt and Weigert, 2022) directly in TrackMate (Tinevez et al., 2017; Ershov et al., 2022), using the following settings: (score threshold = 0.41; overlap threshold = 0.5) and the LAP tracker (linking max distance = 40 µm; track segment splitting = 30 µm). Tracks were filtered in TrackMate so that only the tracks spanning the whole video were included for further analysis. The average ERK-reporter intensity was measured over time using TrackMate, normalized between 0 and 1 and visualized using heatmaps using the PlotTwist shiny app (Goedhart, 2020).

3.3 Tracking of U2OS cells

For tracking the U2OS cells, a custom StarDist model (M6) was first trained using the ZerocostDL4Mic platform (von Chamier et al., 2021). First, a ground truth dataset was created by manually annotating the nuclei of 24 images of U2OS cells expressing clover-ERK-KTR (dimensions: 2048 × 2048 px) using the LOCI plugin in Fiji (Schindelin et al., 2012). The ground truth data was augmented to improve the model performance by generating a dataset of 120 paired images by using the Augmentor ZeroCostDL4Mic notebook (von Chamier et al., 2021; Bloice et al., 2019) by randomly cropping it into 1024 x 1024, rotating, flipping and multiplying the number of the images by 5. Using the new 120 ground truth images (1024 x 1024 px), a custom StarDist model was trained using the StarDist 2D ZeroCostDL4Mic notebook (v1.12.2) with the following parameters: 200 epochs, patch size: 1024 × 1024 with a batch size of 2 and a mae loss function. Key Python packages used included TensorFlow (v1.15), Keras (v2.3.1), CSBdeep (v0.6.1), NumPy (v1.19.5), and CUDA (v11), a C/C++ library for GPU programming. The training was accelerated using a Tesla P100GPU. The performance of the generated model was assessed using the built-in quality assessment metrics of the notebook. The generated StarDist model showed high performance, resulting in an average F1-score of 0.918 when tested with 2 image pairs previously unseen by the network. The generated model was used directly in TrackMate using the custom StarDist detector with the following parameters: score threshold = 0.41, overlap threshold = 0.5, and the LAP tracker linking max distance = 20 µm; track segment gap closing = 25 µm, gap closing max frame gap = 10 frames. Tracks lasting less than 34 frames (2 hours and 40 minutes) were excluded. The average ERK-reporter intensity was measured over time using TrackMate, normalized between 0 and 1 and visualized using heatmaps using the PlotTwist shiny app (Goedhart, 2020).

4 Deep learning model and dataset to showcase DL4MicEverywhere (III)

To showcase how nuclei can be predicted from brightfield images, we trained a pix2pix model (M11) (Isola et al., 2017). To train this model, we prepared 1600 paired brightfield and nuclei images of MDA-MB-231 cells (1022x1024 px) extracted from live cell imaging data. The nuclei were labeled using sir-DNA (Cytoskeleton). Cells were imaged live (37 °C, 5% CO2) using a Nikon Eclipse Ti2-E microscope (Nikon) equipped with an sCMOS Orca Flash4.0 camera (Hamamatsu) and controlled by the NIS-Elements software (Nikon, v 5.11.01). The cells were imaged using a 20× Nikon CFI Plan Apo Lambda objective (NA 0.75), either 1 frame per minute for 2 hours or 1 frame every 5 minutes for 17 hours. This framerate was sufficient to monitor the cells' movements without affecting the cell behavior. In these experiments, a camera binning of 2 × 2 was used to reduce photobleaching. For generating a DL model, all frames were separated as individual images.

The pix2pix model was first trained from scratch with the following settings: 300 epochs, patch size: (512,512) with a batch size of 1, and a vanilla GAN loss function, using the pix2pix ZeroCostDL4Mic notebook (v 1) (von Chamier & Laine et al., 2020). Key Python packages used include TensorFlow (v 2.9.2), NumPy (v1.21.6), torch (v1.12.1), and CUDA (v 11.2.152) C/C++ library for GPU programming. The training was accelerated using a Tesla T4 GPU. The model was re-trained with 100 additional epochs to enhance the model performance. To assess the quality of the final pix2pix model, we prepared a quality control dataset of 15 paired images using the built-in quality assessment metrics of the notebook.

5 Deep learning pipeline to segment and track cancer cell attachment to endothelial cells (unpublished 1)

My lab studies how different Pancreatic Ductal Adenocarcinoma (PDAC) cell lines adhere to and arrest endothelium during cancer metastasis under physiological flow speeds. He generated a dataset (D10) in his research where these events were imaged using a microfluidics setup. My role in the work was to generate DL models for the pipeline to detect cancer cells migrating on top of the endothelium (M5), endothelial cell-cell junctions (Pix2pix BF -> PECAM + Cyto2) (M4 + M2), and endothelial nuclei (M3+M7).

5.1 The HUVEC monolayer dataset (D18)

Microfluidic channels (IBIDI) were coated with 10 µg/ml of fibronectin and seeded with endothelial cells to form an endothelial monolayer. Three different PDAC cell lines (AsPC1, MiaPaca-2, PANC10.05) were perfused at different speeds to study their capacity to adhere on top of the endothelial monolayer. Brightfield images of the cancer cells and the endothelium under flow were acquired using a Nikon Eclipse Ti2-E microscope (Nikon) equipped with an sCMOS Orca Flash4.0 camera (Hamamatsu) and controlled by the NIS-Elements software (Nikon, v 5.11.01) with an exposure time of 5 ms and a frame rate of 40 frames per seconds 2 minutes per flow speed. The resulting 8-minute-long movies were composed of a continuous acquisition of 2 minutes at 4 perfusion speeds: 400, 200, 100 and 400 µm/sec (mean speed measured at the level of the endothelial cells), resulting in approximately 12,000 frames. Raw movies were then cut into 4 standardized extracts of 2250 frames to be analyzed using our DL pipeline.

5.2 Detection and analysis of migrating cancer cells

To detect cancer cells in the brightfield images, a custom StarDist model, StarDist BF cancer cells (M5), was trained. For this model first, a 20 image ground truth dataset was created by manually annotating all the cancer cells floating on top of the endothelium (dimensions: 1024 × 1022 px) using the LOCI plugin in Fiji (Schindelin et al., 2012). Each image had 10-50 annotations. The ground truth data was augmented to improve the model performance by generating a dataset of 160 paired images by using the Augmentor ZeroCostDL4Mic notebook (von Chamier et al., 2021; Bloice et al., 2019) by randomly cropping it into 992 x 992, rotating, flipping and multiplying it by 8. Using the new 160 ground truth images (992 x 992 px), a custom StarDist model was trained using the StarDist 2D ZeroCostDL4Mic notebook (v1.12.2) with the following parameters: 400 epochs, patch size: 992 x 992 with a batch size of 2 and a mae loss function. Key Python packages used include TensorFlow (v2.11.0), CSBdeep (v0.7.3), and CUDA (v11.8.89) C/C++ library for GPU programming. The training was accelerated using a Tesla T4 GPU. The performance of the generated model was assessed using the built-in quality assessment metrics of the notebook. The generated StarDist model showed high performance, resulting in an average F1-score of 0.923 when tested with 4 image pairs previously unseen by the network. The detected cells were Tracked using the TrackMate label detector using the Simple LAP tracker with the following settings: Linking max distance: 15.0 pixels, Gap-closing max distance: 15.0 pixels, Gap-closing max frame gap: 2. The tracks were filtered so that only tracks that had a minimum of 50 spots were considered for the analysis. To analyze the adhesion rate, we included tracks that have a speed of 5 µm/sec or less using TrackMate. For the analysis of the distances between nuclei and the cell-cell borders, we filtered tracks that had an instant velocity above 20 µm/sec at the beginning of the track and 5 µm/sec at the end of the track. This filtering was done using custom Python code. The filtered tracks were analyzed using CellTracksColab (Jacquemet, 2023).

5.3 Detection and analysis of endothelial nuclei and cell-cell junctions

Two DL models were used to detect endothelial nuclei. First, a custom Pix2pix model, Pix2Pix BF -> nuclei (M3), was trained to predict all nuclei, including nuclei of the cancer cells in the brightfield images. For the training ground truth data was created by acquiring 258 paired images (1022x1024 px) of the fixed endothelium with cancer cells stained using sir-DNA (Spirochrome) under brightfield and fluorescence and augmented 8 times by rotating, flipping, randomly zooming, and distorting, shearing, and skewing to generate 2064 unique images. Using these ground truth images, the model was trained from scratch using the following settings: 400 epochs, patch size: (512,512) with a batch size of 1, and a vanilla GAN loss function, using the pix2pix ZeroCostDL4Mic notebook (v 1) (von Chamier et al., 2021). Key Python packages used include TensorFlow (v 2.12.0), NumPy (v1.22.4), torch (v2.0.0), and CUDA (v11.8.89) C/C++ library for GPU programming. The training was accelerated using a Tesla T4 GPU. The performance of the generated model was assessed using the built-in quality assessment metrics of the notebook. The resulting model predicted each nucleus of endothelial and cancer cells in the original brightfield images.

Second, a custom StarDist model (M7) was trained from scratch to segment only the endothelial nuclei from the pix2pix predictions. For this, ground truth data of 17 images were first segmented using the StarDist Versatile nuclei model in Fiji, and the nuclei of cancer cells were manually removed using Fiji, and the dataset was augmented by 4. The custom StarDist model (was trained for 200 epochs), patch size: 1024,1024 with a batch size of 2 and a mae loss function, using the StarDist 2D ZeroCostDL4Mic. notebook (v1) (von Chamier & Laine et al., 2020). Key Python packages used include TensorFlow (v 0.1.12), Keras (v2.3.1), CSBdeep (v 0.6.3), NumPy (v 1.19.5), and CUDA (v 11.0.221) C/C++ library for GPU programming. The training was accelerated using a Tesla K80 GPU. The performance of the generated model was assessed using the built-in quality assessment metrics of the notebook. The generated StarDist model showed high performance, resulting in an average F1-score of 0.952 when tested with 4 image pairs previously unseen by the network.

To artificially label PECAM in the cell-cell junctions, we trained a pix2pix model (M4). The pix2pix model was trained in three rounds due to long processing times, first with 85 epochs and then the model was re-trained for twice 80 epochs. All trainings were performed using 484 paired image patches (1022x1024px), patch size: (512,512), augmented 6 times using rotation, flipping, random zoom magnification, random distortion, image shearing, and image skewing, resulting in 2904 unique images. The model was trained from scratch with a batch size of 1 and a vanilla GAN loss function, using the pix2pix ZeroCostDL4Mic notebook (v1) (von Chamier & Laine et al., 2021). Key Python packages used include TensorFlow (v 2.12.0), NumPy (v1.22.4), torch (v2.0.0), and CUDA (v11.8.89) C/C++ library for GPU programming. The training was accelerated using a Tesla T4 GPU. The performance of the generated model was assessed using the built-in quality assessment metrics of the notebook. To further outline the cell-cell borders, we first summed all the pix2pix predictions in one frame and then used the Cellpose Cyto2 model (M2) to detect the endothelial cell. Cell-cell borders were extracted using CLIJ2 (Haase et al., 2020), using the detectLabelEdges-function followed by thresholding in Fiji. The model performance was visually assessed against a ground truth image where the cell-cell junctions were manually drawn.

To apply the trained DL models to the actual data, a custom-made Google Colaboratory notebook was created. This notebook takes original videos and applies the optimized models and operations in the correct order to generate the desired outputs. For the analysis of the distances between nuclei and the cell-cell borders, we filtered tracks that had an instant velocity above 20 µm/sec at the beginning of the track and 5 µm/sec at the end of the track. This filtering was done using custom Python code. The filtered tracks were analyzed using CellTracksColab (Jacquemet, 2023).

6 Deep learning pipeline to understand drug resistance (unpublished 2)

6.1 The FUCCI dataset (D6)

To study how mutations in cells affect cell motility and cell cycle under drug treatment, we received a live cell imaging dataset from our collaborators. This dataset consisted of videos of non-mutated and mutated PC-9 cells under drug treatment. The specific drug and mutation details are not disclosed here as the data remains unpublished. In the selected dataset, the mutations were performed using CRISPR/CAS9 technology. The first five control wells, C2-C6, served as pooled controls, while wells C7-C11 originated from a single-cell clone derived from the pooled control. The mutation 1 (wells F2-F6) and mutation 2 (wells F7-F11) were single-cell clones originating from CRISPR/CAS9 edited mutation pool and composed of biallelic mutation of a pro-apoptotic signaling pathway protein. All cells were treated with drug A, labeled using the FUCCI cell cycle indicator (Zielke and Edgar, 2015) and imaged using the Incucyte (Stratorius) using phase contrast, green and red channels. Images from three different fields of views per well were captured every 2 hours for 24 days, generating a total of 60 videos with 292 frames. The generated videos were chopped at the time point where cells reached full confluency. The fluctuation of the fluorescent signal caused by the addition of media to the sample was first removed by using the Bleach correction Fiji plugin (method: simple ratio) (Miura, 2020). Data drifting was then corrected by using first generating a sum image of the green and red channels, an image I called "both". The amount of drift was estimated using the StackReg Fiji plugin (Thevenaz et al., 1998) and applied to the other channels by using a custom-made Fiji script.

6.2 DL pipeline to detect nuclei

To segment the cells, a multistep DL pipeline was generated using Google Colaboratory. In this pipeline, we combined five different DL models which output we then combined to a new image I called the "final mask". To segment nuclei of the cells, in the first StarDist model (M12) we used the Versatile nuclei (M10) as a starting point and re-trained the model for 200 epochs by using 10 "both" images paired with manually annotated ground truths (image dimensions: (972, 1296), patch size: (944,944) with a batch size of 2 and a mae loss function, using the StarDist 2D ZeroCostDL4Mic notebook (v 1) (von Chamier & Laine et al., 2020). The data was augmented by a factor of 4. Key Python packages used include TensorFlow (v 2.11.0), Keras (v2.3.1), CSBdeep (v 0.7.3), NumPy (v 1.21.6), and CUDA (v 11.6.124) C/C++ library for GPU programming. The training was accelerated using a Tesla T4 GPU. The performance of the generated model was assessed using the built-in quality assessment metrics of the notebook. The model yielded an average F1 score of 0.92, when tested with 2 image pairs previously unseen by the network.

Using this model, we were unable to segment the nuclei of young daughter cells, as the FUCCI reporter does not express any color when cells undergo cell division at the beginning of G1. Therefore, we used 1740 paired images (image dimensions: (971,972) of bright field and both to train a pix2pix model from scratch to predict all nuclei in the image (M13). The training was done in three iterations using a total of 385 epochs (140+145+100) and using the following parameters: patch size: (512,512) with a batch size of 1 and a vanilla GAN loss function, using the pix2pix ZeroCostDL4Mic notebook (v 1) (von Chamier & Laine et al., 2021). Key Python packages used include TensorFlow (v 2.12.0), NumPy (v1.22.4), Torch (v 2.0.0), and CUDA (v11.8.89) C/C++ library for GPU programming. The training was accelerated using a Tesla T4 GPU. The performance of the generated model was assessed using the built-in quality assessment metrics of the notebook. We selected checkpoint 50 from the third training to be used for the pipeline. We used the same StarDist model from step one (M12) to segment the nuclei from these predictions.

As we still did not reach good enough segmentation quality, we trained another StarDist model to detect rounded cells from the BF image (M14). I trained this model from scratch by using 20 paired images of BF and manually annotated ground truths (972x1296px) using the following parameters: 200 epochs, patch size: (944,944) with a batch size of 2 and a mae loss function, using the StarDist 2D ZeroCostDL4Mic notebook (v 1.18) (von Chamier & Laine et al., 2021). Key Python packages used include TensorFlow (v 2.12.0), CSBDeep (v 0.7.3), and CUDA (v 11.8.89) C/C++ library for GPU programming. The training was accelerated using a Tesla T4 GPU. The performance of the generated model was assessed using the built-in quality assessment metrics of the notebook. The model yielded an average F1 score of 0.71 when tested with 4 image pairs previously unseen by the network.

Finally, we combined the output labels of the three segmentation models and applied the StarDist versatile nuclei segmentation model to form an image we called the “final mask” achieving a sufficient segmentation of the nuclei.

To streamline image processing, we combined all the used models into a single pipeline using a Google Colaboratory notebook. This notebook expects the drift-corrected fluorescent videos as input, applies the models onto the image in the correct order and exports the “final mask” videos for the subsequent tracking step.

6.3 Tracking and analysis

For cell tracking, we merged the final mask with the original phase contrast, red and green channels to form a new image we called the tracking image. We used TrackMate using the label detector to detect the cells before tracking and discarding objects below 59 px. For tracking, we used the LAP detector with settings: linking max distance = 25 px; track segment max gap distance = 30 px, track segment max frame gap = 3, track segment splitting = 20 px. These results were used for the track analysis. We generated another set of results with the same settings, except we didn’t allow segment splitting. These results were used for the peak analysis.

To analyze cell tracks, we used a custom-made Google Colaboratory notebook to analyze the track duration and the division times between conditions directly from the TrackMate output. Next, we extracted the red signal peaks (G1-phase) by using the scipy.signal.find_peaks python function (Virtanen et al., 2020). We then measured the FWHM of the red signal peaks to calculate the duration of the G1-phase, and the distance between the red signal peaks to measure the cell cycle duration. These measurements allowed us to understand the cell movements, division times, if the cell cycle of the treated cells was delayed and if the cells remained longer time in the G1-phase.

Results

1 Fast4DReg helps in the correction of drifty 3D image data

1.1 The Fast4DReg pipeline

Fast4DReg was developed to process drifty 3D video data acquired in our lab. As we produce large numbers of image files that need to be corrected for further analysis, our lab needed a quick and robust tool for removing drift in 3D image data. Previously, in our lab, we used tools such as NanoJ (Laine et al., 2019) for drift correction in 2D data. In the case of 3D data, we have employed Correct 3D Drift (Parslow et al., 2014). We acquired good results with NanoJ, but the need to extend the drift correction to 3D data remained. The starting point for the first publication in this thesis (I) started as a collaboration with the developers of the NanoJ tool and our need to extend the tool to correct drift in 3D data.

The development started with identifying the need for biologists to correct 3D data. Identified needs were 1) the tools need to be flexible and allow selective correction of axial or lateral drift, or both, 2) the need to be able to apply the measured drift to another dataset, for example, another channel, and 3) a user-friendly GUI with the availability of test datasets and documentation. Considering these requirements, we developed a tool that breaks the drift-correction task into two steps: Fast4DReg first estimates an optimal transformation that corrects the drift, followed by applying the determined parameters to produce a corrected image. We called these tasks “time-estimate and apply” and “time-apply”, respectively. The “time-estimate and apply” step first creates intensity projection along the z-axis (Figure 16A, Study I: Fig. 1). Then, it creates intensity projections along the y- or x-axis and estimates and corrects the axial drift. Both steps can also be utilized individually, which can be particularly useful when only one type of drift needs to be corrected. As an output, Fast4DReg creates a new folder with a unique identifier defined by the user and the date of the experiment. This folder contains the corrected images, drift plots (graphs indicating the amount of drift detected), a drift table (drift detected in numerical values), and a settings file containing the selected parameters. Using the “time-apply” tool, the drift table can then be applied to correct other images using the same parameters. The estimation of drift between the reference and moving frames is calculated based on their cross-correlation matrix (CCM). The peak intensity in the CCM then defines the linear shift between the two images (Figure 16B). Fast4DReg allows sub-pixel accuracy by upscaling the CCM via bicubic spline interpolation (Laine et al., 2019).

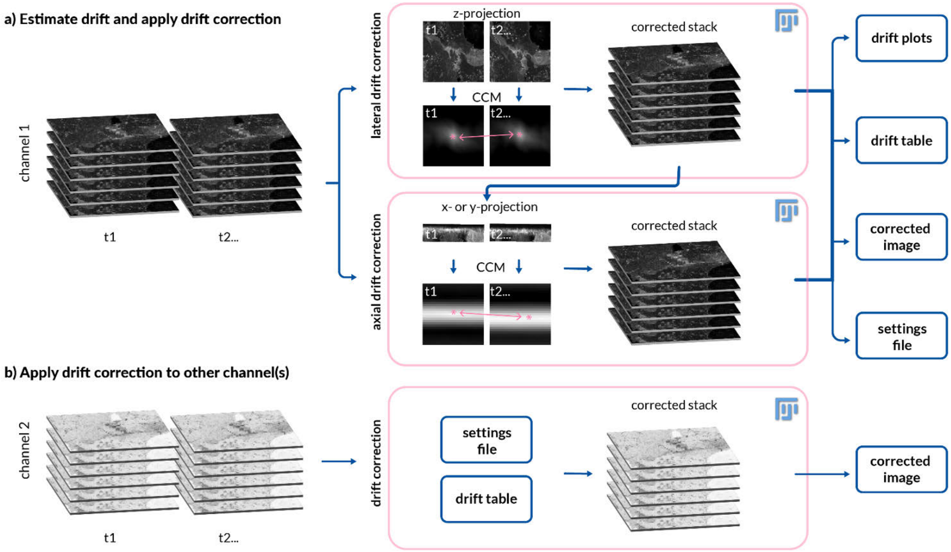


Figure 16: The Fast4DReg pipeline A) Fast4DReg first creates intensity projections along the z-axis, corrects the drift, and repeats the same along the y- or x-axis. The drift between the reference and moving frames is calculated by their cross-correlation matrix (CCM). The location of the peak intensity in the CCM (pink asterisk) defines the linear shift between the two images (as highlighted by the pink double-headed arrow). Fast4DReg outputs the corrected images, the drift plots, a drift table, and a settings file containing all selected parameters and paths to the drift table. B) The settings file inducing the used parameters and path to the drift table can be applied to correct other datasets (i.e., another channel) directly (Study I: Fig. 1).

1.2 The Fast4DReg graphical user interface

To improve user-friendliness, an intuitive GUI was created (Study I: Fig. S2), alongside detailed documentation for each option in the GUI (https://github.com/guijacquemet/Fast4DReg). When optimizing the drift estimation step, several settings can be used. To help the user keep track of the tested metrics and to avoid overwriting the results folder, we included an option to enter numerical experiment numbers in the GUI. This way, the user can always return to the outputs of each experiment to find the optimal parameters. The GUI also allows the selection of the drift correction type (lateral, axial, or both), the projection type (average or maximum), or the reference frame (first or previous). Selecting the reference frame is dependent on the type of video to be corrected. When the reference frame is set to 'first frame (default, better for fixed), every frame will be compared to the first frame to calculate the drift. This option is preferable in videos where there is slow scale drift overlaying the faster motion of the sample, for example, moving cells. In cases where the sample structure is assumed to be static, selecting the first frame as a reference is better to avoid generating additional drift.

Batch processing or scalability of the analysis tools is essential when the aim is to produce new scientific discoveries. To respond to this need, we incorporated a batch processing option into Fast4DReg GUI. In the “estimate+apply” GUI, it is possible to drag-and-drop one or multiple files and correct them in batch using identical parameters. In the “apply” GUI, one settings file is selected, and similarly to the “estimate+apply” functions, all images that need to be corrected in those settings can be dragged and dropped to the GUI. This is especially useful when applying the correction to other channels or when a drift table from a registration slide is applied to biological images to correct multiple misaligned channels.

1.3 Fast4DReg can quickly and reliably remove drift in 3D image data

Live cell imaging data can have multiple features that make downstream analysis difficult; it can, for example, be noisy and have multiple channels. To benchmark Fast4DReg’s capability to correct drift, we compared its performance to two openly available drift correction tools, Correct 3D Drift (Parslow et al., 2014) and Fijiyama (Fernandez and Moisy, 2021). We selected these tools as they have been used in our projects before to remove drift on 4D image data.

To first try the performance of Fast4DReg in a controlled environment, we generated a dataset with synthetic drift with a known amount of drift. We corrected this dataset using Fast4DReg, Fijiyama, and Correct 3D Drift. We showed that Fast4DReg outperformed the other tools (Figure 17, Study I: Fig. 2A-C).

To quantitatively assess the drift correction ability, we generated a Jupyter notebook that measures image similarity metrics between frames of a selected slice. We showed that in the quantitative measurements, Fast4DReg performed better than Correct 3D Drift or Fijiyama on our synthetic dataset (Study I: Fig. 2C).

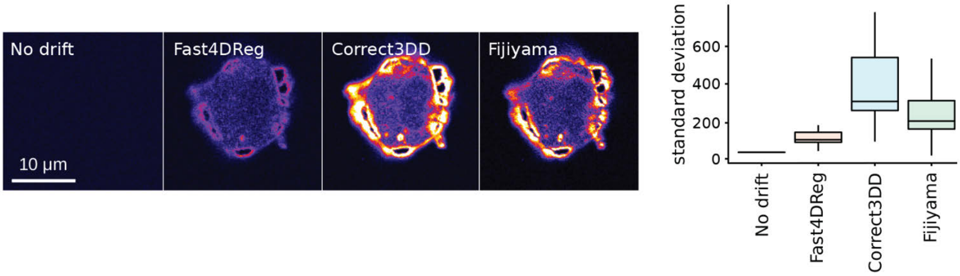


Figure 17: Fast4DReg benchmarking with synthetic drift dataset. Standard deviation time projection was taken of slice 51 (middle of the cell), taking the standard deviation of the pixel intensities through time. Pixels with large intensity through the stack appear brighter in this projection. And black pixels show no variation between the frames over time. For quantification, the standard deviation projection of each z-slice over time was generated and quantified using Fiji, and the results are shown as boxplots created by PlotsOfData (Postma and Goedhart, 2019). The boxes show the 25th and 75th percentiles, the whiskers represent the minimum and maximum values, and the median is marked with a line. No drift shows a high baseline value as specified noise was added during background homogenization. (Study I: Fig. 2 C)

As live cell imaging often suffers from noise, we wanted to test how well the Fast4DReg can correct drift in noisy images. To do this, we generated a series of noisy videos and corrected them using Fast4DReg. To assess the drift correction performance, we applied the estimated drift correction of every noisy image to the original drifty dataset and assessed the corrected images using the Jupyter Notebook measuring image similarity metrics. We showed that Fast4DReg was not affected by noise when the SNR was above 2. With SNR below 2, the drift correction performance decreased. We also found that with very noisy images (SNR below 1.6), Fast4DReg performed much better when using average-intensity projections compared to maximum-intensity projections (Figure 18A-B, Study I: Fig. 3C–F). These findings show that Fast4DReg is relatively resistant to noise and that in cases where high noise levels are inevitable, average intensity projections provide a better drift correction.

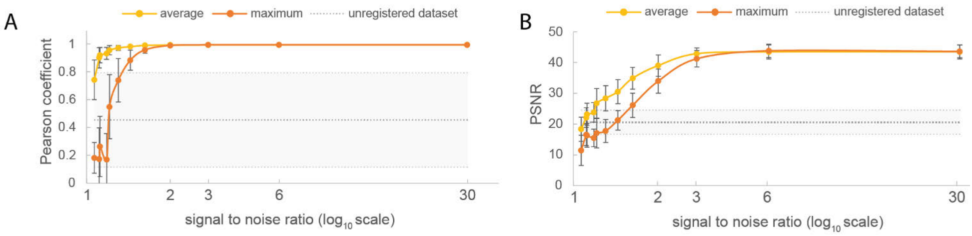


Figure 18: Fast4DReg performance on noisy images. Fast4DReg drift-correction performance for the twelve noisy datasets was assessed using image-similarity metrics. The PCC (A) and PSNR (B) between the first and subsequent frames were calculated for each noise amount. Fast4DReg performance was not affected by noise when the SNR was above 2. With SNR below 2, the drift correction performance decreased. Very noisy images with an SNR below 1.6 Fast4DReg performed better when using average-intensity projections compared to maximum-intensity projections. Higher values in both graphs indicated a better correction ability. (Study I: Fig. 3F).

Next, we wanted to test the drift correction ability of Fast4DReg on a 3D video of a biological sample. We used a long 3D video of a human umbilical vein endothelial cell (HUVEC) monolayer labeled with silicon rhodamine (SiR)-actin and imaged using an Airyscan confocal microscope. This dataset suffered from significant drift in all x-, y- and z-directions mainly due to the unstable microscopy stage. We removed the drift using Fast4DReg, Correct 3D Drift, and Fijiyama. We showed that both Fast4DReg and Correct 3D Drift reduced the amount of axial drift, although the correction was not perfect (Study I: Fig 4 B). We believe that this might be due to a situation where a segment of the data repeatedly exited the imaging volume. The lateral drift in the original dataset was not too severe, but Fast4DReg still improved the data (Study I: Fig. 4A). Interestingly Correct 3D Drift processing led to the monolayer slowly sinking over time, introducing lateral drift rather than removing it (Study I: movie 3). These findings were also quantitatively presented (Study I: Fig. 4C). We failed to obtain reasonable results using Fijiyama as it caused the video to drift even more than the raw data.

Finally, we showed that Fast4DReg can register multichannel 3D videos. We used a movie of cancer cells migrating inside the lung vasculature, which was imaged ex vivo using an Airyscan confocal microscope (Study I: Fig. 5A; movies 4 and 5). These videos suffered from significant drift caused by the sample floating in the sample dish. We used the vasculature images to first estimate the drift and then applied the correction to both channels (Study I: Fig. 5). We showed that this approach can successfully create a 3D video without drift. The ability of Fast4DReg to apply drift tables to other datasets significantly simplifies the drift removal workflow of multichannel data.

1.4 Fast4DReg can align misaligned multichannel images

Images are arrays of data organized in dimensions. In 4D image data, the x-, y-and z-dimensions hold spatial intensity information of the image whereas the time dimension can be used to explore the changes of these intensities over time. Multichannel images are also considered 4D data, but in this case, the channels represent the fourth dimension. As Fast4DReg is designed to correct drift in 4D data, its drift correction function can also be extended to correct drift between misaligned 3D images. To do this, Fast4DReg uses the same pipeline as described for time series but first converts the channels into time frames. We called these functions "channel estimate-apply" and "channel-apply".

Commonly, microscopes suffer from misalignment of channels when acquiring images. This can be caused by chromatic aberrations and when using different cameras for image acquisition. To test the Fast4DReg's ability to correct misalignment between channels we first acquired a 3-channel image of a calibrations slide followed by an image of a biological sample. Both these raw images displayed significant xyz-misalignment. We first used the "channel estimate-apply" function of Fast4DReg to estimate the amount of drift between the channels and then applied the drift correction using the "channels-apply" function. We used line-intensity profiles to visualize the correction ability and we found that Fast4DReg could successfully register this dataset laterally and axially (Figure 19, Study I: Fig. 6A, B). Combined with the Fast4DReg batch-processing mode, this allows automated high-throughput correction of image data acquired in one imaging session.

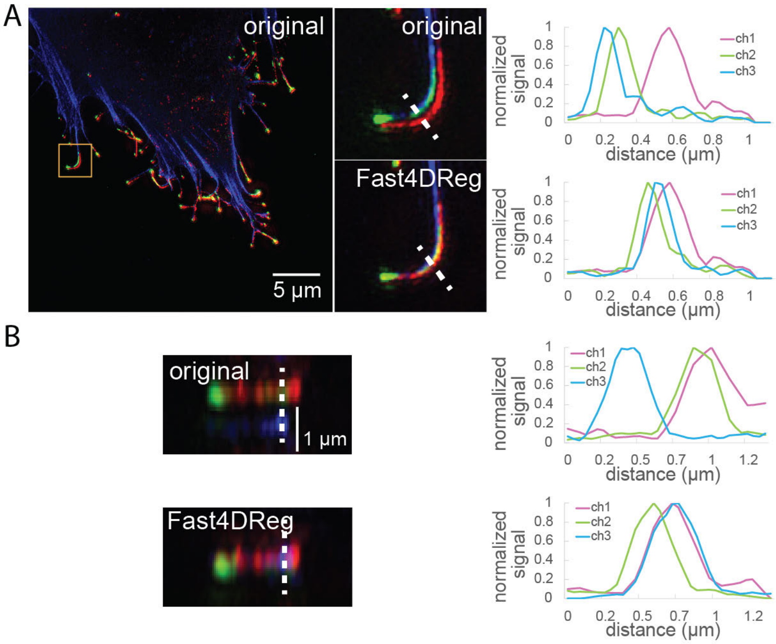


Figure 19: Fast4DReg can align misaligned channels using a calibration slide. A drift table was generated using a calibration slide (Study I: Fig. 6A-B). This drift table was used to correct a 3D SIM image of a U2-OS cell expressing GFP-tagged lamellipodin (RAPH1, red) and MYO10–mScarlet (green), and labeled to visualize its actin cytoskeleton (blue). Line-intensity profiles over the dashed lines show that Fast4DReg could successfully register this dataset laterally (A) and axially (B).

Fast4DReg’s ability to correct misaligned channels becomes useful when preparing training data for DL. To test this, we used a dataset acquired to train a supervised image-restoration DL algorithm. This dataset has two channels, one acquired with high and the other with low SNR. As these channels were acquired using different cameras on the same microscope setup, they had a small shift. We used the “channels estimate-apply” tool to correct the channels and by using line projections we showed that Fast4DReg was able to remove the shift between the channels (Study I: Fig. 6E–F). We envision that by using such pre-processing steps before DL training, the performance of DL models for image restoration can be significantly improved. This approach should be used with caution as Fast4DReg requires structural overlap between the channels to successfully perform the registration.

1.5 Fast4DReg accelerates drift correction tasks

Drift removal from 3D image data, using previously available tools, has been tedious and a slow task. Often the image data is corrected file by file requiring hours of work. We improved the speed of corrections by first performing the drift estimation step by using intensity projections rather than the whole image volume and secondly implementing batch processing.

We used two different computers to assess the time required for Fast4DReg, Correct3DD, and Fijiyama to process the HUVEC dataset. The first computer was a high-performance desktop and the second one a laptop. We found that Fast4DReg was four to nine times faster than Correct 3D Drift and 20 to 90 times faster than Fijiyama when correcting the HUVEC dataset (Table 9, Study I, Fig. 4D, S1D). We also compared the Fast4DReg speed against the RAM saving mode of Fast4DReg, which can be used with computers with limited resources. This mode retains the original bit depth of the image, instead of upscaling it to a 32-bit image. We found that the RAM saving mode performs 1.3 times faster than the regular mode (Table 9). In addition, when using Fijiyama, the file preparation step brings an additional time-consuming step to the process, and all time points need to be separated as individual files before the correction. Overall, in addition to drift correction performance, Fast4DReg also outperformed both, Correct 3D Drift and Fijiyama in speed when correcting the HUVEC dataset.

Table 9: Fast4DReg Correct 3D Drift and Fijiyama processing times. Fast4DReg execution times are much faster compared to the Correct 3Ddrift and Fijiyama. When using the RAM saving mode, which retains the image bit depth, the execution time can be further decreased. The times are the average of three repeats.

	Fast4DReg	**Fast4DReg (RAM save mode)**	**Correct 3D Drift**	**Fijiyama**
Computer 1	1.5 min	1.15 min	12.6 min	123 min
Computer 2	5.9 min	4.5 min	24.2 min	124 min

2 TrackMate v7 enhances tracking analyses (II)

2.1 Integration for improved detectors and analysis tools to TrackMate improves tracking

In life sciences, tracking is commonly used, to follow single particles, subcellular organelles, bacteria, cells, and whole animals to further understand their behavior. In a typical tracking workflow, the objects are first detected and then linked to form tracks. The previous version of the TrackMate (Tinevez et al., 2017) based the detection step LoG and DoG detectors suitable for tracking round blob-like structures, such as filopodia tips (Popović et al., 2023), but performs poorly for textured objects, objects with irregular shapes, and imaging modalities other than fluorescence. These detectors also only measure the objects' position and are not able to measure objects' shape or signal intensity information.

We extended the new TrackMate to incorporate several different segmentation strategies, including the ilastik (Berg et al., 2019) Weka (Arganda-Carreras et al., 2017), Cellpose (Stringer et al., 2021), MorphoLibJ (Legland et al., 2016), and StarDist (Schmidt et al., 2018) detectors, all commonly used in segmentation workflows. TrackMate can also import segmentation results generated by any other segmentation algorithm, allowing the tracking of mask or label images. Additionally, the TrackMate v7 detects object contours in every frame, allowing tracking of changes in the objects' morphological features or signal intensities (Study II: Fig 1).

To improve the usability of the new functions, extensive user documentation (https://imagej.net/plugins/trackmate/detectors/trackmate-v7-detectors, 11.03.2024) was written to complement the original documentation of TrackMate (https://imagej.net/plugins/trackmate, 11.03.2024).

2.2 Analysis of ERK signaling in motile cancer cells

In this work, one of my contributions was to show how ERK signaling can be measured in timelapse videos of motile MDA-MB-231 and U2OS cells expressing an ERK-KTR activity reporter (Kudo et al., 2018). This reporter shuffles between the nucleus and cytoplasm depending on the activation state of ERK (Figure 20). The nuclei were labeled with SiR-DNA.

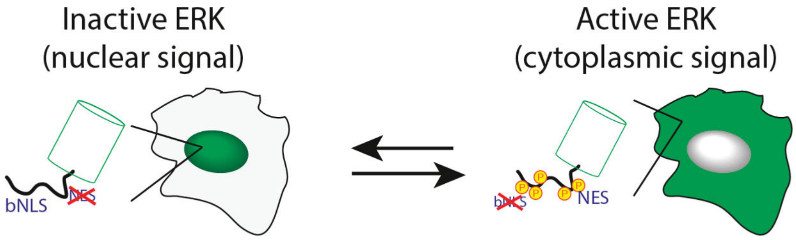


Figure 20: Schematic of the ERK-KTR activity reporter. High ERK activity leads to an increase in phosphorylation and inactivation of the bNLS and shuttling out of the nucleus. Modified from (Conway et al., 2023).

We tracked the nuclei of MDA-MB-231 using the SiR-DNA labeling by using the pre-trained Versatile nuclei model directly in TrackMate. U2OS cells were very sensitive to light and died after only a short acquisition, forcing us to use very low excitation power. The dying cells showed high levels of SiR-DNA which was picked up by the pre-trained Versatile nuclei model. We did not want to include the dying cells in the analysis, so we trained a custom StarDist model using the ZeroCostDL4Mic platform (von Chamier et al., 2021) to track the nuclei of U2OS directly in TrackMate. This new model only detected the SiR-DNA labeled nuclei of healthy cells, with a very low fluorescence signal. In both cell lines, the SiR-DNA signal was used for segmentation, and then measured the average intensity of the ERK reporter (in another channel) over time and the instant velocity of the cells directly in TrackMate. For visualization, we used heat maps by using PlotTwist (Goedhart, 2020) (Figure 21, Study II: Fig. 2 C-D, Study II: Fig. S1 A-B).

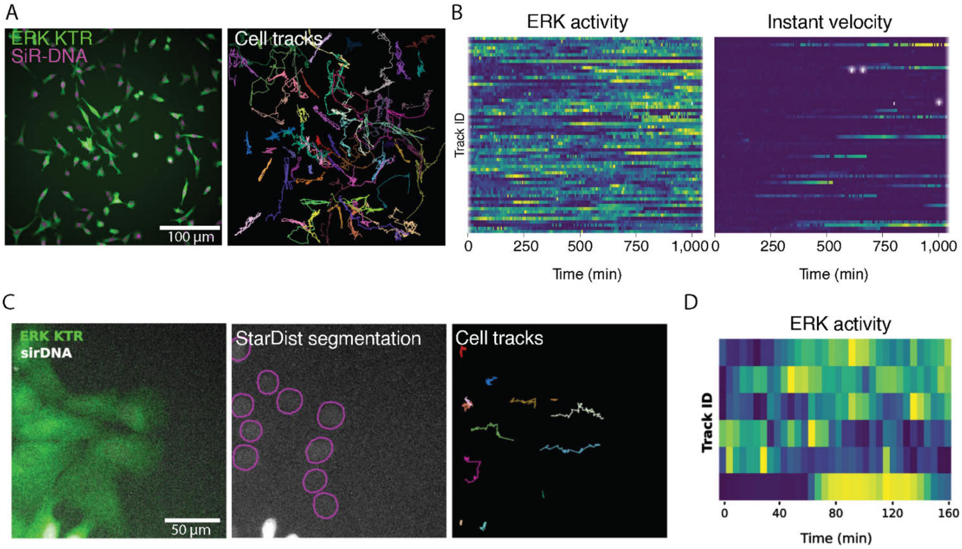


Figure 21: Tracking MDA-MB-231 and U2OS cells using TrackMate. A) MDA-MB-231 cells stably expressing an ERK activity reporter (ERK-KTR-Clover) and labeled using SiR-DNA were recorded live using a widefield fluorescence microscope over 17 hours. Cell nuclei were automatically tracked over time using the Versatile nuclei StarDist model available in TrackMate. B) For each tracked cell, the average intensity of the nuclear ERK reporter was measured using TrackMate. Changes in ERK activity and instant velocity are displayed as heatmaps (blue, high; yellow, low). C) U2OS cells stably expressing an ERK activity reporter (ERK-KTR-Clover) and labeled using SiR-DNA were recorded live using a widefield fluorescence microscope over 3 hours. A custom StarDist model was trained to detect the low-signal U2OS nuclei using the ZeroCostDL4Mic platform. D) For each tracked cell, the average intensity of the ERK reporter was measured in their nucleus over time using TrackMate. Changes in ERK activity are displayed as heatmaps (blue high, yellow low). (Study II: Fig 2 C, Study II: Fig S1 A-B).

This experiment demonstrated how effortless combining DL-powered object detection and tracking can be. Crucially, it also demonstrated the DL models trained in external environments, such as ZeroCostDL4Mic, are compatible with TrackMate. This feature enables users to develop and refine their own DL models and enhances the flexibility of TrackMate by allowing a thorough analysis of diverse morphological and intensity-based metrics throughout different time points. To extend this case study, it would be interesting to include whole cell segmentation in the pipeline, for example, by using Cellpose (Stringer et al., 2021). If the whole cell volume is segmented, it is possible to subtract the nuclei from the whole cell and quantify the ERK signal. By measuring the cytoplasmic and nuclear ERK ratio and incorporating this to track behavior, more information on the ERK shuttling during cell migration could be understood.

2.3 TrackMate helper streamlines the optimization of object detection and tracking algorithms

Cell tracking consists of two consecutive tasks: detection and linking. The quality detection and lining greatly influence the tracking result. Therefore, biologists often struggle with a diversity of available algorithms to decide which algorithm to use in their experiments. We added a TrackMate-Helper module (Ershov et al., 2022) to the TrackMate v7 to help users evaluate which tracking parameters (detectors and tracking algorithms) output a tracking result closest to the ground truth. This module performs parameter sweeps over any combination of detectors and particle-linking algorithms defined by the user. The outputs are then compared with the ground truth and 8 CTC metrics (Table 1) to define the optimal tracking parameters. These parameters can then be used on the whole dataset. In a nutshell, the TrackMate helper systematically optimizes the tracking parameters for a whole dataset.

Being able to optimize cell tracking parameters plays an important role in advancing scientific research, making research processes more reliable, reproducible, and less prone to human error. Overall, the TrackMate helper provides biologists confidence in trusting their tracking results, which directly contributes to advancements in cell biology and biomedical research.

2.4 TrackMate Batcher allows batch processing of cell migration videos

Cell migration data is often produced in a high throughput manner, producing tens or even hundreds of videos. Tracking these videos one by one is by far one of the most tedious and time-consuming tasks. TrackMate Batcher is an extension of TrackMate that allows batch processing of videos using pre-defined tracking parameters. The TrackMate Helper module can manually define or generate these parameters. TrackMate batcher outputs segmentation and tracking results for each input file in .csv files and images. It can also output a tracking file for each file that can later be linked to the original video. TrackMate Batcher saves time and reduces the risk of errors that can occur when processing datasets individually, streamlines the analysis process, and ensures a consistent and reproducible approach to tracking and data analysis.

In this thesis, the TrackMate batcher was used to generate tracking results for the PDAC cells migrating on top of a layer of endothelial cells in the publication unpublished work 1 and to track the final masks of drug-treated cells in the unpublished work 2.

3 Deep learning enables understanding of cancer cell adhesion (unpublished 1)

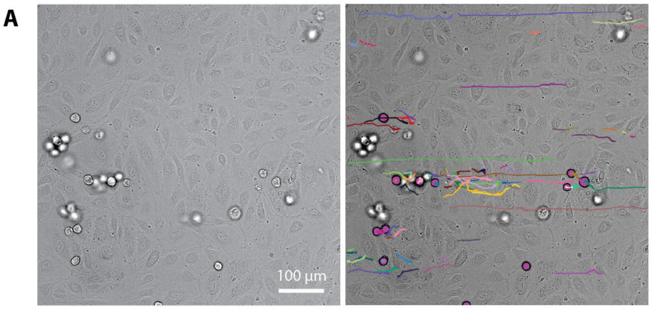
During the metastatic cascade, cancer cells first leave the primary tumor, intravasate to lymphatic or vascular circulation, arrest onto the endothelium, and successfully extravasate for a new tumor at a distal site. This is a multi-step process, where the cancer cells are affected by numerous challenges, and therefore, only a few cancer cells successfully manage to metastasize (Follain et al., 2018). In our lab, we are interested in the metastatic cascade, especially at the points where cancer cells first interact with the endothelium. During this first contact, it has been suggested that the cancer cell adheres to the endothelium using weak transient bonds and then rolls or migrates to find a spot where it adheres using higher adhesion force and prepares to extravasate. Successful intravascular arrest and extravasation are influenced by flow-mediated forces (Follain et al., 2018). Cancer cells must simultaneously resist the shear stress generated by the intravascular flow during the adhesion and retain their contact with the endothelium. If the cancer cell adhesion force is weaker than the shear stress, the cancer cells cannot adhere and are washed away. For the cells to be able to arrest, the cancer cell adhesion force needs to be stronger than the shear stress (Follain et al., 2018; Osmani et al., 2019). It has been shown that membrane proteins CD44 and β1 integrin are key mediators of cancer cell arrest (Osmani et al., 2019), and the cancer cells that successfully adhere to the endothelium extravasate by squeezing through endothelial cells or by integrating into the endothelium (Follain et al., 2018; Arvanitis et al., 2014). The exact mechanism of how cancer cells extravasate remains under further investigation.

To study how and where PDAC cells adhere to endothelial cells, we perfused three different PDAC cell lines using different physiological flow speeds and captured the events using brightfield microscopy. We designed a custom DL and tracking-based analysis pipeline, which allowed us to extract information on the behavior of the perfused PDAC cells. My role in the work was to generate DL models for the pipeline. Here, I highlight how the analysis pipeline was designed and showcase some tentative results. This pipeline and results are not yet published.

3.1 Different PDAC cell lines adhere with different rates under changing flow conditions

To understand how PDAC cells, MiaPaca-2, AsPc1 and Panc10, adhere and migrate on endothelial cells under changing flow conditions, a 4-step flow microfluidics setup was designed based on a system described in (Follain et al., 2018). In our approach, three different PDAC cell lines were perfused using high (H, 400 µm/sec), medium (M, 200 µm/sec), and low (L, 100 µm/sec) speed on top of a HUVEC endothelium, and finally, the cells were washed (400 µm/sec) to ensure that the PDAC cells have truly adhered to the endothelium. A flow velocity of 400 µm/sec matches the flow rate measured in capillary-like vessels and is favored by the extravasation of tumor cells in zebrafish embryos (Follain et al., 2018). The videos were captured using a bright field microscope.

To quantify adhesion events, we first trained a StarDist model (M5) that can detect cancer cells from the captured brightfield images (Figure 22A). As all three PDAC cell lines showed similar appearance, it was possible to use a single StarDist model for all three cell lines (Figure 22B).



B

img	IoU	FP	TP	FN	PR	RC	AC	f1	ntp	npr	MTS	MMC
02	0.80	2	39	4	0.95	0.91	0.87	0.93	43	41	0.80	0.88
08	0.71	5	41	7	0.89	0.85	0.77	0.87	48	46	0.75	0.88
14	0.83	0	29	2	1	0.94	0.94	0.97	31	29	0.83	0.88
20	0.83	3	28	2	0.90	0.93	0.85	0.92	30	31	0.85	0.91

Figure 22: Tracking of adhered cancer cells. A) On the left is a bright field image of PDAC cells (round lighter blobs) migrating on HUVEC endothelium (flat cells on the background). On the right, the PDAC cells were segmented using a custom StarDist model (M5) and tracked using TrackMate. Purple blobs are the segmented cancer cells and colorful lines represent the recorded tracks. B) Quality assessment metrics of the custom StarDist model used to segment PDAC cells were measured using four ground truth images from different cell lines. The model yielded an average F1 score of 0.92. Table abbreviations: img = image, IoU= intersection over union, FP = false positive, TP = true positive, FN = false negative, PR = precision, RC = recall, AC = accuracy, ntp = number of true objects, npr = number of predicted objects, MTS = mean true score, MMC mean matched score.

To remove fast moving, non-adherent cells from the tracking results, we filtered the tracks with a minimum of 50 spots and whose mean speed was 5 µm/sec or less. The number of remaining tracks was plotted over time. We found that the number of adhered MiaPaca-2 and AsPc1 cells increased as the flow rate decreased (Figure 23A). We also found that AsPc1 cells started adhering to the endothelium at medium speed and their adhesion rate increased as the speed decreased (Figure 23A, B). Interestingly, MiaPaca-2 cells showed no dependency on the flow speed, as they started adhering already at the high speed and their adhesion rate remained constant (Figure 23A, B). The final washing step did not wash away the AsPc1 or MiaPaca-2 cells indicating that they were adhered to the endothelium rather than passively arrested (Figure 23A). Panc10 cells did not adhere to the endothelium under any flow conditions (Figure 23A, B).

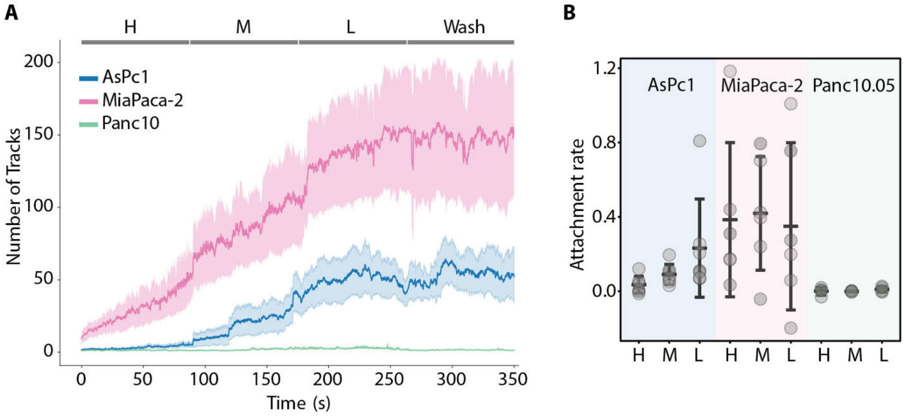


Figure 23: PDAC adhesion to endothelial cells under changing flow conditions. A) The number of AsPc1 (blue), MiaPaca-2 (pink), and Panc10 (green) tracks under high (H), medium (M), and low (L) flow followed by a washing step plotted over time. Tracks were analyzed and the plots were generated using a modified version of CellTracksColab (Jacquemet, 2023). B) Attachment rates of PDAC cells. The change of the curve in A was linearly fitted for every flow speed. Higher values indicated a larger change in the attachment rate between the beginning and the end of a certain speed.

We concluded that the tested DPAC cell lines have unique adhesion characteristics under changing flow conditions. With the overall higher number of adhered cells and independence of flow speed, MiaPaca-2 cells have the most aggressive phenotype over the three compared cell lines.

3.2 Image processing pipeline to study PDAC adhesion to endothelial cells

To study in more detail where the PDAC cell adhesion occurred on the endothelium, we aimed to segment the endothelial cell-cell junctions and the endothelial nuclei. Due to the high speed of acquisition using the lowest possible exposure time, fluorescence-based acquisition was not possible. Additionally, fluorescence illumination can cause phototoxicity, thereby altering the cell's behavior. To overcome this problem, we employed a multistep DL pipeline, which I refer to as the PDAC pipeline in this text. In this pipeline, we trained a pix2pix model to predict the endothelial nuclei (M3) which we further segmented using a custom StarDist model (M7). To find the cell-cell junctions, another StarDist model was trained (M4) and from the summed cell-cell junction predictions, we outlined the endothelial cells using the Cyto2 Cellpose model (M2). We also incorporated the custom StarDist model, which segments the cancer cells (M5) into the pipeline (Figure 24A). We applied these models to the perfusion videos by combining them into a single pipeline using a custom-made Google Colaboratory notebook, which applies all the described operations to the original videos. Using this pipeline, we were able to generate image masks from the original videos in one go for further downstream analysis of the adhesion of PDAC cells (Figure 24B).

Construction of this pipeline demonstrated seamless collaboration with image analysis experts and life science researchers. Here I, as an image analysis expert, used my expertise to guide the researcher to plan the pipeline, generate appropriate training data for the training of DL models and trained the models. As a result, this pipeline allows the researcher to independently process their own data, without the need for continuous support from an image analyst. This is a significant achievement in terms of the efficiency and autonomy of the researcher and aligns with my objective of providing user-friendly analysis tools for life science researchers. This collaboration showcases the potential impact of combining specialized expertise with user-friendly tools in advancing bioimage analysis workflows, yielding results that would not been possible to extract without this pipeline.

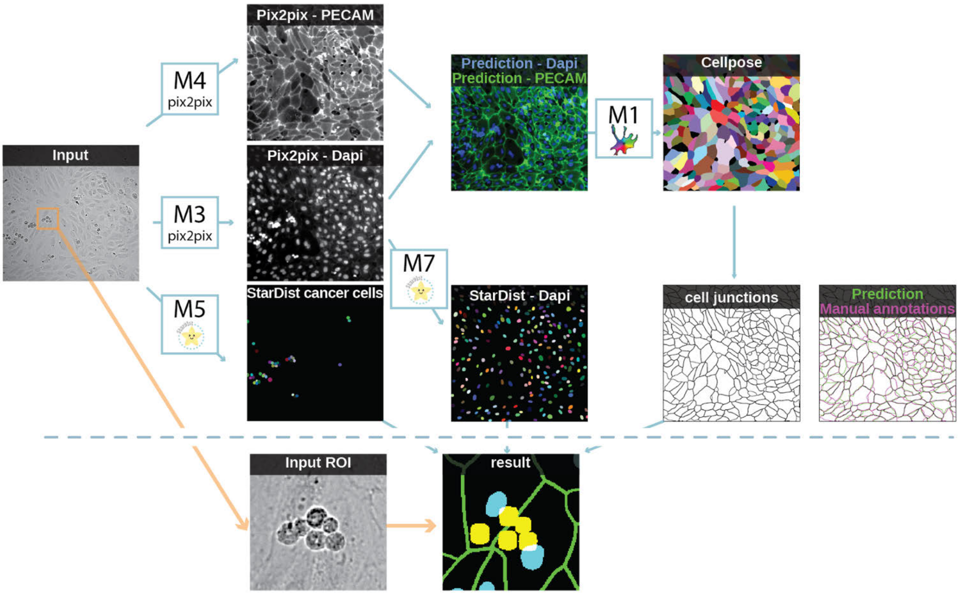
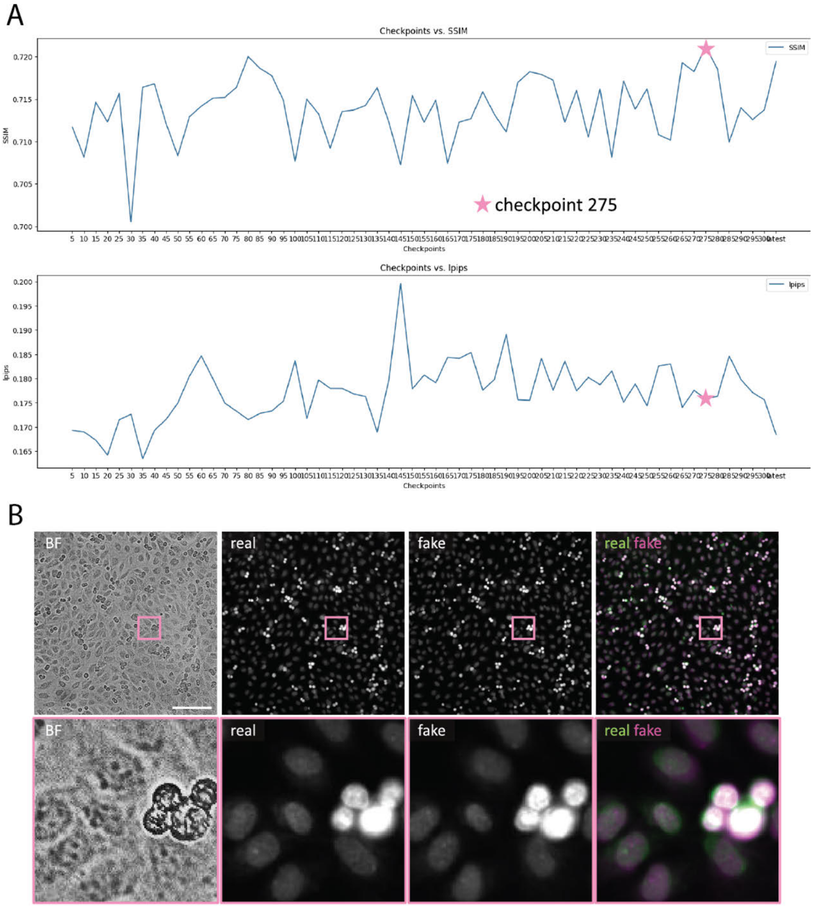


Figure 24: PDAC DL pipeline to study the localization of PDAC cells migrating on top of HUVEC endothelial cells. A) This pipeline allowed us to 1) predict and segment cell-cell junctions using the custom pix2pix (M4) and Cellpose Cyto2 model (M2), 2) predict and segment endothelial nuclei using the custom pix2pix (M3) and custom StarDist model (M7), and 3) segment migrating cancer cells on top of the endothelium (M5). All these models were trained using the ZerocostDL4Mic platform (von Chamier et al., 2021). B) The PDAC pipeline was put together using a custom Google Colaboratory notebook that expects original videos as inputs and outputs images with segmented endothelial nuclei (blue), segmented cell-cell junctions (green), and PDAC cells (yellow).

3.3 Quality assessment of the endothelial nuclei and cell-cell junction detection

As my contribution to this project was to train some of the DL models, I want to describe further how I assessed the quality of each trained model used in the PDAC pipeline (M3, M4, M1). I generated ground truth images for each step and used the built-in assessment tools of the ZeroCostDL4Mic notebooks.

During training, the ZeroCostDL4Mic pix2pix notebook saves model checkpoints every five epochs. Due to the stochastic nature of GAN networks, the last checkpoint is not always the best one to use. GANs are known to hallucinate, which can lead to a mismatch between the generated images and the expected output. Therefore, choosing the most suitable checkpoint to use to make predictions is challenging. By comparing the SSIM and lpips values between the predicted and ground truth images at each checkpoint, followed by a visual inspection, we selected checkpoint 275 to be the best to predict endothelial nuclei (Figure 25A, B). Assessment of the StarDist model for endothelial nuclei segmentation after pix2pix showed high model performance, reaching an average F1 score of 0.972 (Figure 25C).



C

img	IoU	FP	TP	FN	PR	RC	AC	f1	ntp	npr	MTS	MMC
39	0.93	7	254	6	0.97	0.98	0.95	0.98	260	261	0.93	0.95
40	0.93	4	262	8	0.98	0.97	0.96	0.98	270	266	0.92	0.95
50	0.93	7	248	3	0.97	0.99	0.96	0.98	251	255	0.94	0.95
42	0.93	6	234	3	0.98	0.99	0.96	0.98	237	240	0.93	0.95
41	0.91	9	228	8	0.96	0.97	0.93	0.96	236	237	0.91	0.94

Figure 25: Quality assessment for endothelial and PDAC nuclei predictions and segmentation. A) The performance of the endothelial cell-cell junction predicting pix2pix model (M3) was assessed by first observing the SSIM and lpips values

calculated between the predicted and ground truth images at each checkpoint. Checkpoints with high SSIM and low lpips were visually assessed, and checkpoint 275 (B) was selected for the DPAC pipeline. B) Visual assessment of the checkpoint 275. In the first column is the BF image from which nuclei were predicted using M3. The second column (real) shows the ground truth image and truth image, and the third column (fake) shows the image predicted by M3, checkpoint 275. The last column overlays the real (green) and fake (magenta). The bottom row shows a cropped region for easier visualization. C) The performance of the StarDist model, which detected the endothelial nuclei from the pix2pix predictions (M7), was assessed by computing different segmentation quality metrics. Table abbreviations: img = image, IoU= intersection over union, FP = false positive, TP = true positive, FN = false negative, PR = precision, RC = recall, AC = accuracy, ntp = number of true objects, npr = number of predicted objects, MTS = mean true score, MMC = mean matched score.

We used the same approach to determine the best checkpoint for the pix2pix model that predicts the location of endothelial cell-cell junctions and found that the best prediction was generated using the third training round, checkpoint number 60.

We concluded the trained DL models yielded a satisfactory performance and could be combined in one pipeline to generate image data for further downstream analysis of adhesion events in the endothelium.

3.4 PDAC cells adhere in proximity to endothelial cell-cell junctions rather than on top of endothelial nuclei

To understand the initial steps of the PDAC adhesion on endothelial cells, we first tracked the PDAC cells from the perfusion data. Then, we identified complete tracks of cells that at first made contact with the endothelium and then remained adhered by selecting tracks that exhibited an initial velocity above 20 µm/sec and a final velocity below 5 µm/sec and had a minimum of 50 spots. We measured the distances of these tracks from the cell-cell junctions and the endothelial nuclei at the beginning of the track, where the cancer cell first made contact with the endothelium, and at the end of the track, where the cell had adhered. We found that both MiaPaca-2 and AsPc1 cells preferably landed in areas closer to the cell-cell junction rather than closer to the nuclei (Figure 26). Neither MiaPaca-2 nor AsPc1 cells (Figure 26) showed the change in the distances to nuclei or cell-cell borders between the beginning and the end of the track, indicating that they landed directly in preferred areas.

We concluded that both MiaPaca-2 and AsPc1 cells make first contact and then adhere close to the endothelial cell-cell junctions.

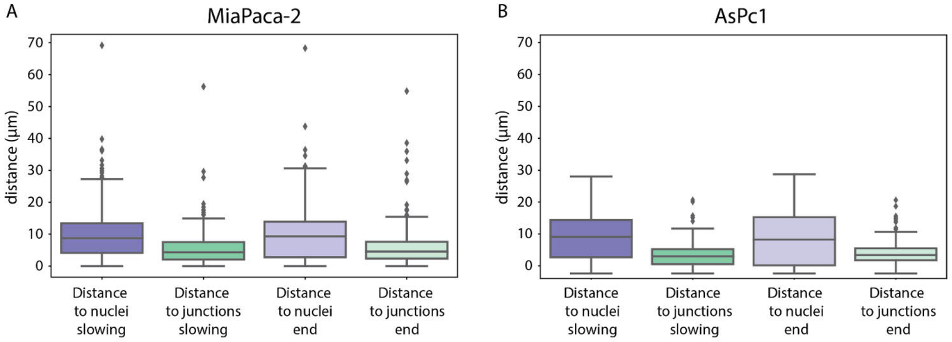


Figure 26: Cancer cells arrest and adhere close to endothelial cell-cell junctions. Quantification of MiaPaca-2 (A) and AsPc1 (B) adhesion to the endothelium at the beginning and the end of the track.

4 Deep learning enables understanding the drug resistance of cancer cells (unpublished 2)

Drug resistance in cancer refers to the ability of cancer cells to survive and continue growing despite exposure to anti-cancer drugs. Despite the initial cancer treading effect, when exposed to a drug for a long time, cancer cells can acquire genetic or non-genetic alterations that make them less responsive to the effects of the used drug. These alterations may affect the drug target directly or the cellular pathways involved in drug response (Alberts et al., 2022).

Our collaborators study the role of phosphorylation of a pro-apoptotic signaling pathway protein (name left out due to the data is yet to be published) in cell migration and the cell cycle under drug treatment. Our collaborators provided us with a dataset where they labeled PC-9 cells using a FUCCI cell cycle reporter, followed by a live cell imaging experiment (Figure 27A). We used a subset of this dataset, in which we could visually observe changes in the cell behavior to design a custom image analysis pipeline. This subset included 10 wells of PC-9 control cells (wells C2-C11) and two monoclonal cell lines originating from CRIPR/CAS9 mediated knock-in, which I call here mutation 1 (wells F2-F6) and mutation 2 (wells F7-F11). All cells were treated with drug A for multiple days. Here, I describe the DL pipeline we created to segment the PC-9 nuclei and show how we extracted information from the segmented nuclei. I also show some results from the selected data. The complete data will be published later by our collaborators.

4.1 Image processing pipeline to study cancer cell behavior under drug treatment

From the dataset received from our collaborators, we wanted to extract information that would give us insight into how the wild-type and mutated cells behave under drug treatment. We first wanted to quantify how fast the cells move and what their mean division time is. We also wanted to use the information provided by the FUCCI reporter. As the FUCCI reporter changes color to indicate the cell cycle phase, we could use this information to measure the duration of a complete cell cycle and the duration of individual cell cycle phases (Figure 27A). We chose to follow the G1 phase as its signal was more stable. To conduct these analyses, we needed to be able to detect every nucleus, track them and measure the G1 signal inside the tracked objects over time.

The dataset that we received from our collaborators had multiple limitations. First, the dataset suffered from translational drift, which complicated the tracking process, and unstable signal levels caused by the opening of the incubator door to add cell culture media. Second, in the dataset, the cells were labeled only using the FUCCI reporter, which is not fluorescent at the beginning of the G1 phase, where the mother cell divides into two daughter cells (Figure 27A). It is not possible to close track gaps and follow track splitting simultaneously, and therefore, the lack of signal at this crucial point of the cell cycle further complicated cell tracking. Although there are a few available pipelines for automated FUCCI analysis (Taïeb et al., 2022; Ghannoum et al., 2021), they were not applicable to our data, as they do not include pre-processing steps for drift or bleach removal, and they rely on long-term nuclear staining. Additionally, although these tools are openly available, they require the use of MATLAB, a resource that is not routinely used in our lab.

To overcome these issues, we designed a multistep image analysis pipeline, which I later refer to as the FUCCI pipeline. Briefly, in this pipeline, the acquired videos were first corrected for bleaching and drifting, and then a multistep DL pipeline was applied to the segmentation of the cell nuclei (Figure 27B, C). Generated nuclei labels were combined with the original BF, red, and green channels (Figure 27D), tracked using TrackMate, and finally, the tracks were analyzed using a custom-made Google Colaboratory notebook.

The development of the DL pipeline to segment nuclei from the FUCCI data was a laborious process. As the cells were only labeled using the FUCCI reporter, which changes color from green to red depending on the cell cycle phase, we started by combining the green and red channels, resulting in an image I call “both” later in the text. We tried classical segmentation methods and the Versatile Nuclei model to segment the nuclei from "both" images, but the generated segmentation results were inadequate. Therefore, we trained our own custom StarDist model to segment the nuclei from the "both" images. As neither green nor red signal is visible when the tracks split (Figure 27A), the generated StarDist model was not able to find nuclei at points where the cells divide (Figure 27C, DL task 1). As it's not possible for TrackMate to close track gaps and follow track splitting simultaneously, we could not generate complete cell tracks. Therefore, we complemented the segmentation pipeline by training a pix2pix model using the original brightfield images and the "both" images to predict the location of the nuclei of dividing cells. We segmented the generated predictions using the same StarDist model we generated to be used directly with the "both" image (Figure 27C, DL task 2). As some dividing nuclei were still not detected, we further complemented the segmentation pipeline with a StarDist model that detected rounded cells from the brightfield images (Figure 27C, DL task 3). The output labels from each task were combined, and the Versatile nuclei StarDist model was used to create the final mask. The final mask was then combined with the original images to form a tracking image for further downstream analysis (Figure 27D).

We found that the combination of the pre-processing steps was essential and sufficient in providing us with usable nuclei labels to be able to conduct reliable cell tracking and analysis using the TrackMate v7 (Ershov et al., 2022).

Similarly, to the PDAC pipeline, the above-described DL pipeline steps were combined to a single Google Colaboratory notebook to generate the "final mask" -image in one go. This integration ultimately enables researchers to independently process their own data, eliminating the need for ongoing assistance from an image analyst. I would find it interesting to test such notebooks combining DL models in the hands of life scientists, as currently this combined analysis is still run by me.

The FUCCI project illustrates the integration of user-friendly open-source methods developed in this thesis, resulting in the generation of novel discoveries that were previously inaccessible, which I further highlight in the following chapters 4.2 and 4.3.

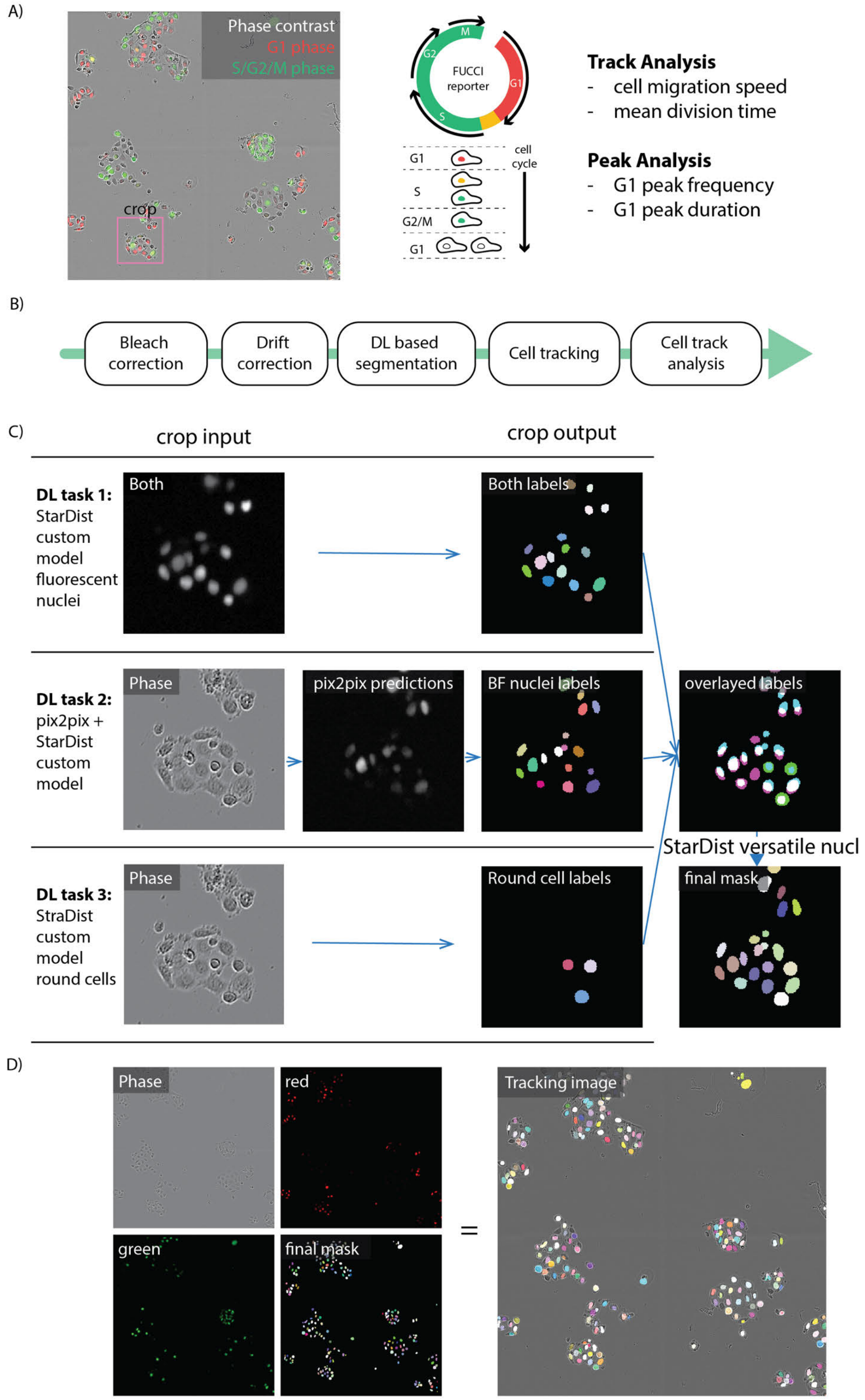

A)
Phase contrast
G1 phase
S/G2/M phase
crop
M
G2
FUCCI reporter
G1
S
G1
S
G2/M
G1
cell cycle
Track Analysis
- cell migration speed
- mean division time
Peak Analysis
- G1 peak frequency
- G1 peak duration
B)
Bleach correction
Drift correction
DL based segmentation
Cell tracking
Cell track analysis
C)
crop input
crop output
DL task 1: StarDist custom model fluorescent nuclei
Both
Both labels
DL task 2: pix2pix + StarDist custom model
Phase
pix2pix predictions
BF nuclei labels
overlayed labels
StarDist versatile nucl
final mask
DL task 3: StraDist custom model round cells
Phase
Round cell labels
D)
Phase
red
green
final mask
=
Tracking image

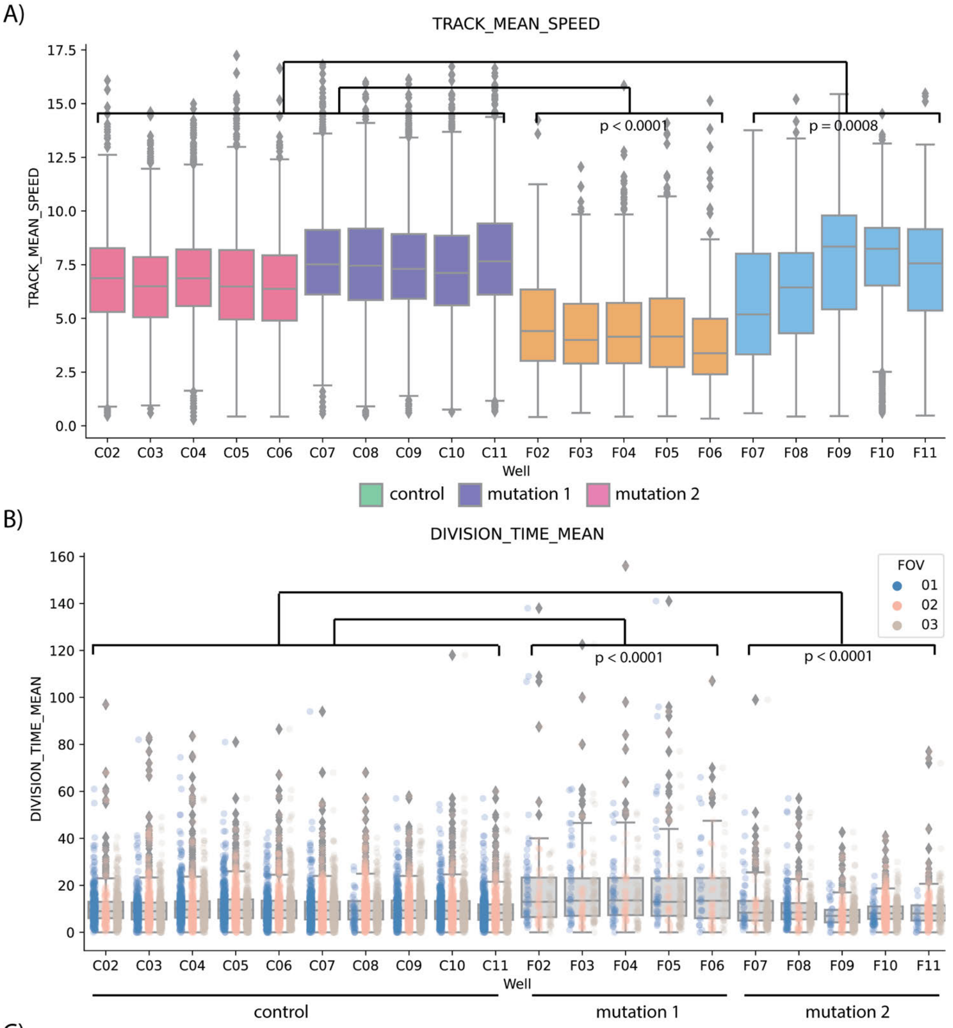
Figure 27: Image analysis pipeline for PC-9 cells expressing a FUCCI reporter. A) CRIPR/CAS9 mediated knock-out PC-9 cells were stained using the FUCCI reporter, which expresses a different combination of fluorescent proteins depending on the cell cycle phase. Cells were images for 6 days using the Incucyte with red and green channels and brightfield. B) Overview of the FUCCI pipeline. First, bleaching in the fluorescent images was removed, followed by drift removal. A DL-based segmentation pipeline was applied to the cells to detect their nuclei, and the generated labels were tracked and the red signal in them was quantified using TrackMate. The tracking results were analyzed using a custom-made Google Colaboratory notebook. C) The cell nuclei were detected using four DL approaches. First, the combined image from the fluorescent channels was segmented using a custom-made StarDist model (M12). Second, a pix2pix model was trained to detect nuclei from brightfield images (M13), and from the predictions, the cell nuclei were detected using the same StarDist model as above (M12). Third, the rounded cells from the brightfield images were segmented using another custom-made StarDist model (M14). The output label images from these three steps were combined, and from those images, the final mask was created using the StarDist Versatile nuclei model (M10). D) The original BF, green, and red images were combined with the final mask to create a tracking image that was then used for tracking.

4.2 Migration and division times are altered in cells harboring mutation 1

To learn how the used drug A affected the cell migration speed and the duration of cell proliferation we tracked the labels of the tracking images (Figure 27D) and analyzed the track mean speed and the time that it took for cells to divide. For this analysis, we used the TrackMate tracks output, in which we allowed track splitting. To compare the conditions, we first calculated the p-values between the conditions and the control. As the number of data points was over 1 million, the p-values were extremely low. Therefore, to better understand the degree of variance between the groups, we calculated Cohen’s d value, which quantifies the effect size between groups. A notable advantage of Cohen’s d value is its insensitivity to effect sizes. The closer the absolute value of Cohen's d is to 1, the larger the effect is.

We found that the cell migration speed was reduced in the cells harboring mutation 1 (Cohen’s d = 1.06) and that mutation 2 did not impact cell migration speed (Cohen’s d = 0.05) (Figure 28A, C).

To understand if the mutations affected the cell division, we plotted the division times from the TrackMate tracks output (Figure 28B). We found that mutation 1 slows down the division time (Cohen’s d = -0.92, Figure 28C), indicating that these cells are affected by the drug. Mutation 2 showed a small difference (Cohen’s d = 0.16, Figure 28C), indicating that PC-9 cells harboring mutation 1 can divide slightly faster than the non-mutated cells. When plotting for the division time, I also wanted to see how well the results of the three fields of view correspond to each other and found that the results within all three fields of view were highly similar, ensuring us that the analysis pipeline gives consistent results (Figure 28B).



C)

Cohen's d	Track speed	Division time
control vs mutation 1	1.06222	-0.91874
control vs mutation 2	0.05305	0.16244

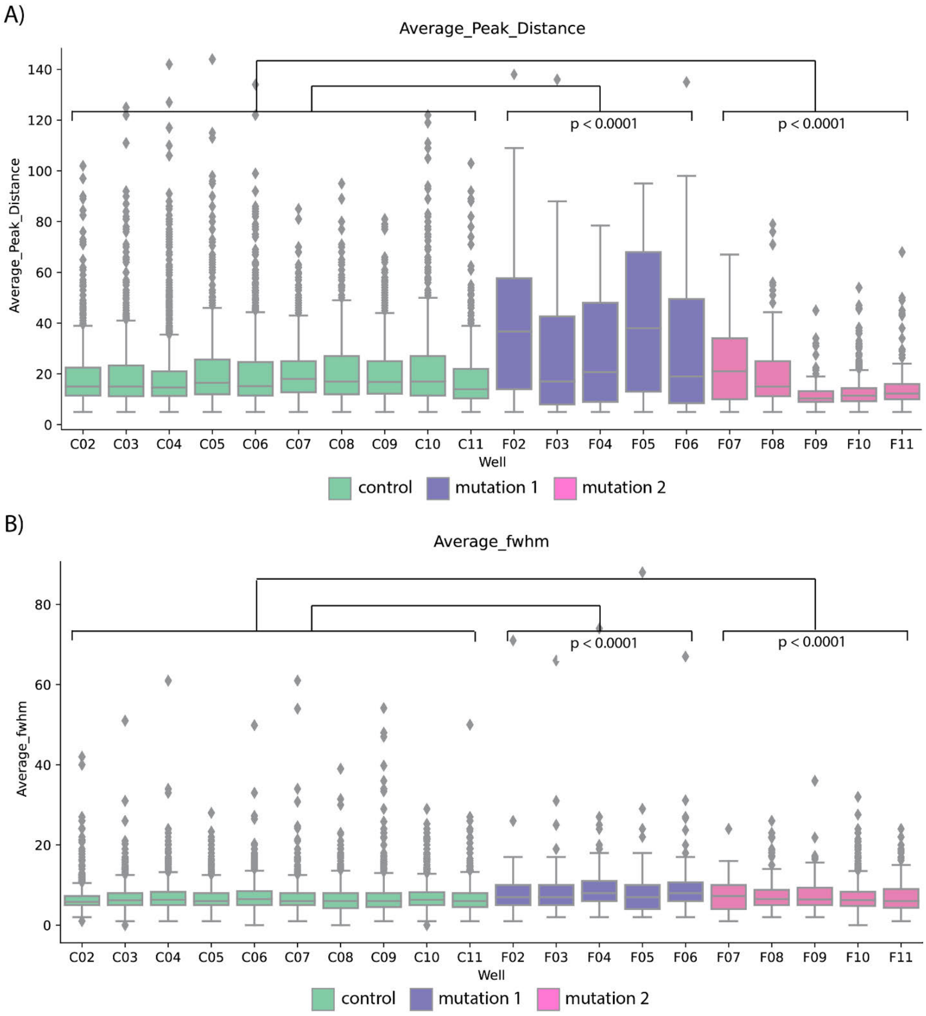
Figure 28: Change in cell migration speed and division times. Track mean speed (A) and mean division times (B) of the tracked cells were plotted using a custom-made Google Colaboratory notebook. C) Cohen's d values for the track speed and track division times were calculated between the control and the mutation groups.

4.3 Cell cycle is delayed and slower in cells harboring mutation 1

We performed a peak analysis to further investigate how the mutations affect the cell cycle under drug treatment. In this analysis, we first quantified the G1 signal in the segmented nuclei over time. As the G1 signal oscillates during the cell cycle, we could use the G1 signal peaks to measure the duration of a complete cell cycle and the duration of the G1 phase. We extracted this information by using the scipy.signal.find_peaks python function (Virtanen et al., 2020). To measure the cell cycle duration, we explored the frequency of the red peaks during tracks to measure how often the cells divide. We found that the average peak distance is longer in the cells harboring mutation 1 (Cohen's d = -0.72, Figure 29A, C), indicating that the cell cycle occurs less frequently. Cells harboring mutation 2 show slightly shorter cell cycle times compared to the control, indicating that they divide faster (Cohen's d = 0.5, Figure 29A, C).

To learn about the duration of the G1 phase, we measured the FWHM of the G1 peaks. We found that the duration of the G1 phase was increased in the mutation 1 cells (Cohen's d = -0.49, Figure 29B, C), and with mutation 2, we saw no change (Cohen's d = -0.09, Figure 29B, C).

Taken together, these results indicate that cells with mutation 1 are more affected by the drug treatment, whereas the mutation 2 cells show resistance to the tested drug.



C)

Cohen's d	Peak distance	fwhm
control vs mutation 1	-0.75044	-0.49080
control vs mutation 2	0.50163	-0.09802

Figure 29: Cell cycle frequency and duration. Average distances (A) and the average duration G1 peaks (B) of the tracked cells were plotted using a custom-made Google Colaboratory notebook. C) Cohen's d values for the G1 peak distances and the peak duration (fwhm) calculated between the control and the mutation groups.

Discussion

Live cell image analysis is challenging, but fun. Despite multiple efforts in the past decades, including the enhancement of instrumentation stability, improvements in detectors and illumination pathways, and the development of new dyes, live cell imaging data still drifts, the dyes still bleach, and the sample health is often compromised. As live cell imaging is an essential part of modern cell biology research, addressing these challenges is crucial for extracting meaningful data from live microscopy experiments. Computational bioimage processing and analysis emerge as promising solutions to solve many of these issues.

The quick development of computational resources has led to the development of more advanced bioimage analysis methods. Despite the rapid creation of new analysis tools, they frequently encounter usability challenges. A significant number of these tools remain inaccessible to life scientist due to their demand for advanced computational skills, which are not typically covered in a standard life science education. Moreover, there are limitations in the proper reporting, documentation, and usability through interactive GUIs, consequently limiting their applicability in life sciences.

My thesis's main objective was to develop user-friendly tools for analyzing live cell imaging data to meet our lab's analysis needs. This resulted in three tools: 1) Fast4DReg, a drift correction tool for 4D live cell imaging data, 2) TrackMate v7, a sophisticated cell tracking tool, and 3) DL4MicEverywhere, a platform for flexible training and deployment of DL models across diverse computational environments. My contribution to each project varied from tasks such as software development, creating datasets and documentation, alpha testing, and addressing issues with the tools. Additionally, I demonstrated the practicality of the developed tools through two case examples where I focused on applying these (and other) tools to create automated image analysis workflows to study cancer biology. To reflect on my diverse experiences throughout these projects, I organized the discussion into key aspects of software development, with a particular emphasis on the user-friendliness of the software.

1 My experience as a software developer

To guide the development of usable image analysis tools, Carpenter et al. defined a list of requirements for usable bioimage analysis software (Carpenter et al., 2012), which included aspects such as user and developer-friendliness, interoperability, modularity and validation of the software. These guidelines are further defined by Levet et al. in the context of software development, especially for the use of life scientists (Levet et al., 2021).

Throughout the development of Fast4DReg, TrackMate v7, and DL4MicEverywhere, I drew inspiration from these publications, making my best effort to fulfill the listed requirements. My particular focus was on incorporating user-friendly features from their work into my image analysis tools.

While in many cases, software development is a lonely process where a single person acts as the developer and a user (Levet et al., 2021), the development of Fast4DReg, where I was the main developer, was a dynamic and interactive process that involved weekly discussions with my supervisor and our collaborators. The work was further supported by the training materials provided by NEUBIAS training schools (Martins et al., 2021) and the image.sc forum (https://forum.image.sc/, 13.03.2024), providing me with the basic knowledge and support to improve the tool. Fellow researchers and the bioimage community can support the development process by providing feedback on the available functionalities, providing examples, and writing documentation (Levet et al., 2021). In the Fast4DReg development, most troubleshooting was conducted by fellow researchers in urgent need of the tool, and valuable information, such as software bugs and illogical order of commands, was acquired to improve the tool. In the case of the development of TrackMate and DL4MicEverywhere, my role was to contribute from a biologist's point of view, providing insights into the needs of biologists to extract useful information from image data.

The development of Fast4DReg started with a test ImageJ macro, where our collaborators helped us to create an intensity projection-based 3D adaptation of a similar method they used for the correction of 2D data (Laine et al., 2019). Although the code worked for our 3D data, it had not yet been validated or widely tested, and it lacked several essential features, such as GUI, before it could be considered a usable image analysis tool. As a result, we developed a drift correction tool for 4D image datasets, which we called Fast4DReg (Pylvänäinen et al., 2023c). This tool can correct drifting or align misaligned channels in 3D image data using intensity projections and cross-correlation. Projection-based 3D drift correction methods have also been reported before in super-resolution microscopy (Mlodzianoski et al., 2011; Huang et al., 2008).

The first version of Fast4DReg was only suited for 3D data. Based on the comments of fellow researchers participating in the Fast4DReg testing phase, we improved the tool to allow the user to use the same tool for both 2D and 3D drift correction. Fast4DReg automatically reads the image dimension and skips the step where it creates intensity projections if a 2D image is detected. In some cases, it is useful to allow selective correction of axial or lateral drift instead of both. We included activating tick boxes in the user interface to enable users to select between lateral, axial correction, or both.

The need for batch processing of multiple files arose from our group members, who wanted to apply the same correction parameters for multiple images. Inspired by this, we further improved Fast4DReg to process multiple image files in batch and export a settings file that could be applied to other datasets with similar drift. To do this, we incorporated a code that saves the correction setting and path to the generated drift tables in a csv file, which can then be applied to another dataset. To avoid overwriting the drift correction results, I wanted to implement an option where a user can give a unique identifier to the correction parameters used. This feature prevents accidental overwrites and allows retracing steps back to the specific settings employed, enhancing reproducibility and traceability, as identified by (Carpenter et al., 2012; Aaron and Chew, 2021). These additions significantly improved the functionality and user-friendliness of the Fast4DReg, making it more versatile and applicable to a broader range of research problems.

During the development of TrackMate v7, a similar need for batch processing arose from our work with multiple live-cell imaging videos where we aimed to apply identical tracking parameters across the datasets. We communicated this need to the leading developer of TrackMate, who implemented the TrackMate batcher tool, which allows batch processing of multiple image files and analyses without the need to code (Ershov et al., 2022). The TrackMate batcher has become the go-to tool for our image analysis pipelines and was used in both unpublished works in this thesis.

A recent effort from the Quality Assessment and Reproducibility for Instruments & Images in Light Microscopy (Quarep-Limi) community summarizes best practices for reporting image analysis workflows (Schmied et al., 2024). Quarep-Limi and (Jamali et al., 2021) recommended that all source code be stored in version-controlled repositories, such as GitHub, to ensure the workflow and code availability, and the code at the time of the original publication to be stored in a data repository, such as Zenodo. We followed these recommendations for all three developed software.

2 User-friendly image analysis software

Life science training usually includes courses in biology fundamentals rather than focusing on learning computational skills. On the contrary, image analysis development is commonly driven by computer scientists. This, combined with the fast development of the image analysis field, has created a gap between tool developers and life scientists, introducing problems in communication and tool usability (Schlaeppi et al., 2022). To successfully navigate in this field requires knowledge of biology, microscopy, and image analysis methods - skills that may not be easily mastered by an individual researcher alone. Therefore, collaboration and common language between software developers and life scientists is essential (Schlaeppi et al., 2022). The search for usable software for biologists has pushed the development of image analysis software in the past decade (Carpenter et al., 2012). This has been improved by developing ontologies (Lindblad et al., 2020) to improve communication and development of guidelines for best practices in publishing image analysis workflows (Schmied et al., 2024), improved documentation, and FAIR sharing image data (Kemmer et al., 2023).

Drawing from existing literature and personal experiences with my favorite user-friendly image analysis tools, such as Cellpose 2.0 (Pachitariu and Stringer, 2022) and ZeroCostDL4Mic (von Chamier et al., 2021), I established my own criteria for the user-friendliness of software, prioritizing their fulfillment for Fast4DReg, TrackMate and DL4MicEverywhere. These included 1) easy and intuitive installation to ensure software adoption and dissemination, 2) intuitive GUI with a logical order of functions and a minimum number of steps required to complete a task, and 3) detailed documentation to minimize the learning curve. These requirements align with earlier published recommendations (Levet et al., 2021) and further endorsed by a community survey where life scientists were asked how software creators could make image analysis more accessible (Jamali et al., 2021).

My first criterion for developing user-friendly software was ensuring that the analysis tools were easily accessible through simple installation. Both Fast4DReg (Pylvänäinen et al., 2023c) and TrackMate (Ershov et al., 2022) are distributed as Fiji plugins (Schindelin et al., 2012). Fiji was selected as the platform, as it is amongst other point-and-click type software most used by life scientists (Jamali et al., 2021). We set up a Fiji update site to ensure easy installation, software adoption, and dissemination of Fast4DReg and TrackMate. Fiji's update sites automatically check for updates, making it easy to adopt any changes made to the code and further enhancing user-friendliness.

During the initial stages of DL4MicEverywhere development, the installation required some manual steps, and the software was launched using the terminal. Subsequently, the installation process was streamlined by packaging all required dependencies into a single Docker container and generating executable files compatible with various operating systems. These improvements allow users to launch the software effortlessly by simply double-clicking on the executable file, thereby reducing the initial hurdles and facilitating a smoother start to using the software.

Secondly, I wanted to create user-friendly GUIs to ensure an easy user experience. A well-designed GUI that allows users to interact with the software platform can maximize the impact and usability of image analysis tools (Levet et al., 2021). Without a GUI, the code needs to be manually adjusted in a situation where any parameters need to be tweaked. Modifying the code without computational skills can be difficult, and several versions of the code make the workflow irreproducible. In the case of Fast4DReg, the GUI needed to be built from scratch. I created a user-friendly GUI that allows the user to tweak any required parameters without coding experience, which can be saved in a separate settings file. This helps when retracing steps back to the used settings. In the case of TrackMate, the previous version (Tinevez et al., 2017) already had a very intuitive GUI, which was extended to provide the new functionalities of the new TrackMate v7. Like TrackMate, DL4MicEverywhere had a basic user interface when I joined the project. In the first version, the user needed to navigate to the software repository through the interface to find the notebook to run. This was later improved so that the notebook paths were incorporated into the GUI as drop-down menus for easier usage.

Lastly, creating detailed documentation is one of the best ways to minimize the learning curve and improve the adoption and dissemination of software (Levet et al., 2021). It can be a time-saver when the documentation indicates whether the software is suited to handle particular image file formats, imaging modalities or analyses. It additionally aids both users and developers in comprehending, replicating, and even expanding the software for analyses performed on their biological images. Adequate documentation ensures transparency, reproducibility, and collaboration within the research community (Carpenter et al., 2012). The documentation for the three presented tools is extensive and made available through the original publications and the publication repositories. Further user support has been provided through the image.sc forum (for example: https://forum.image.sc/t/regarding-error-with-the-stack-in-fast4dreg-drift-tool/92084, 07.03.2024).

In addition to my user-friendliness criteria, some interoperability and modularity features deserve to be mentioned in the context of the developed software's user-friendliness. The modularity of software refers to the design and organization of software that promotes the division of its functions into independent modules, which can then be individually called (Carpenter et al., 2012). In the case of Fast4DReg, we extracted the original 2D drift correction function also used in NanoJ (Laine et al., 2019) as an independent macro-recordable function that can later be integrated into a new piece of software, allowing researchers to create their workflows for their specific purposes. I also demonstrate interoperability in this thesis in both unpublished works, where multiple image processing steps were integrated into a single notebook that outputs the desired image for further tracking experiments and analyses. Using such compiled notebooks enhances workflow efficiency and reproducibility. They also allow researchers without coding experience to apply complex analysis pipelines to their data without continuous support from image analysis experts.

Interoperability of software, on the other hand, refers to the ability of different software systems to work together effectively regardless of their origin (Carpenter et al., 2012). This is especially important when moving from one software to another (Dobson Ellen et al., 2021), as is required when using tools developed in this thesis for live cell imaging studies. To ensure that Fast4DReg can read multiple microscopy file formats, it uses Bio-Formats (Linkert et al., 2010) to load images. Interoperability is also a central feature of TrackMate, which can make use of commonly used ML and DL segmentation models generated using, for example, the DL4MicEverywhere platform (Hidalgo-Cenalmor et al., 2023), ilastik (Berg et al., 2019) or Weka (Arganda-Carreras et al., 2017) and use them directly in TrackMate for object detection. This feature significantly enhances the seamless integration of DL4MicEverywhere and TrackMate, facilitating the user-friendly incorporation of DL and ML models into tracking pipelines.

3 Benchmarking

Image analysis software benchmarking is a process where predetermined ground truth datasets and well-defined performance metrics are used to assess the performance of a particular software and compare it to other similar software. In systematic benchmarking, the software is tested using different operating systems, and aspects like reproducibility, speed and functionality are tested. The goal of software benchmarking is to provide a basis for performance analysis, enabling users to make informed decisions about selecting, optimizing, or improving the software (Kozubek, 2016). Additionally, benchmarking promotes a culture of continuous improvement by helping in software optimization and guiding in finding issues in the software.

One of the main assumptions of benchmarking is that the software being tested should be able to produce expected results consistently and reproducibly under the same conditions. To test this assumption, example datasets that resemble real experiments need to be provided along with well-defined performance metrics (Carpenter et al., 2012). A great example is the Cell Tracking Challenge (Maška et al., 2023), which provides annotated datasets evaluation of segmentation-and-tracking and segmentation-only workflows and performance metrics. We implemented the performance metrics in the TrackMate Helper module and used some of these datasets to assess the TrackMate v7 performance. Another well-annotated test dataset, particularly useful for live cell and DL benchmarking, is the LIVECell dataset, which provides a high-quality, manually annotated and expert-validated datasets of phase-contrast images consisting of over 1.6 million cells from a diverse set of cell morphologies and culture densities (Edlund et al., 2021). This dataset was used to showcase the usability of Cellpose in the DL4MicEverywhere project. The Broad Bioimage Benchmark Collection is worth mentioning. It provides a publicly available collection for testing and validating automated image-analysis algorithms that are particularly useful for high-throughput experiments (Ljosa et al., 2012).

Benchmarking Fast4DReg against other tools was a central part of my thesis work. For benchmarking, I chose two commonly used 3D drift correction tools, Correct 3D Drift (Parslow et al., 2014) and Fijiyama (Fernandez and Moisy, 2021). These two were selected because we used them previously in the lab, and they were openly available through Fiji. Unfortunately, we could not find any available datasets that could be used to benchmark Fast4DReg. To overcome this problem, we created a synthetic drifting dataset with a known amount of drift between frames. This dataset consisted of 25 copies of a 3D image of a single cell. We used this dataset to evaluate the drift correction performance and reproducibility between different operating systems and compared these to other similar tools. We also used a longer video to benchmark the speed and stability of these three software. We evaluated the drift correction performance by comparing different image similarity metrics between consecutive frames before and after drift correction. We showed that Fast4DReg outperforms the other two selected tools in drift correction ability and in speed.

3.1 Fast4DReg usability compared to Correct 3D Drift and Fijiyama

Although we showed that Fast4DReg outperforms Correct 3D Drift and Fijiyama in drift correction performance and speed, each method has unique features. While Correct 3D Drift did not reach as good drift correction results as Fast4DReg, it still had some benefits over Fast4DReg. First, Correct 3D Drift allows input of multichannel 4D data, a feature still missing from Fast4DReg. Adding this to Fast4DReg would make the drift correction process smoother. I considered adding this function to Fast4DReg, but finding the best correction parameters requires optimization; one channel correction is usually enough to assess when the correction ability is sufficient. Then, the correction can be added to the second channel using the setting file generated during optimization. Second, Correct 3D Drift incorporates an edge enhancement option to improve the drift detection.

This very specific operation can work with certain datasets, but denoising or bleach correction algorithms can provide better image enhancement. Pre-processing options were not added to Fast4DReg because I did not want to restrict users from using the selected pre-processing methods. Third, Correct 3D Drift allows the selection of the top and bottom slices to be considered when considering the drift correction. This feature can be handy when performing drift correction directly on 3D data, but as Fast4DReg estimates the drift using intensity projections, the possibility of discarding top and bottom images becomes futile when using maximum intensity projections. In the case of average intensity projections, this feature could prove useful. Lastly, Correct 3D Drift allows adding an ROI to the image, which will be used as a reference for the correction. This feature is extremely useful when using fiducial markers for feature matching. In Fast4DReg, it is possible to add an ROI in x-y-direction, but as this feature still needs to be properly tested, it is not currently included in the documentation. If an x-y ROI is created, it corrects the lateral drift according to the contents of the ROI. I failed to adopt the code so that I could also consider the same ROI for axial correction; despite multiple tries, the axial correction still considered the whole x- or y-projection for estimating the drift. I would like to develop Fast4DReg in the future so that it can correct both lateral and axial drift correction considering a user selected ROI.

Benchmarking with Fijiyama was not a great experience. Fijiyama was not initially developed for time-lapse images, and therefore, it required splitting the image into individual frames before correction, and the frames needed to be named in a specific way. Already, this step reduced the usability of the tool. The Fijiyama user interface is complicated and requires multiple clicks to run even the automatic correction algorithms. Additionally, once the correction pipeline is running, its execution times are very long, and the resulting correction does not reach the performance of Fast4DReg. From my experience, I would not recommend using Fijiyama for 3D drift correction tasks.

Neither Correct 3D Drift nor Fijiyama supports batch processing, a feature we greatly value. Images need to be corrected one by one, and the acquired correction cannot be applied to another dataset. The parameters must be listed manually or through screenshots, which does not align with the recommended image analysis reporting and reproducibility (Aaron and Chew, 2021).

4 Data availability

Bioimage analysis has reached the era of big data, introducing several difficulties in the consistency of data handling, analysis, and management practices. Poor documentation and disorganized management are currently hindering the usage and sharing of image data, DL models, analysis tools, and workflows to their full potential (Kemmer et al., 2023). The imaging community has joined forces and developed FAIR principles to address the challenges in making image data available. These principles provide guidelines to improve the **F**indability, **A**ccessibility, **I**nteroperability, and **R**eusability (=FAIR) of data so that it can be shared, lead to new discoveries and used for the development of new tools (Wilkinson et al., 2016).

To ensure data availability, I followed the FAIR principles of image data publishing (Kemmer et al., 2023). We looked into available data storage options for data, such as OMERO (Allan et al., 2012) and the produced DL models, such as the BioImage Model Zoo (Ouyang et al., 2022), but finally, due to previous experiences and a less cumbersome uploading process, we decided to store all image data used in the publications in properly annotated dataset-specific Zenodo repositories (https://zenodo.org/, 11.03.2024). To ensure findability, the repositories, or links to existing datasets elsewhere, were included to the original publication and the GitHub repositories. From the Zenodo repositories, it is possible to download the images and reuse them in any other software.

5 Limitations of Fast4DReg, TrackMate and DL4MicEverywhere, and future prospects

Software development is never fully complete; it is an ongoing, iterative process involving numerous improvements and adaptations to address emerging challenges. The continuous evolution of the software ensures its relevance alongside advancements in bioimage analysis. Although the tools I presented in this thesis are all openly available for the bioimaging community, they still have their limitations.

We showed that Fast4DReg has superior performance in correcting drift in time-lapse image data compared to two other selected tools. Additionally, it has been well accepted by the imaging community, some labs have adapted it as part of their image analysis pipelines (10046 Fiji update site downloads, https://imagej.net/update-sites/stats, 04.03.2024), and it has one pre-print citation (Rakhymzhan et al., 2023). It has even been proposed as one of the go-to drift correction tools by Andor https://andor.oxinst.com/learning/view/article/drift-and-drift-correction-in-time-lapse-microscopy, 06.03.2024). Despite the performance, Fast4DReg still has several limitations. First, Fast4DReg can only perform translations when correcting a dataset and rotation, scaling or shearing transformations are not available, corrections which, for example, Fijiyama can offer. In most cases, time-lapse data does not suffer from rotation, scaling, or shearing, so we did not try to accommodate these at this point. Second, Fast4DReg expects the frames to have structural similarity between them. This limitation became evident in unpublished work 2, where we created a workflow involving drift correction. Initially, I attempted to employ Fast4DReg for drift removal, but as the intensities of the nuclei varied throughout the video, I was not able to remove the drift. Therefore, I chose StackReg (Thevenaz et al., 1998) for drift correction, successfully achieving the necessary correction. Finally, in the case of channel correction, Fast4DReg requires a calibration slide to estimate the amount of drift. Channel alignment conducted on an original multi-channel image may yield for example in false colocalization, as Fast4DReg tries to register regions with the highest intensities.

After publishing Fast4DReg, it was improved by the community effort to be compatible with headless execution (https://github.com/guijacquemet/Fast4DReg/pull/6, 06.03.2024). Additionally, I envision the following aspects to improve Fast4DReg. First, Fast4DReg does not yet support cropping functions, for example, the selection of a fiducial bead in 3D. Second, as Fast4DReg was developed to work intensity projections, it would be interesting to assess the suitability of using cross-correlation directly to 3D data. Although, in some cases, this might produce better correction results, it will likely increase processing times. Third, Fast4DReg was originally built in the core functions of NanoJ (Laine et al., 2019), which were later implemented in Python through NanoPyx (Saraiva et al., 2023). NanoPyx does not yet incorporate the 3D drift correction, and there are plans to include Fast4DReg as part of NanoPyx.

TrackMate has gained popularity as a tracking tool, and the latest iteration, TrackMate v7, represents a significant leap forward in cell tracking analysis. TrackMate v7 has been cited 318 times (07.04.2024), which can be considered a direct measure of the usability of the software. Despite its widespread recognition and use, the new TrackMate still has limitations. While it supports 3D tracking, it cannot currently track the shapes of 3D objects. Recognizing this gap, the next version of TrackMate is already being planned. The new version will incorporate this crucial feature to enhance its versatility and support 3D analysis.

We recently published DL4MicEverywhere (preprint by Hidalgo-Cenalmor et al., 2023, accepted for publication in Nature Methods in March 2024), which is a major improvement of the popular ZeroCostDL4Mic framework (von Chamier et al., 2021). ZeroCostDL4Mic democratized DL to the use of life scientists without coding skills, and it has been cited 379 times (12.03.2024), demonstrating that it has been well incorporated into life sciences research. DL4MicEverywhere broadens the applicability of the ZeroCostDL4Mic by enabling the deployment of DL for microscopy data across diverse computational environments. This was accomplished by using Docker containers that include all necessary software dependencies and an intuitive GUI. Installation of DL4MicEverywhere is made easy through simple step executable files. DL4MicEverywhere is still in the development phase, and therefore some bugs remain, for example, when running DL4MicEverywhere on the Windows operating system. Additionally, each notebook generates its own Docker image, which require large amounts of space and can quickly clutter the computer's hard drive. Although there are relatively simple ways to clear unnecessary container images from a computer, these need the use of the command line, which is not an evident skill for life scientists. We are looking into modularization to combine Docker images of some notebooks that share similar functions to reduce space requirements.

I envision that all three tools, Fast4DReg, TrackMate, and DL4MicEverywhere, will collectively contribute to advancing the field of bioimage analysis by providing robust and versatile solutions for spatial and temporal analysis, tracking, and DL applications, as already shown in the unpublished scientific work in this thesis. These software tools will provide life scientists with a user-friendly approach to analyzing their live-cell imaging data, enabled by comprehensive documentation and open-source frameworks, fostering a collaborative and comprehensive environment where users can easily address diverse biological questions and challenges in the field of microscopy and image analysis. Through continuous development, community engagement, and following the best practices, these tools will become integral components in the toolkit of live cell image analysis, ultimately accelerating scientific discoveries and innovations in the life sciences.

Concluding remarks

Live cell imaging plays a crucial role when observing multiple biological processes, such as normal development, wound healing, and cancer metastasis, and when combined with modern bioimage analysis tools, visualization and quantification of life become possible. Indeed, bioimage analysis has become an integral part of live cell imaging experiments, providing tools for the extraction of meaningful information from live cell imaging data.

In this thesis, we developed three user-friendly tools for processing and analyzing live cell imaging data. Fast4DReg (I) allows fast and reproducible drift correction and channel alignment for 4D microscopy datasets. DL4MicEverywhere (III) provides a non-code environment for the training and deployment of DL models on multiple computational platforms. TrackMate (II) allows user-friendly tracking and track analysis. Together, these tools create a seamless image analysis workflow for live cell imaging (Figure 30). I further showcased their usability using two cancer biology case examples (unpublished 1 and unpublished 2). These software tools, together with their comprehensive documentation and open-source frameworks, provide life scientists with a user-friendly and intuitive approach to live-cell imaging analysis, allowing new discoveries in life sciences.

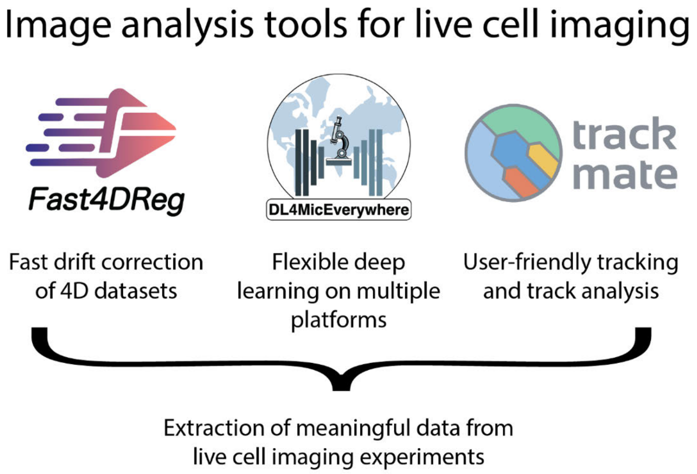


Figure 30: Fast4DReg, DL4MicEverywhere and TrackMate image analysis tools for live cell imaging. These tools provide an easy and user-friendly combination of tools to extract meaningful data from live cell imaging experiments.

Acknowledgments

My PhD journey has been eventful. While the typical PhD path is relatively linear, I chose the scenic route. This route took me through research labs, imaging core facilities and imaging infrastructures, the common nominator still always being imaging. Along the way, I have met fantastic people who all had an important role during my PhD journey and whom I want to now acknowledge.

Let's start.

First, Guillaume. I am beyond grateful for the opportunity to finish my PhD journey in your lab. I admire your passion for science and your commitment to your students. Although my PhD arrangements during the first two years were unusual, we still managed to get me to graduate almost according to the original schedule! I am so grateful for all your guidance and all the opportunities you have offered me along the way. Thank you also for all your wise words and for bringing my feet back on the ground when I lost hope. Thanks for your open-door policy; you always have a minute to chat. These minutes have been valuable for me.

Diana, you have been my mentor for more than ten years. I am grateful for all the support you have given me during the BIMA years as a (co-)supervisor for my licentiate and PhD degrees and my personal life. You have a magical skill always to see the positive in the world, a skill I hope to master one day.

Pasi, thank you for all your support during my TBI years and after. Thanks for your guidance in image analysis and especially in navigating through the jungle of research infrastructures. Thank you for giving me the opportunity to start building IDAT; it significantly clarified my ambitions and helped me pave my way to the future. Thank you also for allowing me to pursue my scientific goals during my TBI years. This arrangement was unique and gave me broad experience, which has given me so much.

I wish to thank Lassi Paavolainen and Pekka Ruusuvuori, for reviewing my thesis and for your constructive comments, which significantly improved my work. I am grateful for Carolina Wählby for accepting the invitation to be my opponent. I warmly remember our fantastic experience in Iceland, exploring Sky Lagoon and the hike to witness an active volcano. During our time together, we engaged in discussions about life and career. It was during these conversations that I decided you would be the perfect opponent for me.

Special thanks go to Pasi Kankaanpää and Anna Aalto for helping me with the Swedish abstract. The same goes for Lilli Pukka and Laura Mairinoja with the Finnish abstract.

No research is done alone. Collaboration makes my field worth pursuing, and it makes the work fun. I have been fortunate to work with some of my image analysis superheroes along my journey! I want to acknowledge all my co-authors who have contributed to this thesis for their efforts and contributions: Romain F. Laine, Bruno M. S. Saraiva, Sujan Ghimire, Gautier Follain, Ricardo Henriques, Dmitry Ershov, Minh-Son Phan, Stéphane U. Rigaud, Laure Le Blanc, Arthur Charles-Orszag, James R. W. Conway, Romain F. Laine, Nathan H. Roy, Daria Bonazzi, Guillaume Duménil, Jean-Yves Tinevez, Iván Hidalgo-Cenalmor, Mariana G Ferreira, Craig T Russell, Ignacio Arganda-Carreras, AI4Life Consortium, Estibaliz Gómez-de-Mariscal, Kari Kurppa and Zejia Song.

I also want to thank all the members of the Cell Migration lab. We have a fantastic crew and rock in everything we do! Thanks, Marie, for being the prepared one; you always have tissues, painkillers, or whatever is needed. Thanks also to the usual lunch team, Ana, Monika, Sujan, Hanna, Johanna, and Diana, for all the lunch discussions; I learned so much from you. Special thanks to Hanna and Gautier for helping me improve my thesis. And finally, thanks, everyone, for listening to my whining, especially during the last weeks of my thesis writing. I must have been a pain :D

Turku BioImaging and Euro-BioImaging have played a significant role during my PhD journey. First, I want to acknowledge John Eriksson. Not only did you create a fantastic imaging universe here in Turku, but you also taught me about research and infrastructure building. I also want to acknowledge Pasi for working hard to build IDAT so that others and I can focus on doing what we love the most. Junel, thank you for teaching me so much about the computational side of bioimage analysis and all our great discussions. I also want to thank the rest of the Turku BioImaging and Euro-BioImaging teams, Dado, Anna, Irina, Tiina, Solveig, Camilo, Anting and Ilari, for sharing your insights into the field of imaging and for your friendship throughout these years.

Laura and Elnaz, my image data goddesses and my whining support group! My PhD journey would have been much more boring without you. Thanks for the fantastic trips we did together and all the wine we consumed on the way. I don't know if we managed to make the world a better place, but at least we found our places in it. I also want to acknowledge other fellow PhD students, Oona and Amna, for sharing your PhD experience with me.

My life outside of my work has been eventful during the PhD journey. I have two fantastic boys, Eikka and Roope, who keep reminding me about life's purpose. I also want to thank you, Harri, for your love and support. Thank you for keeping the house running and taking care of Eikka and Roope, especially during the last days of my writing. Thanks also for letting me crash

on the couch and encouraging me to remember to also take care of myself. I also want to acknowledge my extended family, Leena and Matti, for their help and support.

None of this work would have been possible without the support of my amazing parents, Pekka and Boguslawa. Thanks for teaching me about the world, different cultures and life. I would not be who I am today without you allowing me to try my wings out there, and thanks for catching me when I fell. Your endless support and understanding (even in cases where we disagree) have made me a better person. This thesis is dedicated to you. I also want to thank my brother, Pauli, for being the voice of reason throughout these years. I have the best family, and I love you all so much.

I would also like to acknowledge Varpu, Sonia, Kate and Nina. You are my oldest friends and know me better than everyone. I have been privileged to have you in my life during the good and the difficult times. I hope our friendships last until the end <3.

This thesis was carried out at the Faculty of Science and Engineering at the Department of Cell Biology at Åbo Akademi University and supported by the Turku Doctoral Network in Molecular Biosciences. I want to acknowledge Annika Meinander and Tiina Salminen for their work as heads of subject and for providing a friendly and inspiring work environment. This research was supported by the InFLAMES Flagship Programme of the Academy of Finland (decision numbers: 337531, 357911).

Turku 15.03.2024

Joanna

References

Aaron, J., and T.-L. Chew. 2021. A guide to accurate reporting in digital image processing – can anyone reproduce your quantitative analysis? *J. Cell Sci.* 134. doi:10.1242/jcs.254151.

Abbe, E. 1873. Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung. Arch. Für Mikrosk. Anat. 9:413–468. doi:10.1007/BF02956173.

Ahlers, J., D. Althviz Moré, O. Amsalem, A. Anderson, G. Bokota, P. Boone, J. Bragantini, G. Buckley, A. Burt, M. Bussonnier, A. Can Solak, C. Caporal, D. Doncila Pop, K. Evans, J. Freeman, L. Gaifas, C. Gohlke, K. Gunalan, H. Har-Gil, M. Harfouche, K.I.S. Harrington, V. Hilsenstein, K. Hutchings, T. Lambert, J. Lauer, G. Lichtner, Z. Liu, L. Liu, A. Lowe, L. Marconato, S. Martin, A. McGovern, L. Migas, N. Miller, H. Muñoz, J.-H. Müller, C. Nauroth-Kreß, J. Nunez-Iglesias, C. Pape, K. Pevey, G. Peña-Castellanos, A. Pierré, J. Rodríguez-Guerra, D. Ross, L. Royer, C.T. Russell, G. Selzer, P. Smith, P. Sobolewski, K. Sofiiuk, N. Sofroniew, D. Stansby, A. Sweet, W.-M. Vierdag, P. Wadhwa, M. Weber Mendonça, J. Windhager, P. Winston, and K. Yamauchi. 2023. napari: a multi-dimensional image viewer for Python. doi:10.5281/zenodo.8115575.

Alberts, B., H. Rebecca, J. Alexander, M. David, R. Martin, R. Keith, and W. Peter. 2022. Molecular Biology of the Cell: Seventh Edition. W.W. Norton & Company. 14 pp.

Allan, C., J.-M. Burel, J. Moore, C. Blackburn, M. Linkert, S. Loynton, D. MacDonald, W.J. Moore, C. Neves, A. Patterson, M. Porter, A. Tarkowska, B. Loranger, J. Avondo, I. Lagerstedt, L. Lianas, S. Leo, K. Hands, R.T. Hay, A. Patwardhan, C. Best, G.J. Kleywegt, G. Zanetti, and J.R. Swedlow. 2012. OMERO: flexible, model-driven data management for experimental biology. *Nat. Methods*. 9:245–253. doi:10.1038/nmeth.1896.

Al-Zaben, N., A. Medyukhina, S. Dietrich, A. Marolda, K. Hünniger, O. Kurzai, and M.T. Figge. 2019. Automated tracking of label-free cells with enhanced recognition of whole tracks. *Sci. Rep.* 9:1–10. doi:10.1038/s41598-019-39725-x.

Andreev, A., and D.E.S. Koo. 2020. Practical Guide to Storage of Large Amounts of Microscopy Data. *Microsc. Today*. 28:42–45. doi:10.1017/S1551929520001091.

Anoraganingrum, D. 1999. Cell segmentation with median filter and mathematical morphology operation. *In* Proceedings 10th International Conference on Image Analysis and Processing. 1043–1046.

Aragaki, H., K. Ogoh, Y. Kondo, and K. Aoki. 2022. LIM Tracker: a software package for cell tracking and analysis with advanced interactivity. *Sci. Rep.* 12:2702. doi:10.1038/s41598-022-06269-6.

Archit, A., S. Nair, N. Khalid, P. Hilt, V. Rajashekar, M. Freitag, S. Gupta, A. Dengel, S. Ahmed, and C. Pape. 2023. Segment Anything for Microscopy. 2023.08.21.554208. doi:10.1101/2023.08.21.554208.

Arganda-Carreras, I., V. Kaynig, C. Rueden, K.W. Eliceiri, J. Schindelin, A. Cardona, and H. Sebastian Seung. 2017. Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*. 33:2424–2426. doi:10.1093/bioinformatics/btx180.

Arvanitis, C., S. Khuon, R. Spann, K.M. Ridge, and T.-L. Chew. 2014. Structure and biomechanics of the endothelial transcellular circumferential invasion array in tumor invasion. *PloS One*. 9:e89758. doi:10.1371/journal.pone.0089758.

Arzt, M., J. Deschamps, C. Schmied, T. Pietzsch, D. Schmidt, P. Tomancak, R. Haase, and F. Jug. 2022. LABKIT: Labeling and Segmentation Toolkit for Big Image Data. *Front. Comput. Sci.* 4. doi: 10.3389/fcomp.2022.777728

Balakrishnan, G., A. Zhao, M.R. Sabuncu, J. Guttag, and A.V. Dalca. 2019. VoxelMorph: A Learning Framework for Deformable

Medical Image Registration. *IEEE Trans. Med. Imaging*. 38:1788–1800. doi:10.1109/TMI.2019.2897538.

Bankhead, P., M.B. Loughrey, J.A. Fernández, Y. Dombrowski, D.G. McArt, P.D. Dunne, S. McQuaid, R.T. Gray, L.J. Murray, H.G. Coleman, J.A. James, M. Salto-Tellez, and P.W. Hamilton. 2017. QuPath: Open source software for digital pathology image analysis. *Sci. Rep.* 7:16878. doi:10.1038/s41598-017-17204-5.

Bannon, D., E. Moen, M. Schwartz, E. Borba, T. Kudo, N. Greenwald, V. Vijayakumar, B. Chang, E. Pao, E. Osterman, W. Graf, and D. Van Valen. 2021. DeepCell Kiosk: scaling deep learning–enabled cellular image analysis with Kubernetes. *Nat. Methods*. 18:43–45. doi:10.1038/s41592-020-01023-0.

Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool. 2008. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* 110:346–359. doi:10.1016/j.cviu.2007.09.014.

Berg, S., D. Kutra, T. Kroeger, C.N. Straehle, B.X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J.I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F.A. Hamprecht, and A. Kreshuk. 2019. ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods*. 16:1226–1232. doi:10.1038/s41592-019-0582-9.

Betzig, E., G.H. Patterson, R. Sougrat, O.W. Lindwasser, S. Olenych, J.S. Bonifacino, M.W. Davidson, J. Lippincott-Schwartz, and H.F. Hess. 2006. Imaging intracellular fluorescent proteins at nanometer resolution. *Science*. 313:1642–1645. doi:10.1126/science.1127344.

Bloice, M.D., P.M. Roth, and A. Holzinger. 2019. Biomedical image augmentation using Augmentor. *Bioinformatics*. 35:4522–4524. doi:10.1093/bioinformatics/btz259.

de Boer, P., J.P. Hoogenboom, and B.N.G. Giepmans. 2015. Correlated light and electron microscopy: ultrastructure lights up! *Nat. Methods*. 12:503–513. doi:10.1038/nmeth.3400.

Bragantini, J., M. Lange, and L. Royer. 2023. Large-Scale Multi-Hypotheses Cell Tracking Using Ultrametric Contours Maps.

Breiman, L. 2001. Random Forests. *Mach. Learn.* 45:5–32. doi:10.1023/A:1010933404324.

Burel, J.-M., S. Besson, C. Blackburn, M. Carroll, R.K. Ferguson, H. Flynn, K. Gillen, R. Leigh, S. Li, D. Lindner, M. Linkert, W.J. Moore, B. Ramalingam, E. Rozbicki, A. Tarkowska, P. Walczysko, C. Allan, J. Moore, and J.R. Swedlow. 2015. Publishing and sharing multi-dimensional image data with OMERO. *Mamm. Genome*. 26:441–447. doi:10.1007/s00335-015-9587-6.

Caicedo, J.C., J. Roth, A. Goodman, T. Becker, K.W. Karhohs, M. Broisin, C. Molnar, C. McQuin, S. Singh, F.J. Theis, and A.E. Carpenter. 2019. Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images. *Cytom. Part J. Int. Soc. Anal. Cytol.* 95:952–965. doi:10.1002/cyto.a.23863.

Carpenter, A.E., L. Kamentsky, and K.W. Eliceiri. 2012. A call for bioimaging software usability. *Nat. Methods*. 9:666–670. doi:10.1038/nmeth.2073.

Castello, M., G. Tortarolo, M. Buttafava, T. Deguchi, F. Villa, S. Koho, L. Pesce, M. Oneto, S. Pelicci, L. Lanzanó, P. Bianchini, C.J.R. Sheppard, A. Diaspro, A. Tosi, and G. Vicidomini. 2019. A robust and versatile platform for image scanning microscopy enabling super-resolution FLIM. *Nat. Methods*. 16:175–178. doi:10.1038/s41592-018-0291-9.

Celikay, K., V.O. Chagin, M. Cristina Cardoso, and K. Rohr. 2022. Denoisereg: Unsupervised Joint Denoising and Registration of Time-Lapse Live Cell Microscopy Images Using Deep Learning. *In* 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI). IEEE, Kolkata, India. 1–4. doi:10.1109/ISBI52829.2022.9761507

von Chamier, L., R.F. Laine, J. Jukkala, C. Spahn, D. Krentzel, E. Nehme, M. Lerche, S. Hernández-Pérez, P.K. Mattila, E. Karinou, S. Holden, A.C. Solak, A. Krull, T.-O. Buchholz, M.L. Jones, L.A. Royer, C. Leterrier, Y. Shechtman, F. Jug, M. Heilemann, G. Jacquemet, and R. Henriques. 2021. Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat. Commun.* 12:2276. doi:10.1038/s41467-021-22518-0.

Chaumont, F. de, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y.L. Montagner, T. Lagache, A. Dufour, and J.-C. Olivo-Marin. 2012. Icy: an open bioimage informatics platform for extended reproducible research. *Nat. Methods*. 9:690–696. doi:10.1038/nmeth.2075.

Chen, B.-C., W.R. Legant, K. Wang, L. Shao, D.E. Milkie, M.W. Davidson, C. Janetopoulos, X.S. Wu, J.A. Hammer, Z. Liu, B.P. English, Y. Mimori-Kiyosue, D.P. Romero, A.T. Ritter, J. Lippincott-Schwartz, L. Fritz-Laylin, R.D. Mullins, D.M. Mitchell, J.N. Bembenek, A.-C. Reymann, R. Böhme, S.W. Grill, J.T. Wang, G. Seydoux, U.S. Tulu, D.P. Kiehart, and E. Betzig. 2014. Lattice light-sheet microscopy: Imaging molecules to embryos at high spatiotemporal resolution. *Science*. 346:1257998. doi:10.1126/science.1257998.

Chen, G., W. Xie, and Y. Zhao. 2013. Wavelet-based denoising: A brief review. *In* 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP), Beijing, China, 2013, pp. 570-574, doi:10.1109/ICICIP.2013.6568140.

Chen, J., H. Sasaki, H. Lai, Y. Su, J. Liu, Y. Wu, A. Zhovmer, C.A. Combs, I. Rey-Suarez, H.-Y. Chang, C.C. Huang, X. Li, M. Guo, S. Nizambad, A. Upadhyaya, S.-J.J. Lee, L.A.G. Lucas, and H. Shroff. 2021. Three-dimensional residual channel attention networks denoise and sharpen fluorescence microscopy image volumes. *Nat. Methods*. 18:678–687. doi:10.1038/s41592-021-01155-x.

Chen, R., X. Tang, Y. Zhao, Z. Shen, M. Zhang, Y. Shen, T. Li, C.H.Y. Chung, L. Zhang, J. Wang, B. Cui, P. Fei, Y. Guo, S. Du, and S. Yao. 2023. Single-frame deep-learning super-resolution microscopy for intracellular dynamics imaging. *Nat. Commun.* 14:2854. doi:10.1038/s41467-023-38452-2.

Christiansen, E.M., S.J. Yang, D.M. Ando, A. Javaherian, G. Skibinski, S. Lipnick, E. Mount, A. O'Neil, K. Shah, A.K. Lee, P. Goyal, W. Fedus, R. Poplin, A. Esteva, M. Berndl, L.L. Rubin, P. Nelson, and S. Finkbeiner. 2018. In Silico Labeling: Predicting Fluorescent Labels in Unlabeled Images. *Cell*. 173:792-803.e19. doi:10.1016/j.cell.2018.03.040.

Cimini, B.A., and K.W. Eliceiri. 2023. The Twenty Questions of bioimage object analysis. *Nat. Methods*. 20:976–978. doi:10.1038/s41592-023-01919-7.

Commowick, O., N. Wiest-Daessle, and S. Prima. 2012. Block-matching strategies for rigid registration of multimodal medical images. *In* 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI). IEEE, Barcelona, Spain. 700–703. doi:10.1109/ISBI.2012.6235644

Conway, J.R.W., A. Isomursu, G. Follain, V. Härmä, E. Jou-Ollé, N. Pasquier, E.P.O. Välimäki, J.K. Rantala, and J. Ivaska. 2023. Defined extracellular matrix compositions support stiffness-insensitive cell spreading and adhesion signaling. *Proc. Natl. Acad. Sci.* 120:e2304288120. doi:10.1073/pnas.2304288120.

Crocker, J.C. 1996. Methods of Digital Video Microscopy for Colloidal Studies. *J. Colloid Interface Sci.* 179:298–310. doi:10.1006/jcis.1996.0217.

Cross, S.J., J.D.J.R. Fisher, and M.A. Jepson. ModularImageAnalysis (MIA): Assembly of modularised image and object analysis workflows in ImageJ. *J. Microsc.* n/a. doi:10.1111/jmi.13227.

Dobson Ellen, T.A., B. Cimini, H. Klemm Anna, C. Wählby, E. Carpenter Anne, and W. Eliceiri Kevin. 2021. ImageJ and CellProfiler: Complements in Open Source Bioimage Analysis. *Curr. Protoc.* 1:e89. doi:10.1002/cpz1.89.

Dosovitskiy, A., P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P.V.D. Smagt, D. Cremers, and T. Brox. 2015. FlowNet:

Learning Optical Flow with Convolutional Networks. *In* 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, Santiago. 2758–2766. doi:10.48550/arXiv.1504.06852

Dufour, A., R. Thibeaux, E. Labruyere, N. Guillen, and J.-C. Olivo-Marin. 2011. 3-D Active Meshes: Fast Discrete Deformable Models for Cell Tracking in 3-D Time-Lapse Microscopy. *IEEE Trans. Image Process.* 20:1925–1937. doi:10.1109/TIP.2010.2099125.

Edlund, C., T.R. Jackson, N. Khalid, N. Bevan, T. Dale, A. Dengel, S. Ahmed, J. Trygg, and R. Sjögren. 2021. LIVECell—A large-scale dataset for label-free live cell segmentation. *Nat. Methods*. 18:1038–1045. doi:10.1038/s41592-021-01249-6.

Ershov, D., M.-S. Phan, J.W. Pylvänäinen, S.U. Rigaud, L. Le Blanc, A. Charles-Orszag, J.R.W. Conway, R.F. Laine, N.H. Roy, D. Bonazzi, G. Duménil, G. Jacquemet, and J.-Y. Tinevez. 2022. TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nat. Methods*. 1–4. doi:10.1038/s41592-022-01507-1.

Fan, L., F. Zhang, H. Fan, and C. Zhang. 2019. Brief review of image denoising techniques. *Vis. Comput. Ind. Biomed. Art*. 2:7. doi:10.1186/s42492-019-0016-7.

Fazeli, E., N.H. Roy, G. Follain, R.F. Laine, L. von Chamier, P.E. Hänninen, J.E. Eriksson, J.-Y. Tinevez, and G. Jacquemet. 2020. Automated cell tracking using StarDist and TrackMate. *F1000Research*. 9:1279. doi:10.12688/f1000research.27019.1.

Fernandez, R., A. Crabos, M. Maillard, P. Nacry, and C. Pradal. 2022. High-throughput and automatic structural and developmental root phenotyping on Arabidopsis seedlings. *Plant Methods*. 18:127. doi:10.1186/s13007-022-00960-5.

Fernandez, R., and C. Moisy. 2021. Fijiyama: a registration tool for 3D multimodal time-lapse imaging. *Bioinforma. Oxf. Engl.* 37:1482–1484. doi:10.1093/bioinformatics/btaa846.

Fischer, P., A. Dosovitskiy, and T. Brox. 2014. Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT. *ArXiv*. doi:10.48550/arXiv.1405.5769. This paper has been merged with arXiv:1406.6909 doi:10.48550/arXiv.1406.6909

Follain, G., N. Osmani, A.S. Azevedo, G. Allio, L. Mercier, M.A. Karreman, G. Solecki, M.J. Garcia Leòn, O. Lefebvre, N. Fekonja, C. Hille, V. Chabannes, G. Dollé, T. Metivet, F.D. Hovsepian, C. Prudhomme, A. Pichot, N. Paul, R. Carapito, S. Bahram, B. Ruthensteiner, A. Kemmling, S. Siemonsen, T. Schneider, J. Fiehler, M. Glatzel, F. Winkler, Y. Schwab, K. Pantel, S. Harlepp, and J.G. Goetz. 2018. Hemodynamic Forces Tune the Arrest, Adhesion, and Extravasation of Circulating Tumor Cells. *Dev. Cell*. 45:33-52.e12. doi:10.1016/j.devcel.2018.02.015.

Fox, Z.R., S. Fletcher, A. Fraisse, C. Aditya, S. Sosa-Carrillo, J. Petit, S. Gilles, F. Bertaux, J. Ruess, and G. Batt. 2022. Enabling reactive microscopy with MicroMator. *Nat. Commun.* 13:2199. doi:10.1038/s41467-022-29888-z.

Franz, C.M., G.E. Jones, and A.J. Ridley. 2002. Cell Migration in Development and Disease. *Dev. Cell*. 2:153–158. doi:10.1016/S1534-5807(02)00120-X.

Ghannoum, S., K. Antos, W. Leoncio Netto, C. Gomes, A. Köhn-Luque, and H. Farhan. 2021. CellMAPtracer: A User-Friendly Tracking Tool for Long-Term Migratory and Proliferating Cells Associated with FUCCI Systems. *Cells*. 10:469. doi:10.3390/cells10020469.

Goedhart, J. 2020. PlotTwist: A web app for plotting and annotating continuous data. *PLOS Biol.* 18:e3000581. doi:10.1371/journal.pbio.3000581.

Gómez-de-Mariscal, E., C. García-López-de-Haro, W. Ouyang, L. Donati, E. Lundberg, M. Unser, A. Muñoz-Barrutia, and D. Sage. 2021. DeepImageJ: A user-friendly environment to run deep learning models in ImageJ. *Nat. Methods*. 1–4. doi:10.1038/s41592-021-01262-9.

González, R. 2011. Improving Phase Correlation for Image Registration. González, Rubén. “Improving Phase Correlation for Image Registration.” Image and Vision Computing New Zealand (2011).

Gonzalez, R.C., and R.E. Woods. 2018. Digital image processing. Fourth edition. Pearson, New York, NY. 1168 pp.

Gräf, R., J. Rietdorf, and T. Zimmermann. 2005. Live cell spinning disk microscopy. *Adv. Biochem. Eng. Biotechnol.* 95:57–75. doi:10.1007/b102210.

Gsell, S., S. Tlili, M. Merkel, and P.-F. Lenne. 2023. Marangoni-like tissue flows enhance symmetry breaking of embryonic organoids. 2023.09.22.559003. doi:10.1101/2023.09.22.559003.

Gupta, A., A. Sabirsh, C. Wählby, and I.-M. Sintorn. 2022. SimSearch: A Human-in-The-Loop Learning Framework for Fast Detection of Regions of Interest in Microscopy Images. *IEEE J. Biomed. Health Inform.* 26:4079–4089. doi:10.1109/JBHI.2022.3177602.

Gustafsson, M.G.L. 2000. Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *J. Microsc.* 198:82–87. doi:10.1046/j.1365-2818.2000.00710.x.

Haase, R., E. Fazeli, D. Legland, M. Doube, S. Culley, I. Belevich, E. Jokitalo, M. Schorb, A. Klemm, and C. Tischer. 2022. A Hitchhiker’s guide through the bio-image analysis software universe. *FEBS Lett.* 596:2472–2485. doi:10.1002/1873-3468.14451.

Haase, R., L.A. Royer, P. Steinbach, D. Schmidt, A. Dibrov, U. Schmidt, M. Weigert, N. Maghelli, P. Tomancak, F. Jug, and E.W. Myers. 2020. CLIJ: GPU-accelerated image processing for everyone. *Nat. Methods*. 17:5–6. doi:10.1038/s41592-019-0650-1.

Haase, R., R. Savill, P. Sobolewski, and D. Lee. 2023. haesleinhuepf/napari-assistant: 0.4.6. doi:10.5281/ZENODO.8220110.

Haase, R., and M. Schätz. 2022. haesleinhuepf/the-segmentation-game: 0.2.0. doi:10.5281/ZENODO.7135503.

Han, X., T. Leung, Y. Jia, R. Sukthankar, and A.C. Berg. 2015. MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 3279-3286, doi: 10.1109/CVPR.2015.7298948.

Harris, C.R., K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. Del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T.E. Oliphant. 2020. Array programming with NumPy. *Nature*. 585:357–362. doi:10.1038/s41586-020-2649-2.

Hell, S.W., and J. Wichmann. 1994. Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Opt. Lett.* 19:780–782. doi:10.1364/ol.19.000780.

Hidalgo-Cenalmor, I., J.W. Pylvänäinen, M.G. Ferreira, C.T. Russell, I. Arganda-Carreras, AI4Life Consortium, G. Jacquemet, R. Henriques, and E. Gómez-de-Mariscal. 2023. DL4MicEverywhere: Deep learning for microscopy made flexible, shareable, and reproducible. Bioinformatics.

Hinton, G.E., and R.R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*. 313:504–507. doi:10.1126/science.1127647.

Hu, T., S. Xu, L. Wei, X. Zhang, and X. Wang. 2021. CellTracker: an automated toolbox for single-cell segmentation and tracking of time-lapse microscopy images. *Bioinformatics*. 37:285–287. doi:10.1093/bioinformatics/btaa1106.

Huang, B., W. Wang, M. Bates, and X. Zhuang. 2008. Three-Dimensional Super-Resolution Imaging by Stochastic Optical Reconstruction Microscopy. *Science*. 319:810–813. doi:10.1126/science.1153529.

Huang, X., J. Fan, L. Li, H. Liu, R. Wu, Y. Wu, L. Wei, H. Mao, A. Lal, P. Xi, L. Tang, Y.

Zhang, Y. Liu, S. Tan, and L. Chen. 2018. Fast, long-term, super-resolution imaging with Hessian structured illumination microscopy. *Nat. Biotechnol.* 36:451–459. doi:10.1038/nbt.4115.

Icha, J., M. Weber, J.C. Waters, and C. Norden. 2017. Phototoxicity in live fluorescence microscopy, and how to avoid it. *BioEssays News Rev. Mol. Cell. Dev. Biol.* 39. doi:10.1002/bies.201700003.

Isola, P., J.-Y. Zhu, T. Zhou, and A.A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *In* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/CVPR.2017.632.

Jaccard, P. 1901. Distribution de la flore alpine dans le Bassin des Dranses et dans quelques régions voisines. doi:10.5169/SEALS-266440.

Jacquemet, G. 2023. CellTracksColab, A platform for compiling, analyzing, and exploring tracking data. 2023.10.20.563252. doi:10.1101/2023.10.20.563252.

Jacquemet, G., A.F. Carisey, H. Hamidi, R. Henriques, and C. Leterrier. 2020. The cell biologist's guide to super-resolution microscopy. *J. Cell Sci.* 133. doi:10.1242/jcs.240713.

Jacquemet, G., Minh-Son-Phan, and J.-Y. Tinevez. 2022a. Tracking focal adhesions with TrackMate and Weka - tutorial dataset 2. doi:10.5281/zenodo.5978940.

Jacquemet, G., J.W. Pylvänäinen, and J.-Y. Tinevez. 2022b. Tracking Glioblastoma-astrocytoma cells imaged in brightfield with TrackMate-Cellpose. doi:10.5281/zenodo.5863317.

Jamali, N., E.T.A. Dobson, K.W. Eliceiri, A.E. Carpenter, and B.A. Cimini. 2021. 2020 BioImage Analysis Survey: Community experiences and needs for the future. *Biol. Imaging*. 1:e4. doi:10.1017/S2633903X21000039.

Jaqaman, K., D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S.L. Schmid, and G. Danuser. 2008. Robust single-particle tracking in live-cell time-lapse sequences. *Nat. Methods*. 5:695–702. doi:10.1038/nmeth.1237.

Jensen, E.C. 2013. Overview of Live-Cell Imaging: Requirements and Methods Used. *Anat. Rec.* 296:1–8. doi:10.1002/ar.22554.

Jin, L., B. Liu, F. Zhao, S. Hahn, B. Dong, R. Song, T.C. Elston, Y. Xu, and K.M. Hahn. 2020. Deep learning enables structured illumination microscopy with low light levels and enhanced speed. *Nat. Commun.* 11:1934. doi:10.1038/s41467-020-15784-x.

Jonkman, J., C.M. Brown, G.D. Wright, K.I. Anderson, and A.J. North. 2020. Tutorial: guidance for quantitative confocal microscopy. *Nat. Protoc.* 15:1585–1611. doi:10.1038/s41596-020-0313-9.

Kalman, R.E. 1960. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* 82:35–45. doi:10.1115/1.3662552.

Keller, H.E. 2007. Proper Alignment of the Microscope. *In* Methods in Cell Biology. Academic Press. 43–53. doi:10.1016/S0091-679X(06)81003-4.

Kemmer, I., A. Keppler, B. Serrano-Solano, A. Rybina, B. Özdemir, J. Bischof, A. El Ghadraoui, J.E. Eriksson, and A. Mathur. 2023. Building a FAIR image data ecosystem for microscopy communities. *Histochem. Cell Biol.* 160:199–209. doi:10.1007/s00418-023-02203-7.

Kiepas, A., E. Voorand, F. Mubaid, P.M. Siegel, and C.M. Brown. 2020. Optimizing live-cell fluorescence imaging conditions to minimize phototoxicity. *J. Cell Sci.* doi:10.1242/jcs.242834.

Kozubek, M. 2016. Challenges and Benchmarks in Bioimage Analysis. *In* Focus on Bio-Image Informatics. W.H. De Vos, S. Munck, and J.-P. Timmermans, editors. Springer International Publishing, Cham. 231–262. doi: 10.1007/978-3-319-28549-8_9

Krizhevsky, A., I. Sutskever, and G.E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *In*

Advances in Neural Information Processing Systems. Curran Associates, Inc.

Krull, A., T.-O. Buchholz, and F. Jug. 2019. Noise2Void - Learning Denoising From Single Noisy Images. *In* 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2124–2132. doi: 10.48550/arXiv.1811.10980

Kudo, T., S. Jeknić, D.N. Macklin, S. Akhter, J.J. Hughey, S. Regot, and M.W. Covert. 2018. Live-cell measurements of kinase activity in single cells using translocation reporters. *Nat. Protoc.* 13:155–169. doi:10.1038/nprot.2017.128.

Kuppala, K., S. Banda, and T.R. Barige. 2020. An overview of deep learning methods for image registration with focus on feature-based approaches. *Int. J. Image Data Fusion*. 11:113–135. doi:10.1080/19479832.2019.1707720.

Laine, R.F., I. Arganda-Carreras, R. Henriques, and G. Jacquemet. 2021a. Avoiding a replication crisis in deep-learning-based bioimage analysis. *Nat. Methods*. 18:1136–1144. doi:10.1038/s41592-021-01284-3.

Laine, R.F., G. Jacquemet, and A. Krull. 2021b. Imaging in focus: An introduction to denoising bioimages in the era of deep learning. *Int. J. Biochem. Cell Biol.* 140:106077. doi:10.1016/j.biocel.2021.106077.

Laine, R.F., K.L. Tosheva, N. Gustafsson, R.D.M. Gray, P. Almada, D. Albrecht, G.T. Risa, F. Hurtig, A.-C. Lind\aas, B. Baum, J. Mercer, C. Leterrier, P.M. Pereira, S. Culley, and R. Henriques. 2019. NanoJ: a high-performance open-source super-resolution microscopy toolbox. *J. Phys. Appl. Phys.* 52:163001. doi:10.1088/1361-6463/ab0261.

Laissue, P.P., R.A. Alghamdi, P. Tomancak, E.G. Reynaud, and H. Shroff. 2017. Assessing phototoxicity in live fluorescence imaging. *Nat. Methods*. 14:657–661. doi:10.1038/nmeth.4344.

Le Blanc, L., S. Rigaud, and J.-Y. Tinevez. 2021. Neisseria meningitidis bacterial growth. doi:10.5281/zenodo.5419619.

LeCun, Y., Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature*. 521:436–444. doi:10.1038/nature14539.

Legland, D., I. Arganda-Carreras, and P. Andrey. 2016. MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ. *Bioinformatics*. 32:3532–3534. doi:10.1093/bioinformatics/btw413.

Lemon, W.C., and K. McDole. 2020. Live-cell imaging in the era of too many microscopes. *Curr. Opin. Cell Biol.* 66:34–42. doi:10.1016/j.ceb.2020.04.008.

Leterrier, C., and R.F. Laine. 2020. ZeroCostDL4Mic - DeepSTORM training and example dataset. doi:10.5281/ZENODO.3959089.

Levet, F., A.E. Carpenter, K.W. Eliceiri, A. Kreshuk, P. Bankhead, and R. Haase. 2021. Developing open-source software for bioimage analysis: opportunities and challenges. *F1000Research*. 10:302. doi:10.12688/f1000research.52531.1.

Li, S., S. Besson, C. Blackburn, M. Carroll, R.K. Ferguson, H. Flynn, K. Gillen, R. Leigh, D. Lindner, M. Linkert, W.J. Moore, B. Ramalingam, E. Rozbicki, G. Rustici, A. Tarkowska, P. Walczysko, E. Williams, C. Allan, J.-M. Burel, J. Moore, and J.R. Swedlow. 2016. Metadata management for high content screening in OMERO. *Methods*. 96:27–32. doi:10.1016/j.ymeth.2015.10.006.

Li, X., Y. Hu, X. Gao, D. Tao, and B. Ning. 2010. A multi-frame image super-resolution method. *Signal Process.* 90:405–414. doi:10.1016/j.sigpro.2009.05.028.

Linkert, M., C.T. Rueden, C. Allan, J.-M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. MacDonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K.W. Eliceiri, and J.R. Swedlow. 2010. Metadata matters: access to image data in the real world. *J. Cell Biol.* 189:777–782. doi:10.1083/jcb.201004104.

Ljosa, V., K.L. Sokolnicki, and A.E. Carpenter. 2012. Annotated high-throughput microscopy image sets for validation. *Nat. Methods*. 9:637–637. doi:10.1038/nmeth.2083.

Lucas, A.M., P.V. Ryder, B. Li, B.A. Cimini, K.W. Eliceiri, and A.E. Carpenter. 2021. Open-source deep-learning software for bioimage segmentation. *Mol. Biol. Cell*. 32:823–829. doi:10.1091/mbc.E20-10-0660.

Lukinavičius, G., C. Blaukopf, E. Pershagen, A. Schena, L. Reymond, E. Derivery, M. Gonzalez-Gaitan, E. D'Este, S.W. Hell, D. Wolfram Gerlich, and K. Johnsson. 2015. SiR-Hoechst is a far-red DNA stain for live-cell nanoscopy. *Nat. Commun.* 6:8497. doi:10.1038/ncomms9497.

Magidson, V., and A. Khodjakov. 2013. Chapter 23 - Circumventing Photodamage in Live-Cell Microscopy. *In* Methods in Cell Biology. G. Sluder and D.E. Wolf, editors. Academic Press. 545–560.

Martins, G.G., F.P. Cordelières, J. Colombelli, R. D'Antuono, O. Golani, R. Guiet, R. Haase, A.H. Klemm, M. Louveaux, P. Paul-Gilloteaux, J.-Y. Tinevez, and K. Miura. 2021. Highlights from the 2016-2020 NEUBIAS training schools for Bioimage Analysts: a success story and key asset for analysts and life scientists. doi:10.12688/f1000research.25485.1.

Maška, M., V. Ulman, P. Delgado-Rodriguez, E. Gómez-de-Mariscal, T. Nečasová, F.A. Guerrero Peña, T.I. Ren, E.M. Meyerowitz, T. Scherr, K. Löffler, R. Mikut, T. Guo, Y. Wang, J.P. Allebach, R. Bao, N.M. Al-Shakarji, G. Rahmon, I.E. Toubal, K. Palaniappan, F. Lux, P. Matula, K. Sugawara, K.E.G. Magnusson, L. Aho, A.R. Cohen, A. Arbelle, T. Ben-Haim, T.R. Raviv, F. Isensee, P.F. Jäger, K.H. Maier-Hein, Y. Zhu, C. Ederra, A. Urbiola, E. Meijering, A. Cunha, A. Muñoz-Barrutia, M. Kozubek, and C. Ortiz-de-Solórzano. 2023. The Cell Tracking Challenge: 10 years of objective benchmarking. *Nat. Methods*. 20:1010–1020. doi:10.1038/s41592-023-01879-y.

McGorty, R., D. Kamiyama, and B. Huang. 2013. Active microscope stabilization in three dimensions using image correlation. *Opt. Nanoscopy*. 2:3. doi:10.1186/2192-2853-2-3.

Miihkinen, M., M.L.B. Grönloh, A. Popović, H. Vihinen, E. Jokitalo, B.T. Goult, J. Ivaska, and G. Jacquemet. 2021. Myosin-X and talin modulate integrin activity at filopodia tips. *Cell Rep.* 36:109716. doi:10.1016/j.celrep.2021.109716.

Miura, K. 2020. Bleach correction ImageJ plugin for compensating the photobleaching of time-lapse sequences. *F1000Research*. 9:1494. doi:10.12688/f1000research.27171.1.

Mlodzianoski, M.J., J.M. Schreiner, S.P. Callahan, K. Smolková, A. Dlasková, J. Šantorová, P. Ježek, and J. Bewersdorf. 2011. Sample drift correction in 3D fluorescence photoactivation localization microscopy. *Opt. Express*. 19:15009. doi:10.1364/OE.19.015009.

Moen, E., D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen. 2019. Deep learning for cellular image analysis. *Nat. Methods*. 16:1233–1246. doi:10.1038/s41592-019-0403-1.

Muehlboeck, J.-S., E. Westman, and A. Simmons. 2014. TheHiveDB image data management and analysis framework. *Front. Neuroinformatics*. 7. doi:10.3389/fninf.2013.00049.

Nan A., Tennant M., Rubin., Ray N. 2020. DRMIME: Differentiable Mutual Information and Matrix Exponential for Multi-Resolution Image Registration. doi: 10.48550/arXiv.2001.09865.

Neil, M.A., R. Juskaitis, and T. Wilson. 1997. Method of obtaining optical sectioning by using structured light in a conventional microscope. *Opt. Lett.* 22:1905–1907. doi:10.1364/ol.22.001905.

Nketia, T.A., H. Sailem, G. Rohde, R. Machiraju, and J. Rittscher. 2017. Analysis of live cell images: Methods, tools and opportunities. *Methods San Diego Calif.* 115:65–79. doi:10.1016/j.ymeth.2017.02.007.

Ollion, J., J. Cochennec, F. Loll, C. Escudé, and T. Boudier. 2013. TANGO: a generic tool for high-throughput 3D image analysis for studying nuclear organization. *Bioinformatics*. 29:1840–1841. doi:10.1093/bioinformatics/btt276.

Osmani, N., G. Follain, M.J. García León, O. Lefebvre, I. Busnelli, A. Larnicol, S. Harlepp, and J.G. Goetz. 2019. Metastatic Tumor Cells Exploit Their Adhesion Repertoire to Counteract Shear Forces during Intravascular Arrest. *Cell Rep.* 28:2491-2500.e5. doi:10.1016/j.celrep.2019.07.102.

Otsu, N. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* 9:62–66. doi:10.1109/TSMC.1979.4310076.

Ounkomol, C., S. Seshamani, M.M. Maleckar, F. Collman, and G.R. Johnson. 2018. Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nat. Methods*. 15:917–920. doi:10.1038/s41592-018-0111-2.

Ouyang, W., F. Beuttenmueller, E. Gómez-de-Mariscal, C. Pape, T. Burke, C. Garcia-López-de-Haro, C. Russell, L. Moya-Sans, C. de-la-Torre-Gutiérrez, D. Schmidt, D. Kutra, M. Novikov, M. Weigert, U. Schmidt, P. Bankhead, G. Jacquemet, D. Sage, R. Henriques, A. Muñoz-Barrutia, E. Lundberg, F. Jug, and A. Kreshuk. 2022. BioImage Model Zoo: A Community-Driven Resource for Accessible Deep Learning in BioImage Analysis. 2022.06.07.495102. doi:10.1101/2022.06.07.495102.

Ouyang, W., K.W. Eliceiri, and B.A. Cimini. 2023. Moving beyond the desktop: prospects for practical bioimage analysis via the web. *Front. Bioinforma.* 3. doi:10.3389/fbinf.2023.1233748

Ouyang, W., and C. Zimmer. 2017. The imaging tsunami: Computational opportunities and challenges. *Curr. Opin. Syst. Biol.* 4:105–113. doi:10.1016/j.coisb.2017.07.011.

Øvrebø, Ø., M. Ojansivu, K. Kartasalo, H.M.G. Barriga, P. Ranefall, M.N. Holme, and M.M. Stevens. 2023. RegiSTORM: channel registration for multi-color stochastic optical reconstruction microscopy. *BMC Bioinformatics*. 24:237. doi:10.1186/s12859-023-05320-1.

Pachitariu, M., and C. Stringer. 2022. Cellpose 2.0: how to train your own model. *Nat. Methods*. 1–8. doi:10.1038/s41592-022-01663-4.

Pan, S.J., and Q. Yang. 2010. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* 22:1345–1359. doi:10.1109/TKDE.2009.191.

Parslow, A., A. Cardona, and R.J. Bryson-Richardson. 2014. Sample drift correction following 4D confocal time-lapse imaging. *J. Vis. Exp. JoVE*. doi:10.3791/51086.

Paul-Gilloteaux, P., S. Tosi, J.-K. Hériché, A. Gaignard, H. Ménager, R. Marée, V. Baecker, A. Klemm, M. Kalaš, C. Zhang, K. Miura, and J. Colombelli. 2021. Bioimage analysis workflows: community resources to navigate through a complex ecosystem. doi:10.12688/f1000research.52569.1.

Pawley, J.B. ed. . 2006. Handbook Of Biological Confocal Microscopy. Springer US, Boston, MA.

Peng, H. 2008. Bioimage informatics: a new area of engineering biology. *Bioinforma. Oxf. Engl.* 24:1827–1836. doi:10.1093/bioinformatics/btn346.

Perez-Careta, E., M. Torres-Cisneros, O. Debeir, J. Sanchez-Mondragon, J.G. Avina-Cervantes, O.G. Ibarra-Manzano, D. May-Arrioja, E. Perez-Pantoja, and G. Negrete-Romero. 2008. Cell tracking by normalized cross-correlation with image processing. *In* 2008 Digest of the IEEE/LEOS Summer Topical Meetings. 43–44. doi: doi: 10.1109/LEOSST.2008.4590480.

Pettersen, H.S., I. Belevich, E.S. Røyset, E. Smistad, M.R. Simpson, E. Jokitalo, I. Reinertsen, I. Bakke, and A. Pedersen. 2022. Code-Free Development and Deployment of Deep Segmentation Models for Digital Pathology. *Front. Med.* 8. doi:10.3389/fmed.2021.816281.

Peuhu, E., G. Jacquemet, C.L.G.J. Scheele, A. Isomursu, M.-C. Laisne, L.M. Koskinen, I. Paatero, K. Thol, M. Georgiadou, C. Guzmán,

S. Koskinen, A. Laiho, L.L. Elo, P. Boström, P. Hartiala, J. van Rheenen, and J. Ivaska. 2022. MYO10-filopodia support basement membranes at pre-invasive tumor boundaries. *Dev. Cell*. 57:2350-2364.e7. doi:10.1016/j.devcel.2022.09.016.

Pielawski, N., A. Andersson, C. Avenel, A. Behanova, E. Chelebian, A. Klemm, F. Nysjö, L. Solorzano, and C. Wählby. 2023. TissUUmaps 3: Improvements in interactive visualization, exploration, and quality assessment of large-scale spatial omics data. *Heliyon*. 9:e15306. doi:10.1016/j.heliyon.2023.e15306.

Pietzsch, T., S. Preibisch, P. Tomancák, and S. Saalfeld. 2012. ImgLib2--generic image processing in Java. *Bioinforma. Oxf. Engl.* 28:3009–3011. doi:10.1093/bioinformatics/bts543.

Popović, A., M. Miihkinen, S. Ghimire, R. Saup, M.L.B. Grönloh, N.J. Ball, B.T. Goult, J. Ivaska, and G. Jacquemet. 2023. Myosin-X recruits lamellipodin to filopodia tips. *J. Cell Sci.* 136:jcs260574. doi:10.1242/jcs.260574.

Postma, M., and J. Goedhart. 2019. PlotsOfData—A web app for visualizing data together with their summaries. *PLOS Biol.* 17:e3000202. doi:10.1371/journal.pbio.3000202.

Preibisch, S., S. Saalfeld, and P. Tomancak. 2009. Globally optimal stitching of tiled 3D microscopic image acquisitions. *Bioinformatics*. 25:1463–1465. doi:10.1093/bioinformatics/btp184.

Priessner, M., D.C.A. Gaboriau, A. Sheridan, T. Lenn, J.R. Chubb, U. Manor, R. Vilar, and R.F. Laine. 2021. Content-aware frame interpolation (CAFI): Deep Learning-based temporal super-resolution for fast bioimaging. 2021.11.02.466664. doi:10.1101/2021.11.02.466664.

Pylvänäinen, J.W., J.R. Conway, and G. Jacquemet. 2023a. ERK activity in migrating MDA-MB-231 cells (clover-ERK-KTR + sir-DNA). doi:10.5281/ZENODO.8347775.

Pylvänäinen, J.W., E. Gómez-de-Mariscal, R. Henriques, and G. Jacquemet. 2023b. Live-cell imaging in the deep learning era. *Curr. Opin. Cell Biol.* 85:102271. doi:10.1016/j.ceb.2023.102271.

Pylvänäinen, J.W., R.F. Laine, S. Ghimire, G. Follain, and G. Jacquemet. 2022. Fast4DRegistration. doi:10.5281/zenodo.6534570.

Pylvänäinen, J.W., R.F. Laine, B.M.S. Saraiva, S. Ghimire, G. Follain, R. Henriques, and G. Jacquemet. 2023c. Fast4DReg – fast registration of 4D microscopy datasets. *J. Cell Sci.* 136:jcs260728. doi:10.1242/jcs.260728.

Qiao, C., D. Li, Y. Guo, C. Liu, T. Jiang, Q. Dai, and D. Li. 2021. Evaluation and development of deep neural networks for image super-resolution in optical microscopy. *Nat. Methods*. 18:194–202. doi:10.1038/s41592-020-01048-5.

Rakhymzhan, A., A.F. Fiedler, R. Günther, S. Domingue, L. Wooldridge, R. Leben, Y. Cao, A. Bias, J. Roodselaar, R. Köhler, C. Ulbricht, J. Heidelin, V. Andresen, I. Beckers, A. Haibel, G. Duda, A.E. Hauser, and R.A. Niesner. 2023. In vivo optimized three-photon imaging of intact mouse tibia links plasma cell motility to functional states in the bone marrow. 2023.12.16.571998. doi:10.1101/2023.12.16.571998.

Rasal, T., T. Veerakumar, B.N. Subudhi, and S. Esakkirajan. 2022. A new approach for reduction of the noise from microscopy images using Fourier decomposition. *Biocybern. Biomed. Eng.* 42:615–629. doi:10.1016/j.bbe.2022.05.001.

Regot, S., J.J. Hughey, B.T. Bajar, S. Carrasco, and M.W. Covert. 2014. High-Sensitivity Measurements of Multiple Kinase Activities in Live Single Cells. *Cell*. 157:1724–1734. doi:10.1016/j.cell.2014.04.039.

Riedl, J., A.H. Crevenna, K. Kessenbrock, J.H. Yu, D. Neukirchen, M. Bista, F. Bradke, D. Jenne, T.A. Holak, Z. Werb, M. Sixt, and R. Wedlich-Soldner. 2008. Lifeact: a versatile marker to visualize F-actin. *Nat. Methods*. 5:605. doi:10.1038/nmeth.1220.

Ronneberger, O., P. Fischer, and T. Brox. 2015. U-Net: Convolutional Networks for

Biomedical Image Segmentation. doi:10.48550/arXiv.1505.04597.

Royer, L.A. 2023. Omega – Harnessing the Power of Large Language Models for Bioimage Analysis. doi:10.5281/ZENODO.8240289.

Rueden, C.T., M.C. Hiner, E.L. Evans, M.A. Pinkert, A.M. Lucas, A.E. Carpenter, B.A. Cimini, and K.W. Eliceiri. 2022. PyImageJ: A library for integrating ImageJ and Python. *Nat. Methods*. 19:1326–1327. doi:10.1038/s41592-022-01655-4.

Rust, M.J., M. Bates, and X. Zhuang. 2006. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nat. Methods*. 3:793–796. doi:10.1038/nmeth929.

Sakaue-Sawano, A., H. Kurokawa, T. Morimura, A. Hanyu, H. Hama, H. Osawa, S. Kashiwagi, K. Fukami, T. Miyata, H. Miyoshi, T. Imamura, M. Ogawa, H. Masai, and A. Miyawaki. 2008. Visualizing spatiotemporal dynamics of multicellular cell-cycle progression. *Cell*. 132:487–498. doi:10.1016/j.cell.2007.12.033.

Salmon, E.D., and P. Tran. 1998. High-resolution video-enhanced differential interference contrast (VE-DIC) light microscopy. *Methods Cell Biol.* 56:153–184. doi:10.1016/s0091-679x(08)60426-4.

Sanderson, J. 2020. Fundamentals of Microscopy. *Curr. Protoc. Mouse Biol.* 10:e76. doi:10.1002/cpmo.76.

Sandler, O., S.P. Mizrahi, N. Weiss, O. Agam, I. Simon, and N.Q. Balaban. 2015. Lineage correlations of single cell division time as a probe of cell-cycle dynamics. *Nature*. 519:468–471. doi:10.1038/nature14318.

Schermelleh, L., A. Ferrand, T. Huser, C. Eggeling, M. Sauer, O. Biehlmaier, and G.P.C. Drummen. 2019. Super-resolution microscopy demystified. *Nat. Cell Biol.* 21:72–84. doi:10.1038/s41556-018-0251-8.

Schindelin, J., I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D.J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona. 2012. Fiji: an open-source platform for biological-image analysis. *Nat. Methods*. 9:676–682. doi:10.1038/nmeth.2019.

Schlaeppi, A., W. Adams, R. Haase, J. Huisken, R.B. MacDonald, K.W. Eliceiri, and E.C. Kugler. 2022. Meeting in the Middle: Towards Successful Multidisciplinary Bioimage Analysis Collaboration. *Front. Bioinforma.* 2:889755. doi:10.3389/fbinf.2022.889755.

Schmidt, F., A. Evers, A. Amberg, G. Hessler, C. Robles, and K.-H. Baringhaus. 2011. Phototoxicity – from molecular descriptors to classification models. *J. Cheminformatics*. 3:O10. doi:10.1186/1758-2946-3-S1-O10.

Schmidt, U., and M. Weigert. 2022. StarDist Fluorescence Nuclei Segmentation. doi:10.5281/zenodo.6348085.

Schmidt, U., M. Weigert, C. Broaddus, and G. Myers. 2018. Cell Detection with Star-Convex Polygons. *In* Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. A.F. Frangi, J.A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, editors. Springer International Publishing, Cham. 265–273. doi: 10.1007/978-3-030-00934-2_30

Schmied, C., M. Ebner, P.S. Ferré, V. Haucke, and M. Lehmann. 2023. OrgaMapper: A robust and easy-to-use workflow for analyzing organelle positioning. *bioRxiv*, doi: 10.1101/2023.07.10.548452.

Schmied, C., M.S. Nelson, S. Avilov, G.-J. Bakker, C. Bertocchi, J. Bischof, U. Boehm, J. Brocher, M.T. Carvalho, C. Chiritescu, J. Christopher, B.A. Cimini, E. Conde-Sousa, M. Ebner, R. Ecker, K. Eliceiri, J. Fernandez-Rodriguez, N. Gaudreault, L. Gelman, D. Grunwald, T. Gu, N. Halidi, M. Hammer, M. Hartley, M. Held, F. Jug, V. Kapoor, A.A. Koksoy, J. Lacoste, S. Le Dévédec, S. Le Guyader, P. Liu, G.G. Martins, A. Mathur, K. Miura, P. Montero Llopis, R. Nitschke, A. North, A.C. Parslow, A. Payne-Dwyer, L. Plantard, R. Ali, B. Schroth-Diez, L. Schütz, R.T. Scott, A. Seitz, O. Selchow, V.P. Sharma, M. Spitaler, S. Srinivasan, C. Strambio-De-Castillia, D. Taatjes, C. Tischer, and H.K.

Jambor. 2024. Community-developed checklists for publishing images and image analyses. *Nat. Methods*. 21:170–181. doi:10.1038/s41592-023-01987-9.

Selzer, G.J., C.T. Rueden, M.C. Hiner, E.L. Evans, K.I.S. Harrington, and K.W. Eliceiri. 2023. napari-imagej: ImageJ ecosystem access from napari. *Nat. Methods*. doi:10.1038/s41592-023-01990-0.

Senft, R.A., B. Diaz-Rohrer, P. Colarusso, L. Swift, N. Jamali, H. Jambor, T. Pengo, C. Brideau, P.M. Llopis, V. Uhlmann, J. Kirk, K.A. Gonzales, P. Bankhead, E.L.E. Iii, K.W. Eliceiri, and B.A. Cimini. 2023. A biologist's guide to planning and performing quantitative bioimaging experiments. *PLOS Biol.* 21:e3002167. doi:10.1371/journal.pbio.3002167.

Shannon, M.J., S.E. Eisman, A.R. Lowe, T. Sloan, and E.M. Mace. 2023. cellPLATO: an unsupervised method for identifying cell behaviour in heterogeneous cell trajectory data. *bioRxiv*. 2023.10.28.564355. doi:10.1101/2023.10.28.564355.

Sies, H., and D.P. Jones. 2020. Reactive oxygen species (ROS) as pleiotropic physiological signalling agents. *Nat. Rev. Mol. Cell Biol.* 21:363–383. doi:10.1038/s41580-020-0230-3.

Silberberg, M., and H.E. Grecco. 2024. Binlets: Data fusion-aware denoising enables accurate and unbiased quantification of multichannel signals. *Inf. Fusion*. 101:101999. doi:10.1016/j.inffus.2023.101999.

Sivagurunathan, S., S. Marcotti, C.J. Nelson, M.L. Jones, D.J. Barry, T.J.A. Slater, K.W. Eliceiri, and B.A. Cimini. 2023. Bridging imaging users to imaging analysis - A community survey. *J. Microsc.* doi:10.1111/jmi.13229.

Soltwedel, J.R., and R. Haase. 2023. Challenges and opportunities for bioimage analysis core-facilities. *J. Microsc.* jmi.13192. doi:10.1111/jmi.13192.

Soulez, F., L. Denis, Y. Tourneur, and É. Thiébaut. 2012. Blind deconvolution of 3D data in wide field fluorescence microscopy. *In* 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI). 1735–1738.

Spahn, C., E. Gómez-de-Mariscal, R.F. Laine, P.M. Pereira, L. von Chamier, M. Conduit, M.G. Pinho, G. Jacquemet, S. Holden, M. Heilemann, and R. Henriques. 2022. DeepBacs for multi-task bacterial image analysis using open-source deep learning approaches. *Commun. Biol.* 5:1–18. doi:10.1038/s42003-022-03634-z.

Speiser, A., L.-R. Müller, P. Hoess, U. Matti, C.J. Obara, W.R. Legant, A. Kreshuk, J.H. Macke, J. Ries, and S.C. Turaga. 2021. Deep learning enables fast and dense single-molecule localization with high accuracy. *Nat. Methods*. 18:1082–1090. doi:10.1038/s41592-021-01236-x.

Stehbens, S., H. Pemble, L. Murrow, and T. Wittmann. 2012. Imaging intracellular protein dynamics by spinning disk confocal microscopy. *Methods Enzymol.* 504:293–313. doi:10.1016/B978-0-12-391857-4.00015-X.

Stirling, D.R., M.J. Swain-Bowden, A.M. Lucas, A.E. Carpenter, B.A. Cimini, and A. Goodman. 2021. CellProfiler 4: improvements in speed, utility and usability. *BMC Bioinformatics*. 22:433. doi:10.1186/s12859-021-04344-9.

Stringer, C., T. Wang, M. Michaelos, and M. Pachitariu. 2021. Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods*. 18:100–106. doi:10.1038/s41592-020-01018-x.

Sugawara, K., Ç. Çevrim, and M. Averof. 2022. Tracking cell lineages in 3D by incremental deep learning. *eLife*. 11:e69380. doi:10.7554/eLife.69380.

Sun, H. 2002. The Hartley transform applied to particle image velocimetry. *Meas. Sci. Technol.* 13:1996. doi:10.1088/0957-0233/13/12/326.

Szabó, G., P. Bonaiuti, A. Ciliberto, and A. Horváth. 2023. Enhancing Cell Tracking with a Time-Symmetric Deep Learning Approach. doi:10.48550/arXiv.2308.03887.

Taïeb, H.M., L. Bertinetti, T. Robinson, and A. Cipitria. 2022. FUCCItrack: An all-in-one software for single cell tracking and cell cycle analysis. *PLOS ONE*. 17:e0268297. doi:10.1371/journal.pone.0268297.

Thevenaz, P., U.E. Ruttimann, and M. Unser. 1998. A pyramid approach to subpixel registration based on intensity. *IEEE Trans. Image Process.* 7:27–41. doi:10.1109/83.650848.

Tinevez, J.-Y. 2016. TrackMate documentation.

Tinevez, J.-Y., G. Jacquemet, and N.H. Roy. 2021a. T cells migration followed with TrackMate. doi:10.5281/ZENODO.5206119.

Tinevez, J.-Y., N. Perry, J. Schindelin, G.M. Hoopes, G.D. Reynolds, E. Laplantine, S.Y. Bednarek, S.L. Shorte, and K.W. Eliceiri. 2017. TrackMate: An open and extensible platform for single-particle tracking. *Methods San Diego Calif.* 115:80–90. doi:10.1016/j.ymeth.2016.09.016.

Tinevez, J.-Y., and J.W. Pylvänäinen. 2021. Cell migration with ERK signalling. doi:10.5281/ZENODO.5205955.

Tinevez, J.-Y., J.W. Pylvänäinen, and G. Jacquemet. 2021b. Segmenting cells in a spheroid in 3D using 2D StarDist within TrackMate. doi:10.5281/zenodo.5220610.

Tong, X., Z. Ye, Y. Xu, S. Gao, H. Xie, Q. Du, S. Liu, X. Xu, S. Liu, K. Luan, and U. Stilla. 2019. Image Registration With Fourier-Based Image Correlation: A Comprehensive Review of Developments and Applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 12:4062–4081. doi:10.1109/JSTARS.2019.2937690.

Tsai, H.-F., J. Gajda, T.F.W. Sloan, A. Rares, and A.Q. Shen. 2019. Usiigaci: Instance-aware cell tracking in stain-free phase contrast microscopy enabled by machine learning. *SoftwareX*. 9:230–237. doi:10.1016/j.softx.2019.02.007.

Ulicna, K., G. Vallardi, G. Charras, and A.R. Lowe. 2021. Automated Deep Lineage Tree Analysis Using a Bayesian Single Cell Tracking Approach. *Front. Comput. Sci.* 3.

Ulman, V., M. Maška, K.E.G. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, I. Smal, K. Rohr, J. Jaldén, H.M. Blau, O. Dzyubachyk, B. Lelieveldt, P. Xiao, Y. Li, S.-Y. Cho, A.C. Dufour, J.-C. Olivo-Marin, C.C. Reyes-Aldasoro, J.A. Solis-Lemus, R. Bensch, T. Brox, J. Stegmaier, R. Mikut, S. Wolf, F.A. Hamprecht, T. Esteves, P. Quelhas, Ö. Demirel, L. Malmström, F. Jug, P. Tomancak, E. Meijering, A. Muñoz-Barrutia, M. Kozubek, and C. Ortiz-de-Solorzano. 2017. An objective comparison of cell-tracking algorithms. *Nat. Methods*. 14:1141–1152. doi:10.1038/nmeth.4473.

Virtanen, P., R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*. 17:261–272. doi:10.1038/s41592-019-0686-2.

Wagner, N., F. Beuttenmueller, N. Norlin, J. Gierten, J.C. Boffi, J. Wittbrodt, M. Weigert, L. Hufnagel, R. Prevedel, and A. Kreshuk. 2021. Deep learning-enhanced light-field imaging with continuous validation. *Nat. Methods*. 18:557–563. doi:10.1038/s41592-021-01136-0.

Wäldchen, S., J. Lehmann, T. Klein, S. van de Linde, and M. Sauer. 2015. Light-induced cell damage in live-cell super-resolution microscopy. *Sci. Rep.* 5:15348. doi:10.1038/srep15348.

van der Walt, S., J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D. Warner, N. Yager, E. Gouillart, T. Yu, and scikit-image contributors. 2014. scikit-image: image processing in Python. *PeerJ*. 2:e453. doi:10.7717/peerj.453.

Wang, Y., J. Schnitzbauer, Z. Hu, X. Li, Y. Cheng, Z.-L. Huang, and B. Huang. 2014. Localization events-based sample drift correction for localization microscopy with redundant cross-correlation algorithm. *Opt. Express*. 22:15982–15991. doi:10.1364/OE.22.015982.

Wang, Z., A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* 13:600–612. doi:10.1109/TIP.2003.819861.

von Wangenheim, D., R. Hauschild, M. Fendrych, V. Barone, E. Benková, and J. Friml. 2017. Live tracking of moving samples in confocal microscopy for vertically grown roots. *eLife*. 6:e26792. doi:10.7554/eLife.26792.

Waters, J.C. 2007. Live-Cell Fluorescence Imaging. *Methods in Cell Biology*. doi:10.1016/B978-0-12-407761-4.00006-3.

Weigert, M., U. Schmidt, T. Boothe, A. Müller, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broaddus, S. Culley, M. Rocha-Martins, F. Segovia-Miranda, C. Norden, R. Henriques, M. Zerial, M. Solimena, J. Rink, P. Tomancak, L. Royer, F. Jug, and E.W. Myers. 2018. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nat. Methods*. 15:1090–1097. doi:10.1038/s41592-018-0216-7.

Wester, M.J., D.J. Schodt, H. Mazloom-Farsibaf, M. Fazel, S. Pallikkuth, and K.A. Lidke. 2021. Robust, fiducial-free drift correction for super-resolution imaging. *Sci. Rep.* 11:23672. doi:10.1038/s41598-021-02850-7.

Wiggins, L., A. Lord, K.L. Murphy, S.E. Lacy, P.J. O'Toole, W.J. Brackenbury, and J. Wilson. 2023. The CellPhe toolkit for cell phenotyping using time-lapse imaging and pattern recognition. *Nat. Commun.* 14:1854. doi:10.1038/s41467-023-37447-3.

Wilkinson, M.D., M. Dumontier, Ij.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L.B. da Silva Santos, P.E. Bourne, J. Bouwman, A.J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C.T. Evelo, R. Finkers, A. Gonzalez-Beltran, A.J.G. Gray, P. Groth, C. Goble, J.S. Grethe, J. Heringa, P.A.C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S.J. Lusher, M.E. Martone, A. Mons, A.L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M.A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, and B. Mons. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data*. 3:160018. doi:10.1038/sdata.2016.18.

Wolf, D.E. 2007. Fundamentals of Fluorescence and Fluorescence Microscopy. *In* Methods in Cell Biology. Academic Press. 63–91.

Wolf, S., Y. Wan, and K. McDole. 2021. Current approaches to fate mapping and lineage tracing using image data. *Development*. 148:dev198994. doi:10.1242/dev.198994.

Wolff, C., J.-Y. Tinevez, T. Pietzsch, E. Stamataki, B. Harich, L. Guignard, S. Preibisch, S. Shorte, P.J. Keller, P. Tomancak, and A. Pavlopoulos. 2018. Multi-view light-sheet imaging and tracking with the MaMuT software reveals the cell lineage of a direct developing arthropod limb. *eLife*. 7:e34410. doi:10.7554/eLife.34410.

Wortel, I.M.N., A.Y. Liu, K. Dannenberg, J.C. Berry, M.J. Miller, and J. Textor. 2021. CelltrackR: An R package for fast and flexible analysis of immune cell migration data. *ImmunoInformatics*. 1. doi:10.1016/j.immuno.2021.100003.

Wu, Y., A. Ghitani, R. Christensen, A. Santella, Z. Du, G. Rondeau, Z. Bao, D. Colón-Ramos, and H. Shroff. 2011. Inverted selective plane illumination microscopy (*i* SPIM) enables coupled cell identity lineaging and neurodevelopmental imaging in *Caenorhabditis elegans*. *Proc. Natl. Acad. Sci.* 108:17708–17713. doi:10.1073/pnas.1108494108.

Xu, H., C. Lu, R. Berendt, N. Jha, and M. Mandal. 2017. Automatic Nuclei Detection Based on Generalized Laplacian of Gaussian Filters. *IEEE J. Biomed. Health Inform.*

21:826–837. doi:10.1109/JBHI.2016.2544245.

Zargari, A., G.A. Lodewijk, N. Mashhadi, N. Cook, C.W. Neudorf, K. Araghbidikashani, R. Hays, S. Kozuki, S. Rubio, E. Hrabeta-Robinson, A. Brooks, L. Hinck, and S.A. Shariati. 2023. DeepSea is an efficient deep-learning model for single-cell segmentation and tracking in time-lapse microscopy. *Cell Rep. Methods*. 3:100500. doi:10.1016/j.crmeth.2023.100500.

Zernike, F. 1942. Phase contrast, a new method for the microscopic observation of transparent objects. *Physica*. 9:686–698. doi:https://doi.org/10.1016/S0031-8914(42)80035-X.

Zhang, R., P. Isola, A.A. Efros, E. Shechtman, and O. Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *In* 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, Salt Lake City, UT. doi:10.48550/arXiv.1801.03924

Zhao, S., T. Lau, J. Luo, E.I.-C. Chang, and Y. Xu. 2020. Unsupervised 3D End-to-End Medical Image Registration With Volume Tweening Network. *IEEE J. Biomed. Health Inform.* 24:1394–1404. doi:10.1109/JBHI.2019.2951024.

Zielke, N., and B.A. Edgar. 2015. FUCCI sensors: powerful new tools for analysis of cell proliferation. *WIREs Dev. Biol.* 4:469–487. doi:10.1002/wdev.189.

Zimmer, C., B. Zhang, A. Dufour, A. Thebaud, S. Berlemont, V. Meas-Yedid, and J.-C.O. Marin. 2006. On the digital trail of mobile cells. *IEEE Signal Process. Mag.* 23:54–62. doi:10.1109/MSP.2006.1628878.

Zitová, B., and J. Flusser. 2003. Image registration methods: a survey. *Image Vis. Comput.* 21:977–1000. doi:10.1016/S0262-8856(03)00137-9.

Original Publications

I

Pylvänäinen, J.W., Laine, R. F., Saraiva, B. M. S., Ghimire, S., Follain, G.,
Henriques, R., & Jacquemet, G., (2023).

Fast4DReg - fast registration of 4D microscopy datasets.

Journal of Cell Science, doi: 10.1242/jcs.260728

© 2023. Published by The Company of Biologists Ltd | Journal of Cell Science (2023) 136, jcs260728. doi:10.1242/jcs.260728

TOOLS AND RESOURCES

Fast4DReg – fast registration of 4D microscopy datasets

Joanna W. Pylvänäinen[1,2,3], Romain F. Laine[4,5,*], Bruno M. S. Saraiva[6], Sujan Ghimire[1,3], Gautier Follain[1,3], Ricardo Henriques[6] and Guillaume Jacquemet[1,2,3,7,‡]

ABSTRACT

Unwanted sample drift is a common issue that plagues microscopy experiments, preventing accurate temporal visualization and quantification of biological processes. Although multiple methods and tools exist to correct images post acquisition, performing drift correction of three-dimensional (3D) videos using open-source solutions remains challenging and time consuming. Here, we present a new tool developed for ImageJ or Fiji called Fast4DReg that can quickly correct axial and lateral drift in 3D video-microscopy datasets. Fast4DReg works by creating intensity projections along multiple axes and estimating the drift between frames using two-dimensional cross-correlations. Using synthetic and acquired datasets, we demonstrate that Fast4DReg can perform better than other state-of-the-art open-source drift-correction tools and significantly outperforms them in speed. We also demonstrate that Fast4DReg can be used to register misaligned channels in 3D using either calibration slides or misaligned images directly. Altogether, Fast4DReg provides a quick and easy-to-use method to correct 3D imaging data before further visualization and analysis.

KEY WORDS: 3D drift correction, Live imaging, Image analysis, ImageJ, Fiji

Introduction

Live imaging is essential in biomedical research, enabling scientists to follow biological processes over time. Despite being heavily used, performing live-imaging experiments using fluorescence microscopy remains technically challenging. The user must carefully balance illumination power and acquisition speed while maintaining specimen health. In addition, imaging is often prone to drift. Drift can be caused, for example, by temperature changes leading to thermal expansion of the microscope mechanical components or by the movement of the sample itself.

Multiple software and hardware solutions have been developed to minimize drifting during acquisition. For instance, axial drifting can be limited using an infrared light that is reflected on the glass-sample interface and captured by a detector (e.g. Leica's Adaptive Focus Control or Nikon's Perfect Focus System). Lateral drift due to sample movement can also be compensated by tracking algorithms that follow the sample over time and move the microscope stage accordingly (Fox et al., 2022; von Wangenheim et al., 2017). Yet, drifting is rarely entirely eliminated at the acquisition stage, especially when acquiring multiple positions for an extended period. Therefore, it is often necessary to perform drift correction (via image registration) as a post-processing step before image visualization and quantification. Beyond live imaging, drift correction and/or channel registration is a crucial processing step for multiple image-analysis pipelines, including colocalization analysis (using calibration slides) or the reconstruction of super-resolution microscopy images.

Most drift-correction/registration algorithms work sequentially by comparing a reference image to a moving image and estimating the movement between these two images to correct the drift. Multiple open-source tools capable of correcting four-dimensional (4D) datasets already exist. Popular tools include, for instance, Insight ToolKit (McCormick et al., 2014), elastix (Klein et al., 2010), Multiview Reconstruction (Preibisch et al., 2010; 2014), Fijiyama (Fernandez and Moisy, 2021) or Correct 3D drift (Correct3DD) (Parslow et al., 2014). However, except for Multiview Reconstruction and Correct3DD, these tools are geared toward correcting medical imaging datasets and can be impractical to use for the correction of long three-dimensional (3D) videos. Multiview Reconstruction, which was designed to register large light-sheet fluorescence microscopy datasets, uses interest points (e.g. fluorescent beads, nuclei or membrane markers) in the imaging volume to perform the 3D registration, which are not always available (Preibisch et al., 2010; 2014). Although we routinely use Correct3DD, we felt limited by its speed and available features.

Here, prompted by a need to correct our 3D videos more easily and more efficiently, we developed Fast4DReg, a fast two-dimensional (2D) and 3D video drift-correction tool. Using multiple datasets, we show that Fast4DReg can outperform two state-of-the-art 3D video drift-correction tools available in Fiji, namely, Correct3DD (Parslow et al., 2014) and Fijiyama (Fernandez and Moisy, 2021). In addition, we show that Fast4DReg can register misaligned channels in 3D using either calibration slides or misaligned images directly. Fast4DReg is fast and has an easy-to-use graphical interface. These features make Fast4DReg a versatile and easy-to-use open-source 2D/3D drift-correction tool.

RESULTS

The Fast4DReg pipeline

Fast4DReg breaks the drift-correction task into two steps: image registration (estimation of a transformation that corrects the drift optimally) followed by image transformation (applying the determined parameters to produce a corrected image). To estimate

---

[1]Åbo Akademi University, Faculty of Science and Engineering, Biosciences, Turku 20520, Finland. [2]Turku Bioimaging, University of Turku and Åbo Akademi University, Turku 20520, Finland. [3]Turku Bioscience Centre, University of Turku and Åbo Akademi University, Turku 20520, Finland. [4]MRC Laboratory for Molecular Cell Biology, University College London, London WC1E 6BT, UK. [5]The Francis Crick Institute, London NW1 1AT, UK. [6]Instituto Gulbenkian de Ciência, Oeiras 2780-156, Portugal. [7]InFLAMES Research Flagship Center, Åbo Akademi University, Turku 20520, Finland.
*Present address: Micrographia Bio, Translation and Innovation Hub, 84 Wood Lane, London, UK.

‡Author for correspondence (guillaume.jacquemet@abo.fi)

G.J., 0000-0002-9286-920X

Handling Editor: Michael Way
Received 16 October 2022; Accepted 25 January 2023

the drift of a 3D video in the *x*-, *y*- and *z*-coordinates, Fast4DReg sequentially estimates the lateral drift, corrects the lateral drift, then estimates and corrects the axial drift (Fig. 1). Lateral and axial drift corrections can also be performed independently, which can be particularly useful when only the axial drift needs to be corrected. As an output of the drift estimation step, Fast4DReg creates a new folder containing the corrected images, drift plots (graphs indicating the amount of drift detected), a drift table (drift detected in numerical values) and a settings file containing the selected parameters. Notably, the drift table can then be applied to correct other images using the same parameters (i.e. to correct another channel). Indeed, when correcting multichannel 3D videos, the user needs to choose one channel to use to estimate the drift. The other channel(s) can then be corrected using the same drift table (Fig. 1). To estimate the lateral or axial drift of a 3D video, Fast4DReg creates *z*- or *y*-intensity projections for each time point to create a 2D video. Fast4DReg then estimates the linear drift between the reference and moving frames by calculating their cross-correlation matrix (CCM) (see Materials and Methods for more details). The location of the peak intensity in the CCM defines the linear shift between the two images. Sub-pixel accuracy is accomplished by upscaling the CCM via bicubic spline interpolation (as demonstrated by Laine et al., 2019). Depending on their data, users can choose the first frame (best for fixed data) or consecutive frames from the movie (best for live-imaging data) as the reference frame.

Fast4DReg outperforms Correct3DD or Fijiyama on our synthetic dataset

To assess the capabilities of Fast4DReg to correct 3D videos, we compared Fast4DReg results to two other state-of-the-art drift-correction methods available in Fiji (Schindelin et al., 2012): Correct3DD (Parslow et al., 2014) and Fijiyama (Fernandez and Moisy, 2021). For this purpose, two synthetic videos with known amounts of drift were created: one with no drift and another with a large amount of drift (Fig. 2A; Fig. S1A). As these videos were generated by duplicating an acquired single 3D stack and adding artificial drift, a perfect drift correction should generate near identical time frames as only the background noise will differ.

Visually, all three tools corrected the artificially drifting 3D videos regardless of the amount of drift (Fig. 2B; Movie 1). To carefully quantify the performance of these three software, we selected a *z*-slice and plotted the standard deviation projection of the corrected stack (Fig. 2C). Next, we calculated multiple image-similarity metrics between the first and each subsequent frame (Fig. 2D,E; Fig. S1B,C). Both assessment methods indicated that Fast4DReg performed better than Correct3DD or Fijiyama on our synthetic dataset (Movie 2; Fig. 2B–E; Fig. S1B,C). Importantly, these results demonstrated that using 2D intensity projections followed by 2D cross-correlation is a suitable method to correct drifting 3D videos.

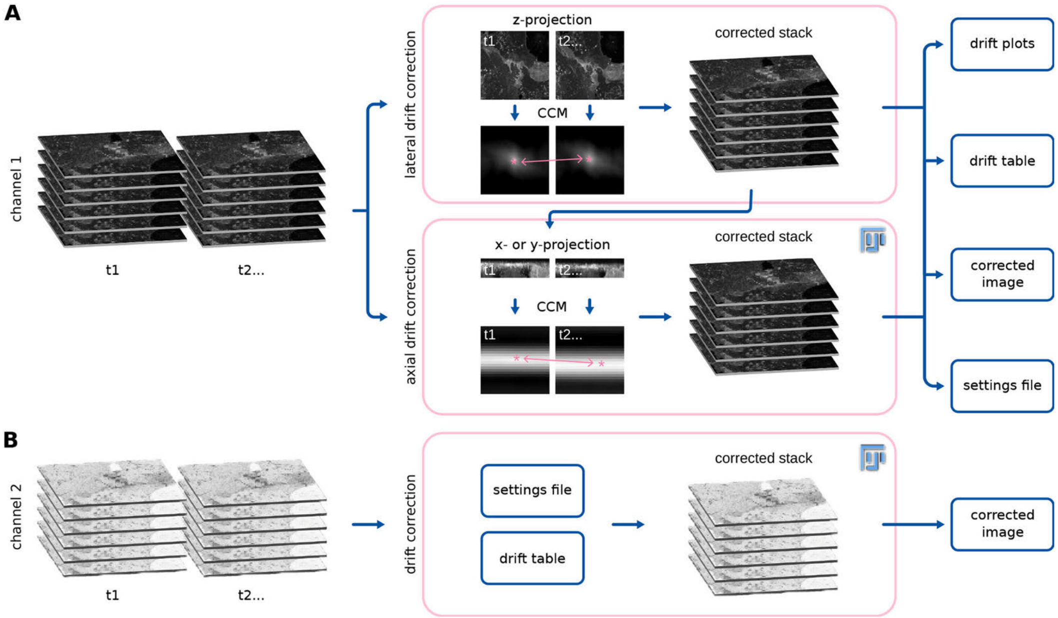


Fig. 1. Drift correction of 3D videos using Fast4DReg. Scheme highlighting the Fast4DReg pipeline. (A) Fast4DReg sequentially estimates the lateral drift, corrects the lateral drift, and then estimates and corrects the axial drift. Fast4DReg creates intensity projections along multiple axes and estimates the drift between the reference and moving frames by calculating their cross-correlation matrix (CCM). The location of the peak intensity in the CCM (pink asterisk) defines the linear shift between the two images (as highlighted by the pink double-headed arrow). Fast4DReg outputs the corrected images, the drift plots, a drift table, and a settings file containing all selected parameters and paths to the drift table. (B) The settings file inducing the used parameters and path to the drift table can then be applied to correct other datasets (i.e. another channel) directly.

Journal of Cell Science (2023) 136, jcs260728. doi:10.1242/jcs.260728

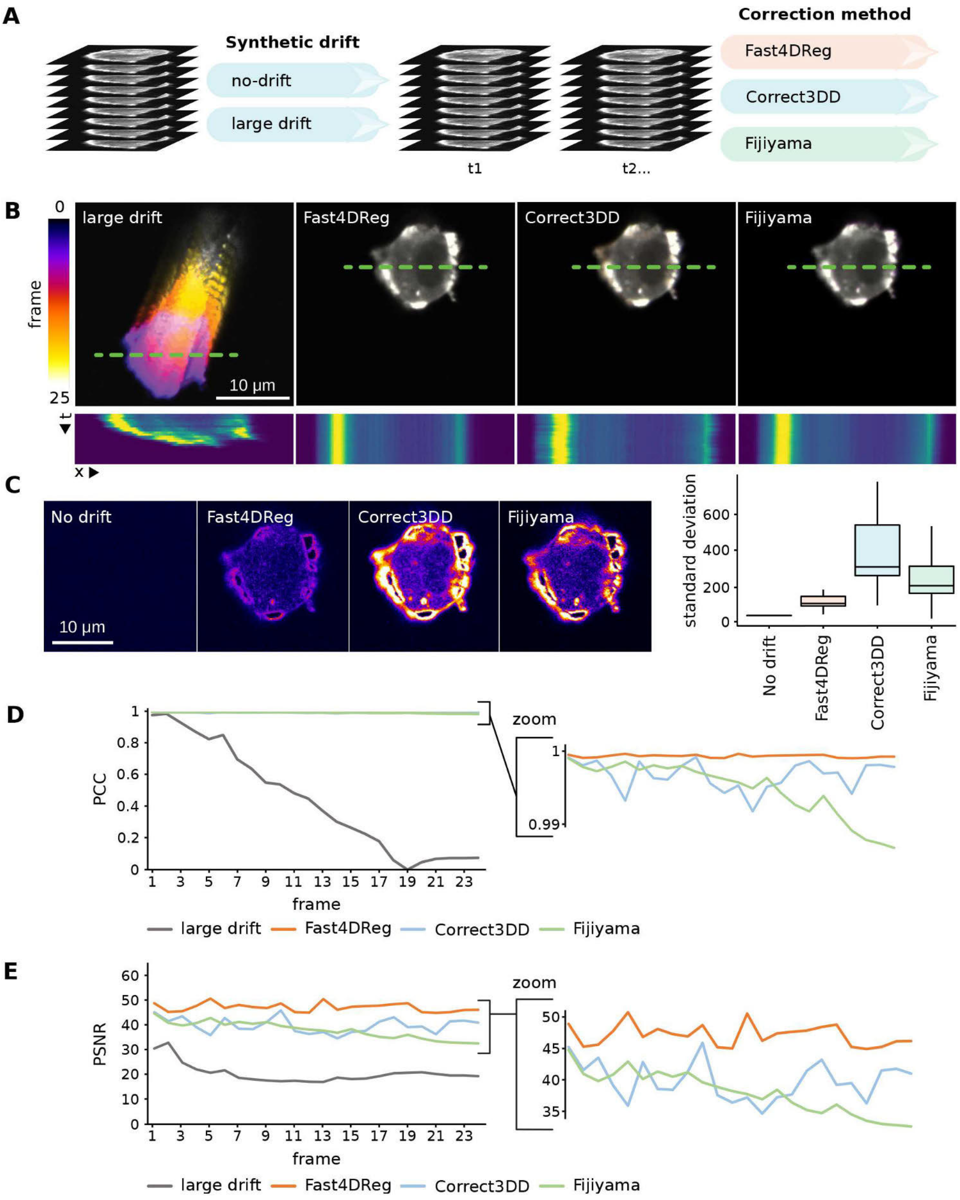


Fig. 2. See next page for legend.

Fast4DReg is relatively resistant to noise

Live fluorescence imaging often requires using low illumination levels to avoid harming the sample, which can result in the acquisition of noisy images. In order to evaluate the sensitivity of Fast4DReg to noise, synthetic datasets with varying levels of noise were generated and processed using Fast4DReg. To assess and compare the results using image-similarity metrics, we applied the drift tables calculated by Fast4DReg to the original dataset, then calculated the image-similarity metrics on these corrected datasets (Fig. 3A,B). These analyses indicated that Fast4DReg was not affected by noise when the signal-to-noise ratio (SNR) was above 2. When the SNR was below 2, a decrease in

Fig. 2. Fast4DReg outperforms Correct3DD and Fijiyama on a synthetic dataset. (A) Two synthetic 3D video datasets were created, one with no drift and another with a large amount of drift (see also Fig. S1). The large drift dataset was then corrected using Fast4DReg, Correct3DD and Fijiyama. (B) The drift-correction performance of the three algorithms was assessed using temporal color projections of a selected *z*-slice (middle of the cell) and kymographs (along the green dashed lines; dimensions, 25 µm × 25 frames). (C) Standard deviation time projection of the middle slice of the cell. This projection takes the standard deviation of the pixel intensities through time. Positions with large differences in the pixel intensities through the stack appear brighter in this projection. Therefore, a black image highlights no variation between the frames over time (perfect drift correction), whereas signals highlight slight errors in the drift correction. For each *z*-slice, the standard deviation projection over time was generated and quantified using Fiji, and the results are shown as boxplots created by PlotsOfData (Postma and Goedhart, 2019), in which the boxes show the 25th and 75th percentiles, the whiskers represent the minimum and maximum values, and the median is marked with a line. No drift shows a high baseline value as specified noise was added during background homogenization. (D,E) The Pearson's correlation coefficient (PCC) (D) and peak signal-to-noise ratio (PSNR) (E) between the first and each subsequent frame were calculated. For PSNR, a higher value indicates a better drift correction. For Pearson's correlation coefficient, a value of 1 indicates perfect drift correction. For all panels, scale bars: 10 µm.

performance could be observed. Interestingly, when the images started to be very noisy (SNR below 1.6), Fast4DReg performed much better when using average-intensity projections instead of maximum-intensity projections (Fig. 3C–F). Taken together, these results indicate that Fast4DReg is relatively resistant to noise and that, when correcting noisy data, it is more effective to use average-intensity projections.

Fast4DReg is fast and successfully corrects drift from acquired 3D videos

Next, we assessed the suitability of Fast4DReg to correct drifts in acquired 3D biological images. We used a long 3D video of a human umbilical vein endothelial cell (HUVEC) monolayer labeled with silicon rhodamine (SiR)-actin and imaged using an Airyscan confocal microscope. We also registered this dataset with Correct3DD and Fijiyama. Although both Fast4DReg and Correct3DD produced good results when assessed visually (Movie 3), we failed to generate meaningful results with Fijiyama as the processing made the video drift even more than the raw data (data not shown).

To estimate the correction efficiency of Fast4DReg and Correct3DD on this dataset, we first searched for a structure that should remain immobile across multiple time points in the movie and chose a large stress fiber. We then color-coded three consecutive frames (one color per frame) and observed the overlaps of this stable structure between frames using line profiles (Fig. 4A). In the uncorrected movie, the stress fiber did not overlap in these three frames, clearly indicating drift. In the movies corrected by Fast4DReg and Correct3DD, the stress fiber overlap between frames improved, showing that the drift correction worked in both cases. Interestingly, the drift correction provided by Fast4DReg was superior here as the stress fiber overlap between the three frames was greater (Fig. 4A).

To visualize the axial drift-correction efficiency of Fast4DReg and Correct3DD on this dataset, we generated kymographs from the *y*-projections (Fig. 4B). In the original data, the kymograph showed a clear band pattern due to the microscope stage jumping cyclically. This banding pattern was improved in the movies corrected by Fast4DReg and Correct3DD. Still, it did not entirely disappear, indicating that although both registration methods worked well on this dataset, the correction was not perfect (Fig. 4B; Movie 3). This was perhaps because part of the data went out of the imaging volume several times. Of note, Correct3DD processing led to the monolayer slowly sinking over time, which is less desirable.

To obtain a quantitative estimate of the performance of Fast4DReg and Correct3DD on the HUVEC dataset, we measured the image-similarity metric between each adjacent frame pair for a selected *z*-plane in the corrected videos. Indeed, efficient drift correction should make successive frames more similar despite the inevitable biological changes. These biological changes would, however, lead to lower image-similarity metrics than those measured with our synthetic dataset (Figs 2 and 3), even if the data was perfectly registered. Using these metrics, we found that Fast4DReg and Correct3DD improved adjacent frame similarity on this dataset and performed similarly (Fig. 4C).

Finally, we assessed the computing time required by Fast4DReg, Correct3DD and Fijiyama to process the HUVEC dataset using two different computers. We found that Fast4DReg was four to nine times faster than Correct3DD and 20 to 90 times faster than Fijiyama when correcting the HUVEC dataset (Fig. 4D). These differences are relevant as the registration of datasets often requires tweaking hyperparameters to obtain the best possible results. Overall, Fast4DReg outperformed Correct3DD when correcting the HUVEC dataset, as the corrected data did not sink over time and the processing time was faster.

Fast4DReg can register multichannel 3D videos

Next, we assessed the ability of Fast4DReg to correct acquired multichannel 3D videos. We used a movie of cancer cells migrating inside the lung vasculature, which was imaged *ex vivo* using an Airyscan confocal microscope (Fig. 5A; Movies 4 and 5). In this dataset, both the cancer cells and the vasculature were labeled, and a noticeable *xyz*-drift perturbated the visualization (Fig. 5; Movies 4 and 5). To correct these data with Fast4DReg, we first estimated the drift using the vasculature images. Once the drift was estimated, the same drift tables were then applied to the cancer cell images (Fig. 5B). Visually, Fast4DReg corrected this dataset very well (Movies 4 and 5). Using time projection of a selected *z*-plane and line-intensity profiles, we found that Fast4DReg could indeed successfully register the images of the vasculature (Fig. 5C). Importantly, applying the same drift tables also efficiently corrected the drift in the cancer cell images (Fig. 5D). We believe that the ability of Fast4DReg to apply drift tables to other datasets will significantly simplify the registration of multichannel data.

Fast4DReg can also register misaligned 3D channel stacks

Finally, we tested whether Fast4DReg could be used to align 3D multichannel images instead of 3D videos. In this case, Fast4DReg uses the same pipeline as described for time series but first converts the channels into time frames.

To test this approach, we registered, using Fast4DReg, a three-channel 3D image of a calibration slide. In this dataset, the raw images displayed significant *xyz*-misalignment owing to chromatic aberrations and the fact that the channels were acquired using different cameras (Fig. 6A,B). Using line-intensity profiles, we found that Fast4DReg could successfully register this dataset laterally and axially (Fig. 6A,B). Importantly, the drift tables measured using the calibration slide could then be used to correct any microscopy images acquired using the same conditions (Fig. 6C,D). Combined with the Fast4DReg batch-processing mode, we envision that the indirect channel alignment approach

Journal of Cell Science (2023) 136, jcs260728. doi:10.1242/jcs.260728

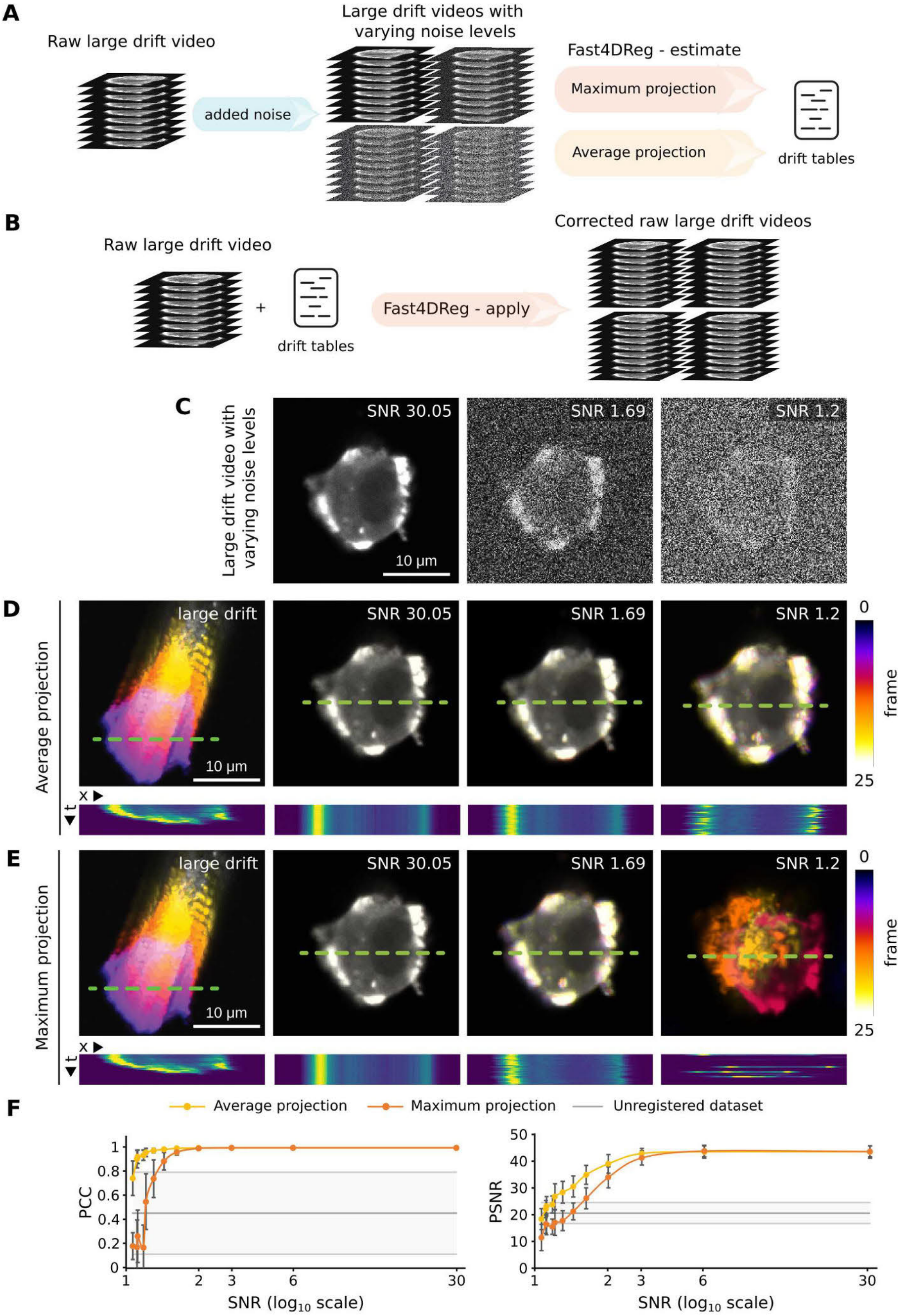


Fig. 3. See next page for legend.

Fig. 3. Fast4DReg is relatively resistant to noise. Twelve synthetic 3D video datasets with varying amounts of noise were created and corrected using Fast4DReg, either using maximum- or average-intensity projections. The drift tables were then applied to the original data to assess drift-correction accuracy. (A,B) Schematic illustrating the pipeline used to assess Fast4DReg sensitivity to noise. (C) Example of three noisy datasets used to assess Fast4DReg sensitivity to noise. (D,E) Fast4DReg drift-correction performance for three noisy datasets (C) was assessed using temporal color projections of a selected *z*-slice (middle of the cell) and kymographs (along the green dashed lines; dimensions, 25 µm × 25 frames). Note that Fast4DReg fails to register the images with an SNR of 1.2 when using maximum-intensity projections (E). (F) Fast4DReg drift-correction performance for the twelve noisy datasets was assessed using image-similarity metrics. The PSNR and PCC between the first and subsequent frames were calculated for each noise amount. For all panels, scale bars: 10 µm.

described here will be advantageous when performing colocalization analyses.

However, not all datasets have a corresponding calibration slide or bead image, so we also tested the ability of Fast4DReg to correct misaligned channels directly. In this case, we used a two-channel 3D image that was acquired to train a supervised image-restoration deep-learning algorithm (von Chamier et al., 2021; Weigert et al., 2018). As with the previous example, Fast4DReg performed well in registering this dataset (Fig. 6E,F). It is worth noting that a direct channel alignment approach might only work for some datasets, as Fast4DReg requires structural overlap between the channels to perform the registration. Additionally, we do not recommend directly aligning images aimed for colocalization analysis as it might lead to artefactual results.

DISCUSSION

In live microscopy, sample drifting can be a significant challenge, and implementing post-processing drift-correction pipelines is not always fast or straightforward. Here, we developed Fast4DReg, an ImageJ or Fiji-based tool that can quickly correct axial and lateral drift in 2D and 3D videos. We show that Fast4DReg can outperform two open-source 3D drift-correction tools on our test datasets. A significant advantage of Fast4DReg is that it can correct 3D videos in a fraction of the time compared to other tested tools and comes with an easy-to-use graphical user interface (Fig. S2). Fast4DReg speed is likely due to two factors: (1) using 2D projections greatly simplifies the computations required and (2) using CPU multithreading further accelerates the 2D cross-correlation process. In the future, Fast4DReg speed could be further improved by enabling graphic card acceleration.

Despite its performance, Fast4DReg has several limitations. Firstly, Fast4DReg can only perform translations when correcting a dataset. Rotation, scaling or shearing transformations are not supported, although these should not be required to correct most time-course video or multichannel microscopy datasets. Secondly, the channel alignment is limited to images with structural conservation between channels or requires calibration slides or beads images to compute the shift maps.

With Fast4DReg, we demonstrate that using intensity projections followed by 2D cross-correlation is a quick and efficient way to register various multidimensional data types, including 3D videos and 3D multichannel datasets. In the future, it would be interesting to assess the suitability of using 3D cross-correlation directly to register similar images. But using 3D cross-correlation will likely impede processing times.

To promote adoption by the community, Fast4DReg is available through a Fiji update site, GitHub and Zenodo. We also provide test datasets and detailed step-by-step instructions. With Fast4DReg, we hope to make the process of multidimensional data registration more straightforward and faster, and, therefore, more accessible to the community.

MATERIALS AND METHODS

Algorithms

To estimate the lateral or axial drift of a 3D video, Fast4DReg creates *z*- or *y*-intensity projections for each time point to create a 2D video. Fast4DReg then estimates the linear drift between the reference and moving frames by calculating their CCM. In Fast4DReg, in a similar fashion to the work of Sun (2002), the cross-correlation between two images is calculated by performing a discrete Hartley transform on both images, followed by a multiplication of one of the transformed images by the complex conjugate of the other. The result of this multiplication is then inversely transformed back to real space, generating the CCM. A bicubic spline interpolation is then used to upscale the CCM and achieve subpixel precision. The upscaled CCM is normalized by calculating the Pearson's correlation coefficient between the two images shifted according to the minimum and maximum values of the upscaled CCM. Finally, the linear shift between the two images is then calculated by taking the global maximum peak of the normalized up-scaled CCM (as demonstrated by Laine et al., 2019).

Fast4DReg can also be used to register channels from misaligned 3D stacks. In this case, Fast4DReg simply converts the channels into time frames before applying the Fast4DReg drift-correction pipeline described above. As a note of caution, cross-correlation only works well to register channels in which similar structures or cells are labeled. Importantly Fast4DReg can also register 2D video and 2D multichannel images either one at a time or in batches.

Fast4DReg can run on any computer on which Fiji (Schindelin et al., 2012) and the Bio-Formats (Linkert et al., 2010) plugin are installed. Fast4DReg also has a memory-saving mode that allows the registration of larger datasets using a computer with limited resources (processing time available in Fig. S1D).

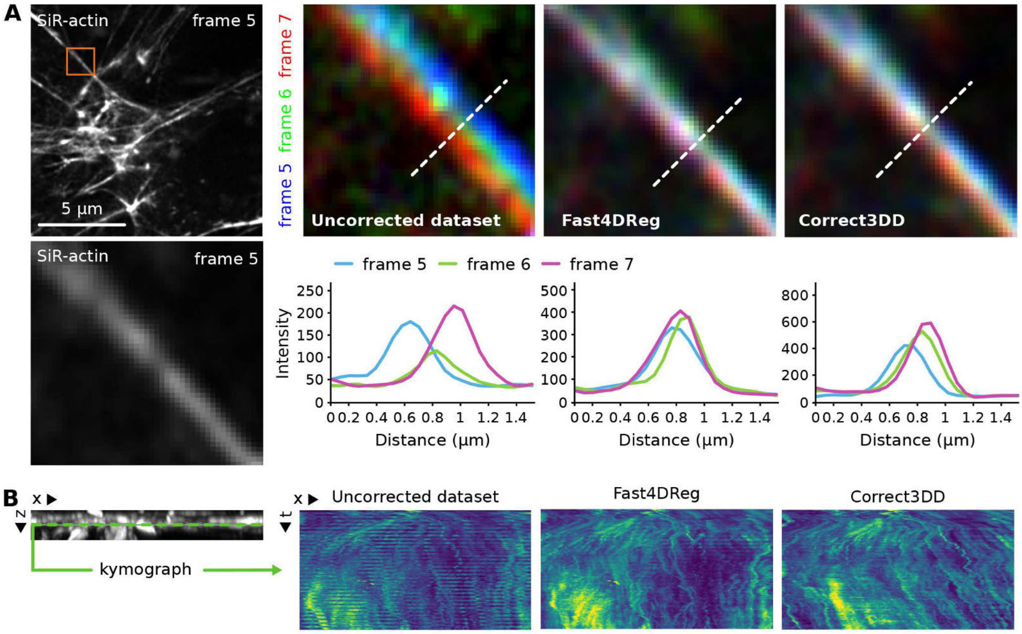
Fast4DReg expects as input one or multiple single-channel 2D or 3D videos. Fast4DReg outputs corrected files, drift tables, drift plots and a settings file. Owing to Bio-Formats (Linkert et al., 2010), Fast4DReg can handle various image formats as input. Fast4DReg can be tested using our test datasets available on Zenodo (https://zenodo.org/record/7514913). Fast4DReg is written using a combination of an ImageJ macro and Java, and is distributed via an ImageJ update site. The installation procedure and up-to-date, step-by-step instructions are available on the Fast4DReg GitHub page (https://github.com/guijacquemet/Fast4DReg).

Cells

AsPC1 cells (a pancreatic ductal adenocarcinoma cell line) were purchased from American Type Culture Collection (CRL-1682) and grown in Roswell Park Memorial Institute medium (Thermo Fisher Scientific, 11875093) supplemented with 10% fetal bovine serum (FBS) (Biowest, S1860). HUVECs were purchased from PromoCell (C-12203) and grown in endothelial cell growth medium (PromoCell, C-22010). U2-OS osteosarcoma cells were purchased from the Leibniz Institute DSMZ, German Collection of Microorganisms and Cell Cultures (Braunschweig, Germany; ACC 785) and grown in Dulbecco's modified Eagle medium (DMEM; Merck, D5671) supplemented with 10% FBS. MCF10DCIS.com Lifeact–RFP cells were generated previously (Jacquemet et al., 2017) and cultured in a 1:1 mix of DMEM and F12 (Merck, 51651C) supplemented with 5% horse serum (Gibco, 16050122), 20 ng/ml human epidermal growth factor (Merck, E9644), 0.5 mg/ml hydrocortisone (Merck, H0888-1G), 100 ng/ml cholera toxin (Merck, C8052-1MG), 10 mg/ml insulin (Merck, I9278-5ML) and 1% (vol/vol) penicillin/streptomycin (Merck, P0781-100ML). All cell lines tested negative for mycoplasma. AsPC1 cells were authenticated by DSMZ. All other cell lines were not authenticated.

Datasets with synthetic drift

The synthetic drift datasets were created by duplicating a 3D stack image 25 times and artificially adding a known amount of *x*-, *y*- and *z*-drift between



C

Metric	Uncorrected dataset	Fast4DReg	Correct3DD	Fijiyama
PCC	0.8112 ± 0.072	0.9199 ± 0.026	0.9161 ± 0.043	N.D
mSSIM	0.676 ± 0.069	0.782 ± 0.039	0.773 ± 0.051	N.D
NRMSE	0.135 ± 0.0135	0.135 ± 0.0135	0.140 ± 0.029	N.D
PSNR	24.926 ± 2.15	28.496 ± 1.543	28.086 ± 2.136	N.D

D

	Fast4DReg	Correct3DD	Fijiyama
Computer 1	1.4 min	12.6 min	123 min
Computer 2	5.9 min	24.2 min	124 min

Fig. 4. Fast4DReg can rapidly correct axial and lateral drift in 3D videos. A 3D video of HUVECs cells labeled with SiR-actin displaying a large *xyz*-drift was corrected with Fast4DReg and Correct3DD. (A) A region of interest containing a stress fiber that should remain immobile across multiple time points was chosen. Three consecutive frames were pseudo-colored blue, green and red, and merged. White indicates structural overlaps between the three frames. Line profiles along the dashed white lines to further study the overlap between frames were drawn as shown. (B) A kymograph (dimensions, 15 µm × 7146 s) of a selected line (monolayer ventral plane) in the *y*-projection was created to visualize the *z*-drift over time. (C) Image similarity metrics were calculated between each consecutive frame pair. Values show the mean±s.d. PCC, Pearson's correlation coefficient; mSSIM, mean structural similarity index; NRMSE, normalized root mean squared error; PSNR, peak signal-to-noise ratio. N.D., not determined. (D) Two computers, computer 1 (a high-performance desktop) and computer 2 (a laptop), were used to measure the speed of the correction methods. Shown values are the average times of three measurements.

each frame. The original image was that of an AsPC1 cell expressing Lifeact–mScarletI migrating inside the vasculature of a zebrafish embryo. This image was acquired using a 3i CSU-W1 spinning-disk confocal microscope equipped with a 40× water immersion objective (NA 1.15) and a Hamamatsu sCMOS Orca Flash camera. The microscope was controlled using the Slidebook 6 software (Intelligent Imaging Innovations, Inc.; https://www.intelligent-imaging.com/slidebook).

The amount of drift added corresponds to the drift typically observed in our live-cell imaging experiments. Using this method, two videos were created: one with no drift (ground-truth video) and one with a large drift (across the field of view) (see also Fig. S1A). After the drift was simulated, the image background was made homogeneous via pixel intensity subtraction and by adding specified noise using Fiji ('add specified noise' function). The *xy*- and *z*-drift in these synthetic datasets was corrected using Fast4DReg, Correct3DD and Fijiyama. For each software, the parameters providing the best possible drift correction were chosen (the settings used are described in Table S1).

After correcting the drift in the synthetic datasets, the images were first cropped to be the same size (352×275 pixels, 69 *z*-slices, 25 frames) using Fiji. The drift-correction performance was then quantified by measuring image-similarity metrics between frames (the reference frame was the first frame) of a selected z-slice (*z*-slice 51) using a custom-made Jupyter notebook (available in Zenodo). This *z*-slice was selected as it was in the middle of the cell.

The noisy synthetic drift dataset

To generate the 12 noisy synthetic drift datasets, a specified amount of Gaussian noise was added to the original synthetically drifting dataset using Fiji ('add specified noise' function). The added selected Gaussian noise had

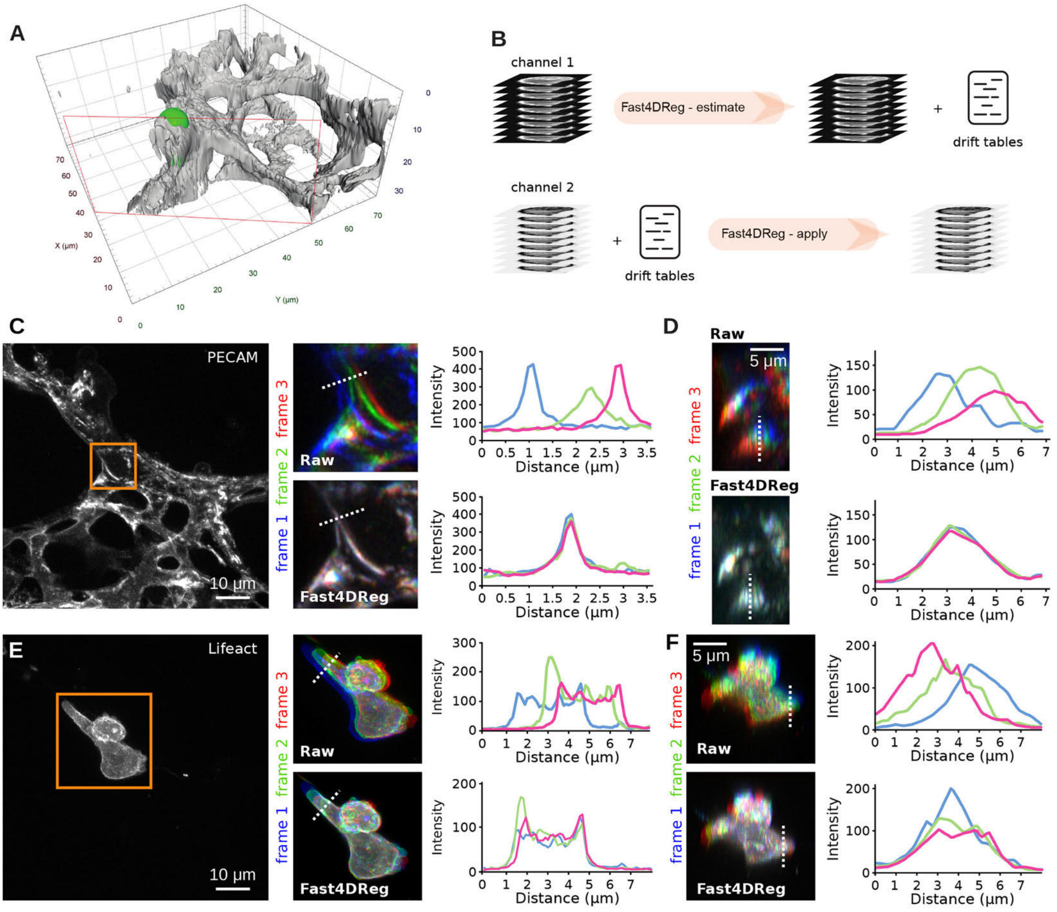


Fig. 5. Registration of a 3D multichannel video using Fast4DReg. A 3D multichannel video of cancer cells migrating inside the lung vasculature was corrected using Fast4DReg. (A) 3D surface rendering of a selected time point (see also Movie 4) created using Arivis Vision4D. The lung vasculature is shown in grey, and the cancer cell (AsPC1) is in green. The red rectangle indicates the clipping plane used to render the image. (B) Schematic illustrating the pipeline used to correct a multichannel 3D video using Fast4DReg. For this dataset, the drift was first estimated using the vasculature images (channel 1), and the resulting drift table was then applied to the cancer cell images (channel 2). (C–E) Three consecutive frames of the vasculature (C) and cancer cell images (D) were pseudo-colored blue, green and red, and merged. White indicates structural overlaps between the three frames. Line profiles along the dotted white lines to further study the overlap between frames were drawn as shown. (C,E) *Z*-projections are displayed to visualize the lateral misalignments corrected by Fast4DReg. Scale bars: 10 µm. (D,F) *Y*-projections are displayed to visualize the axial misalignment corrected by Fast4DReg. Scale bars: 5 µm.

standard deviations of 0, 5000, 10,000, 15,000, 20,000, 25,000, 30,000, 35,000, 40,000, 45,000, 50,000 and 60,000, yielding images with SNR of 30.053, 5.586, 2.964, 2.111, 1.686, 1.478, 1.327, 1.227, 1.196, 1.127, 1.119 and 1.070, respectively. The SNR was calculated by dividing the mean cell signal by the mean background signal. All 12 generated noisy synthetic drifts were corrected using Fast4DReg using maximum- or average-intensity projections (the settings used are described in Table S1). The generated drift tables were then used to correct the original large drift dataset (Fig. 3A,B). Corrected images were then cropped to be the same size (192×192 pixels, 69 *z*-slices, 25 frames) using Fiji. The drift-correction performance was then quantified by measuring image-similarity metrics between frames (the reference frame was the first frame) of a selected *z*-slice (z-slice 51) using a custom-made Jupyter notebook (available in Zenodo). This *z*-slice was selected as it was in the middle of the cell.

The HUVEC monolayer dataset

The HUVEC monolayer dataset consists of a 3D video of HUVECs labeled with SiR-actin (Spirochrome). The video was acquired using a laser scanning confocal LSM880 microscope (Zeiss) equipped with an Airyscan detector (Carl Zeiss) and a 63× oil (NA 1.4) objective. The microscope was controlled using Zen Black (2.3) (Zeiss), and the Airyscan detector was used in standard super-resolution mode. This dataset has 200 frames (488×488 pixels) and 24 *z*-slices. This dataset was corrected using Fast4DReg, Correct3DD and Fijiyama using the parameters providing the best possible drift correction (the settings used are described in Table S1). The correction performance was quantified by measuring image-similarity metrics between adjacent frames (the reference frame was the previous frame) of a selected *z*-slice (*z*-slice 8) using a custom-made Jupyter notebook.

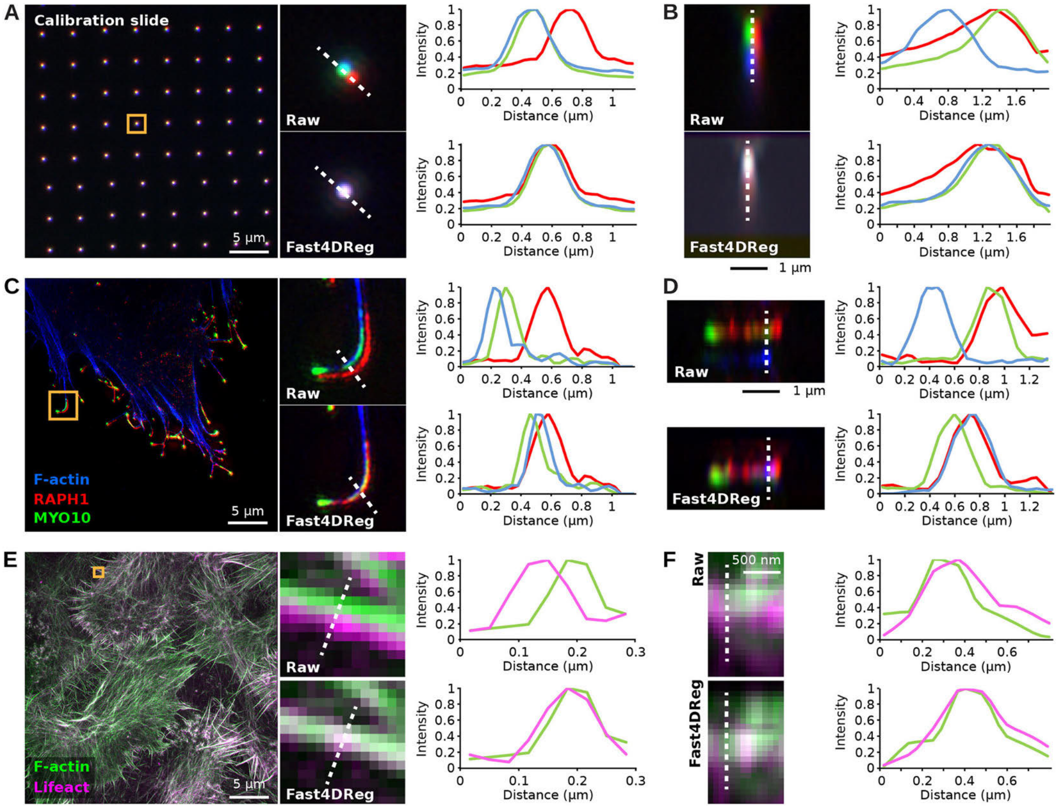


Fig. 6. Fast4DReg can align 3D multichannel images. (A,B) A three-channel 3D calibration slide image was aligned using Fast4DReg. Merged images and line-intensity profiles (along the dashed white lines) are displayed to highlight the level of overlap between the three channels. (A) A *z*-projection is displayed to visualize the lateral misalignment corrected by Fast4DReg. (B) A *y*-projection of one of the calibration slide spots is displayed to illustrate the axial misalignment corrected by Fast4DReg. (C,D) The drift table generated in A,B was then used to correct a 3D SIM image of a U2-OS cell expressing GFP-tagged lamellipodin (RAPH1, red) and MYO10–mScarlet (green), and labeled to visualize its actin cytoskeleton (blue). (C) A *z*-projection is displayed to visualize the lateral misalignment, evident in filopodia, corrected by Fast4DReg. (D) A *y*-projection of one filopodium visualizes the axial misalignment corrected by Fast4DReg. (E,F) A 3D SIM image of MCF10DCIS.com cells expressing RFP–Lifeact (magenta) and stained to visualize F-actin (green) was aligned using Fast4DReg directly. (E) A *z*-projection is displayed to visualize the lateral misalignment corrected by Fast4DReg. (F) A *y*-projection visualizes the slight axial misalignment corrected by Fast4DReg.

Two computers were used to compare the execution times of all compared methods: computer 1 (operating system, Windows; processor, AMD Ryzen 7 5800X 8-Core; graphics card, GeForce GTX 3080; RAM, 32 GB; Fiji version 1.53q) and computer 2 [operating system, macOS; processor, M1 chip (8-core CPU, 8-core GPU); RAM: 16 GB; Fiji version 1.53q].

The mouse lung dataset

The mouse lung dataset (624×626 pixels, 55 *z*-slices, eight frames, two channels) consists of a 3D video of an AsPC1 cell expressing Lifeact–mNeonGreen migrating inside the lung vasculature. Briefly, labeled AsPC1 cells were injected into the tail vein of a 10-week-old immunocompromised female mouse (Hsd: Athymic Nude-*Foxn1*^{nu} strain). The mouse was euthanized shortly after injection, and precision-cut lung slices were prepared (Puttur et al., 2019). The National Animal Experiment Board authorized all animal studies, and per The Finnish Act on Animal Experimentation (animal license number 12558/2021). The lung endothelial cells were labeled using an Alexa Fluor 488-conjugated anti-PECAM antibody (1:100, BioLegend, 102413). The precision-cut lung slices were then imaged using an Airyscan confocal LSM880 microscope (Carl Zeiss) equipped with a 63× water (NA 1.15) objective. The *xy*- and *z*-drift in this dataset was corrected with Fast4DReg using the PECAM staining (settings used described in Table S1). The drift table generated by Fast4DReg was then applied to the second channel (cancer cells), after which these channels were merged using Fiji. Line-intensity profiles of three consecutive frames of selected structures were measured using Fiji to quantify the correction. Arivis Vision4D (v 3.5.0, Zeiss, https://www.zeiss.com/microscopy/en/products/software/arivis-vision4d.html) was used for the 3D reconstruction of the time-lapse movies. Surface rendering was performed using the 'Extract Isosurface' function.

The calibration slide dataset

The calibration slide dataset (1024×1024 pixels, 25 *z*-slices, three channels) was created by imaging a channel calibration slide (Argolight HM) using a DeltaVision OMX v4 (GE Healthcare Life Sciences) microscope fitted with a 60× Plan-Apochromat objective lens, 1.42 NA (immersion oil refractive index of 1.516), used in wide-field illumination mode. The emitted light was collected on a front-illuminated pco.edge sCMOS camera (pixel size 6.5 mm, readout speed 95 MHz; PCO) controlled by SoftWorx (AppliedPrecision). The *xy*- and *z*-drift in this dataset was corrected using Fast4DReg (the settings used are described in Table S1).

The filopodia dataset

The filopodia dataset (1024×1024 pixels, 17 *z*-slices, three channels) consists of a 3D structured illumination microscopy (SIM) image of a U2-OS cell expressing GFP-tagged lamellipodin and MYO10–mScarlet, and labelled with SiR-actin (Popovic et al., 2023). This dataset was acquired using a DeltaVision OMX v4 (GE Healthcare Life Sciences) microscope fitted with a 60× Plan-Apochromat objective lens, 1.42 NA (immersion oil refractive index of 1.516) used in SIM illumination mode (five phases and three rotations). The emitted light was collected on a front-illuminated pco.edge sCMOS camera (pixel size 6.5 mm, readout speed 95 MHz; PCO) controlled by SoftWorx. The *xy*- and *z*-drift in this dataset was corrected using Fast4DReg using the drift table computed using the calibration slide dataset (the settings used are described in Table S1).

The DCIS.com filopodia dataset

The DCIS.com Filopodia dataset consists of a SIM image of MCF10DCIS.COM Lifeact–RFP cells labeled with phalloidin. Briefly, MCF10DCIS.COM Lifeact–RFP cells (Jacquemet et al., 2017) were grown on high-tolerance glass-bottomed dishes (MatTek Corporation, coverslip 1.5). Cells were fixed and permeabilized simultaneously using 4% (wt/vol) paraformaldehyde and 0.25% (vol/vol) Triton X-100 for 10 min. Cells were then washed with PBS, quenched using a solution of 1 M glycine for 30 min, and incubated with Alexa Fluor 488 phalloidin (1:200 in PBS; A12379, Thermo Fisher Scientific) at 4°C overnight until imaging. Samples were washed three times in PBS, mounted in Vectashield (Vectorlabs), and imaged using a DeltaVision OMX v4 (GE Healthcare Life Sciences) used in SIM illumination mode (five phases and three rotations). The microscope was fitted with a ×60 Plan-Apochromat objective lens (1.42 NA, immersion oil refractive index of 1.516). The fluorescent light was collected using front-illuminated pco.edge sCMOS camera (pixel size 6.5 µm, readout speed 95 MHz; PCO). The high SNR ratio images were acquired from the phalloidin-488 staining using acquisition parameters optimal to obtain high-quality SIM images (50 ms of exposure time, 10% laser power). The low SNR ratio images were acquired from the Lifeact-RFP channel using acquisition parameters more suitable for live-cell imaging (100 ms of exposure time, 1% laser power). The *xy*- and *z*-drift in this dataset was corrected using Fast4DReg (the settings used are described in Table S1).

Metrics

To quantitatively assess the drift-correction performance of Fast4DReg and the other tools tested, four image-similarity metrics were used. These metrics were calculated using a custom-made Jupyter notebook (modified from Laine et al., 2021). This notebook is available on Zenodo (https://zenodo.org/record/7514913).

Pearson's correlation coefficient (PCC) measures the linear correlation between two images. A PCC value of 1 indicates a perfect linear relationship, or perfect similarity, between the two images. The mean structural similarity index (mSSIM) evaluates the similarity of two images based on their contrast, luminance and structural content. An mSSIM value of 1 indicates that the two images are perfectly similar. The peak SNR ratio (PSNR) is a metric that compares the peak signal amplitudes of two images and is typically expressed in decibels. A higher PSNR value indicates greater similarity between the two images. The normalized root mean squared error (NRMSE) measures the average difference between the pixel intensity in two images. A lower NRMSE value indicates greater similarity between the images.

Fast4DReg downloads and source code

Fast4DReg, the generator of synthetic drift, and the notebook used to make the image-similarity measurements (all under MIT licenses) are available on GitHub (https://github.com/guijacquemet/Fast4DReg) and their source code is archived on Zenodo (https://zenodo.org/record/7514913). Fast4DReg is also available through a Fiji update site (https://imagej.net/plugins/fast4dreg).

Acknowledgements
We acknowledge the Cell Imaging and Cytometry Core facility (Turku Bioscience, University of Turku, Åbo Akademi University and Biocenter Finland) and Turku Bioimaging for their services, instrumentation and expertise. We thank Junel Solis and Anting Li for testing Fast4DReg and providing feedback.

Competing interests
The authors declare no competing or financial interests.

Author contributions
Conceptualization: R.F.L., G.J.; Methodology: J.W.P., R.F.L., G.J.; Software: J.W.P., R.F.L., B.M.S.S., R.H., G.J.; Validation: G.J.; Formal analysis: J.W.P., G.J.; Investigation: J.W.P., S.G., G.F., G.J.; Resources: G.J.; Data curation: J.W.P., G.J.; Writing - original draft: J.W.P., G.J.; Writing - review & editing: J.W.P., R.F.L., B.M.S.S., G.F., R.H., G.J.; Visualization: J.W.P., S.G., G.F., G.J.; Supervision: R.H., G.J.; Project administration: G.J.; Funding acquisition: R.H., G.J.

Funding
This study was supported by the Academy of Finland (338537 to G.J.), the Cancer Society of Finland (Syöpäjärjestöt; to G.J.) and the Solutions for Health strategic funding to Åbo Akademi University (to G.J.). J.W.P. was supported by Health Campus Turku 2.0 funded by the Academy of Finland. R.F.L. was supported by a Medical Research Council Skills development fellowship (MR/T027924/1). G.F. was supported by an Academy of Finland postdoctoral fellowship (332402). R.H. is supported by the Gulbenkian Foundation (Fundação Calouste Gulbenkian) and received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement number 101001332), the European Molecular Biology Organization (EMBO) Installation Grant (EMBO-2020-IG4734) and the Chan Zuckerberg Initiative Visual Proteomics Grant (vpi-0000000044). This research was supported by InFLAMES Flagship Programme of the Academy of Finland (decision number: 337531). Open access funding provided by the University of Turku. Deposited in PMC for immediate release.

Data availability
All datasets used in this study and the code used to generate them are available on Zenodo (https://zenodo.org/record/7514913).

First Person
This article has an associated First Person interview with the first author of the paper.

Peer review history
The peer review history is available online at https://journals.biologists.com/jcs/article-lookup/doi/10.1242/jcs.260728.reviewer-comments.pdf

References

Fernandez, R. and Moisy, C. (2021). Fijiyama: a registration tool for 3D multimodal time-lapse imaging. *Bioinformatics* **37**, 1482-1484. doi:10.1093/bioinformatics/btaa846

Fox, Z. R., Fletcher, S., Fraisse, A., Aditya, C., Sosa-Carrillo, S., Petit, J., Gilles, S., Bertaux, F., Ruess, J. and Batt, G. (2022). Enabling reactive microscopy with MicroMator. *Nat. Commun.* **13**, 1. doi:10.1038/s41467-022-29888-z

Jacquemet, G., Paatero, I., Carisey, A. F., Padzik, A., Orange, J. S., Hamidi, H. and Ivaska, J. (2017). FiloQuant reveals increased filopodia density during breast cancer progression. *J. Cell Biol.* **216**, 3387-3403. doi:10.1083/jcb.201704045

Klein, S., Staring, M., Murphy, K., Viergever, M. A. and Pluim, J. P. W. (2010). elastix: a toolbox for intensity-based medical image registration. *IEEE Trans. Med. Imaging* **29**, 196-205. doi:10.1109/TMI.2009.2035616

Laine, R. F., Tosheva, K. L., Gustafsson, N., Gray, R. D. M., Almada, P., Albrecht, D., Risa, G. T., Hurtig, F., Lindås, A.-C., Baum, B. et al. (2019). NanoJ: a high-performance open-source super-resolution microscopy toolbox. *J. Phys. D Appl. Phys.* **52**, 163001. doi:10.1088/1361-6463/ab0261

Laine, R. F., Arganda-Carreras, I., Henriques, R. and Jacquemet, G. (2021). Avoiding a replication crisis in deep-learning-based bioimage analysis. *Nat. Methods* **18**, 1136-1144. doi:10.1038/s41592-021-01284-3

Linkert, M., Rueden, C. T., Allan, C., Burel, J.-M., Moore, W., Patterson, A., Loranger, B., Moore, J., Neves, C., Macdonald, D. et al. (2010). Metadata matters: access to image data in the real world. *J. Cell Biol.* **189**, 777-782. doi:10.1083/jcb.201004104

Mccormick, M., Liu, X., Ibanez, L., Jomier, J. and Marion, C. (2014). ITK: enabling reproducible research and open science. *Front. Neuroinform.* **8**, 13. doi:10.3389/fninf.2014.00013 https://www.frontiersin.org/articles/10.3389/fninf.2014.00013.

Parslow, A., Cardona, A. and Bryson-Richardson, R. J. (2014). Sample drift correction following 4D confocal time-lapse imaging. *J. Vis. Exp.* **86**, 51086. doi:10.3791/51086

Popovic, A., Miihkinen, M., Ghimire, S., Saup, R., Grönloh, M., Ball, N., Goult, B.T., Ivaska, J. and Jacquemet, G. (2023). Myosin-X recruits lamellipodin to filopodia tips. *J. Cell Sci.* **136**, jcs260574. doi:10.1242/jcs.260574

Postma, M. and Goedhart, J. (2019). PlotsOfData—A web app for visualizing data together with their summaries. *PLoS Biol.* **17**, e3000202. doi:10.1371/journal.pbio.3000202

Preibisch, S., Saalfeld, S., Schindelin, J. and Tomancak, P. (2010). Software for bead-based registration of selective plane illumination microscopy data. *Nat. Methods* **7**, 6. doi:10.1038/nmeth0610-418

Preibisch, S., Amat, F., Stamataki, E., Sarov, M., Singer, R. H., Myers, E. and Tomancak, P. (2014). Efficient Bayesian-based multiview deconvolution. *Nat. Methods* **11**, 645-648. doi:10.1038/nmeth.2929

Puttur, F., Denney, L., Gregory, L. G., Vuononvirta, J., Oliver, R., Entwistle, L. J., Walker, S. A., Headley, M. B., Mcghee, E. J., Pease, J. E. et al. (2019). Pulmonary environmental cues drive group 2 innate lymphoid cell dynamics in mice and humans. *Sci. Immunol.* **4**, eaav7638. doi:10.1126/sciimmunol.aav7638

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B. et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676-682. doi:10.1038/nmeth.2019

Sun, H. (2002). The Hartley transform applied to particle image velocimetry. *Meas. Sci. Technol.* **13**, 1996. doi:10.1088/0957-0233/13/12/326

Von Chamier, L., Laine, R. F., Jukkala, J., Spahn, C., Krentzel, D., Nehme, E., Lerche, M., Hernández-Pérez, S., Mattila, P. K., Karinou, E. et al. (2021). Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat. Commun.* **12**, 2276. doi:10.1038/s41467-021-22518-0

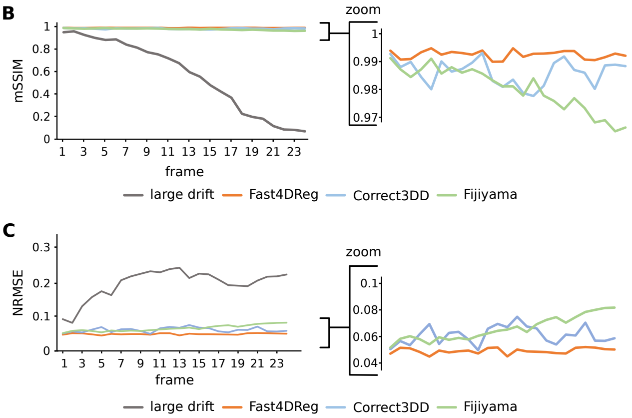
Von Wangenheim, D., Hauschild, R., Fendrych, M., Barone, V., Benková, E. and Friml, J. (2017). Live tracking of moving samples in confocal microscopy for vertically grown roots. *ELife* **6**, e26792. doi:10.7554/eLife.26792

Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S. et al. (2018). Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nat. Methods* **15**, 1090-1097. doi:10.1038/s41592-018-0216-7

A

Large drift (pixels)

Frame	X-drift	Y-drift	Z-drift
1	-0.1846	-0.842	0.7736
2	1.1534	0.69	-1.0418
3	0.3099	1.261	-0.5265
4	1.3837	1.961	-2.517
5	2.8785	1	-3.3751
6	3.221	-0.072	-4.0643
7	4.83	-2.012	-2.4789
8	6.2691	-4.293	-4.9687
9	7.1845	-7.21	-5.6736
10	8.8191	-11.375	-7.2127
11	12.1096	-13.948	-6.6842
12	13.6142	-18.579	-7.6793
13	15.5335	-25.244	-7.4236
14	18.8507	-32.933	-7.7894
15	21.5718	-37.042	-9.3144
16	25.3066	-45.681	-8.9694
17	28.4077	-52.989	-9.1761
18	32.8398	-60.376	-9.9103
19	34.7591	-71.216	-10.8287
20	40.0535	-79.337	-10.9413
21	42.8704	-90.376	-11.5039
22	47.4757	-100.918	-13.4116
23	51.8525	-112.391	-14.2919
24	56.8755	-124.903	-14.3824
25	62.7437	-137.609	-14.8706

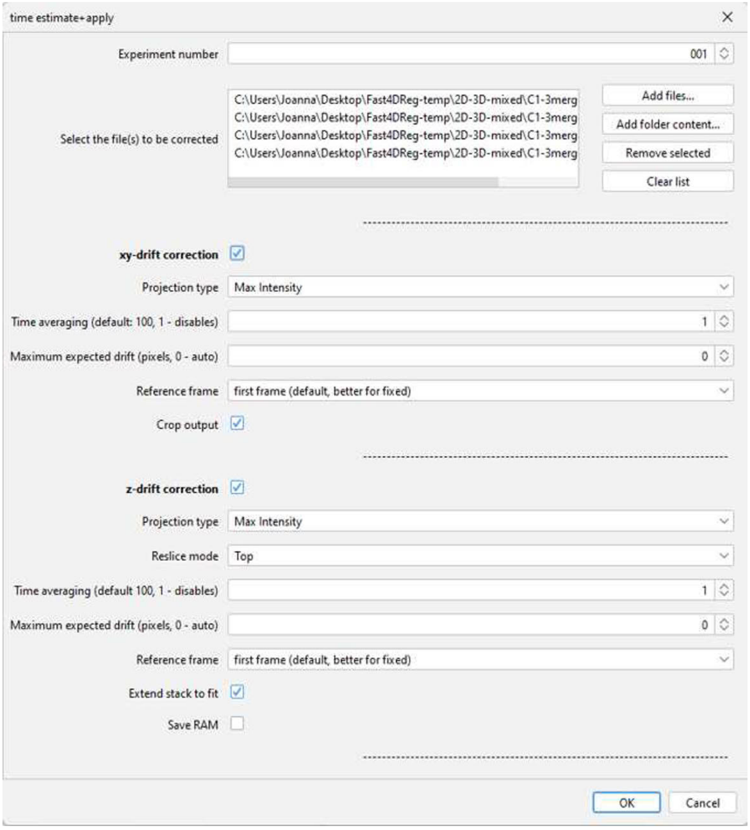


D

	Computer 1	Computer 2
Fast4DReg	1.4 min	5.9 min
Fast4DReg (RAM saving mode)	1.15 min	4.5 min

Fig. S1. Fast4DReg outperforms Correct3DD and Fijiyama on a synthetic 3D + t dataset. (**A-C**) Two synthetic 3D video datasets were created, one with no drift, and another with a large amount of drift (see also Fig. 2). (**A**) Drift table indicating the amount of drift added to create the large drift dataset. (**B and C**) The large drift dataset was then corrected using Fast4DReg, Correct3DD, and Fijiyama, and the registration quality assed using image similarity metrics. The mSSIM (**B**) and NRMSE metrics (**C**) between the first and each subsequent frame were calculated. For mSSIM a value of 1 indicates perfect drift correction, for NRMSE, a lower value indicates a better drift correction. (**D**) Comparison of Fast4DReg processing time when the RAM saving mode is enabled (HUVEC monolayer -dataset).

J. Cell Sci.: doi:10.1242/jcs.260728: Supplementary information



Experiment number: 001

Select the file(s) to be corrected:
C:\Users\Joanna\Desktop\Fast4DReg-temp\2D-3D-mixed\C1-3merg
C:\Users\Joanna\Desktop\Fast4DReg-temp\2D-3D-mixed\C1-3merg
C:\Users\Joanna\Desktop\Fast4DReg-temp\2D-3D-mixed\C1-3merg
C:\Users\Joanna\Desktop\Fast4DReg-temp\2D-3D-mixed\C1-3merg

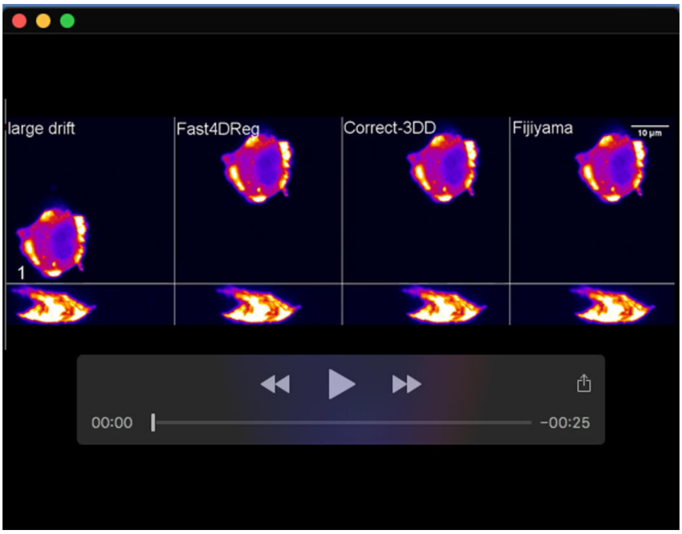
xy-drift correction ☑
Projection type: Max Intensity
Time averaging (default: 100, 1 - disables): 1
Maximum expected drift (pixels, 0 - auto): 0
Reference frame: first frame (default, better for fixed)
Crop output ☑

z-drift correction ☑
Projection type: Max Intensity
Reslice mode: Top
Time averaging (default 100, 1 - disables): 1
Maximum expected drift (pixels, 0 - auto): 0
Reference frame: first frame (default, better for fixed)
Extend stack to fit ☑
Save RAM ☐

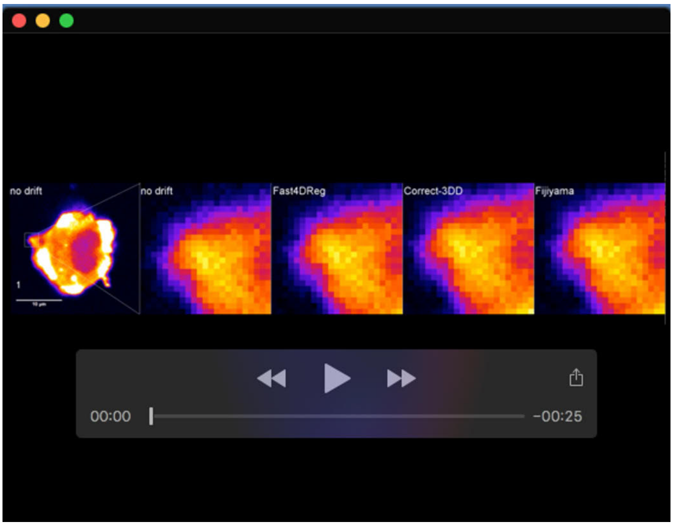
Fig. S2. Fast4DReg user interface. Screenshot highlighting Fast4DReg user interface.

Table S1. Table S1 lists the Fast4DReg, Corect3DD, and Fijiyama settings used to analyze the various datasets presented in this study.

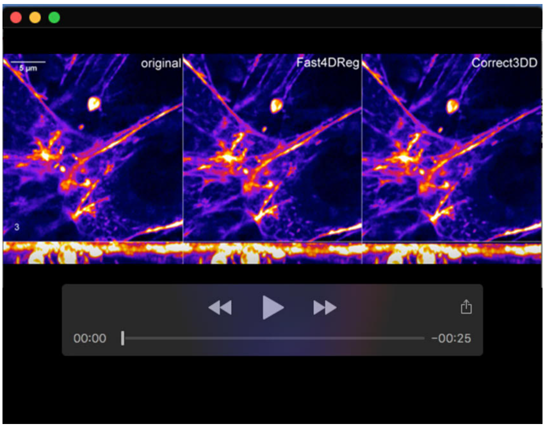
Click here to download Table S1



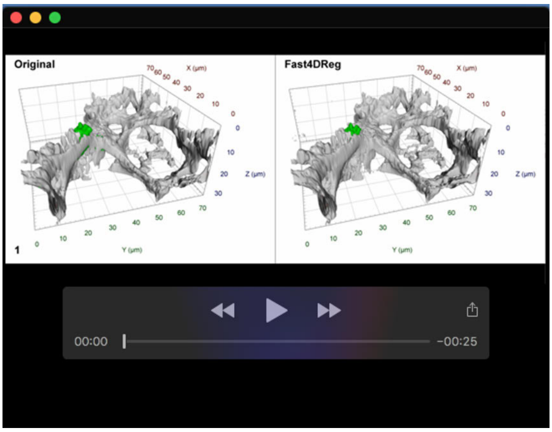
Movie 1. Dataset with a large amount of drift corrected using Fast4DReg, Correct3DD, and Fijiyama. The top panels show a selected z-plane of the uncorrected and corrected videos. To visualize the axial drift, the bottom panels show the y projection of the uncorrected and corrected videos. Note that the drift correction tools generate a cropped image, causing a mismatch in the location of the cell compared to the raw data. Scale bar 10 µm.



Movie 2. Comparison of the drift correction performance of Fast4DReg Correct3DD and Fijiyama. A selected region of ground truth and drift-corrected videos was magnified to visualize the drift-correction performance of Fast4DReg, Correct3DD, and Fijiyama. Scale bar 10 µm.

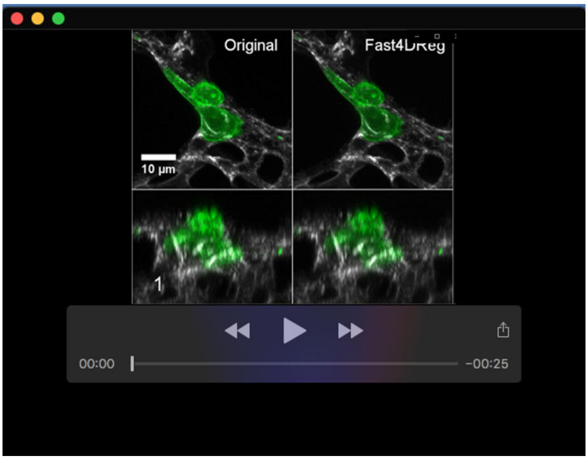


Movie 3. HUVEC-monolayer dataset corrected using Fast4DReg and Correct3DD. HUVEC cells labeled with SiR-actin were imaged in 3D using a Zeiss 880 Airyscan confocal microscope. This dataset suffered from significant drift, corrected using Fast4DReg and Correct3DD. The top panels show a selected z-plane of the uncorrected and corrected videos. The bottom panels show the y-projection of the uncorrected and corrected videos to visualize the axial drift. Scale bar 5 µm.



Movie 4. 3D surface rendering of the mouse lung dataset. AsPC1 cancer cell migrating inside the mouse lung vasculature was imaged ex-vivo in 3D using a Zeiss 880 Airyscan confocal microscope. The original dataset suffered from lateral and axial drift, which was corrected using Fast4DReg. 3D surface renderings of the original and Fast4DReg corrected 3D videos were created using Arivis Vision4D.

J. Cell Sci.: doi:10.1242/jcs.260728: Supplementary information



Movie 5. The mouse lung dataset corrected using Fast4DReg. AsPC1 cancer cell migrating inside the mouse lung vasculature was imaged ex-vivo in 3D using a Zeiss 880 Airyscan confocal microscope. The original dataset suffered from lateral and axial drift, which was corrected using Fast4DReg. The top panels show z-projection of the uncorrected and corrected videos. The bottom panels show the y-projection of the uncorrected and corrected videos to visualize the axial drift. Scale bar 10 µm.

Ershov, D*, Phan, M-S*, Pylvänäinen, J.W.*, Rigaud, S.U.*, Le Blanc, L,
Charles-Orszag, A, Conway, J.R.W., Laine, RF, Roy, NH, Bonazzi, D, Duménil,
G, Jacquemet, G & Tinevez, J-Y., (2022).

TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines.

Nature Methods, doi: 10.1038/s41592-022-01507-1

**equal contribution*

nature methods

BRIEF COMMUNICATION

https://doi.org/10.1038/s41592-022-01507-1

TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines

Dmitry Ershov[1,2,11], Minh-Son Phan[1,11], Joanna W. Pylvänäinen[3,4,5,11], Stéphane U. Rigaud[1,11], Laure Le Blanc[6], Arthur Charles-Orszag[6], James R. W. Conway[3], Romain F. Laine[7,8,10], Nathan H. Roy[9], Daria Bonazzi[6], Guillaume Duménil[6], Guillaume Jacquemet[3,4,5] ✉ and Jean-Yves Tinevez[1] ✉

TrackMate is an automated tracking software used to analyze bioimages and is distributed as a Fiji plugin. Here, we introduce a new version of TrackMate. TrackMate 7 is built to address the broad spectrum of modern challenges researchers face by integrating state-of-the-art segmentation algorithms into tracking pipelines. We illustrate qualitatively and quantitatively that these new capabilities function effectively across a wide range of bio-imaging experiments.

In the life sciences, tracking is used, for instance, to follow single particles, subcellular organelles, bacteria, cells, and whole animals. Owing to the sheer diversity of images, no single software can address every tracking challenge. This has prompted the development of flexible and extensible software tracking platforms[1–5] that enable biologists to build automated tracking pipelines tailored to a specific problem. Most tracking algorithms proceed in two steps. First, a detection algorithm detects or segments individual objects at each time point. Second, a linking algorithm links the detections to build tracks that follow each object over time. Importantly, the accurate detection of objects is crucial for the tracking process[6]. However, the low signal-to-noise ratio (SNR) that is typical of live-cell fluorescence microscopy often makes segmentation challenging. Aberrant object detection then leads to missing links and the generation of tracks that end prematurely, with multiple short tracks representing an individual object over time. Objects at high density can also be challenging to segment owing to overlap or close contact. Most detection algorithms will treat tightly packed objects as a single entity, resulting in breaks in tracks or single tracks linking groups of objects. Modern segmentation algorithms, in particular those based on machine learning (ML) and deep learning (DL) approaches, can address these challenges, because they excel at image-segmentation tasks in low-SNR and high-density images[7]. TrackMate[4] is a user-friendly Fiji[8] plugin for tracking objects in fluorescence microscopy images. TrackMate offers automated and semi-automated tracking algorithms, together with advanced visualization and analysis tools. However, until now, TrackMate detectors were solely based on the Laplacian of Gaussian (LoG) filter that is efficient against sub-resolved particles[9] or other blob-like objects, but performs poorly for textured objects, objects with complex shapes, and imaging modalities other than fluorescence. These detectors are also limited to measuring the objects' position and not their shape.

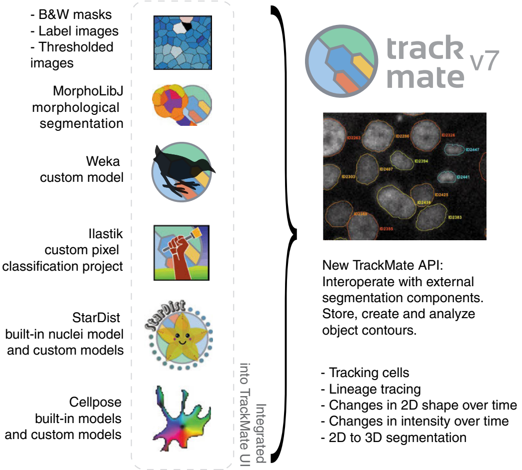
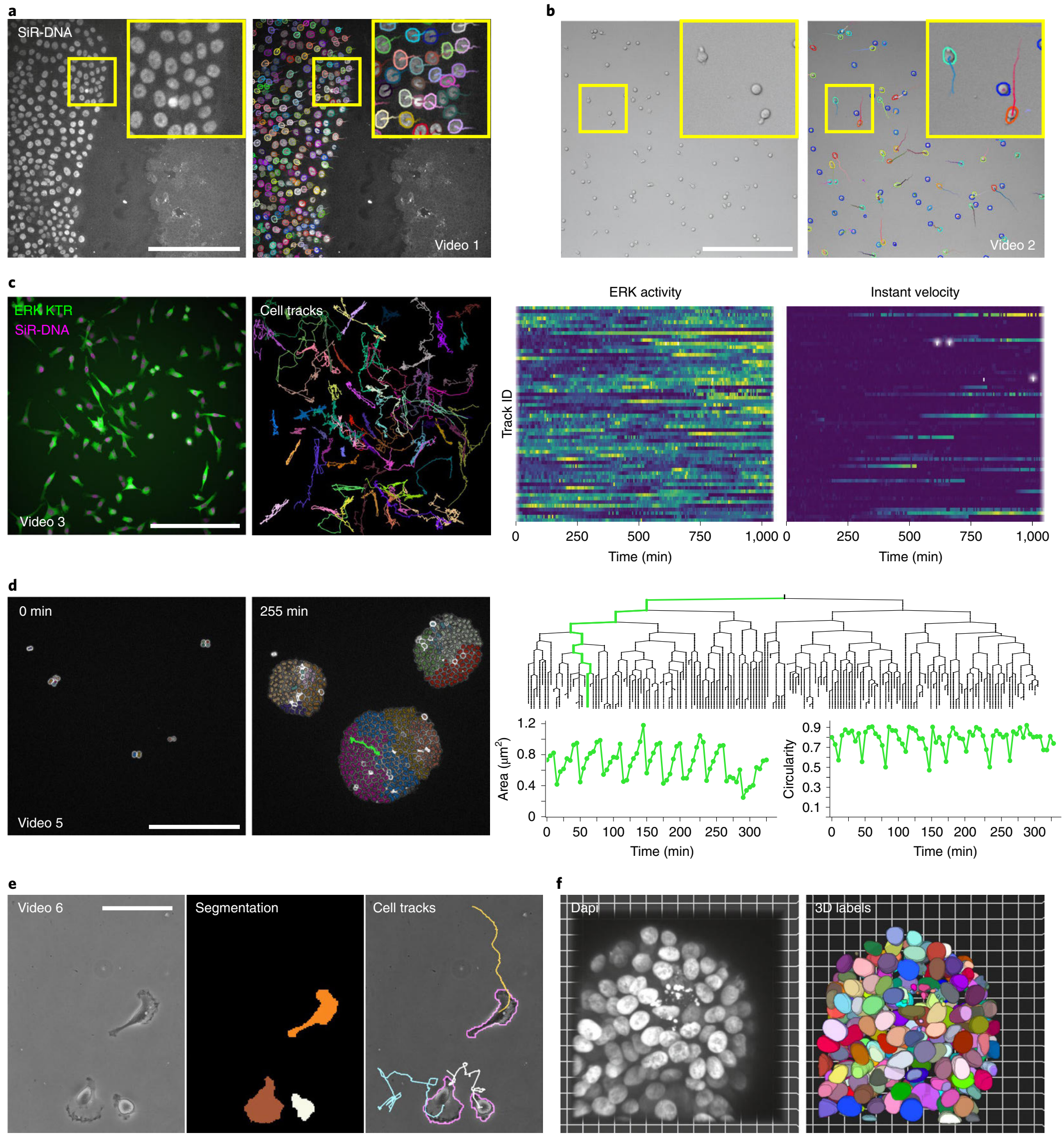


Fig. 1 | The new capabilities of TrackMate. TrackMate can now create, use, analyze, and store object contours segmented from 2D images. These contours enable TrackMate to extract morphological features of the tracked objects over time. We also wrote a new application programming interface (API) to allow the integration of external components in TrackMate. We use this API to incorporate popular segmentation tools including ilastik, the Weka Trainable-Segmentation Fiji plugin, cellpose, StarDist, and the morphological segmentation tool MorphoLibJ within TrackMate. TrackMate can also import segmentation results as masks or label images and use them for tracking, making it compatible with any segmentation algorithm. B&W, Black and White.

---

[1]Institut Pasteur, Université de Paris Cité, Image Analysis Hub, Paris, France. [2]Institut Pasteur, Université de Paris Cité, Biostatistics and Bioinformatic Hub, Paris, France. [3]Turku Bioscience Centre, University of Turku and Åbo Akademi University, Turku, Finland. [4]Åbo Akademi University, Faculty of Science and Engineering, Biosciences, Turku, Finland. [5]Turku Bioimaging, University of Turku and Åbo Akademi University, Turku, Finland. [6]Institut Pasteur, Université Paris Cité, INSERM UMR1225, Pathogenesis of Vascular Infections unit, Paris, France. [7]MRC Laboratory for Molecular Cell Biology, University College London, London, UK. [8]The Francis Crick Institute, London, UK. [9]Department of Microbiology and Immunology, SUNY Upstate Medical University, Syracuse, NY, USA. [10]Present address: Micrographia Bio, Translation and Innovation Hub, London, UK. [11]These authors contributed equally: Dmitry Ershov, Minh-Son Phan, Joanna W. Pylvänäinen, Stéphane U. Rigaud. ✉e-mail: guillaume.jacquemet@abo.fi; jean-yves.tinevez@pasteur.fr



Here, we introduce a new version of TrackMate (TrackMate 7) that has been rewritten to improve performance, usability, and versatility, all of which present several advantages over other available tracking tools (Supplementary Table 1). In particular, we developed a new application programming interface (API) that enables developers to integrate segmentation tools as TrackMate detectors. As examples, we provide detectors based on ilastik[10], Weka[11], cellpose[12], MorphoLibJ[13], and StarDist[14]. While the training of custom ML and DL models must be performed with external tools (using, for example, ilastik or the ZeroCostDL4Mic[15] platform for StarDist and cellpose), popular built-in models are now fully integrated into TrackMate with a user-friendly interface and scripting capabilities. TrackMate can also import segmentation results as mask or label images for tracking, making it possible to perform tracking with any segmentation algorithm. Notably, as TrackMate now detects object contours in every frame, we reconfigured the TrackMate data model to store, display and analyze two-dimensional (2D) morphological features of the tracked objects over time (Fig. 1). The new detectors work for 2D and three-dimensional (3D) images when possible, but the analysis of object contours is currently limited to 2D images.

These new features widely increase the breadth of TrackMate applications and capabilities (Fig. 2, Supplementary Videos 1–11, Extended Data Figs. 1–4, Supplementary Manual, and Supplementary Tutorials) and its tracking performance (Supplementary Table 2).

Fig. 2 | TrackMate can be used to track objects from a wide variety of bio-imaging experiments. a, Migration of MCF10DCIS.com cells, labeled with SiR-DNA, recorded using a spinning disk confocal microscope and automatically tracked using a custom StarDist model loaded in TrackMate (see Supplementary Video 1). Detected cells and their local tracks (colors indicate track ID) are displayed. Scale bar, 250 µm. **b**, The migration of activated T cells plated on ICAM-1 was recorded using a brightfield microscope and automatically tracked using a custom StarDist model loaded in TrackMate (see Supplementary Video 2). Detected cells (colors indicate the mean track speed: blue, slow-moving cells; red, fast-moving cells) and their local tracks (colors indicate track ID) are displayed. Scale bar, 250 µm. **c**, MDA-MB-231 cells stably expressing an ERK activity reporter (ERK-KTR-Clover) and labeled using SiR-DNA were recorded live using a widefield fluorescence microscope over 17 hours. Cell nuclei were automatically tracked over time using a StarDist model available in TrackMate (see Supplementary Video 3). For each tracked cell, the average intensity of the ERK reporter was measured in their nucleus over time (directly in TrackMate). Changes in ERK activity and in instant velocity are displayed as heatmaps (blue, high; yellow, low). **d**, The growth of *Neisseria meningitidis* expressing PilQ-mCherry was recorded using a spinning-disk confocal microscope. An ilastik pixel classifier, trained to segment individual bacteria, was loaded into TrackMate to follow bacteria growth. Representative fields of view and the lineage tree of the bacteria highlighted in green are displayed (see Supplementary Video 5). Changes in area and circularity of a bacterium over the tracking period are also highlighted (green track). Cell division events translate into sharp decreases in area, followed by a quasi-linear increase. The circularity roughly plateaus during cell growth then decreases before cell division. Scale bar, 25 µm. **e**, Glioblastoma cells migrating on a polyacrylamide gel were automatically segmented using a custom cellpose model trained in the ZeroCostDL4Mic platform. The resulting label images were automatically tracked using TrackMate (see Supplementary Fig. 1 and Supplementary Video 6). Example raw and label images, as well as cell tracks, are displayed. **f**. MCF10DCIS.com 3D spheroids were stained for DAPI and imaged using a spinning disk confocal microscope. Across the *Z* volume, nuclei were detected at each *Z* plane using StarDist and tracked (all performed in TrackMate). Tracked nuclei were then exported as a label image to create 3D labels (see Supplementary Video 9).

For instance, the StarDist integration offers efficient and versatile nuclei detection in fluorescence images via the built-in model (from image set BBBC038v1 in ref. [16]). Our integration also provides an interface to use custom StarDist models. To illustrate this, we used custom StarDist models trained with the ZeroCostDL4Mic platform[15] to track fluorescently labeled nuclei of collectively migrating breast cancer cells, or rapidly migrating T cells from brightfield images (Fig. 2a,b and Supplementary Videos 1 and 2). Before this integration, fully automated tracking of label-free cells was not possible in TrackMate.

As TrackMate supports multi-dimensional images, users can now track objects using one channel while measuring the changing intensities of the tracked objects in separate channels over time. As an example, we tracked the nuclei of breast cancer cells expressing a kinase translocation reporter, following changes in ERK activity in single cells as they migrated (Fig. 2c, Extended Data Fig. 1, and Supplementary Video 3).

To further showcase the versatility of TrackMate, we used a Weka model (trained using the Weka Fiji plugin) together with the new overlap tracker (linking algorithm based on object overlap between consecutive frames) to follow focal adhesions in endothelial cells (Extended Data Fig. 2 and Supplementary Video 4). We also used an ilastik pixel classifier (trained using ilastik) to follow *Neisseria meningitidis* growth and correlate lineage information to single bacteria morphological measurements (Fig. 2d and Supplementary Video 5). To showcase that TrackMate can now import segmentation results directly, then follow the imported objects, we tracked migrating cancer cells (fluorescent images and brightfield images), and hematopoietic stem cells (ref. [17], brightfield images) previously segmented using cellpose[12] (Fig. 2e, Extended Data Fig. 3, and Supplementary Videos 6–8).

TrackMate's new detectors can also be used to perform 3D segmentation. Indeed, by swapping the *z* dimension of the source image with time, TrackMate can propagate 2D segmentation results across *z* planes and generate a 3D segmentation result from successive annotated 2D planes. This new feature makes the segmentation of 3D objects accessible, flexible, and possible without programming knowledge (Fig. 2f, Extended Data Fig. 4, and Supplementary Videos 9–11).

TrackMate v7 currently offers a choice of ten segmentation detectors (plus the integration of custom models for some of them) and five particle-linking algorithms for tracking the detected objects. To facilitate choosing an optimal combination for a specific dataset, we developed an additional module, the TrackMate helper (Supplementary Manual and Extended Data Fig. 5). This module is a user-friendly application that performs parameter sweeps over any combination of detectors and particle-linking algorithms. Using the ground truth provided by the user, TrackMate helper computes the Cell-Tracking-Challenge (CTC) metrics[18] for each parameter combination and reports the optimal one for each of the CTC metrics (Supplementary Table 2). In a nutshell, TrackMate helper allows the optimization of the tracking parameters for a whole dataset systematically.

Altogether, TrackMate now enables powerful segmentation approaches for tracking purposes directly in Fiji within a user interface already familiar to many. We envision that, by enabling scientists to resolve complex tracking problems more efficiently, this new version of TrackMate will accelerate biological discovery. We expect that TrackMate will continue to evolve in the years to come. In particular, as core libraries handling 3D objects are further developed in Fiji, the analysis of 3D object contours and shapes during tracking could become an invaluable addition. TrackMate was also built as a software platform to be extended by others independently and is documented as such. As DL-based segmentation and tracking algorithms are being developed, we hope contributors will consider TrackMate as a platform to accelerate the dissemination of their work to researchers in the Life Sciences and beyond[18,19].

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-022-01507-1.

Received: 20 September 2021; Accepted: 25 April 2022;
Published online: 2 June 2022

References

1. Sbalzarini, I. F. & Koumoutsakos, P. Feature point tracking and trajectory analysis for video imaging in cell biology. *J. Struct. Biol.* **151**, 182–195 (2005).
2. Chenouard, N., Bloch, I. & Olivo-Marin, J.-C. Multiple hypothesis tracking for cluttered biological image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 2736–3750 (2013).
3. Piccinini, F., Kiss, A. & Horvath, P. CellTracker (not only) for dummies. *Bioinformatics* **32**, 955–957 (2016).
4. Tinevez, J.-Y. et al. TrackMate: an open and extensible platform for single-particle tracking. *Methods* **115**, 80–90 (2017).
5. McQuin, C. et al. CellProfiler 3.0: Next-generation image processing for biology. *PLoS Biol.* **16**, e2005970 (2018).

6. Chenouard, N. et al. Objective comparison of particle tracking methods. *Nat. Methods* **11**, 281–289 (2014).
7. Moen, E. et al. Deep learning for cellular image analysis. *Nat. Methods* **16**, 1233–1246 (2019).
8. Schindelin, J. et al. Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012).
9. Sage, D., Neumann, F. R., Hediger, F., Gasser, S. M. & Unser, M. Automatic tracking of individual fluorescence particles: application to the study of chromosome dynamics. *IEEE Trans. Image Process.* **14**, 1372–1383 (2005).
10. Berg, S. et al. ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods* **16**, 1226–1232 (2019).
11. Arganda-Carreras, I. et al. Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics* **33**, 2424–2426 (2017).
12. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods* **18**, 100–106 (2021).
13. Legland, D., Arganda-Carreras, I. & Andrey, P. MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ. *Bioinformatics* **32**, 3532–3534 (2016).
14. Schmidt, U., Weigert, M., Broaddus, C. & Myers, G. Cell detection with star-convex polygons. in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018* (eds. Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C. & Fichtinger, G.) 265–273 (Springer International Publishing, 2018).
15. von Chamier, L. et al. Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat. Commun.* **12**, 2276 (2021).
16. Caicedo, J. C. et al. Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl. *Nat. Methods* **16**, 1247–1253 (2019).
17. Lutolf, M. P., Doyonnas, R., Havenstrite, K., Koleckar, K. & Blau, H. M. Perturbation of single hematopoietic stem cell fates in artificial niches. *Integr. Biol. Quant. Biosci. Nano Macro* **1**, 59–69 (2009).
18. Haase, R. et al. CLIJ: GPU-accelerated image processing for everyone. *Nat. Methods* **17**, 5–6 (2020).
19. Haase, R. *clij/TrackMate-clij2: 2.5.1.3-doi*. (Zenodo, 2022); https://doi.org/10.5281/zenodo.5983244

Methods

Cells and reagents. MDA-MB-231 and U2OS cells were engineered to express ERK-KTR by first producing lentiviral particles in HEK 293FT packaging cells (Thermo Fisher, R70007). Cells were co-transfected with the third-generation lentiviral packaging system composed of pMDLg / pRRE (Addgene plasmid 12251), pRSV-Rev (Addgene plasmid 12253), and pMD2.G (Addgene plasmid 12259), along with the pLentiPGK Puro DEST ERK KTRClover (a kind gift from M. Covert; Addgene plasmid 90227) transfer plasmid, using Lipofectamine 3000 (ThermoFisher) in OptiMEM (Gibco, 31985070), as per the manufacturer's protocol[20,21]. After 24 hours, the medium was replaced with complete growth medium and incubated for 24 hours, at which point the medium was collected and filtered through a 0.45-µm syringe filter. MDA-MB-231 and U2OS cells were transduced with lentivirus for 48 hours in the presence of polybrene (8 µg/ml; Sigma, TR-1003-G), before washing and selection of stable positive cells using puromycin (2 µg/ml). Cells were then sorted by fluorescence-activated cell sorting (FACS) to isolate a population within a similar fluorescence range. MCF10 DCIS.COM cells were cultured in a 1:1 mix of DMEM (Sigma Aldrich) and F12 (Sigma Aldrich) supplemented with 5% horse serum (16050-122; Gibco BRL), 20 ng/ml human EGF (E9644; Sigma Aldrich), 0.5 mg/ml hydrocortisone (H0888-1G; Sigma Aldrich), 100 ng/ml cholera toxin (C8052-1MG; Sigma Aldrich), 10 µg/ml insulin (I9278-5ML; Sigma Aldrich), and 1% (vol/vol) penicillin/streptomycin (P0781-100ML; Sigma Aldrich). All cell lines were regularly checked for mycoplasma contamination, and all cell lines tested negative. MDA-MB-231 triple-negative human breast adenocarcinoma cells were acquired from ATCC (catalog number: HTB-26). U2OS osteosarcoma cells were provided by the Leibniz Institute DSMZ-German Collection of Microorganisms and Cell Cultures (cat. no. ACC 785). MCF10DCIS.COM cells were provided by J. F. Marshall (Barts Cancer Institute, Queen Mary University of London).

Tracking migrating breast cancer cells. Migrating MCF10DCIS.com cells were tracked using either StarDist directly implemented within TrackMate or using Cellpose and then TrackMate. To track MCF10DCIS.com cells labeled with sir-DNA using StarDist and TrackMate, a custom StarDist model was generated using the ZeroCostDL4Mic platform[14,15]. This custom StarDist model was trained for 100 epochs on 72 paired image patches (image dimensions: 1024 × 1024, patch size: 1024 × 1024) with a batch size of 2 and a mae loss function, using the StarDist 2D ZeroCostDL4Mic notebook (v1.12.2). The StarDist 'Versatile fluorescent nuclei' model was used as a training starting point. Key python packages used include TensorFlow (v1.15), Keras (v2.3.1), CSBdeep (v0.6.1), NumPy (v1.19.5), and Cuda (v10.1.243). The training was accelerated using a Tesla P100 GPU. This model generated excellent segmentation results on our test dataset (average F_1 score > 0.96). This model, the training dataset, and the code used for training are available on Zenodo[22]. In TrackMate, the StarDist detector custom model (score threshold = 0.41 and overlap threshold = 0.5) and the LAP tracker (linking max distance = 30 µm; track segment splitting = 15 µm) were used. Tracks were filtered in the function of their total distance traveled, and tracks shorter than 80 µm were excluded.

To track MCF10DCIS.com cells expressing lifeact-RFP (cell line described in ref. [23]) and labeled with sir-DNA, cells were first segmented using the ZeroCostDL4Mic cellpose 2D notebook (v1.12, refs. [12,15]). The cellpose model Cyto was used for the segmentation and the lifeact staining was used as the main segmentation channel. The Sir-DNA channel was used as the secondary segmentation channel. The following cellpose parameters were used: flow threshold = 0.4, cell probability threshold = 0, object diameter = 50. This approach generated excellent segmentation results on our test dataset (average F_1 score > 0.93). In TrackMate, the label image detector and the LAP tracker (linking max distance = 30 µm; track segment gap closing = 15 µm and 2 frames; track segment splitting = 15 µm) were used. Tracks were filtered in the function of the total number of spots detected, and tracks with fewer than 40 spots were excluded. This dataset is available on Zenodo[24,25].

Tracking migrating T cells. T cells migrating on ICAM-1 were automatically tracked using StarDist directly implemented within TrackMate. The StarDist model used was described previously[26] and is publicly available on Zenodo[27,28]. This model generated excellent segmentation results on our test dataset (F_1 score > 0.99). In TrackMate, the StarDist detector custom model (score threshold = 0.41 and overlap threshold = 0.5) and the Simple LAP tracker (linking max distance = 30 µm; gap closing max distance = 15 µm, gap closing max frame gap = 2 frames) were used. This dataset is available on Zenodo[27].

Following ERK activity in migrating cells. MDA-MD-231 or U2OS cells stably expressing clover-ERK-KTR were seeded on fibronectin-coated (1 µg /ml) Ibidi 8-well slides (Ibidi) 1 day before imaging. Four hours before imaging, the medium was supplemented with 250 nM sir-DNA (Cytoskeleton) and 25 mM HEPES (Sigma). Cells were then imaged live (37 °C, 5% CO_2) using a Nikon Eclipse Ti2-E microscope (Nikon) equipped with an sCMOS Orca Flash4.0 camera (Hamamatsu) and controlled by the NIS-Elements software (Nikon, v 5.11.01). MDA-MD-231 cells were imaged using a 20× Nikon CFI Plan Apo Lambda objective (NA 0.75), either 1 frame per minute for 2 hours or 1 frame every 5 minutes for 17 hours. In these experiments, a camera binning of 2 × 2 was used. U2OS cells were imaged using a 10× Nikon CFI Plan-Fluor objective (NA 0.3) every 5 minutes for 3 hours. Cell nuclei were automatically tracked over time by using StarDist in TrackMate.

To track the nuclei of U2OS cells, a custom StarDist model was trained using the ZeroCostDL4Mic platform[15]. The training source for the model was generated from 25 manually annotated images (dimensions: 2048 × 2048) using the LOCI plugin in Fiji. The generated training source and target were randomly cropped into size 1024 × 1024, rotated, flipped, and multiplied by 5 using the Augmentor ZeroCostDL4Mic notebook[15,29] to generate a dataset of 120 paired images. The custom StarDist model was trained for 200 epochs on the 120 paired image patches (image dimensions: 1024 × 1024, patch size: 1024 × 1024) with a batch size of 2 and a mae loss function, using the StarDist 2D ZeroCostDL4Mic notebook (v1.12.2)[15]. Key python packages used include TensorFlow (v1.15), Keras (v2.3.1), CSBdeep (v0.6.1), NumPy (v1.19.5), and Cuda (v11). The training was accelerated using a Tesla P100GPU. This model generated excellent segmentation results on our test dataset (average F_1 score > 0.918). In TrackMate, the StarDist detector custom model (score threshold = 0.41 and overlap threshold = 0.5) and the LAP tracker (linking max distance = 20 µm; track segment gap closing = 25 µm, gap closing max frame gap = 10 frames) were used. Tracks were filtered in function of their track duration and tracks shorter than 34 frames (2 hours and 40 minutes) were excluded. This dataset is available on Zenodo.

To track the nuclei of MDA-MB-231 cells, the 'Versatile fluorescent nuclei' StarDist model was used. In TrackMate, the StarDist detector (score threshold = 0.41; overlap threshold = 0.5) and the LAP tracker (linking max distance = 40 µm; track segment splitting = 30 µm) were used. Tracks were filtered using their duration, and only the tracks spanning the whole video were considered for further analysis (directly in TrackMate). For each tracked cell, the average intensity of the ERK reporter was measured in their nucleus over time (directly in TrackMate). To visualize the changes in ERK activity over time, results were uploaded to PlotTwist[30], and data were rescaled between 0 and 1 and visualized as heatmaps. This dataset is available on Zenodo[31].

Tracking mouse hematopoietic stem cells migrating in hydrogel microwells. Mouse hematopoietic stem cells migrating in a hydrogel microwell[17] were automatically segmented using cellpose (Cyto model) implemented in the ZeroCostDL4Mic platform[12,15]. The following Cellpose parameters were used: flow threshold = 0.4 and cell probability threshold = 0, object diameter = 17. This approach generated excellent segmentation results on our test dataset (average F_1 score > 0.93). The resulting label images were automatically tracked using TrackMate. In TrackMate, the label image detector and the LAP tracker (linking max distance = 30 µm; track segment gap closing = 15 µm and 2 frames; track segment splitting = 15 µm) were used. Spots were filtered by their circularity and area. Tracks were filtered by the total distance traveled; tracks shorter than 80 µm were excluded. This dataset is available from the Cell Tracking Challenge website (http://celltrackingchallenge.net/).

Tracking glioblastoma-astrocytoma cells migrating on a polyacrylamide gel. Glioblastoma-astrocytoma U373 cells migrating on a polyacrylamide gel were automatically segmented using a custom cellpose trained using the ZeroCostDL4Mic platform[12,15]. This cellpose model was trained for 500 epochs on 214 paired image patches (image dimensions: 520 × 696), with a batch size of 8, using the cellpose ZeroCostDL4Mic notebook (v1.13). The cellpose Cyto2 model was used as a training starting point. The training was accelerated using a Tesla K80 GPU. The following cellpose parameters were used for inference: flow threshold = 0.4 and cell probability threshold = 0, object diameter = 100. This approach generated excellent segmentation results on our test dataset (average F_1 score > 0.97). The resulting label images were automatically tracked using TrackMate. In TrackMate, the label image detector and the simple LAP tracker (linking max distance = 100 µm; track segment gap closing = 100 µm and 10 frames) were used. This dataset is available from the Cell Tracking Challenge website (http://celltrackingchallenge.net/) and on Zenodo[32].

***Neisseria meningitidis* sample preparation and imaging.** The *Neisseria meningitidis* strain 2C43 (ref. [33]) *pilQ/pilQ-mCherry*$_{ind}$ was grown on GCB agar plates (Difco) containing Kellog's supplements, 3 µg/ml vancomycin and 5 µg/ml chloramphenicol at 37 °C in a moist atmosphere containing 5% CO_2. The pMGC17 plasmid was designed in order to generate the 2C43 *pilQ/pilQ-mCherry*$_{ind}$ strain allowing IPTG-inducible expression of the type IV pilus secretin protein PilQ with a carboxy-terminal fusion to mCherry expressed from the *Neisseria meningitidis* chromosome.

First, *pilQ* was PCR-amplified from *Neisseria meningitidis* chromosomal DNA with primers pilQ-F: TTAATTAAAGGAGTAATTTTATGAATACCAAACTGAC AAAAATCGTCGACTCAATAGCGCAGGCTGTTGC

This PCR fragment was cloned in a pCRII-Blunt-TOPO vector (Invitrogen). Then, the mCherry ORF was PCR-amplified with a forward primer containing a region homologous to the 3' of *pilQ* (minus the stop codon) as well as a Gly-Ser-Gly linker, and a reverse primer containing a SalI restriction site and a region homologous to the TOPO vector (MUTmChCT-F: AGCCTGCGCTATGGTT CCGGTGTGAGCAAGGGC, MUTmChCT-R: CTGCAGAATTCGCCCTTGTC GACTCACTTGTACAG).

This PCR fragment was used as a mutagenesis megaprimer to amplify *pilQ* from the TOPO vector[34]. Finally, this vector was digested with PacI and SalI restriction enzymes and the resulting insert ligated into pMGC10. The pMGC10 plasmid was generated by inserting the *lacI* gene and the *lac* promoter in the pMGC3 plasmid[35]. The fragment of interest was PCR-amplified from the pMMB207 plasmid[36] using primers (LacIF2: GAA TTC GCT AAC TTA CAT TAA TTG CGT TGC, LacIPR: GTC GAC GAT CTT AAT TAT TTC CTG TGT GAA ATT GTT ATC CG) and cloned in pMGC3 using EcoRI and SalI restriction. The pMGC17 plasmid was used to transform *Neisseria meningitidis*, generating an intermediate strain that carries both a native *pilQ* and *pilQ-mCherry*. This strain was then transformed with chromosomal DNA from a *pilQ* mutant strain[37,38].

Bacteria in exponential phase from a 2-hour pre-culture in RPMI + 10% FBS supplemented with 100 µM IPTG at 37 °C and 5% CO$_2$ were diluted to an optical density of 0.015 (1.5×10^7 bacteria/ml) and dropped onto a 2% agarose gel supplemented with 100 µM IPTG. Once the bacteria-containing droplet had dried up, the agar pad was flipped down onto a Fluorodish (Ibidi, 60-µm dish, 35-mm-high glass-bottom). Fluorescently labeled proliferating bacteria were acquired using an inverted spinning-disk confocal microscope (Nikon, TI Eclipse) equipped with a 100× immersion objective (Plan-Fluor, NA = 0.5–1.3) at 37 °C in the presence of 5% CO$_2$. Bacterial fluorescence was imaged in time-lapse at a 5-min frame rate with an exposure time of 300 ms for 5.5 hours and recorded with a CMOS Camera (Photometrics, 95BPrime) using Metamorph Imaging Software (Molecular Devices, MetaMorph v7.10.4.407). The focus was maintained with the Perfect Focus System (PFS, Nikon).

We trained an ilastik model using the Pixel Classification workflow using images from these experiments. This model was then used in the TrackMate-Ilastik detector, with a threshold on the probability map of 0.5. Spurious detections larger than 4,000 µm^2 were removed prior to linking. We used the LAP tracker for linking, using a max linking distance of 1 µm, with a max gap-closing distance of 2 µm over a maximum of 2 frames, and detecting cell divisions over a maximal distance of 2 µm. We then filtered out tracks that start after 43 minutes. This dataset is available on Zenodo[39].

Tracking focal adhesions in endothelial cells. Live imaging of the endothelial cells expressing paxillin-eGFP was described previously[40]. Briefly, human dermal microvascular blood endothelial cells expressing paxillin were imaged using a Marianas spinning disk confocal microscope. This microscope was controlled by SlideBook 6 (Intelligent Imaging Innovations), equipped with a Yokogawa CSU-W1 scanning unit, an inverted Zeiss Axio Observer Z1 body, and a 100×, NA-1.4 oil (Plan-Apochromat, M27) objective. Images were acquired every two minutes using an Orca Flash4 sCMOS camera (chip size 2048 × 2048; 2 × 2 camera binning enabled; Hamamatsu Photonics), at 37 °C and in the presence of 5% CO$_2$. Acquired images were then processed using Fiji to remove background (rolling ball radius: 10 pixels), compensate for bleaching (exponential fit method), and correct drifting (StackReg, Rigid body). A custom Weka pixel classifier was then trained in Fiji to segment the focal adhesions. In TrackMate, the Weka detector (threshold on probability = 0.5) and the overlap tracker (min IoU = 0.3, scale factor = 1) were used. This dataset is available on Zenodo[41].

3D segmentation based on the association of 2D segmentation results. To form spheroids, MCF10 DCIS.com cells were seeded as single cells, in standard growth medium, at low density (3,000 cells per well) on growth factor reduced (GFR) Matrigel-coated glass-bottom dishes (coverslip No. 0; MatTek). After 12 h, the medium was replaced by a normal growth medium supplemented with 2% (vol/vol) GFR matrigel. After 6 days, spheroids were fixed with 4% PFA for 10 min at room temperature and labeled using DAPI. Spheroids were then imaged using a spinning-disk confocal microscope (Z step = 0.5 µm). The spinning-disk confocal microscope used was a Marianas spinning disk imaging system with a Yokogawa CSU-W1 scanning unit on an inverted Zeiss Axio Observer Z1 microscope (Intelligent Imaging Innovations) equipped with a 100× (NA 1.4) oil, Plan-Apochromat, M27 (Zeiss). To generate 3D labels, nuclei were detected and tracked across the Z volume using StarDist implemented in TrackMate.

In TrackMate, the StarDist detector (score threshold = 0.41 and overlap threshold = 0.5) and the LAP tracker (linking max distance = 1 µm, track merging and splitting enabled) were used. Detected spots were filtered in function of their mean intensity to exclude spots with weak intensities. Tracks were filtered in function of the number of spots per track, and only the tracks with more than three spots were considered for further analysis (directly in TrackMate). In TrackMate, tracked nuclei were then exported as a label image to create 3D labels. 3D labels were then visualized using the FPBioimage software[42]. The video was also generated using the FPBioimage software. This dataset is available on Zenodo[43].

Confocal images of *Arabidopsis thaliana* floral meristem[44] and light-sheet images of a developing *Drosophila melanogaster* embryo[45,46] were automatically segmented using cellpose (Cyto2 model) implemented in the ZeroCostDL4Mic platform[12,15]. The following cellpose parameters were used: flow threshold = 0.4 and cell probability threshold = 0, object diameter = 0. This approach generated excellent segmentation results on our test datasets (*Arabidopsis thaliana* floral meristem, average F_1 score > 0.97; *Drosophila melanogaster* embryo, average F_1 score > 0.89). To generate 3D labels, the 2D label images were tracked using TrackMate. In TrackMate, the label image detector and the simple LAP tracker were used. The videos were generated using the FPBioimage software.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data Availability

All of the new data used in this article are available on Zenodo, under a dedicated collection (https://zenodo.org/communities/trackmate). They are publicly available under the Creative Commons Attribution 4.0 International license.

Code Availability

TrackMate 7 and TrackMate-Helper introduced and used in this article are open-source software (GNU General Public License v3.0). Their source code is available on GitHub (https://github.com/fiji/TrackMate and https://github.com/tinevez/TrackMate-CTCRunner). TrackMate 7 is directly available in the Fiji software by simply updating it. TrackMate is documented on the ImageJ wiki: https://imagej.net/plugins/trackmate/ and the documentation for the new features can be accessed from https://imagej.net/plugins/trackmate/trackmate-v7-detectors.

References

20. Regot, S., Hughey, J. J., Bajar, B. T., Carrasco, S. & Covert, M. W. High-sensitivity measurements of multiple kinase activities in live single cells. *Cell* **157**, 1724–1734 (2014).
21. Kudo, T. et al. Live-cell measurements of kinase activity in single cells using translocation reporters. *Nat. Protoc.* **13**, 155–169 (2018).
22. Jacquemet, G. *Combining StarDist and TrackMate Example 1—Breast Cancer Cell Dataset* (2020); https://doi.org/10.5281/zenodo.4034976
23. Jacquemet, G. et al. FiloQuant reveals increased filopodia density during breast cancer progression. *J. Cell Biol.* **216**, 3387–3403 (2017).
24. Jacquemet, G., Pylvänäinen, J. W. & Tinevez, J.-Y. *Tracking Breast Cancer Cells Migrating Collectively and Imaged in Fluorescence with TrackMate-Cellpose* (2022); https://doi.org/10.5281/zenodo.5864646
25. Tinevez, J.-Y., Jacquemet, G. & Pylvänäinen, J. W. *Tracking Label Images with TrackMate* (2021); https://doi.org/10.5281/zenodo.5221424
26. Fazeli, E. et al. Automated cell tracking using StarDist and TrackMate. *F1000Res.* **9**, 1279 (2020).
27. Tinevez, J.-Y., Jacquemet, G. & Roy, N. H. *T cells Migration Followed with TrackMate* (2021); https://doi.org/10.5281/zenodo.5206119
28. Roy, N. H. & Jacquemet, G. *Combining StarDist and TrackMate Example 2—T Cell Dataset* (2020); https://doi.org/10.5281/zenodo.4034929
29. Bloice, M. D., Roth, P. M. & Holzinger, A. Biomedical image augmentation using Augmentor. *Bioinformatics* **35**, 4522–4524 (2019).
30. Goedhart, J. PlotTwist: A web app for plotting and annotating continuous data. *PLoS Biol.* **18**, e3000581 (2020).
31. Tinevez, J.-Y. & Pylvänäinen, J. W. *Cell Migration with ERK Signalling* (2021); https://doi.org/10.5281/zenodo.5205955
32. Jacquemet, G., Pylvänäinen, J. W. & Tinevez, J.-Y. *Tracking Glioblastoma–Astrocytoma Cells Imaged in Brightfield with TrackMate-Cellpose* (2022); https://doi.org/10.5281/zenodo.5863317
33. Nassif, X. et al. Antigenic variation of pilin regulates adhesion of *Neisseria meningitidis* to human epithelial cells. **8**, 719–725 (1993).
34. Ke, S.-H. & Madison, E. L. Rapid and efficient site-directed mutagenesis by single-tube 'megaprimer' PCR method. *Nucleic Acids Res.* **25**, 3371–3372 (1997).
35. Soyer, M. et al. Early sequence of events triggered by the interaction of Neisseria meningitidis with endothelial cells. *Cell. Microbiol.* **16**, 878–895 (2014).
36. Morales, V. M., Bäckman, A. & Bagdasarian, M. A series of wide-host-range low-copy-number vectors that allow direct screening for recombinants. *Gene* **97**, 39–47 (1991).
37. Geoffroy, M.-C., Floquet, S., Métais, A., Nassif, X. & Pelicic, V. Large-scale analysis of the meningococcus genome by gene disruption: resistance to complement-mediated lysis. *Genome Res.* **13**, 391–398 (2003).
38. Georgiadou, M., Castagnini, M., Karimova, G., Ladant, D. & Pelicic, V. Large-scale study of the interactions between proteins involved in type IV pilus biology in *Neisseria meningitidis*: characterization of a subcomplex involved in pilus assembly. *Mol. Microbiol.* **84**, 857–873 (2012).
39. Le Blanc, L., Rigaud, S. & Tinevez, J.-Y. Neisseria meningitidis *Bacterial Growth* (2021); https://doi.org/10.5281/zenodo.5419619
40. Hakanpaa, L. et al. Targeting β1-integrin inhibits vascular leakage in endotoxemia. *Proc. Natl Acad. Sci. USA* **115**, E6467–E6476 (2018).
41. Jacquemet, G., Minh-Son-Phan & Tinevez, J.-Y. *Tracking Focal Adhesions with TrackMate and Weka—Tutorial Dataset 2* (2022); https://doi.org/10.5281/zenodo.5978940
42. Fantham, M. & Kaminski, C. F. A new online tool for visualization of volumetric data. *Nat. Photonics* **11**, 69–69 (2017).

43. Tinevez, J.-Y., Pylvänäinen, J. W. & Jacquemet, G. *Segmenting Cells in a Spheroid in 3D using 2D StarDist within TrackMate* (2021); https://doi.org/10.5281/zenodo.5220610
44. Kar, A. *Original Stacks and Segmented Data* (2021); https://doi.org/10.6084/m9.figshare.14447079.v1
45. Ulman, V. et al. An objective comparison of cell-tracking algorithms. *Nat. Methods* **14**, 1141–1152 (2017).
46. Maška, M. et al. A benchmark for comparison of cell tracking algorithms. *Bioinformatics* **30**, 1609–1617 (2014).

Acknowledgements

The integration of existing algorithms as new detectors in TrackMate has been made possible thanks to the high quality of the code, documentation, and support provided by their respective authors. In particular, we would like to thank A. Kreshuk, D. Legland, D. Kutra, I. Arganda-Carreras, C. Stringer, M. Pachitariu, M. Weigert, S. Culley, and U. Schmidt. We can only hope for TrackMate to reach such a standard of quality to become a better tool of science. We are also grateful for the support and help of the bioimage analysis community, in particular C. Rueden, J. Eglinger, N. Chiaruttini, R. Guiet, O. Burri, V. Ulman, T. Pietzsch, and P. Tomancak. We thank H. Blau for giving us the permission to use the 'mouse hematopoietic stem cells in hydrogel microwells' dataset made available on the Cell Tracking Challenge website. The authors thank H. Hamidi for her critical reading of the manuscript. This study was supported by France BioImaging (Investissement d'Avenir; ANR-10-INBS-04, J.-Y. T.), the Academy of Finland (338537, G. J.), the Sigrid Juselius Foundation (G. J.), the Cancer Society of Finland (G. J.), the Åbo Akademi University Research Foundation (G. J., CoE CellMech), the Drug Discovery and Diagnostics strategic funding to Åbo Akademi University (G. J.) and the European Union's Horizon 2020 research and innovation program under Marie Sklodowska-Curie grant agreement 841973 (J. R. W. C.). J. W. P. was supported by Health Campus Turku 2.0 funded by the Academy of Finland. R. F. L. was supported by an MRC Skills development fellowship (MR/T027924/1). The Cell Imaging and Cytometry Core facility (Turku Bioscience, University of Turku, Åbo Akademi University, and Biocenter Finland) and Turku Bioimaging are acknowledged for services, instrumentation, and expertise.

Author contributions

G. J. and J.-Y. T. conceived the project; J.-Y. T. wrote the source code; G. J., J. W. P., N. H. R., and L. L. B. performed the image acquisition of the test and example data; G. J., J. W. P., R. F. L., J.-Y. T., M.-S. P., D. E., and S. U. R. tested the code; J. R. W. C., D. B., G. D. and A. C.-O. provided critical reagents; G. J., J. W. P., J.-Y. T., M.-S. P., D. E., S. U. R., and J.-Y. T. wrote the documentation and tutorials.; G. J. and J.-Y. T. wrote the manuscript with input from all co-authors.

Competing interests

The authors declare no competing interests.

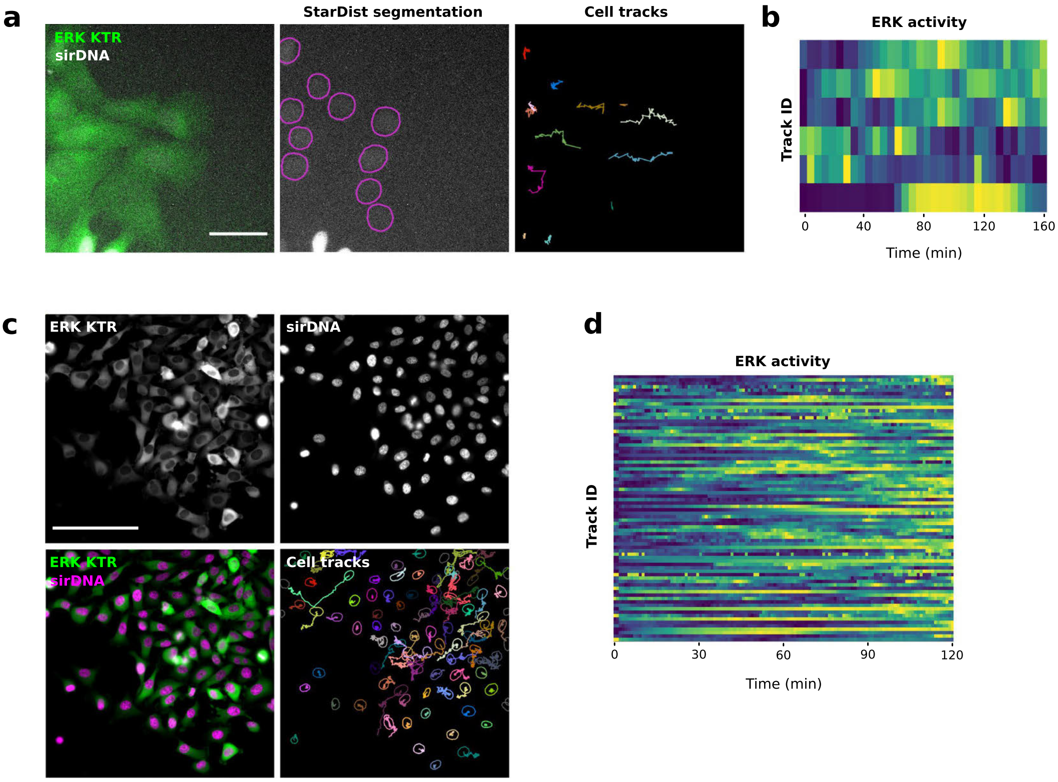
Additional information

Extended data is available for this paper at https://doi.org/10.1038/s41592-022-01507-1.

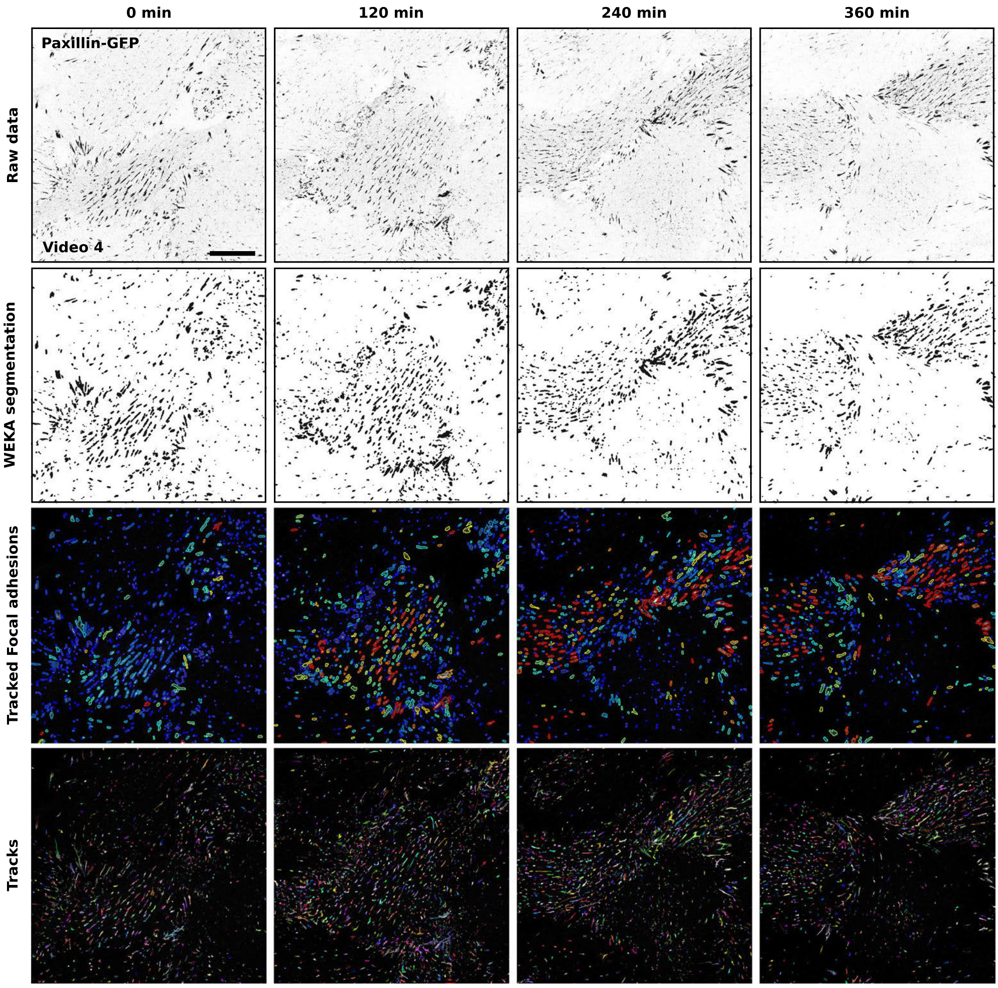
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-022-01507-1.

Correspondence and requests for materials should be addressed to Guillaume Jacquemet or Jean-Yves Tinevez.

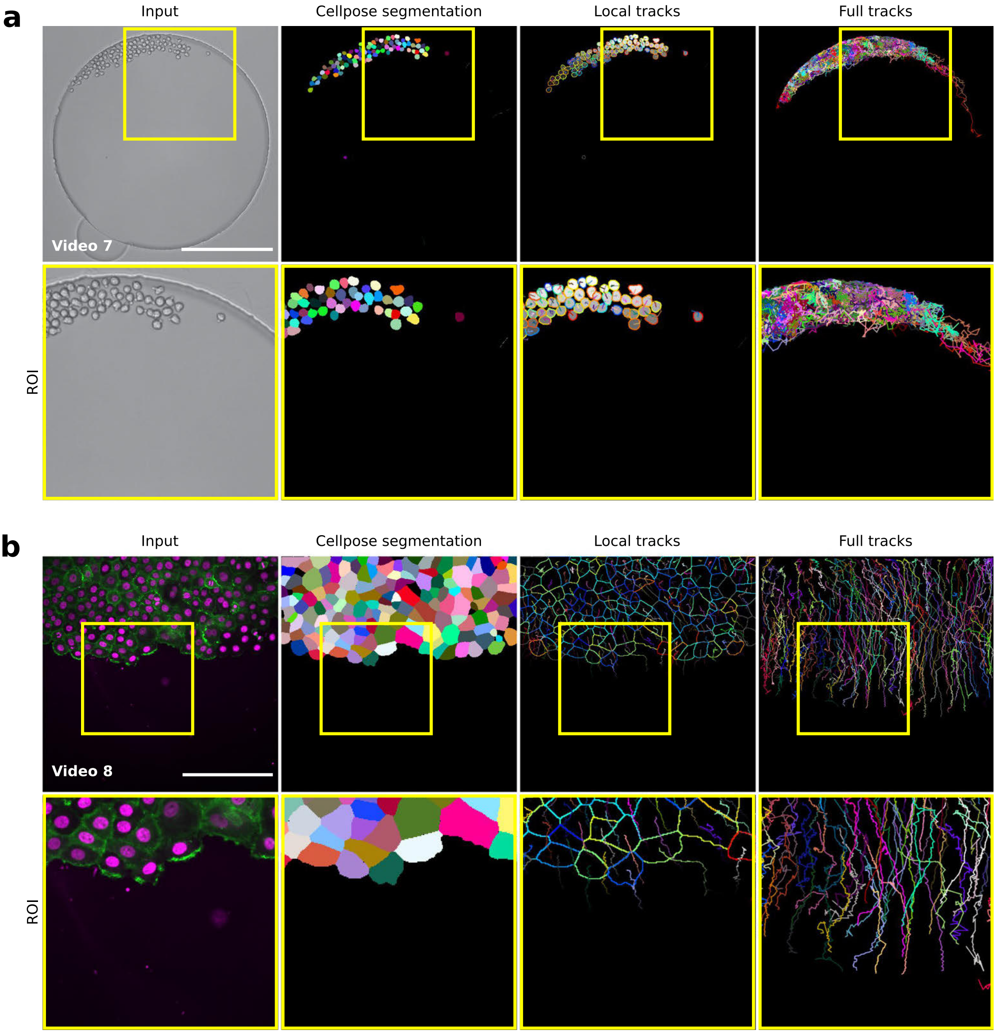
Peer review information *Nature Methods* thanks the anonymous reviewers for their contribution to the peer review of this work. Rita Strack was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.



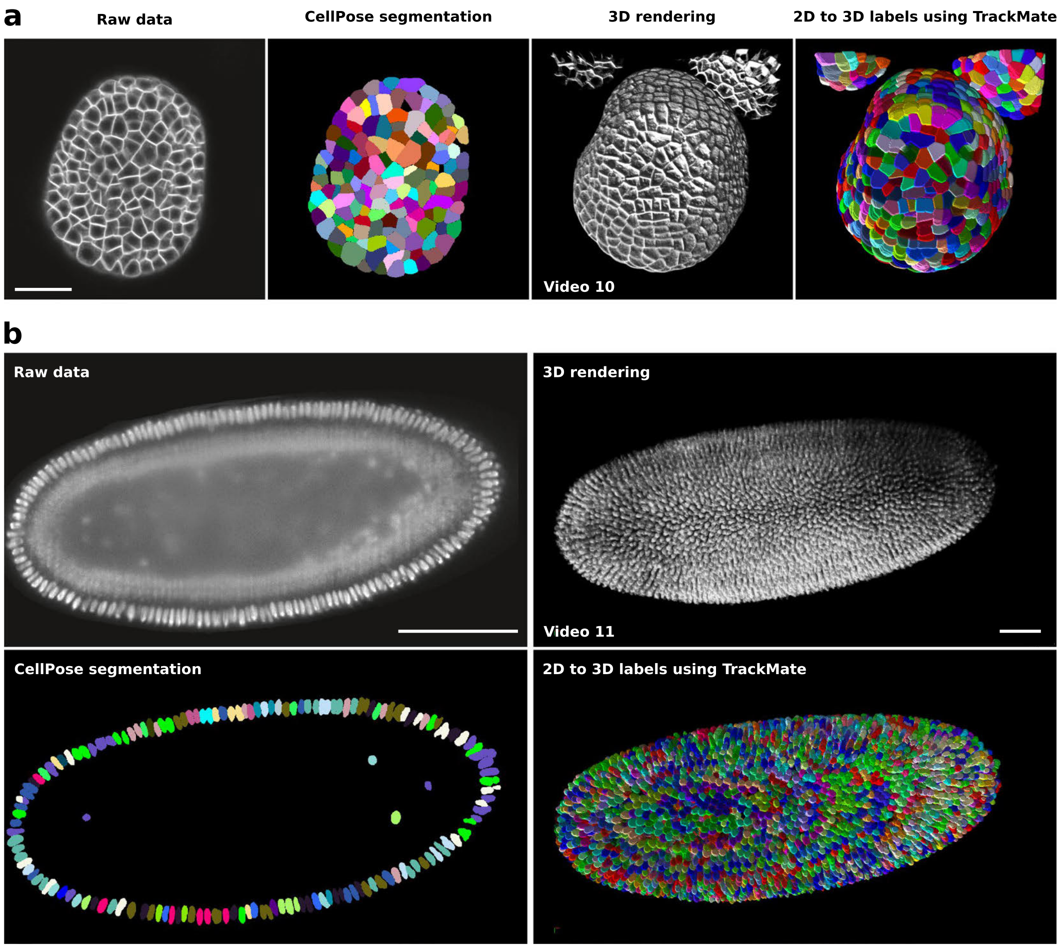
Extended Data Fig. 1 | Following ERK activity in migrating cells. U2OS (**a**, **b**.) and MDA-MB-231 cells (c. and d.) stably expressing an ERK activity reporter (ERK-KTR-Clover) and labeled using SiR-DNA were recorded live using a widefield fluorescence microscope. U2OS cells were recorded live over 3 hours (1 image every 5 minutes) and MDA-MB-231 cells were recorded live over 2 hours (1 image every minute). Cell nuclei were automatically tracked over time by using StarDist in TrackMate. A custom StarDist model was trained to detect the U2OS nuclei using the ZeroCostDL4Mic platform. The "Versatile fluorescent nuclei" StarDist model was used to track the MDA-MB-231 cell nuclei. For each tracked cell, the average intensity of the ERK reporter was measured in their nucleus over time (directly in TrackMate). Changes in ERK activity are displayed as heatmaps (blue high, yellow low). Heatmaps were generated using PlotTwist. Scale bar = 250 µm.



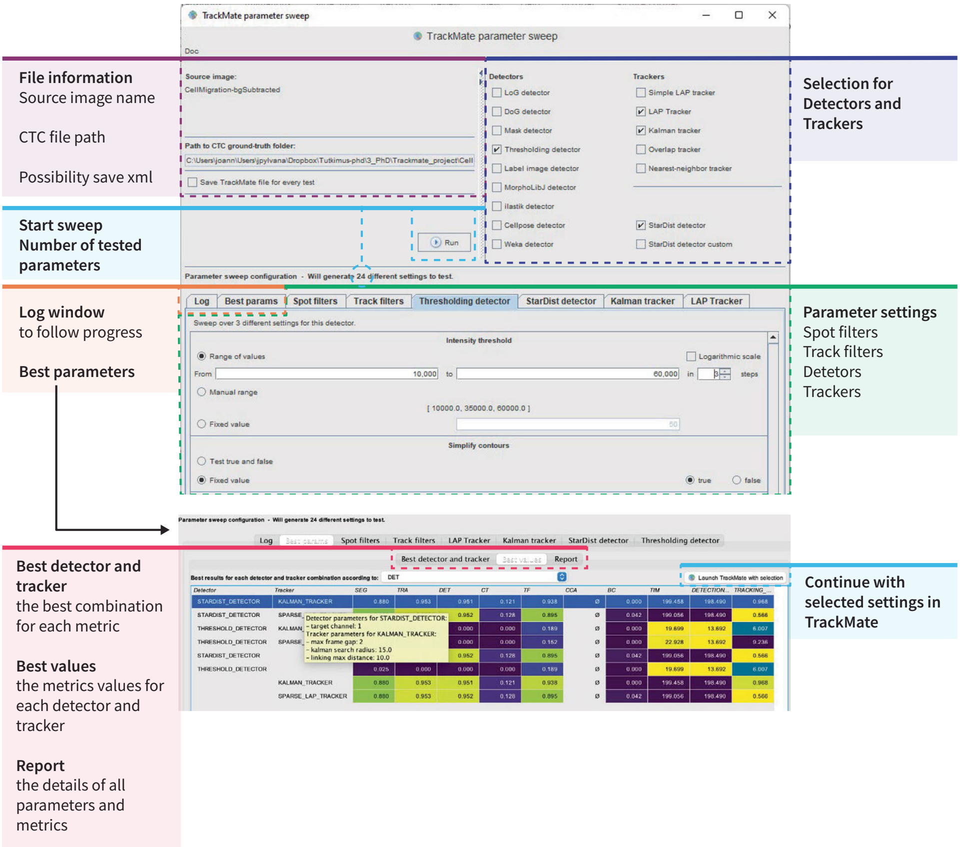
Extended Data Fig. 2 | Tracking focal adhesions in endothelial cells using Weka and TrackMate. Endothelial cells expressing paxillin-GFP were recorded live using a spinning disk confocal microscope. Focal adhesions were then segmented and tracked using Weka integrated within TrackMate (Movie 4). Raw data (inverted LUT), Weka segmentation results, tracked focal adhesion, and the focal adhesion tracks are displayed for selected time points. Tracked focal adhesions are color-coded to indicate their lifetime (red, long-lived, blue short-lived). In the bottom panel, track colors indicate ID. Scale bar = 25 µm.



Extended Data Fig. 3 | Tracking label images using TrackMate. a. Mouse hematopoietic stem cells migrating in a hydrogel microwell were automatically segmented using cellpose (Cyto model) implemented in the ZeroCostDL4Mic platform. The resulting label images were automatically tracked using TrackMate (Movie 7). Example raw and label images, as well as local and full cell tracks, are displayed. Yellow squares highlight regions of interest that are magnified. Scale bar = 250 µm. This dataset is available from the Cell Tracking Challenge. **b**. MCF10DCIS.com cells stably expressing lifeact-RFP and labeled with SiR-DNA were recorded live using a spinning disk confocal microscope. Cells were segmented using cellpose (Cyto model) implemented in the ZeroCostDL4Mic platform. The resulting label images were tracked using TrackMate (Movie 8). Example raw and label images, as well as local and full cell tracks, are displayed. Yellow squares highlight regions of interest that are magnified. Scale bar = 250 µm.



Extended Data Fig. 4 | Tracking 2D labels to generate 3D labels using TrackMate. (**a**.) Confocal images of Arabidopsis thaliana floral meristem and (**b**.) light-sheet images of a developing Drosophila melanogaster embryo were automatically segmented using cellpose 2D (Cyto2 model) implemented in the ZeroCostDL4Mic platform. Representative single Z plane and the corresponding cellpose segmentation results are displayed. To generate 3D labels, cellpose 2D segmentation results were then tracked using TrackMate. 3D rendering of the raw data and the 3D segmentation results are also shown. Scale bars: (a) = 25 µm, (b) = 100 µm.



File information
Source image name

CTC file path

Possibility save xml

Start sweep
Number of tested parameters

Log window
to follow progress

Best parameters

Selection for Detectors and Trackers

Parameter settings
Spot filters
Track filters
Detetors
Trackers

Best detector and tracker
the best combination for each metric

Best values
the metrics values for each detector and tracker

Report
the details of all parameters and metrics

Continue with selected settings in TrackMate

TrackMate parameter sweep

Source image:
CellMigration-bgSubtracted

Path to CTC ground-truth folder:
C:\Users\joann\Users\jpylvana\Dropbox\Tutkimus-phd\3_PhD\Trackmate_project\Cell

☐ Save TrackMate file for every test

Detectors
☐ LoG detector
☐ DoG detector
☐ Mask detector
☑ Thresholding detector
☐ Label image detector
☐ MorphoLibJ detector
☐ ilastik detector
☐ Cellpose detector
☐ Weka detector

Trackers
☐ Simple LAP tracker
☑ LAP Tracker
☑ Kalman tracker
☐ Overlap tracker
☐ Nearest-neighbor tracker
☑ StarDist detector
☐ StarDist detector custom

Parameter sweep configuration - Will generate 24 different settings to test.

Sweep over 3 different settings for this detector.

Intensity threshold
● Range of values ☐ Logarithmic scale
From 10,000 to 60,000 in 3 steps
○ Manual range
[10000.0, 35000.0, 60000.0]
○ Fixed value 50

Simplify contours
○ Test true and false
● Fixed value ● true ○ false

Parameter sweep configuration - Will generate 24 different settings to test.

Best results for each detector and tracker combination according to: DET

Detector	Tracker	SEG	TRA	DET	CT	TF	CCA	BC	TIM	DETECTION...	TRACKING_...
STARDIST_DETECTOR	KALMAN_TRACKER	0.880	0.953	0.951	0.121	0.938	Ø	0.000	199.458	198.490	0.968
STARDIST_DETECTOR	SPARSE_			0.952	0.128	0.895	Ø	0.042	199.056	198.490	0.566
THRESHOLD_DETECTOR	KALMAN_			0.000	0.000	0.189	Ø	0.000	19.699	13.692	6.007
THRESHOLD_DETECTOR	SPARSE_			0.000	0.000	0.152	Ø	0.000	22.928	13.692	9.236
STARDIST_DETECTOR				0.952	0.128	0.895	Ø	0.042	199.056	198.490	0.566
THRESHOLD_DETECTOR		0.025	0.000	0.000	0.000	0.189	Ø	0.000	19.699	13.692	6.007
	KALMAN_TRACKER	0.880	0.953	0.951	0.121	0.938	Ø	0.000	199.458	198.490	0.968
	SPARSE_LAP_TRACKER	0.880	0.953	0.952	0.128	0.895	Ø	0.042	199.056	198.490	0.566

Detector parameters for STARDIST_DETECTOR:
- target channel: 1
Tracker parameters for KALMAN_TRACKER:
- max frame gap: 2
- kalman search radius: 15.0
- linking max distance: 10.0

Extended Data Fig. 5 | The TrackMate Helper module. Screenshot highlighting the user interface of TrackMate helper, a module that performs systematic parameter sweeps over any user-defined combination of TrackMate detectors and particle-linking algorithms. Using the ground truth provided by the user, TrackMate helper computes the Cell-Tracking -Challenge metrics to help users choose the best detector/tracker combination for their data.

Hidalgo I., Pylvänäinen, J.W., Ferreira M.G., Russell C.T., Arganda-Carreras I., AI4Life Consortium, Jacquemet G., Henriques, R., Gómez-de-Mariscal, E.G., (2023).

DL4MicEverywhere: Deep learning for microscopy made flexible, shareable, and reproducible.

bioRxiv; doi: 10.1101/2023.11.19.567606

Accepted for publication in Nature Methods in March 2023

DL4MicEverywhere: Deep learning for microscopy made flexible, shareable, and reproducible

Iván Hidalgo-Cenalmor[1], **Joanna W Pylvänäinen**[2], **Mariana G Ferreira**[1], **Craig T Russell**[3], **Ignacio Arganda-Carreras**[4,5,6,7], **AI4Life Consortium**[*], **Guillaume Jacquemet**[2,8,9,10] ✉, **Ricardo Henriques**[1,11,] ✉, **and Estibaliz Gómez-de-Mariscal**[1] ✉

[1]Optical cell biology group, Instituto Gulbenkian de Ciência, Oeiras, Portugal
[2]Faculty of Science and Engineering, Cell Biology, Åbo Akademi University, Turku, Finland
[3]European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Genome Campus, United Kingdom
[4]Dept. Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Spain
[5]IKERBASQUE, Basque Foundation for Science, Spain
[6]Donostia International Physics Center (DIPC), Spain
[7]Biofisika Institute, Spain
[8]Turku Bioscience Centre, University of Turku and Åbo Akademi University, Turku, Finland
[9]Turku Bioimaging, University of Turku and Åbo Akademi University, Turku, Finland
[10]InFLAMES Research Flagship Center, Åbo Akademi University
[11]MRC Laboratory for Molecular Cell Biology, University College London, London, United Kingdom
[*]Authors of the AI4Life Consortium listed at the end

Deep learning has revolutionised the analysis of extensive microscopy datasets, yet challenges persist in the widespread adoption of these techniques. Many lack access to training data, computing resources, and expertise to develop complex models. We introduce DL4MicEverywhere, advancing our previous ZeroCostDL4Mic platform, to make deep learning more accessible. DL4MicEverywhere uniquely allows flexible training and deployment across diverse computational environments by encapsulating methods in interactive Jupyter notebooks within Docker containers –a standalone virtualisation of required packages and code to reproduce a computational environment–. This enhances reproducibility and convenience. The platform includes twice as many techniques as originally provided by ZeroCostDL4Mic and enables community contributions via automated build pipelines. DL4MicEverywhere empowers participatory innovation and aims to democratise deep learning for bioimage analysis.

phototoxicity | live microscopy | machine learning | cell division

Correspondence: *(G. Jacquemet) guillaume.jacquemet@abo.fi, (R. Henriques) rjhenriques@igc.gulbenkian.pt (E. Gómez-de-Mariscal) egomez@igc.gulbenkian.pt*

Introduction

Deep learning enables the transformative analysis of large multidimensional microscopy datasets, but barriers remain in implementing these advanced techniques (3, 4). Many researchers lack access to sufficient annotated data, high-performance computing resources, and expertise to develop, train, and deploy complex deep-learning models. In recent years, several approaches have been developed to democratise the usage of deep learning for microscopy (4). Multiple tools, such as BioImage.io, facilitate sharing and reusing broadly useful, previously trained deep learning models, distributing them as one-click image analysis solutions (1, 5). Yet often, deep learning models need to be trained or fine-tuned on the end user dataset to perform well (1, 4, 6). We previously released ZeroCostDL4Mic (2), an online platform

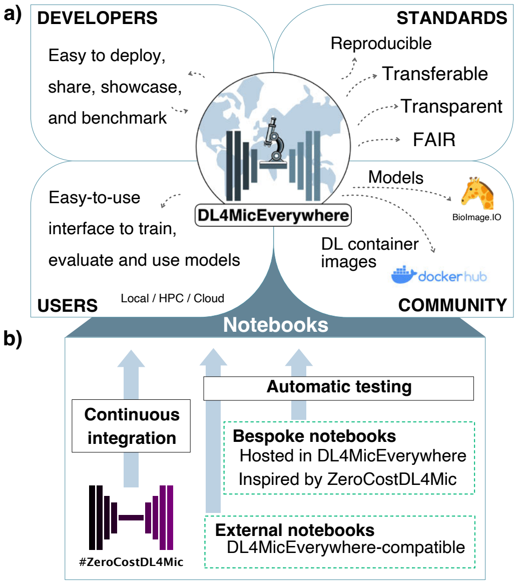


Fig. 1. DL4MicEverywhere platform. a) DL4MicEverywhere eases deep learning workflow sharing, deployment, and showcasing by providing a user-friendly interactive environment to train and use models. Enabling cross-platform compatibility ensures deep-learning model training reproducibility. DL4MicEverywhere contributes to deep learning standardisation in bioimage analysis by promoting transferable, FAIR, and transparent pipelines. The platform exports models compatible with the BioImage Model Zoo(1) and populates the Docker hub with free and open source (FOSS) container images that developers can reuse, incrementing the list of available workflows. b) DL4MicEverywhere accepts three types of notebook contributions: ZeroCostDL4Mic(2) notebooks, bespoke notebooks inspired by ZeroCostDL4Mic(2), and notebooks hosted in external repositories that are compliant with our format. These contributions are automatically tested to ensure the correct requirements and format.

Hidalgo-Cenalmor *et al.*, November 19, 2023 | 1–5

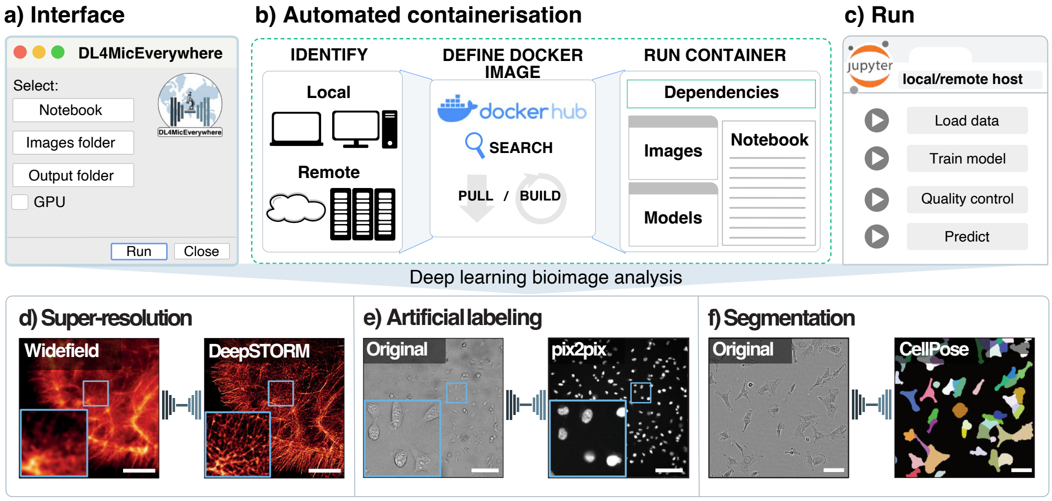


Fig. 2. a) When running DL4MicEverywhere, the user interacts with an interface to choose a notebook, image and output folder, and choose a GPU running model if possible. b) DL4MicEverywhere will automatically identify the system architecture and requirements to build a Docker container image. If the image is not available in the Docker hub, it is built in the user's machine. This image is used to create a Docker container: a functional instance of the image that gathers the code environment to use the chosen notebook. c) A Jupyter lab session is launched inside the Docker container to train, evaluate or use the chosen deep learning model. DL4MicEverywhere notebooks are also interactive and equivalent to ZeroCostDL4Mic (2) notebooks. d-f) DL4MicEverywhere enables the use of the same notebooks in different local or remote infrastructures such as workstations, the cloud or high-performance computing clusters. This is, researchers could run exactly the same d) super-resolution, e) artificial-labelling or g) segmentation pipelines, among many others, in different systems.

relying on Google Colab that helped democratise deep learning by providing a zero-code interface to train and evaluate models capable of performing various bioimage analysis tasks, such as segmentation, object detection, denoising, super-resolution microscopy, and image-to-image translation. Here, we introduce DL4MicEverywhere, a major advancement of the ZeroCostDL4Mic (2) framework (**Fig.**1).

DL4MicEverywhere allows users the flexibility to train and deploy their models across various computational environments, including Google Colab, their own computational resources (*e.g.*, desktop or laptop), or high-performance computing systems. This flexibility is made possible by enclosing each deep learning technique in an interactive Jupyter notebook, which is then contained in a Docker (7)-based environment. This enables users to install and interact with deep learning techniques easily. Incorporating cross-platform containerisation technology boosters the long-term platform's stability and reproducibility and enhances user convenience (8).

Results

DL4MicEverywhere introduces a novel and user-friendly graphical interface that enables users to easily access and launch a comprehensive collection of interactive Jupyter notebooks. Each notebook comes packaged into a Docker container with all necessary software dependencies, as illustrated in **Fig.** 2a-c.

DL4MicEverywhere has gone beyond simply containerising notebooks, providing a zero-code interface that handles all behind-the-scenes complexities. Users are not required to deal with the intricacies of Docker or configuring deep learning frameworks. The intuitive interface abstracts away these technical details, while the Docker encapsulation provides a standardised and rich environment for executing advanced techniques reliably (**Figure** 2b). Researchers can select a notebook, choose computing resources, and run the corresponding deep learning-powered analysis with just a few clicks. The platform handles deploying the encapsulated coding environment seamlessly in the background. This allows users to train and apply models on various computing resources they control, eliminating reliance on third-party platforms. Furthermore, researchers can launch a notebook on local or remote systems with GPU acceleration on clusters whenever available, without worrying about complex software dependencies, docker container management or losing access to deep-learning frameworks (**Fig.** 2d-f). DL4MicEverywhere offers twice the number of deep learning approaches than what was initially available in ZeroCostDL4Mic. The platform is designed to encourage sharing and reuse of models via the BioImage Model Zoo. DL4MicEverywhere's infrastructure is strengthened by automated build pipelines (9), which allows for the seamless integration of new trainable models contributed by the community (10–13) (as shown in **Fig.** 1b). These contributions are further facilitated through user-friendly templates, allowing new notebooks to be added independently of the original ZeroCostDL4Mic framework. By empowering participatory innovation in an open and flexible platform, DL4MicEverywhere aims to make deep learning more accessible for bioimage analysis. Developers can share a notebook based on our template and metadata for their method, and DL4MicEverywhere handles the testing and building of fully documented and open-source containerisation. Note that notebook containerisation allows others to reliably replicate analyses and build on the latest methods. The highly flexible nature of Docker containers encapsulating notebooks enhances long-term reproducibility across operating systems and computing environments. Researchers can easily share not just code, but the full software environment required to run it reliably. This reusable encapsulation empowers others to replicate analysis, evaluate methods, and build on research.

Discussion

Deep learning is revolutionising microscopy through data-driven analysis and discovery (14). However, significant barriers persist in accessing these advanced techniques, including a lack of training data, computing resources, and expertise (4, 6, 14). Proprietary platforms create technological and cultural obstacles, while complex workflows impede adoption by non-experts. DL4MicEverywhere is an initiative that aims to make deep learning accessible to everyone by providing a flexible and community-driven platform. Encapsulating software in Docker containers makes it possible to integrate new methods and enrich the microscopy community through participatory innovation. Intuitive graphical user interfaces also lower the barriers to entry, making it easier for non-experts to use the platform. Users can rely on shared techniques while customising models across diverse hardware, retaining control over data and analysis. The platform will particularly be useful with the increasing development and use of cutting-edge foundation models (15). By bundling these sophisticated models into shareable containers, researchers can customise and exploit them in their microscopy applications. DL4MicEverywhere also simplifies complex deep learning workflows for non-experts through automated pipelines, and is optimised for use with local computational resources, high-performance computing, and cloud-based solutions. This flexibility is precious for 1) sensitive biomedical data, where privacy risks may limit reliance on public cloud platforms, and 2) continuously scaling data such as time-lapse volumetric images or high-throughput high-content imaging data, where storage, dissemination and access rely on institutional infrastructures with specific data sharing protocols. DL4MicEverywhere also adheres to FAIR principles, enhancing discoverability and interoperability. We expect DL4MicEverywhere to represent an important step towards reliable, transparent, and participatory artificial intelligence in microscopy.

Code availability. The source code, documentation, and tutorials for DL4MicEverywhere can be found at `https://github.com/HenriquesLab/DL4MicEverywhere`. DL4MicEverywhere is made available under the Creative Commons CC-BY-4.0 license.

ACKNOWLEDGEMENTS

I.H.C., M.G.F., C.T.R., R.H., and E.G.M. received funding from the European Commission through the Horizon Europe program (AI4LIFE project with grant agreement 101057970-AI4LIFE, and RT-SuperES project with grant agreement 101099654-RT-SuperES to R.H.). I.H.C., M.G.F., E.G.M. and R.H. also acknowledge the support of the Gulbenkian Foundation (Fundação Calouste Gulbenkian) and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101001332 to R.H.). Funded by the European Union. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. This work was also supported by the European Molecular Biology Organization (EMBO) Installation Grant (EMBO-2020-IG-4734 to R.H.), the EMBO Postdoctoral Fellowship (EMBO ALTF 174-2022 to E.G.M.), the Chan Zuckerberg Initiative Visual Proteomics Grant (vpi-0000000044 with DOI:10.37921/743590vtudfp to R.H.) and the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundations (Chan Zuckerberg Initiative Napari Plugin Foundations Grant Cycle 2, NP2-0000000085 granted to R.H.). R.H. also acknowledges the support of LS4FUTURE Associated Laboratory (LA/P/0087/2020). This work is partially supported by grant GIU19/027 (to I.A.C.) funded by the University of the Basque Country (UPV/EHU), grant PID2021-126701OB-I00 (to I.A.C.) funded by the Ministerio de Ciencia, Innovación y Universidades, AEI, MCIN/AEI/10.13039/501100011033, and by "ERDF A way of making Europe" (to I.A.C.). This study was also supported by the Academy of Finland (338537 to G.J.), the Sigrid Juselius Foundation (to G.J.), the Cancer Society of Finland (Syöpäjärjestöt; to G.J.), and the Solutions for Health strategic funding to Åbo Akademi University (to G.J.). This research was supported by InFLAMES Flagship Programme of the Academy of Finland (decision number: 337531). We would like to thank Amin Rezaei, Ainhoa Serrano, Pablo Alonso, Urtzi Beorlegui, Andoni Rodriguez, Erlantz Calvo, Soham Mandal, and Virginie Uhlmann for their contributions to the ZeroCostDL4Mic notebook collection.

Bibliography

1. 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, Dominik Kutra, Maksim Novikov, Martin Weigert, Uwe Schmidt, Peter Bankhead, Guillaume Jacquemet, Daniel Sage, Ricardo Henriques, Arrate Muñoz-Barrutia, Emma Lundberg, Florian Jug, Anna Kreshuk, and Chan Zuckerberg Biohub. BioImage Model Zoo: A community-driven resource for accessible deep learning in bioimage analysis. *bioRxiv*, page 2022.06.07.495102, June 2022. doi: 10.1101/2022.06.07.495102. Publisher: Cold Spring Harbor Laboratory.
2. 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, December 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-22518-0.
3. Erick Moen, Dylan Bannon, Takamasa Kudo, William Graf, Markus Covert, and David Van Valen. Deep learning for cellular image analysis. *Nature Methods*, 16(12):1233–1246, December 2019. ISSN 1548-7091. doi: 10.1038/s41592-019-0403-1.
4. Joanna W. Pylvänäinen, Estibaliz Gómez-de Mariscal, Ricardo Henriques, and Guillaume Jacquemet. Live-cell imaging in the deep learning era. *Current Opinion in Cell Biology*, 85: 102271, December 2023. ISSN 0955-0674. doi: 10.1016/j.ceb.2023.102271.
5. Estibaliz Gómez-de Mariscal, Carlos García-López-de Haro, Wei Ouyang, Laurène Donati, Emma Lundberg, Michael Unser, Arrate Muñoz-Barrutia, and Daniel Sage. DeepImageJ: A user-friendly environment to run deep learning models in ImageJ. *Nature Methods*, 18(10): 1192–1195, October 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01262-9. Number: 10 Publisher: Nature Publishing Group.
6. Romain F. Laine, Ignacio Arganda-Carreras, Ricardo Henriques, and Guillaume Jacquemet. Avoiding a replication crisis in deep-learning-based bioimage analysis. *Nature Methods 2021 18:10*, 18(10):1136–1144, October 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01284-3. Publisher: Nature Publishing Group.
7. Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux j*, 239(2):2, 2014.
8. David Moreau, Kristina Wiebels, and Carl Boettiger. Containers for computational reproducibility. *Nature Reviews Methods Primers*, 3(1):1–16, July 2023. ISSN 2662-8449. doi: 10.1038/s43586-023-00236-9. Number: 1 Publisher: Nature Publishing Group.
9. Brett K. Beaulieu-Jones and Casey S. Greene. Reproducibility of computational workflows is automated using continuous analysis. *Nature Biotechnology*, 35(4):342–346, April 2017. ISSN 1546-1696. doi: 10.1038/nbt.3780. Number: 4 Publisher: Nature Publishing Group.
10. 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):688, July 2022. ISSN 2399-3642. doi: 10.1038/s42003-022-03634-z.
11. Artur Speiser, Lucas-Raphael Müller, Philipp Hoess, Ulf Matti, Christopher J. Obara, Wesley R. Legant, Anna Kreshuk, Jakob H. Macke, Jonas Ries, and Srinivas C. Turaga. Deep learning enables fast and dense single-molecule localization with high accuracy. *Nature Methods*, 18(9):1082–1090, September 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01236-x. Number: 9 Publisher: Nature Publishing Group.
12. Afshin Khadangi, Thomas Boudier, and Vijay Rajagopal. EM-stellar: benchmarking deep learning for electron microscopy image segmentation. *Bioinformatics*, 37(1):97–106, April 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa1094.

13. Martin Priessner, David C. A. Gaboriau, Arlo Sheridan, Tchern Lenn, Jonathan R. Chubb, Uri Manor, Ramon Vilar, and Romain F. Laine. Content-aware frame interpolation (CAFI): Deep Learning-based temporal super-resolution for fast bioimaging. *bioRxiv 2021*, 2021. doi: https://doi.org/10.1101/2021.11.02.466664.
14. Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972): 47–60, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06221-2. Number: 7972 Publisher: Nature Publishing Group.
15. Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, July 2022. arXiv:2108.07258 [cs].

Authors of the AI4Life Horizon Europe program consortium.

Arrate Muñoz-Barrutia
Bioengineering Department, Universidad Carlos III de Madrid, Leganes, Spain; Instituto de Investigación Sanitaria Gregorio Marañón, Madrid, Spain
Beatriz Serrano-Solano
Euro-BioImaging ERIC Bio-Hub, European Molecular Biology Laboratory (EMBL) Heidelberg, Heidelberg, Germany
Caterina Fuster Barcelo
Bioengineering Department, Universidad Carlos III de Madrid, Leganes, Spain; Instituto de Investigación Sanitaria Gregorio Marañón, Madrid, Spain
Constantin Pape
Georg-August-University Göttingen, Institute of Computer Science, Göttingen, Germany
Emma Lundberg
Department of Pathology, Stanford University School of Medicine, Stanford, CA, USA; Science for Life Laboratory, School of Engineering Sciences in Chemistry, Biotechnology and Health, KTH Royal Institute of Technology, Stockholm, Sweden; Department of Bioengineering, Stanford University, Stanford, CA, USA
Florian Jug
Fondazione Human Technopole, Milan, Italy
Joran Deschamps
Fondazione Human Technopole, Milan, Italy
Matthew Hartley
European Molecular Biology Laboratory, European Bioinformatics Institute, EMBL-EBI, Wellcome Genome Campus, Cambridge, UK
Mehdi Seifi
Fondazione Human Technopole, Milan, Italy
Teresa Zulueta-Coarasa
European Molecular Biology Laboratory, European Bioinformatics Institute, EMBL-EBI, Wellcome Genome Campus, Cambridge, UK
Vera Galinova
Fondazione Human Technopole, Milan, Italy
Wei Ouyang
Department of Applied Physics, Science for Life Laboratory, KTH Royal Institute of Technology, Stockholm, Sweden

Methods

DL4MicEverywhere Platform Implementation. The core DL4MicEverywhere platform was implemented in Bash packaging and managing Python notebook workflows through Docker containers. An overview of the key technical components is provided below.

Docker Containerization. Each notebook is encapsulated into a Docker container, including all dependencies required for smooth runtime (Docker v24.0.5, Docker Inc.). These containers are functional instances of Docker images –software units that contain the virtualisation of a specific computational environment, with all the specified dependencies and packages included. Images were built from Ubuntu (v20.04/22.04) base images, with optional Nvidia CUDA support for GPU acceleration. Python (v3.7/3.8/3.10), deep learning packages (TensorFlow, Keras or Pytorch), and notebook packages were installed according to the requirements into the containers. Unique containers were constructed for each notebook using a parameterised Docker file build process, taking metadata like notebook URL and software versions as input. These images are uploaded to Docker hub so they can be distributed as free and open source (FOSS) and belong to the Open Container Initiative (OCI) (https://opencontainers.org/).

Launch Script and GUI. A Bash shell script launch.sh was implemented to manage the building, running, and monitoring of the notebook containers based on user input. Key functions included argument parsing, installation checking, Docker image building, and Jupyter Lab invocation within the container. A graphical user interface was additionally created using Wish (a Tcl/Tk application) to enable intuitive notebook and parameter selection through a desktop window. This is invoked by the launch script and passed user selections.

Configuration Metadata. Inspired by the BioImage Model Zoo (1) specifications, notebook container construction was driven by human-readable YAML configuration files specifying necessary build metadata for each notebook, including the URL of the notebook itself, Python requirements, and Docker parameters. These configurations were loaded by the launch script when initialising a container. This format establishes a basis for a seamless connection with the BioEngine of the Zoo.

Testing and Deployment Pipelines. GitHub Actions workflows were implemented to automatically build and publish container images for each new notebook, handling testing across platforms like AMD64 and ARM64. Images were versioned based on notebook metadata and published to DockerHub for distribution. Strict conventions enforced by templates facilitated notebook contributions from the community. These contributions are further checked via GitHub Actions to assert that they follow the specified format with valid URLs and that it is possible to build a Docker image.

Jupyter Notebooks and Widgets. Notebooks were adapted from the ZeroCostDL4Mic Colab format to interactive Jupyter notebooks leveraging *ipywidgets* for a simplified user interface requiring no coding. Parameters could be configured via graphical elements rather than edits to code.

Joanna Pylvänäinen

Bioimage analysis for life scientists: tools for live cell imaging

Live imaging is essential in visualizing biological processes such as normal tissue development, wound healing, and cancer — processes too small for the bare human eye to observe. Optical microscopy has enabled the magnification of these processes, and the integration of sensitive digital cameras has enabled the acquisition of images for subsequent observation and analysis. Extracting meaningful information from live imaging poses several challenges. Despite the availability of several tools to improve live cell imaging, challenges persist. Live cell imaging data often drifts, dyes bleach, and sample health is frequently compromised, hindering the extraction of meaningful information from microscopy experiments. Computational bioimage processing and analysis emerge as promising solutions to address many of these issues.

This thesis presents user-friendly live cell image prosessing and analysis tools for life scientists. First, Fast4DReg swiftly corrects drifting in 4D images, second, DL4MicEverywhere allows implementation of deep learning in various computational environments without coding, and third, TrackMate v7 tracking software integrates cutting-edge segmentation algorithms into tracking pipelines. To ensure usability, comprehensive documentation and tutorials have been created. This thesis demonstrates the use of these tools in studies of pancreatic cancer cell interactions during metastasis and drug resistance mechanisms. In summary, the user-friendly image analysis tools developed in this thesis offer efficient and accessible solutions for processing and analyzing live cell imaging data, benefiting researchers across various fields and contributing to our understanding of cell behavior and disease processes.

ISBN 978-952-12-4372-1