

# An Open-Source Modular Framework for Automated Pipetting and Imaging Applications

Wei Ouyang, Richard W. Bowman, Haoran Wang, Kaspar E. Bumke, Joel T. Collins, Ola Spjuth, Jordi Carreras-Puigvert, and Benedict Diederich\*

The number of samples in biological experiments is continuously increasing, but complex protocols and human error in many cases lead to suboptimal data quality and hence difficulties in reproducing scientific findings. Laboratory automation can alleviate many of these problems by precisely reproducing machine-readable protocols. These instruments generally require high up-front investments, and due to the lack of open application programming interfaces (APIs), they are notoriously difficult for scientists to customize and control outside of the vendor-supplied software. Here, automated, high-throughput experiments are demonstrated for interdisciplinary research in life science that can be replicated on a modest budget, using open tools to ensure reproducibility by combining the tools OpenFlexure, Opentrons, ImJoy, and UC2. This automated sample preparation and imaging pipeline can easily be replicated and established in many laboratories as well as in educational contexts through easy-to-understand algorithms and easy-to-build microscopes. Additionally, the creation of feedback loops, with later pipetting or imaging steps depending on the analysis of previously acquired images, enables the realization of fully autonomous “smart” microscopy experiments. All documents and source files are publicly available to prove the concept of smart lab automation using inexpensive, open tools. It is believed this democratizes access to the power and repeatability of automated experiments.

## 1. Introduction

One of the core interests in modern life sciences is to understand how cells interact with each other and form highly organized biological systems.<sup>[1,2]</sup> Tools such as microscopy help to discover cell dynamics on very small scales in both time and

space to provide evidence for previously formulated hypotheses.<sup>[3]</sup>

It is common to repeat experiments many times or to use small variations in environmental parameters to understand causal relationships in increasingly complex biological systems.<sup>[4]</sup> This produces a lot of data, leading to challenges in data management and analysis. In most cases such experiments involve a lot of human labor carrying out protocols designed for and by humans, which is error-prone and might lead to data of lower quality and sometimes contribute to poorly reproducible results.<sup>[5]</sup>

The resulting vastly improved datasets have the potential to lead to new insights. Here, a hypothesis is initially formulated and the experiment or parts of the experiment is designed and autonomously carried out by machines.<sup>[2,6–10]</sup> This is particularly useful when the same task has to be performed repeatedly over long periods of time, which is the case of multiple antibody-labeling (e.g., multiplexing).

The application of machine learning algorithms in microscopy image analysis potentially helps to find correlations in data with previously trained networks<sup>[11,12]</sup> or in a completely unsupervised manner.<sup>[13]</sup> Notable examples include tracking morphological changes in cells,<sup>[14]</sup> improving the signal-to-noise ratio in low-photon dose imaging<sup>[15–18]</sup> or even performing augmented microscopy tasks. On-line analysis allows us to record only those events of interest, such as

W. Ouyang  
Science for Life Laboratory  
School of Engineering Sciences in Chemistry  
Biotechnology and Health  
KTH – Royal Institute of Technology  
Stockholm 114 28, Sweden

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/adbi.202101063>.

© 2021 The Authors. Advanced Biology published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/adbi.202101063

R. W. Bowman, K. E. Bumke, J. T. Collins  
Department of Physics  
University of Bath  
Bath BA2 7AY, UK

H. Wang, B. Diederich  
Leibniz Institute for Photonic Technology  
Albert-Einstein-Str. 9, 07749 Jena, Germany  
E-mail: [benedict.diederich@leibniz-ipht.de](mailto:benedict.diederich@leibniz-ipht.de)

H. Wang, B. Diederich  
Institute of Physical Chemistry  
Friedrich-Schiller-Universität Jena  
Helmholtzweg 4, 07743 Jena, Germany

O. Spjuth, J. Carreras-Puigvert  
Department of Pharmaceutical Biosciences and Science for Life Laboratory  
Uppsala University  
Box 591, Uppsala SE-75124, Sweden

mitosis or apoptosis, and greatly reduce data storage challenges. Overall, laboratory automation minimizes time spent by highly trained researchers on routine tasks and significantly improves reproducibility by eliminating human error. Automated analysis during or after experiments can fine-tune parameters in future experimental steps.<sup>[19]</sup>

The increasing complexity and associated costs of experiments, due to the increased use of laboratory equipment, as well as the knowledge required to properly operate all the necessary instruments, make high-throughput research very exclusive.<sup>[20,21]</sup> In turn, it is particularly challenging to train students to use such methods.<sup>[22]</sup> Our aim in this work is to prove the principle of lab automation using open, accessible tools, with the ultimate goal of making this technology more widespread and easier to use.

In the past, much open-source software and, more recently, open hardware projects have shown that by creating a community of developers and allowing them to develop the projects together, a high level of professionalism and usability is created, with the help being offered free of charge. In terms of the modern bio-lab, many research groups have already shown that different do-it-yourself (DIY) hardware can also be applied to biological questions and that it is even possible to build customized solutions using 3D printing and electronic setups, which significantly simplify the laboratory process.<sup>[14,22–27]</sup> The quality of documentation, and the skills required to replicate a given solution are often quite variable, which currently limits the uptake of this approach.

On the other hand, projects such as the 3D-printed OpenFlexure Microscope,<sup>[25]</sup> the Octopi/SQUID project,<sup>[14]</sup> and the cellSTORM microscope<sup>[24]</sup> come with a fully illustrated manual that enables high-performance microscopy on a stand-alone or modular scale. These projects have all been replicated by the growing community of microscopy users. For a great overview of the subject, please refer to Rosario et al.<sup>[21]</sup>

In most cases, commercial devices lack open hardware or software interfaces for customization or automation, making it difficult or impossible to use them outside of the intended application for the devices. The Opentrons OT-2 pipetting robot<sup>[6,28]</sup> which we use in this manuscript is an example of commercially produced open-source hardware. Its commercial success shows that collaborative development also works in an industry context and additional functions from the developer community can be quickly incorporated.

Open-source software projects, such as the data analysis tool CellProfiler<sup>[29]</sup> or the image processing tool Fiji,<sup>[30]</sup> have created vibrant communities where the core software is enhanced with a huge library of contributed plugins. This common interface can accelerate the research process enormously by making it much easier for researchers to implement a published method, or to integrate multiple plugins together in new work. On close observation, such an interface between individual open hardware projects seems to be lacking at present. With UC2, we have already shown that it is possible to create a framework to connect different components from other projects or commercial sources so that a variety of optical experiments are possible.<sup>[22]</sup> Tools such as GitHub/Gitlab repositories in combination with proper open-source licenses help to organize such collaborations in the institutional context and beyond. This way, researchers and enthusiasts can participate and the lifetime of such a project very often outlives classical research projects.

In this work, we ask the question, how different expertise from several open-source hard- and software projects can be merged to generate a professional laboratory quality research tool. We show how the interaction between open-source tools such as the Opentrons OT-2 pipetting robot (Opentrons, New York, USA), the OpenFlexure Microscope (OFM Server<sup>[31]</sup>), ImJoy,<sup>[11]</sup> and UC2,<sup>[22]</sup> each expert in liquid handling, robotic microscopy, web-based image processing, and modular optics solutions, respectively, can help democratization of smart lab automation with a close-to turn-key solution. We will present two new compact slide-scanning (fluorescence) microscopes that can be placed directly in the Opentrons robot or cell culture incubators. Furthermore, we show an example of how a complete protocol from staining the microtubule network of fixed HeLa cells, to simultaneous in situ observation via the microscope, to quantitative analysis which can be realized via a simple interface in a web browser.

## 2. Experimental Section

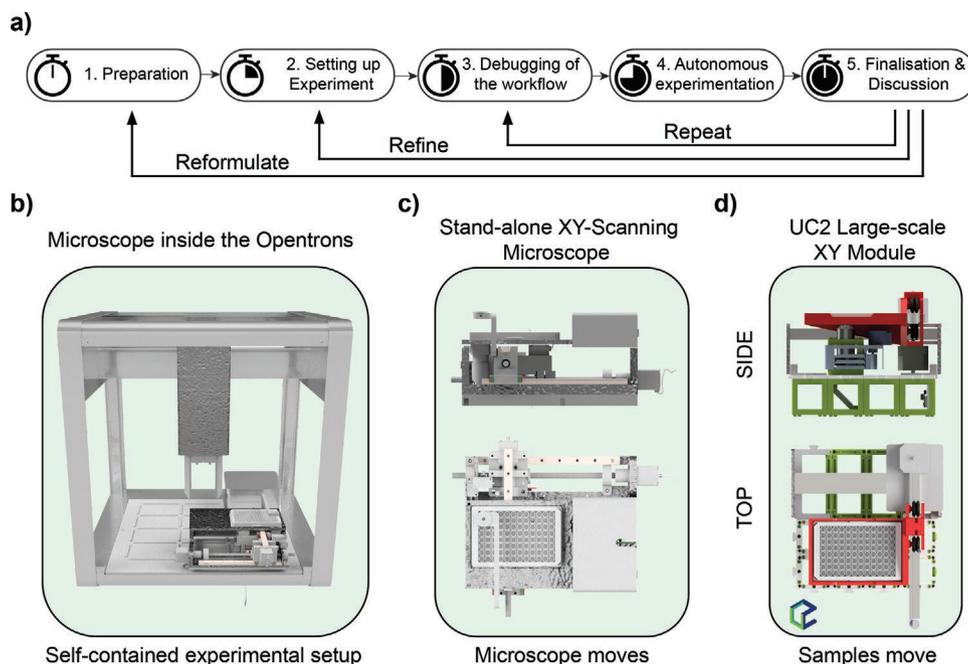
Here, a brief description of the software pipeline, the automated pipetting system (Figure 1a) as well as the two different DIY high-throughput microscopy imaging systems (Figure 1c,d) for on-site microscopic imaging and image analysis inside the Opentrons pipetting robot (Figure 1b) is given. Additionally, it is shown how all components can easily be integrated into a common workflow using Jupyter notebook to design complex biological protocols as depicted in Figure 1a. The soft- and hardware used in this project is summarized in Table 1. An extensive overview of alternative open-source DIY labware is given in refs. [14,21].

The first stand-alone microscope (Figure 1c) with a fixed sample stage was used as a development device for the software in several different locations (e.g., Sweden, Germany). From the experience gained from prototype development, the second UC2-based device (Figure 1d) moved the sample and was found much easier to reproduce.

### 2.1. Microscope Design

To perform cell observation both during the pipetting process and after completion of the protocol, two compact microscopes with cellular resolution ( $\approx 2\text{--}3\ \mu\text{m}$ ) capable of recording time series of individual wells were developed. The necessary parameters identified for the desired microscope for prototyping were

- 1) flat design to fit into lateral flow hoods, biological incubators, liquid handling robots (i.e., Opentrons OT-2),
- 2) scanning of multi-well plates (6, 24, 96), as well as individual sample, slides with high speed and reproducibility with respect to sample coordinates,
- 3) transmission brightfield and optional fluorescence imaging,
- 4) easily reproducible, expandable, and cost-effective to quickly build a variety of instruments,
- 5) an easy-to-use control system that can be easily integrated into existing workflows,



**Figure 1.** Components for a fully autonomous pipetting and imaging pipeline. a) A typical workflow starts with sketching the biological protocol and implementing it in python-based code for the opentrons/microscope. After simulating the code and setting up the experiment which typically involves placing reagents and samples inside the volume of the OT2, a first run can be started with some dummy reagents to debug the experiment. The autonomous experimentation includes robotic pipetting steps, continuous cell observation and processing on-the-fly image which can be controlled remotely. After collecting the data and an adjacent analysis, a hypothesis can be proven, the experiment repeated or refined. b) The microscope should fit into the working volume of the Opentrons OT2 pipetting robot to be integrated into a common pipetting workflow. c) The standalone high-throughput microscope “OpenmiTronScope” moves the optical assembly around a fixed sample plate. d) An extension to the modular optical toolbox “UC2” and is based on a widely available laser engraving x/y table to realize high accuracy at a low price, where the sample is moved in x/y.

- 6) simple integration of image processing tasks into the software,
- 7) perform in vitro experiments at temperature around 37 °C as well as high humidity for long periods of time with automatic focus.

Recent work such as the Incubot<sup>[34]</sup> or the open-source Lab Platform<sup>[35]</sup> had already shown how parts from a 3D printer or computer numerical control (CNC) machine could be used to create low-cost, high-throughput microscopes. The ideas presented there were taken up and two new compact 3D printed devices with a similar range of functions were demonstrated. Optimal use of the limited working volume within the OT-2 was made and the maximum pipetting height of 150 mm measured from the base was considered.

The standalone “OpenmiTronScope” (Figure 1c and Section S1, Supporting Information) offered a static sample well, moved the camera in XYZ to generate large-scale microscope images, and used a white-light light-emitting diode (LED) to realize transmission brightfield imaging at cellular resolution. A more generic solution, where customized imaging schemes such as fluorescence or multiple objective lenses for different magnification and optical resolutions were desired, was provided by the UC2-based<sup>[22,36]</sup> system “Hi2” (Figure 1d). The functionality of recently introduced open-source modular optical toolbox UC2 with the capability of large-scale fluorescence microscopy with the development of a laser engraver-based XY stage (Figure 2a) and a novel Z-focusing unit (Figure 2b) was extended. Because most parts came pre-assembled, the complexity to replicate

such devices was significantly reduced, hence enabling increased accessibility and wide-spread dissemination.

An in-depth description and analysis of the mechanical properties can be found in Sections S1 and S2 in the Supporting Information, while a set of instructions on how to manufacture and assemble the parts as well as a detailed bill of materials (BOM) can be found on the project page <https://beniroquai.github.io/Hi2>.

## 2.2. Software for Hardware Control

For software control, it was important that all components, namely, the pipetting robot, the image processing pipeline, the microscope, and any additional components such as sensors or laboratory devices, could communicate with each other so that a fully automated workflow could be created. For this purpose, integration via a local area network (LAN) using Ethernet or WiFi, was a convenient option that allowed the use of existing software to run each device, rather than requiring a single integrated application. Furthermore, intuitive control using a graphical user interface (GUI) or scripting interface such as Jupyter Notebook enabled users to get started quickly with familiar tools.

In addition to an online protocol designer, the Raspberry Pi-based Opentrons OT2 could be controlled using a Python application programming interface (API) from a Jupyter Notebook hosted on the OT2 or using a representational state transfer (REST) API via hypertext transfer protocol (HTTP) requests.<sup>[28]</sup> Similarly, the OFM server offered control through a browser-based

**Table 1.** Summary of the open-source hard- and software tools that have been combined to form the automated labeling and imaging pipeline.

Category	Name	Description	Source
Image Processing (IP)	ImJoy.io	Browser-based image processing that can use GitHub-hosted plugins; enables cloud-computing	<a href="https://imoy.io">https://imoy.io</a> <sup>[11]</sup>
	Fiji/ImageJ.js	Online-based Java image processing-tool that enables versatile processing using large variety of available plugins in the browser	<a href="https://doi.org/10.5281/zenodo.4944985">https://doi.org/10.5281/zenodo.4944985</a> <sup>[32]</sup>
	Jupyter Notebook	A Python-based notebook that organizes experimental workflows and controls execution of pipetting/imaging tasks as well as plans upcoming tasks based on image processing results	<a href="https://jupyter.org/">https://jupyter.org/</a>
Control	OFM Server	Web-server/browser-based control of microscopy hardware using IoT standards	<a href="https://gitlab.com/openflexure/openflexure-microscope-server">https://gitlab.com/openflexure/openflexure-microscope-server</a> <sup>[33]</sup>
	OFM Client	A python client that controls the OpenFlexure microscope server using the labthings-based REST API	<a href="https://doi.org/10.5281/zenodo.5534322">https://doi.org/10.5281/zenodo.5534322</a> <a href="https://gitlab.com/openflexure/openflexure-microscope-pyclient">https://gitlab.com/openflexure/openflexure-microscope-pyclient</a> <sup>[33]</sup>
Microscopy	OpenFlexure Microscope	3D-printable flexure bearing-based microscope for high-resolution light microscopy	<a href="https://openflexure.org/">https://openflexure.org/</a> <sup>[31]</sup>
	UC2-Hi2	Modular open-source microscope for high-throughput fluorescence microscopy	<a href="https://github.com/openUC2/UC2-Hi2">https://github.com/openUC2/UC2-Hi2</a> <sup>[22]</sup>
Automatic Pipetting	Opentrons OT2	Commercially available open-source pipetting robot with a Python programming interface to design biological protocols	<a href="https://opentrons.com/">https://opentrons.com/</a> <sup>[28]</sup>
Documentation	General Setup	A description of how all the different components can be used in one single Jupyter Notebook	<a href="https://beniroquai.github.io/Hi2/">https://beniroquai.github.io/Hi2/</a>

GUI and using a REST API, which allowed, e.g., the initiation of scans and captured via the browser<sup>[31]</sup> or from a Python module running anywhere on the network.

The OFM server supported the software extensions so that, e.g., the “GRBL” interface<sup>[37]</sup> used here for the stepper motor control running on a serial-connected Arduino and the USB3 camera could be integrated easily. The server was designed for low computing resources and could be used on single-board computer (SBC) such as the Raspberry Pi (v3b, UK) or the Nvidia Jetson Nano (Santa Clara, USA). Because the GUI ran in the browser, the hardware could also be controlled remotely with suitable (i.e., secured) network configuration. The core HTTP server code is available as the flask-labthings Python package.<sup>[38]</sup>

### 2.3. Software for On-Site Image Processing

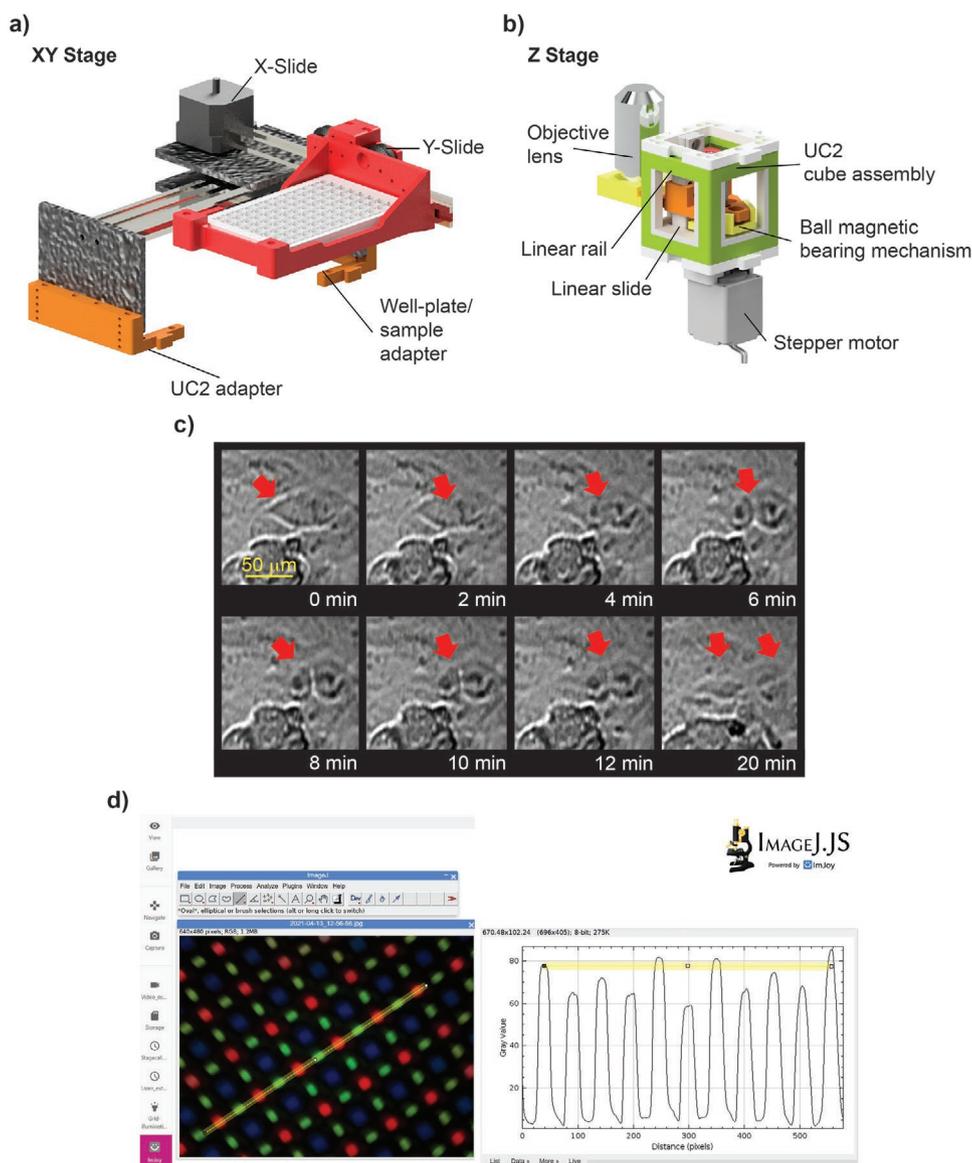
In order to process the generated image data directly without transferring each image from the microscope to an external processing machine or to involve imaging data into automated decision making for future pipetting events, the browser- and web-based image processing tool ImJoy<sup>[11]</sup> was integrated directly into the OFM server. With this, it was, e.g., possible to evaluate captured images directly in the browser using ImageJ.js<sup>[32]</sup> or already available plugins such as the ITK/VTK Viewer. This was, e.g., useful if the pixel size might be calibrated (as shown in Figure 2d), image tiles had to be stitched to form a larger field of view (FOV) or time-lapse series had to be combined into a video. ImJoy also offered a large number of plugins developed by the community which could directly be accessed from within the browser. These included previously trained networks, denoising or deconvolution algorithms to partially compensate the loss of quality caused by the low-cost hardware. ImJoy could use the computing resources of the browser in which the plugin was operated, as well as external computer clusters, e.g., for training neural networks and was therefore not limited by the low computational resources by the SBCs.

## 3. Results

### 3.1. Open Collaboration for Distributed Hardware Development

The ability to develop hardware in a decentralized fashion has proven particularly helpful in the still ongoing COVID-19 pandemic, where many scientists have limited access to the lab. A discussion of an idea of building a compact microscope for the integration in the pipetting robot was followed by first prototyping of the “OpenmiTronscope” in Jena (Germany). Two prototypes were sent to JCP and WO in Sweden in an early stage of development. With this functional prototype, it was possible to work on the further development of the software at several locations in parallel and to incorporate possible hardware and software optimizations into the next optimization iteration. Using tools such as remote coding sessions in VSCode (<https://code.visualstudio.com/>, Microsoft, Seattle, USA), versioning tools such as Github/Gitlab, and video conferencing such as Teams or ZOOM, the integration of the various components was thus realized from multiple locations even though the project partners never met in person. Any hardware changes can be quickly implemented using 3D printing or off-the-shelf components so that upgrades can be performed in situ. The iterative development process led to the UC2-based system, where all issues from the first version have been resolved to maximize user experience and stability. This approach could serve as a blueprint for future projects since it offers a very fast development process from the first prototype in February 2021 to a fully working solution in May 2021. Development was simplified by adopting an “openly developed hardware”<sup>[20]</sup> approach, removing the overhead of restricting access to designs.

The selection of the projects presented in Table 1 followed specific criteria which we will briefly discuss below. An in-depth discussion about detailed requirements for the optical system and the software can be found in adjacent sections and in the Supporting Information.



**Figure 2.** a) For the UC2-based high-throughput microscope “Hi2,” we developed two additional modules where the XY stage derived from a commercially available laser engraving machine can simply be integrated into the existing cube-based framework using a minimum of 3D-printed parts. b) The novel Z-stage features low wobbling along the optical axis with the help of a magnetic ball-bearing decoupling mechanism in combination with linear rail bearings. The hardware module is optimized for simple replication, long lifetime, and low price. c) A long-term time-lapse image series of *in vivo* HeLa cells conducted inside a cell culture incubator revealed a mitotic event. d) With the help of a newly created ImJoy extension for the OpenFlexure server, image processing, such as pixel size calibration using Imagej.js, can directly be conducted inside the browser without transferring the data to external computers.

### 3.1.1. Microscope

The microscope must be compact and adaptable to experimental conditions through simple modification. UC2 offers a modular optical approach to quickly change modules such as excitation wavelength to observe different fluorophores. At the same time, it offers a cost-effective, high-quality solution that integrates into existing laboratory workflows via OpenFlexure’s REST API.

### 3.1.2. Pipetting Robot

Although there are a large number of open-source tools for sample preparation of 96-well plates, they are often not capable

of washing samples or transferring reagents across multiple well locations. The Opentrons OT2 is capable of both by providing accurate aspirating and dispensing steps and, like the microscope, has a REST API that allows it to be controlled over the network for integration into an automated workflow.

### 3.1.3. Image Processing

The goal is to operate all devices from the browser, so that copying data for the purpose of analysis becomes unnecessary. ImJoy.io as a browser-based image processing software can therefore be integrated very well into the workflow. Through an integration in Imagej.js, which is integrated as a native “APP”

in the OpenFlexure GUI, everything can be found in one place, thus increasing the user-friendliness.

### 3.1.4. Control Software

Jupyter Notebook is used to unite all the participants in a biological workflow, such as pipetting robot, the microscope, and processing software, and to share the results reproducibly with the research community as a human- and machine-readable protocol. Using Python scripting language, the individual devices can be controlled and connected to each other both via a Python API (Opentrons OT2, ImJoy) and via a REST API (OpenFlexure /Hi2). This way, an image acquisition can be requested, processed via ImJoy and updated control commands can be sent to the pipetting robot or the microscope.

A comprehensive list of additional labware for individual experimental needs is offered by the databases from Open Know-How,<sup>[39]</sup> Thingiverse<sup>[40]</sup> or the NIH 3D print exchange.<sup>[41]</sup>

## 3.2. Using the Hi2 Microscope for Live-Cell and High-Throughput Imaging

For a high-throughput microscope, we identify the following criteria for use in biological experiments:

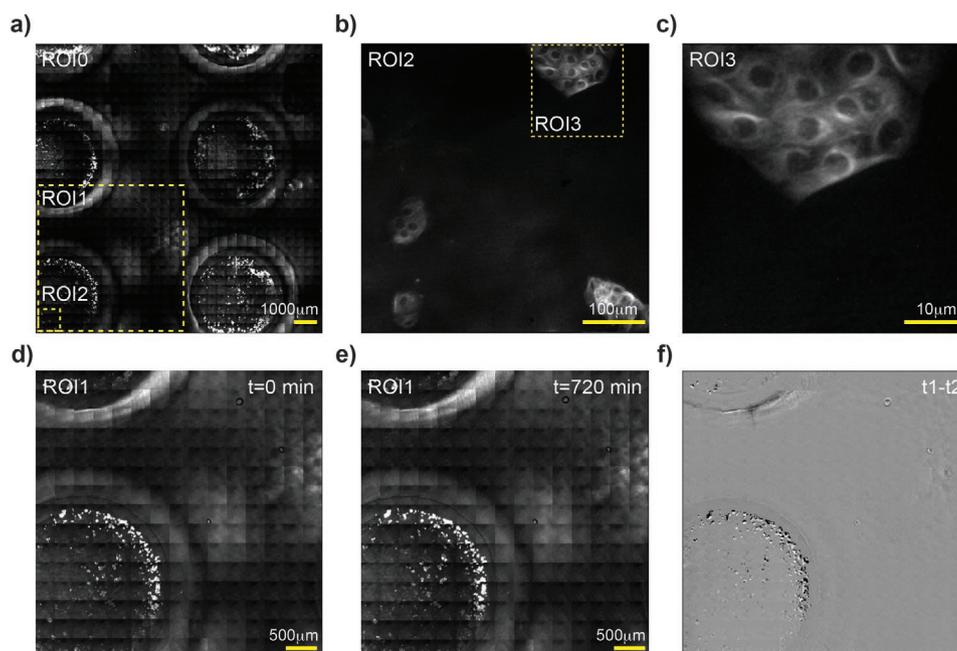
- 1) Long-term imaging in the incubator at high temperature/humidity environment
- 2) Permanent in-focus across the full well plate

- 3) Reproducibility of the targeted XYZ coordinates over multiple runs overtime to locate individual cells over time reproducibly
- 4) Ability to stitch a larger FOV from multiple image tiles
- 5) Fluorescence imaging with at least one excitation channel

3D printing material with increased glass transition temperature such as polyethylene terephthalate glycol (PETG) in combination with an autofocus algorithm, that compensates for temporal focus drift and tilted plates (see Section S3, Supporting Information), helps to perform long-term imaging experiments of living organisms in cell culture incubators, where 37 °C and high humidity represent challenging conditions for electronics and thermoplastics.

To estimate the mean displacement of multiple regions of interest within one well plate, a long-term experiment, where the microscope periodically ( $t = 5$  min) scans 32 out of the 96 wells, was performed at room temperature. The measurement in Section S3 in the Supporting Information suggests an average displacement of less than 32  $\mu\text{m}$  (4% for a region of interest (ROI)) across an 18 h measurement, leading to the possibility to continuously monitor cells at multiple locations (e.g., wells). Fluorescent imaging (Figure 3) of the robot-labeled HeLa cell sample (Thermo Fisher alpha Tubulin Monoclonal Antibody cat. Number: A11126, Alexa Fluor 647 Secondary Antibody, cat. Number: A-21235), the ability to stitch multiple image tiles to form a larger FOV directly on the device as shown in the zoomed ROI, is illustrated in Figure 3.

A detailed discussion about the selection of individual hardware components can be found in Section S1/S2 in the Supporting Information.



**Figure 3.** a) A representative stitched tile scan of a  $\mu\text{Slide}$  15-well sample slide, where HeLa cells were labeled with anti-tubulin AF647 primary/secondary antibodies using the Opentrons OT2. c) The zoomed-in version (ROI3) of b) ROI2, suggests the presence of fibrous structures which can be identified as the microtubule network. d,e) Two roi-scans were performed 12 h apart to demonstrate the location error computed in (f) as the difference between the two timestamps. Only small variations of the sample, mostly due to photobleaching can be observed.

### 3.3. Making the Labware Talk to Each Other

Until now, the individual components, such as the microscope, the GUI, and the image processing worked independently from each other. In the following, we want to show how to connect them to create a fully automated workflow. This could in turn mean that pipetting steps can be done based on the previously obtained and processed imaging results to, e.g., adjust pipetting volumes or have better environmental conditions for future experiments (e.g., buffer and antibody concentration).

A router (Netgear Nighthawk AC1900 R7000, 100€) ensures a stable connection of all devices over ethernet (Wifi for the Opentrons OT2) to ensure command as well as data transfer. An additional laptop or the Jetson Nano running the OFM Server is used to render the OFM GUI as well as the Jupyter notebook from the Opentrons. An in-detail description of how to set up the environment, perform simple tasks, and a selection of ready-to-use pipetting protocols can be found on our project webpage (<https://beniroquai.github.io/Hi2>).

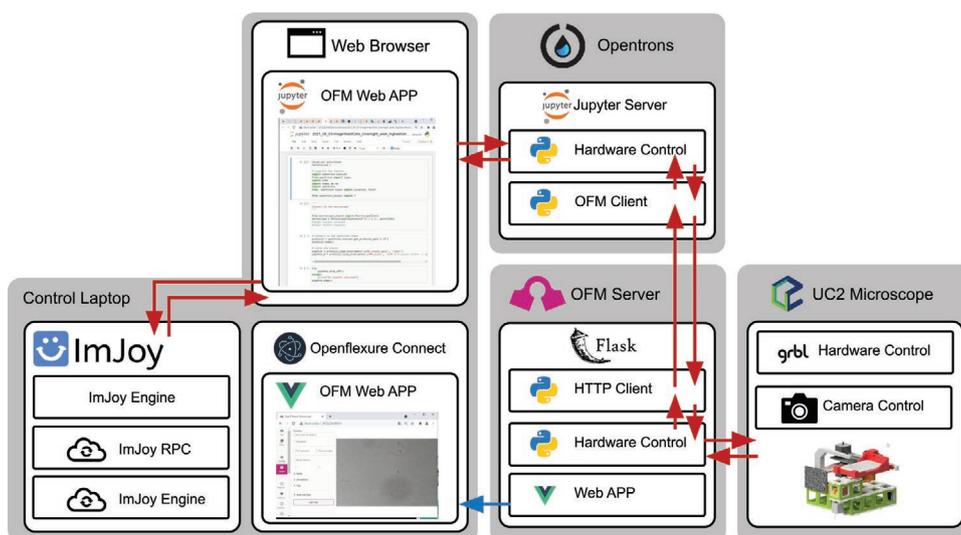
Using the Opentrons Python API, arbitrarily complex pipetting operations can be formulated in an easily readable code framework and executed both in standalone Python scripts as well as in dynamic browser-based Jupyter notebook. The latter has the advantage that each step is clearly logged, can be added to the lab notebook in a graphically easy-to-understand form, and facilitates reproducibility by sharing directly in the browser (Figure 4).

The basic code structure for a protocol is divided into the definition of the labware (pipette tip rack, well plate, microscope, etc.), which are assigned to the deck coordinates and the execution of steps such as aspiration, dispensing, or the movement from tip location A1 to well location B2, e.g. To this end, common Python libraries (e.g., numpy<sup>[42]</sup>) can be easily integrated, which is useful for controlling the Hi2 microscope or doing calculations based on computed results. Using the OFM client library (Figure 4, bottom),<sup>[43]</sup> Python commands such as

image capture, move to coordinate, or laser on/off are wrapped in human-readable protocol steps and executed via the REST API. This enables time-lapse imaging series during incubation times or fluorescence imaging after a staining process to check if the experiment worked on site. Custom labware, such as the microscopes or 3D-printed tip racks, can be added using custom labware definitions or manually set XYZ coordinates.

The ImJoy Jupyter Notebook Plugin simplifies the execution of available plugins directly in the automated workflow. A typical use case is on-the-fly image processing, where the robot requests an image capture, downloads the data from the OFM server, and transfers it to an externally running (e.g., browser of the laptop/computer cluster) ImJoy Plugin. This way, processing steps requiring high computational resources (e.g., segmentation by neural networks) can be realized without storing large data sets continuously. This enables the use of external libraries that cannot be installed on the read-only file system from the Opentrons and is especially useful for graphics rendering applications, where the ITK/VTK Viewer in the form of an ImJoy plugin helps illustrate results for further debugging.

Alternatively, e.g., in case an OT2 is not available, the microscope can conveniently be used as a standalone device, where ImJoy is integrated into the OFM web app, allowing, e.g., image processing using ImageJ.js (Figure 2d) or other ImJoy plugins. The OFM GUI can be accessed through the browser or an Electron app, providing a live view of the microscope's camera feed and controls for common imaging operations. It is possible to control the Opentrons robot using an experimental HTTP interface, moving overall control to a different device. We describe this in Section S4 in the Supporting Information but note that it is not yet feature-complete. Section S4 in the Supporting Information describes alternative configurations, e.g., using the experimental HTTP interface on the Opentrons robot to move overall control to the OpenFlexure software, but this is not yet able to access the full functionality of the Opentrons software.



**Figure 4.** Diagram for intra-device communication, where the Opentrons OT2 pipetting robot's Jupyter server controls the experiment. The protocol is formulated using human-readable python syntax, where on-the-fly image processing can be performed using available ImJoy plugins directly in the browser. Additional hardware, such as the GRBL stage and camera can be connected to the Jetson Nano, which also runs the OFM server.

### 3.4. Imaging Results from a Fully Automated Antibody Labeling Workflow

After presenting the individual building blocks for the automated pipetting, microscopic imaging, image analysis, and hardware control, the open question remains, can we get the hardware work together in order to perform a fully autonomous biological experiment? For this, we perform a standard, often very time-consuming laboratory experiment using primary and secondary antibodies for fluorescent immunostaining. A complete protocol can be found in the online repository protocols.io (see Section S5, Supporting Information). This is an ideal use case, where human errors, like varying pipetting volumes or wrong timings, can successfully be avoided since every step is properly tracked in the digital protocol. A repeated run with another fluorophore does not require immense labor work since, because the protocol can simply be run again. As shown in Figure 5, the protocol consists of several pipetting and washing steps, with image acquisition during the waiting times to observe possible reactions of samples to reagents. In the experiment presented here, only microtubules (anti-tubulin) are labeled with an Alexa Fluor 647 conjugated (anti-mouse) antibody and then automatically imaged with the fluorescence unit (see Figure 5, right). After the image acquisition, the results are directly processed using ImJoy to apply a look-up-table (LUT), perform a pixel size calibration, and quantify the cell's features (see also Section S5, Supporting Information).

A critical point of such an experiment is the debugging of the protocol without living samples and expensive reagents. For this purpose, the protocol can first be simulated before it is run on the robot without samples to detect and eliminate any errors in advance.

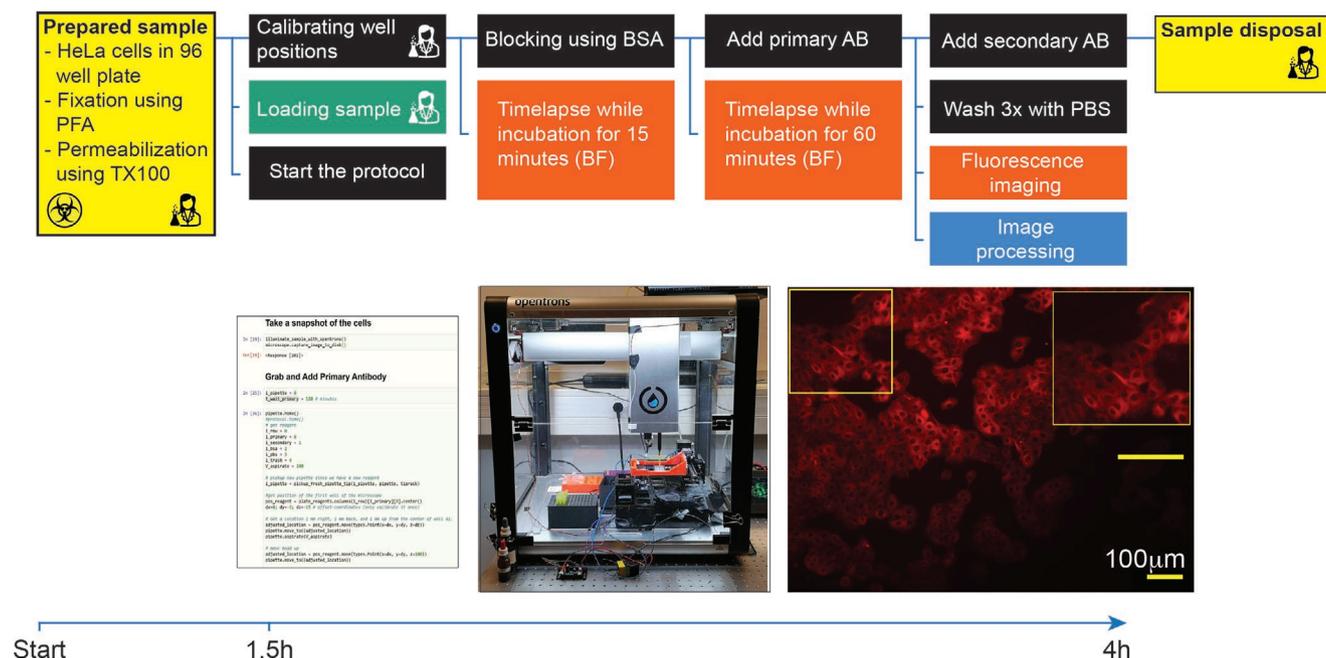
Live cell experiments inside the robot at 37 °C ambient temperature and sufficiently high humidity are possible in principle. For this purpose, similar to ref. [44], we have placed a hotplate heated to 75 °C with a 2 L water vessel in the working volume of the robot to achieve a constant temperature of about 35–37 °C with the housing completely closed.

### 3.5. Performing “Smart Microscopy” by Finding Highest Cell Density

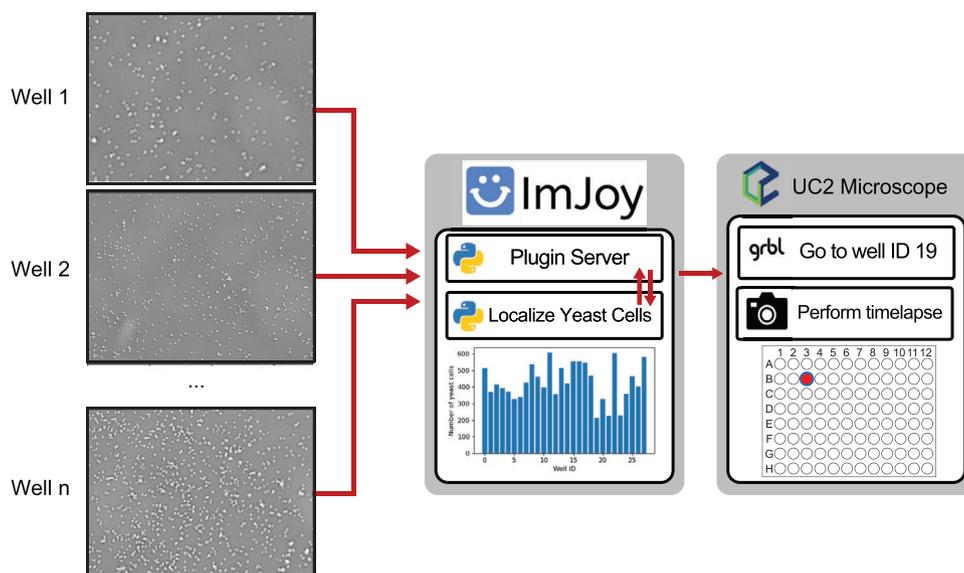
A unique feature of the closed-loop pipetting, imaging, and processing pipeline presented here is the ability to plan future steps within a protocol based on a computer-aided decision. We give a simple example in which the pipetting robot seeds an unknown quantity of yeast cells, which are imaged by the microscope after sedimentation (Figure 6). One image from each of the 96 wells is sent to a customized ImJoy plugin written in Python running on the laptop, which returns the number of yeast cells as the result. After cell preparation of all wells, the microscope moves to the position that has the lowest cell density and performs a long-time series. In this way, unknown cell densities could easily be calibrated, and subsequent experiments can be better selected in advance, e.g., to better observe the growth rate at the most ideal cell density, which can impact the biological outcome of a given experiment.

## 4. Discussion and Outlook

In this work, we demonstrate how a fusion of knowledge from open-source projects, each hosting a large and active developer



**Figure 5.** A common automated workflow that involves robot-assisted immunostaining, microscopic imaging, and image processing. The control blocks are formulated in the form of a Jupyter Notebook running on the OpenTrons OT2. All steps, except sample preparation involving fixation and permeabilization as well as sample disposal, are conducted directly at the robot.



**Figure 6.** After seeding an unknown number of yeast cells using the pipetting robot, the microscope performs a whole plate scan before a dedicated ImJoy plugin running on a separated server computes the cell density and decides which well will be analyzed in long-time experiments. In this case, well number 19 showed lowest cell density.

community, leads to a powerful tool that makes the increasingly important field of lab automation available and accessible. The total cost, including the commercial OT-2 ( $\approx 50\text{k€}$  vs  $>50\text{k€}$ , e.g., Eppendorf Eppimotion, Hamburg, Germany), the microscope ( $\approx 2\text{k€}$  vs  $>50\text{k€}$ , e.g., Biosense oCelloscope, Denmark or Cytosmart Omni, Netherlands), and the software (free vs  $\approx 20\text{k€}$ ), is about  $7\text{k€}$ , plus individual labor costs of around  $6\text{k€}$  if a Ph.D. student works on it fulltime for 2 months. This stands in contrast to  $>120\text{k€}$  and thus more than one order of magnitude less expensive than a comparable commercial setup with similar functionality. More importantly, the here presented open-source solution for the pipetting robot-contained microscope offers the ability to be readily customized since the software and hardware can be adapted and extended to individual needs. The low price and high accessibility make it ideal for the use in low-resource laboratories.

Our solution includes browser-based GUI software for controlling the microscope, Jupyter notebook, and ImJoy plugins for the reproducible execution of pipetting protocols and processing of image data. Imaging is performed on easily reproducible microscopes from the UC2 toolkit. This demonstrates that smart microscopy experiments can be realized without great effort and costs. We rely on existing projects and many off-the-shelf components like the commercially available open-source pipetting robot, laser engraving stages, and pre-assembled UC2 modules to fully concentrate on the experiment.

The additional 3D-printed parts can be easily adapted to the conditions of the experiment and replaced if necessary. However, this flexibility and the low price of the components are accompanied by significantly reduced stability compared to metal machined parts. In particular, the cantilevered sample holder in the Hi2 microscope is susceptible to low-frequency vibrations. This is particularly noticeable through small variations in the FOV and is especially visible when the robot is moving or when vibrations prevail in the building. 3D-printed parts made from

PLA start to soften at a temperature around  $37\text{ °C}$  (e.g., inside an incubator). Using PETG with a much higher glass transition temperature solves the problem, although a microscope warm-up phase with corresponding temperature-induced drift has to be considered. The low positioning error of less than  $\pm 4\%$  within the FOV with repeated movements over 18 h was a surprising finding for the low-cost x/y laser engraving table and shows the strength from the already well-engineered maker hardware for use in the scientific context.<sup>[45]</sup>

Other critical points within an experiment are possible malfunctioning behavior of components within an ongoing experiment leading to a cancelation of the process, as reagents such as antibodies are often very expensive and should not be wasted. This can be remedied by early debugging in the form of a protocol run with “dummy reagents” such as dyed water and checking the intermediate results or evaluating a series of time lapses made using a cell phone camera.

The low-cost optics used here is easy to obtain and can be replaced by, e.g., higher quality lenses to achieve a higher optical resolution and better optical performance. Additional UC2 modules such as different excitation lasers and filter cubes can enable additional functionalities such as multicolor fluorescence imaging based on the building block-based principle.

The same applies to the here made integration of the web-based image processing tool ImJoy. Available libraries, code snippets, and plugins for Fiji, Python, or JavaScript can be integrated into the workflow, allowing the measured value in the form of 2D images of the image to be quantified and used for the next experiment or an adaptation of the previous workflow. However, a truly functioning feedback loop has not yet been established.

The open-source nature of all projects in this study pays off since they allow rapid integration and provide a legally valid framework through corresponding open-source licenses. The model of open, collaborative science we demonstrate here relies

on the participation of many research groups. Often, direct commercial exploitation of research results still prevails, which can lock up intellectual property, and prevent projects like this one from using those results.<sup>[46]</sup>

## 5. Conclusion

The effort to plan an already existing biological protocol, such as the antibody labeling shown here, and to transform it into a machine-readable code is currently much larger and more time-consuming than the manual approach. If the planning phase is included, the time required is an order of magnitude higher than conducting the pipetting manually. One can therefore ask the question of whom laboratory automation will be useful. However, with the workflows presented here making this technology available to a wide audience, the ability to share protocols that have already been performed, and the prospect of better training in this area already in the university context, it seems likely that automated performance of experiments will prevail in the long term. The ability to perform many experiments in parallel or replicate an experiment exactly will greatly improve both the quality of data and the reproducibility of studies. Expensive, sophisticated automation systems are often underutilized, because many biologists are not trained in their use, there are insufficient specialist technicians available to support them. The system as presented in this manuscript is not yet a step-change in user friendliness, but our eventual goal is that, by vastly increasing the accessibility of automation, we will help it to become a more mainstream technique. Building a large community of users and developers will, in turn, help to develop training material and share know-how to improve the uptake of these powerful tools—both the open and low-cost ones, and the existing commercial systems. Our next goal is to perform closed loop experiments where, based on the data recorded in the experiments, trained neural networks can help to make decisions about future experiments and to plan and execute them autonomously.

## Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

## Acknowledgements

The authors thank Rainer Heintzmann for reviewing the draft of this manuscript. The authors want to thank Opentrons for supporting this project. The authors thank the Free State of Thuringia for funding BD. The authors acknowledge the ZIM project ZF4006820DF9 for funding H.W. Additionally the authors thank the Royal Society (URF\R1\180153, RGF\EA\181034), EPSRC (EP/R013969/1, EP/R011443/1) for funding R.B., K.B., and J.C.

Open access funding enabled and organized by Projekt DEAL.

## Conflict of Interest

The authors declare no conflict of interest.

## Data Availability Statement

All raw and processed data associated with this manuscript are accessible in the following Zenodo repository: <https://zenodo.org/record/5016986>. All design files for the hardware, as well as manuals to install the software can be accessed through the GitHub repository <https://github.com/openUC2/UC2-Hi2> or the project webpage <https://beniroquai.github.io/Hi2/>.

## Keywords

high-throughput, lab automation, machine learning, open source, smart microscopy

Received: June 23, 2021

Revised: September 14, 2021

Published online: October 24, 2021

- [1] E. Armingol, A. Officer, O. Harismendy, N. E. Lewis, *Nat. Rev. Genet.* **2021**, 22, 71.
- [2] L. A. Royer, W. C. Lemon, R. K. Chhetri, Y. Wan, M. Coleman, E. W. Myers, P. J. Keller, *Nat. Biotechnol.* **2016**, 34, 1267.
- [3] M. K. Driscoll, A. Zaritsky, *J. Cell Sci.* **2021**, 134, jcs254292.
- [4] M. Wright Muelas, F. Ortega, R. Breitling, C. Bendtsen, H. V. Westerhoff, *Sci. Rep.* **2018**, 8, 3029.
- [5] M. Baker, D. Penny, *Nature* **2016**, 533, 452.
- [6] T. Sanderson, J. C. Rayner, *bioRxiv* **2017**, 241596. <https://doi.org/10.1101/241596>.
- [7] D. Waithe, J. M. Brown, K. Reglinski, I. Diez-Sevilla, D. Roberts, C. Eggeling, *J. Cell Biol.* **2020**, 219, e201903166.
- [8] B. Burger, P. M. Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. Wang, X. Li, B. M. Alston, B. Li, R. Clowes, N. Rankin, B. Harris, R. Sebastian Sprick, A. I. Cooper, *Nature* **2020**, 583, 237.
- [9] S. Eggert, P. Mieszczanek, C. Meinert, D. W. Huttmacher, *HardwareX* **2020**, 8, e00152.
- [10] S. H. Needs, T. T. Diep, S. P. Bull, A. Lindley-Decaire, P. Ray, A. D. Edwards, *PLoS One* **2019**, 14, e0224878.
- [11] W. Ouyang, F. Mueller, M. Hjelmare, E. Lundberg, C. Zimmer, ImJoy: An Open-Source Computational Platform for the Deep Learning Era, <http://arxiv.org/abs/1905.13105> (accessed: August 2019).
- [12] L. von Chamier, 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, R. Henriques, *Nat. Commun.* **2021**, 12, 2276.
- [13] K. Sanchez, I. Kim, B. Chun, J. Pucilowska, W. L. Redmond, W. J. Urba, M. Martel, Y. Wu, M. Campbell, Z. Sun, G. Grunkemeier, S. C. Chang, B. Bernard, D. B. Page, *Breast Cancer Res.* **2021**, 23, 2.
- [14] H. Li, D. Krishnamurthy, E. Li, P. Vyas, N. Akireddy, C. Chai, M. Prakash, *bioRxiv* **2020**, 2020.12.28.424613. <https://doi.org/10.1101/2020.12.28.424613> bioRxiv.
- [15] M. Weigert, U. Schmidt, T. Boothe, A. Müller, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broadus, S. Culley, M. Rocha-Martins, F. Segovia-Miranda, C. Norden, R. Henriques, M. Zerial, M. Solimena, J. Rink, P. Tomancak, L. Royer, F. Jug, E. W. Myers, *Nat. Methods* **2018**, 15, 1090.
- [16] A. Krull, T. O. Buchholz, F. Jug, in *Proc. the IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2019**, pp. 2124–2132, <https://doi.org/10.1109/CVPR.2019.00223>.
- [17] D. P. Sullivan, E. Lundberg, *Cell* **2018**, 173, 546.
- [18] E. M. Christiansen, S. J. Yang, D. M. Ando, L. L. Rubin, P. Nelson, S. Finkbeiner, *Cell* **2018**, 173, 792.e19.

- [19] G. Lippi, G Da Rin, *Clin. Chem. Lab. Med.* **2019**, *57*, 802.
- [20] J. Stirling, R Bowman, *Des. J.* **2020**, *24*, 299.
- [21] M. Del Rosario, H. S. Heil, A. Mendes, V. Saggiomo, R. Henriques, The field guide to 3D printing in microscopy **2021**, 2021050352, Preprints, <https://doi.org/10.20944/preprints202105.0352.v1>.
- [22] B. Diederich, R. Lachmann, S. Carlstedt, B. Marsikova, H. Wang, X. Uwurukundo, A. S. Mosig, R. Heintzmann, *Nat. Commun.* **2020**, *11*, 5979.
- [23] Welcome to – Open Culture Technology, <http://www.gaudi.ch/GaudiLabs/> (accessed: May 2021).
- [24] B. Diederich, Ø. Helle, P. Then, P. Carravilla, K. O. Schink, F. Hornung, S. Deinhardt-Emmer, C. Eggeling, B. S. Ahluwalia, R. Heintzmann, *bioRxiv* **2020**, 2020.09.04.283085. <https://doi.org/10.1101/2020.09.04.283085>.
- [25] J. P. Sharkey, D. C. W. Foo, A. Kabla, J. J. Baumberg, R. W Bowman, *Rev. Sci. Instrum.* **2016**, *87*, 025104.
- [26] A. Maia Chagas, L. L. Prieto-Godino, A. B. Arrenberg, T. Baden, *PLoS Biol.* **2017**, *15*, e2002702.
- [27] Q. Lu, G. Liu, C. Xiao, C. Hu, S. Zhang, R. X. Xu, K. Chu, Q. Xu, Z. J. Smith, *PLoS One* **2018**, *13*, e0194063.
- [28] Opentrons | Open-source Lab Automation, starting at \$5,000 | Opentrons, <https://opentrons.com/> (accessed: May 2021).
- [29] C. McQuin, A. Goodman, V. Chernyshev, L. Kamensky, B. A. Cimini, K. W. Karhohs, M. Doan, L. Ding, S. M. Rafelski, D. Thirstrup, W. Wiegraebe, S. Singh, T. Becker, J. C. Caicedo, A. E. Carpenter, *PLoS Biol.* **2018**, *16*, e2005970.
- [30] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, A. Cardona, *Nat. Methods* **2012**, *9*, 676.
- [31] OpenFlexure /openflexure-microscope-server · GitLab, <https://gitlab.com/openflexure/openflexure-microscope-server> (accessed: May 2021).
- [32] W. Ouyang, YW. imjoy-team/imajoy.js: Image.JS v0.4.0, <https://doi.org/10.5281/ZENODO.4944985> (accessed: June 2021).
- [33] J. T. Collins, J. Knapper, J. Stirling, S. McDermott, R Bowman, Modern Microscopy with the Web of Things: The OpenFlexure Microscope Software Stack. arXiv preprint arXiv:2101.00933, **2021**, 1.
- [34] G. O. T. Mercés, C. Kennedy, B. Lenoci, E. G. Reynaud, N. Burke, M Pickering, *HardwareX* **2021**, *9*, e00189.
- [35] P. Katunin, J. Zhou, O. M. Shehata, A. A. Cadby, A. Nikolaev, *Front. Cell Dev. Biol.* **2021**, *9*, 697584.
- [36] H. Wang, R. Lachmann, B. Marsikova, R. Heintzmann, B Diederich, UCsim2: 2D structured illumination microscopy using UC2, *bioRxiv* **2021**.
- [37] grbl/grbl: An open source, embedded, high performance g-code-parser and CNC milling controller written in optimized C that will run on a straight Arduino, <https://github.com/grbl/grbl> (accessed: May 2021).
- [38] labthings/python-labthings: Python implementation of LabThings, based on the Flask microframework, <https://github.com/labthings/python-labthings> (accessed: June 2021).
- [39] Open Know-How, <https://openknowhow.org/> (accessed: July 2021).
- [40] Thingiverse – Digital Designs for Physical Objects, <https://www.thingiverse.com/> (accessed: July 2021).
- [41] NIH 3D Print Exchange | A Collection of Biomedical 3D Printable Files and 3D Printing Resources Supported by the National Institutes of Health (NIH), <https://3dprint.nih.gov/> (accessed: August 2021).
- [42] C. R. Harris, 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, T. E. Oliphant, *Nature* **2020**, *585*, 357.
- [43] OpenFlexure /openflexure-microscope-pyclient · GitLab, <https://gitlab.com/openflexure/openflexure-microscope-pyclient/> (accessed: May 2021).
- [44] R. H. Vera, E. Schwan, N. Fatsis-Kavalopoulos, J Kreuger, *PLoS One* **2016**, *11*, e0167583.
- [45] S. Baas, V Saggiomo, *HardwareX* **2021**, *10*, e00219.
- [46] A. Parker, A. Novak, Building Blocks for Better Science: Case Studies in Low-Cost and Open Tools for Science | Wilson Center, <https://www.wilsoncenter.org/publication/building-blocks-better-science-case-studies-low-cost-and-open-tools-science> (accessed: June 2021).
- [47] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, *TensorFlow: A System for Large-Scale Machine Learning*, <https://research.google/pubs/pub45381/> (accessed: June 2021).
- [48] 3.0 O. Feature Detection and Description — OpenCV 3.0.0-dev documentation, [http://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_table\\_of\\_contents\\_feature2d/py\\_table\\_of\\_contents\\_feature2d.html](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_table_of_contents_feature2d/py_table_of_contents_feature2d.html) (accessed: May 2016).
- [49] UC2 – Automated Immunostaining Protocol using the Opentrons OT2 and UC2 High-throughput microscope – V1, <https://www.protocols.io/view/uc2-automated-immunostaining-protocol-using-the-op-bw8dphs6> (accessed: August 2021).