Check for updates

SOFTWARE TOOL ARTICLE

REVISED Qudi-HiM: an open-source acquisition software package for highly multiplexed sequential and combinatorial optical imaging

[version 2; peer review: 2 approved]

Franziska Barho[1], Jean-Bernard Fiche[1], Marion Bardou[1], Olivier Messina[1], Alexandre Martiniere[2], Christophe Houbron[1], Marcelo Nollmann[1]

[1]Centre de Biologie Structurale, Centre National de la Recherche Scientifique, UMR5048, Montpellier, 34090, France
[2]Institut Agro Montpellier, INRAE, Montpellier, 34000, France

v2 **First published:** 07 Apr 2022, **2**:46
https://doi.org/10.12688/openreseurope.14641.1
Latest published: 19 Jul 2022, **2**:46
https://doi.org/10.12688/openreseurope.14641.2

Abstract

Multiplexed sequential and combinatorial imaging enables the simultaneous detection of multiple biological molecules, *e.g.* proteins, DNA, or RNA, enabling single-cell spatial multi-omics measurements at sub-cellular resolution. Recently, we designed a multiplexed imaging approach (Hi-M) to study the spatial organization of chromatin in single cells. In order to enable Hi-M sequential imaging on custom microscope setups, we developed Qudi-HiM, a modular software package written in Python 3. Qudi-HiM contains modules to automate the robust acquisition of thousands of three-dimensional multicolor microscopy images, the handling of microfluidics devices, and the remote monitoring of ongoing acquisitions and real-time analysis. In addition, Qudi-HiM can be used as a stand-alone tool for other imaging modalities.

Keywords

optical microscopy, multiplexed imaging, chromosome organization, transcription

This article is included in the European Research Council (ERC) gateway.

Open Peer Review

Approval Status ✓ ✓

	1	2
version 2 (revision) 19 Jul 2022	✓ view	✓ view
version 1 07 Apr 2022	? view	? view

1. **Ricardo Henriques**, Instituto Gulbenkian de Ciência, Oeiras, Portugal
Simao Coelho, Instituto Gulbenkian de Ciência, Oeiras, Portugal

2. **Romain Laine**, MicrographiaBio, London, UK

Any reports and responses or comments on the article can be found at the end of the article.



This article is included in the Medical Biotechnology gateway.



This article is included in the Horizon 2020 gateway.



This article is included in the Genetics and Genomics gateway.



This article is included in the Analytical Techniques collection.



This article is included in the Diagnostic Imaging Technology collection.



This article is included in the Epigenetics collection.



This article is included in the Advances in Optics collection.

Corresponding author: Marcelo Nollmann (martinez@cnrs.fr)

Author roles: Barho F: Conceptualization, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Fiche JB**: Conceptualization, Formal Analysis, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Bardou M**: Validation; **Messina O**: Validation; **Martiniere A**: Validation; **Houbron C**: Validation; **Nollmann M**: Conceptualization, Funding Acquisition, Methodology, Project Administration, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under the grant agreement Numbers: 862151 (Next Generation Imaging [NGI]) & 724429 (Epigenomics and chromosome architecture one cell at a time [EpiScope]) (M.N.). We also acknowledge the Bettencourt-Schueller Foundation for their prize 'Coup d'élan pour la recherche Française', and the France-BioImaging infrastructure supported by the French National Research Agency (grant ID ANR-10-INBS-04, "Investments for the Future").
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2022 Barho F *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Barho F, Fiche JB, Bardou M *et al.* **Qudi-HiM: an open-source acquisition software package for highly multiplexed sequential and combinatorial optical imaging [version 2; peer review: 2 approved]** Open Research Europe 2022, **2**:46 https://doi.org/10.12688/openreseurope.14641.2

First published: 07 Apr 2022, **2**:46 https://doi.org/10.12688/openreseurope.14641.1

REVISED **Amendments from Version 1**

The manuscript has been adapted to address the remarks of the reviewers. Specifically, we added a new table describing the hardware components handled by Qudi-HiM, a new section with instructions on how to add new hardware components, a more extensive explanation of reporting methods and how these can be used to assess experiments, a new figure describing the workflow used by Qudi-HiM to setup, run and control automatized experiments, in addition to several text revisions.

Any further responses from the reviewers can be found at the end of the article

Motivation and significance

Cutting-edge developments of novel fluorescence microscopy methods have flourished over the last decade, particularly those aimed at revealing the organization and dynamics of biological systems at multiple scales[1–3]. These developments focused particularly on critical technological limitations, such as spatial resolution, imaging depth, imaging speed, or photo-toxicity[4]. An additional inherent limitation of fluorescence microscopy methods relies on their ability to image multiple molecular species at once. Typically, the restriction arises from the overlap in the emission spectra of available fluorophores. Recently, this limitation was addressed by several technologies that combine liquid-handling robots to widefield fluorescence microscopy to perform sequential or combinatorial acquisition of different molecular species, such as RNA, DNA or proteins[5–13] (Figure 1A). These multiplexing imaging technologies now enable the detection of hundreds to thousands of different biological objects at once[14,15].

The simultaneous detection of multiple targets is particularly important to understand the organization and function of chromatin at the single-cell level[9–12]. In eukaryotes, chromatin is organized at multiple length scales, from nucleosomes to chromosome territories[16]. Multi-scale three-dimensional

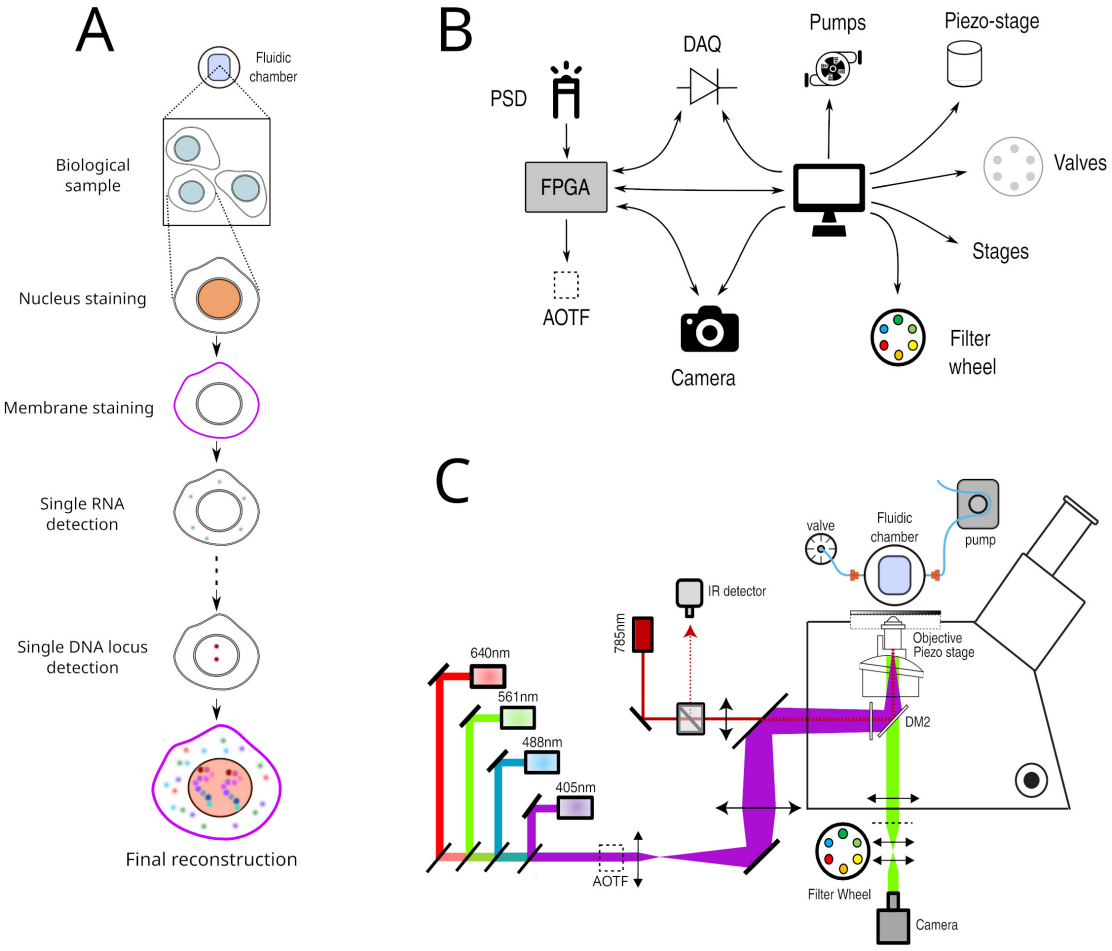


Figure 1. A. Principle of Hi-M imaging. A single region of interest within the biological sample contains several cells. In cycle 0, a DNA specific dye is used to label the nucleus and other markers. In the following cycles, different probes are injected and imaged. Data from all cycles are combined to reconstruct the final image. **B**. Overview of hardware devices that are addressed by Qudi-HiM. **C**. Schematic drawing of a Hi-M microscope. The setup shown here includes 4 excitation lasers (405, 488, 561, 640 nm), an acousto-optic tunable filter, flat and dichroic mirrors, a microscope objective, emission filters, microscope camera, a 785 nm infrared laser and detector, sample holder with microfluidic chamber, sample positioning stage, and a liquid handling robot.

(3D) conformation is involved and influences many biological processes, such as transcription[17]. Thus, it is of particular importance to be able to simultaneously visualize 3D chromatin organization and transcriptional output in single cells.

We recently developed a multiplexed imaging approach (Hi-M) to simultaneously capture the conformation of chromatin and the transcription of genes in single cells[5,10], and used it to investigate whether and how single-cell chromatin structure influences transcription[18]. In Hi-M, as in the other multiplexing methods mentioned above, each cycle of acquisition requires the coordinated control of several fluid-handling hardware components (*e.g.* pumps, valves, motorized stages) to deliver and incubate different fluids into the sample, and of multiple devices (*e.g.* acousto-optic tunable filters, piezoelectric and motor stages, cameras) to acquire 3D images at multiple regions of interest (ROIs) (Figures 1B, 1C). Acquisition of each of these sequential cycles can require tens of minutes, thus a typical Hi-M experiment with tens of cycles can last several days of continuous and unsupervised acquisition (depending on sample and number of ROIs). As a consequence, automatic, flexible, and robust experimental control software is required for efficient and reproducible multiplexed measurements.

Here, we present Qudi-HiM: an open-source, modular, user-friendly and robust acquisition interface written in Python 3 to perform sequential and high-throughput fluorescence imaging. The software was developed with a focus on multiplexed imaging of DNA, RNA and proteins, but can be easily extended to other fluorescence imaging modalities such as single-molecule localization microscopy[19].

Software description

Software architecture

Python software packages, a popular alternative to commercial environments. Developing custom software is often necessary when implementing new microscopy technology, as this provides enough flexibility for interfacing different hardware components and optimizing custom acquisition workflows. Common approaches used to develop custom software include the use of commercial environments, such as LabView (National Instruments, US) (RRID:SCR_014325), that can interface with APIs (application programming interfaces) or SDKs (software development kits) to control hardware. However, LabView requires expensive licenses, depends on upstream development from the hardware constructors, and is less adaptable to version control systems compared to textual programming languages due to its binary content.

As an alternative, we turned to Python (RRID:SCR_008394), a popular programming language that is intelligible due to its high level of abstraction and that benefits from a dynamic and expanding scientific user community[20]. This is further illustrated by the fact that most hardware components can now be interfaced using Python packages, either developed by the constructors or end-users, and are often freely available on GitHub. A large set of Python packages are continuously developed for image analysis, for graphical user interface design and for hardware control. For example, the majority of novel image analysis involving artificial intelligence methods typically provide Python APIs[21–23]. Finally, Python can directly interface with SDK libraries to control hardware devices and offers the possibility to combine data acquisition and real-time analysis using Python APIs from robust, community-led packages (*e.g.* NumPy (RRID:SCR_008633)[20], scikit-image (RRID:SCR_021142)[23] or astropy (RRID:SCR_018148)[24,25]) or from newly developed image analysis methods. Python is therefore an appealing choice for designing home-build microscopy control software.

Several software packages have been recently developed using Python for various imaging modalities. These include Qudi[26], the software package Qudi-HiM is based on; Python Microscopy; Oxford Microscope-Cockpit[27]; Tormenta[28]; Storm-Control; and MicroMator[29].

Qudi. Qudi is a modular laboratory experiment management suite written in Python 3 and the Qt Python-binding pyQt. It provides a stable and comprehensive framework for the development of custom graphical user interfaces (GUI) and hardware interfacing. The Qudi architecture consists of a core infrastructure that envelopes application-specific science modules.

Qudi's core contains the general functionalities for program execution, logging, error handling and configuration reading. Its structure is not limited to a particular scientific application. Hence, it was possible to reuse it as the backbone for Qudi-HiM[30]. Its central part is a manager module that uses a setup-specific configuration file to load and connect hardware components and control them through so-called science modules.

The science modules provide all the functionalities that are required for the control of a specific instrument. These modules are organized into three categories following the model-view-controller design pattern: hardware modules, experimental logic, and GUI. Hardware modules represent the physical devices and make the basic hardware functionalities accessible. The experimental logic assembles hardware commands into higher-level operations to control experiments in a convenient manner. Experimental logic modules serve as a communication channel between GUI and hardware. Due to the use of an abstract parent class for hardware devices that provide the same functionalities (*e.g.* different camera technologies or different types of actuators for controlling sample position), interchanging hardware modules is completely transparent for the experimental logic layer. Finally, the GUI provides a means to control the experiments and to directly visualize data without an in-depth knowledge of the underlying code. This software architecture enables rapid adaptation and portability to microscope setups with different hardware components, and facilitates long-term maintenance, adaptation to new user needs, and development of new functionalities. Finally, the use of the same GUI and underlying logic for different microscopy setups (*e.g.* widefield, time-lapse or Hi-M imaging) simplifies user training, which is advantageous for imaging facilities open to multiple users with different backgrounds.

Software functionalities

Multiplexed imaging requires not only the selection and synchronization of light sources, filters, detectors and the control of sample displacements, but also the handling of injection of liquids by a robotic system consisting of valves, pumps, and motorized stages (Figures 1B, 1C). To decouple these different operations and to make the acquisition of Hi-M datasets user-friendly and flexible, we developed separate modules for imaging, sample positioning and for fluidics handling. The GUI modules and their main functionality are summarized in Table 1 and described in the following sections.

Basic tools for imaging

Basic imaging toolbox. The Basic imaging module is a polyvalent tool for custom microscopes that provides all fundamental operations needed for imaging on an intuitive user interface (Figure 2A). In the context of multiplexed imaging, this includes the visual inspection of the sample to assess its quality using fluorescence and brightfield imaging, such as checking for labeling efficiency and specificity, cell density, tissue integrity, *etc*. Specifically, this module allows for the selection of intensity and wavelength of the light sources, the switching of filters, and the setting of acquisition parameters for the camera, such as exposure time, gain, acquisition mode, and sensor temperature, independently of the camera technology (emCCD, sCMOS, CMOS) deployed on a setup. Besides real-time inspection, imaging data can be saved using multiple common formats such as TIFF, FITS (FITS Documentation Page) or NumPy arrays[20].

ROI selector. Multiplexed microscopy involves the repeated acquisition of images in several regions of interest per cycle. Hence, a module was developed to manually or automatically select the ROIs that will be acquired during a multiplexed acquisition (Figure 2B). Automatic selection of ROIs includes mosaic and interpolation tools to explore larger portions of a sample efficiently using snake-like patterns. Mosaics are constructed by defining the central position, and the number of tiles. Alternatively, the user can define the edges of a region to be explored and the module will automatically interpolate tiles to cover the entire region. For both automatic ROI selection methods, an overlap area between tiles can be parametrized to optimize tile realignment and reassembly.

Focus toolbox. The Focus module (Figure 2C) is used to change the focal plane by translating the objective or sample relative to each other by means of a piezo-electric or motorized stage. In addition, this module maintains the selected focal plane by using a feedback system. The autofocus system uses the reflection of a near-infrared laser (785 nm) to detect changes in the objective-sample distance[31] and actively maintains the setpoint using a proportional-integral-differential (PID) controller. Thus, depending on the sample and the requirements of each experiment, this module can either be used to maintain focus over time or to locally define the focal plane for each ROI. For both applications, several modalities have been developed depending on the detectors available on the microscope (*e.g.* CMOS camera, position sensitive device, *etc*.).

Fluidics toolbox. The purpose of the fluidics GUI (Figure 3A) is to control the devices that deliver different liquid solutions into the microfluidics chamber. The hardware components include modular valve positioners, pumps, flow-rate sensors, as well as motorized stages, which together form a custom-made liquid handling and delivery robot. Using these devices and the

Table 1. Overview of Qudi-HiM graphical user interface (GUI) modules with their key functionalities. The corresponding graphical user interfaces are shown in Figure 2 and Figure 3. Colorcode: yellow – imaging modules, light red – fluidics modules, blue – automation modules.

GUI Module	Functionality
Basic imaging toolbox	Real-time acquisition Camera settings and status Illumination control (laser & white light source) Filter control
Region of interest (ROI) selector	Spatial position navigation Registration of ROI lists Mosaic and interpolation tools
Focus toolbox	Manual focus Autofocus calibration Focus stabilization mode Local focus search
Fluidics toolbox	Valve position selector 3-axes positioning stage for probe selection Pressure control, flow rate and volume measurement
Injections configurator	Definition of injection sequences for Hi-M experiments
Experiment configurator	Variable form for easy configuration of various experiment types

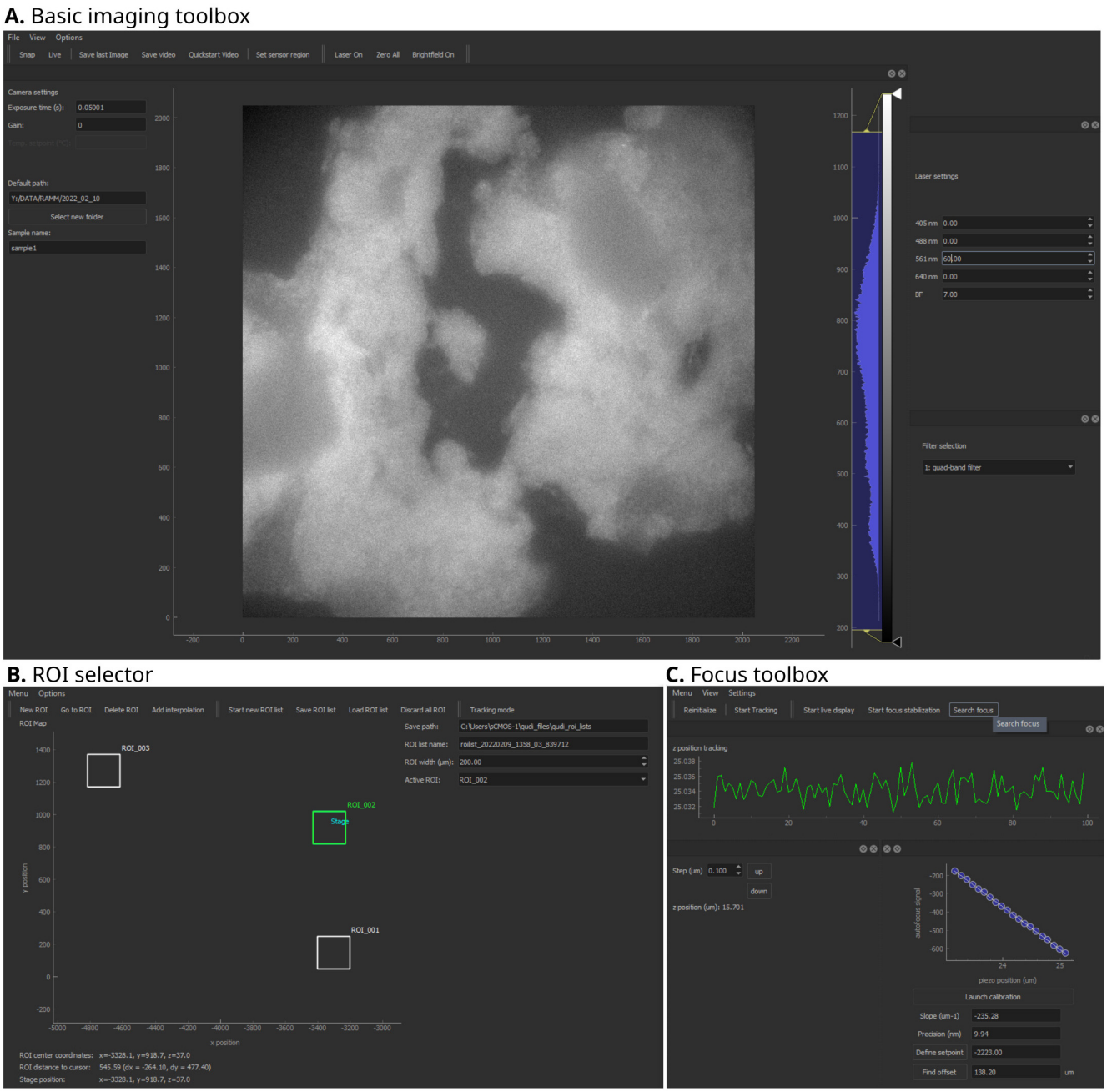


Figure 2. Qudi-HiM imaging graphical user interface (GUI) modules. A.Basic imaging toolbox. **B**. Region of interest (ROI) selector. **C**. Focus toolbox. Key functionalities of each module are summarized in Table 1.

operations defined in the underlying experimental logic, injection sequences can be performed in a controlled and reproducible manner. Fine tuning of the flow rate and injected volumes is achieved using a PID feedback loop on the pump system.

Injections configurator. A sequential acquisition experiment requires repeating a series of injection and incubation steps in a well-controlled manner that can include: hybridization of imaging probes, washing steps to remove unbound probes, chemical bleaching, and introduction of buffers to prevent photobleaching during imaging. Qudi-HiM provides a GUI (Figure 3B) to configure injection sequences, including information on the products (*e.g.* names, valve positions, etc.), the target volume, required flow rate and eventually, incubation time.

Automatized experiments

Qudi natively provides a handler for iterative experiments: the Task runner. This module is based on a state machine that handles a sequence of steps such as 'starting', 'running', 'pausing', 'resuming', 'finishing', 'stopped', *etc*. Custom experiments,

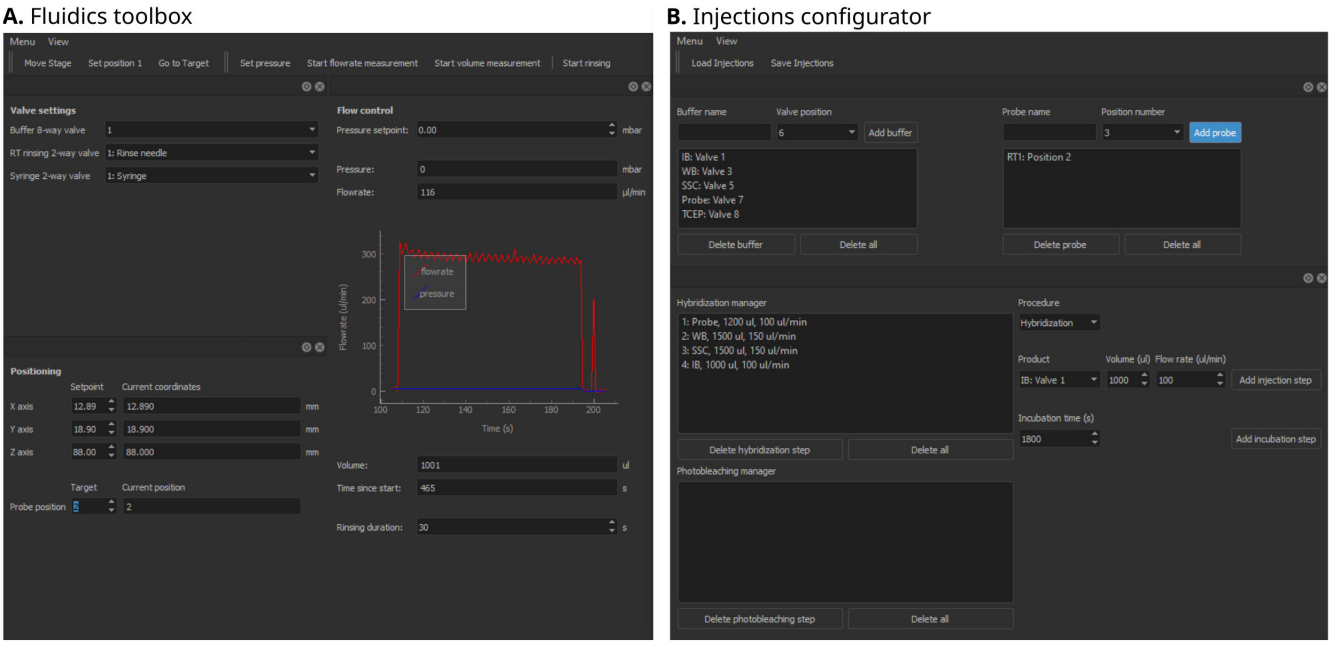


Figure 3. Qudi-HiM fluidics graphical user interface (GUI) modules. A. Fluidics toolbox. **B**. Injections configurator. Key functionalities of each module are summarized in Table 1.

called "Tasks", are defined by the actions performed on each state transition.

Several modules of a typical Hi-M experiment (*e.g.* acquisition of multicolor stacks of images, injection sequences, *etc*.) as well as the complete pipeline for a Hi-M acquisition were implemented and made available in the Task runner. The idea behind this implementation was to turn Qudi-HiM into a versatile imaging tool: efficient for running complex experiments such as Hi-M and helpful for executing simpler acquisitions, such as those required during experiment setup, protocol optimization or exploratory tests.

Each task requires a set of user parameters that needs to be easily accessible and adapted according to the sample and the experimental conditions. For this reason, an experiment configurator module for the setup of experiments is included in Qudi-HiM. Depending on the selected task, the configurator GUI displays a form that provides access to all required parameters and verifies their validity before saving the configuration (Figure 4). The configuration is stored as a YAML file and is subsequently loaded by the Python script that contains all information for the task execution. The script itself is controlled via the Task runner GUI: Here, the user selects which task to run and has the ability to run, stop, pause or resume the task. The task script connects to all required experimental logic modules to combine the necessary high level hardware actions.

Since multiplexed acquisitions can last for several days, it is convenient for the users to track the current status of their experiment. Besides visualization on the Qudi-HiM GUIs and a textual information status on the log for local observation, we also provided a file exchange-based interface from Qudi-HiM software to a status-tracker application using Bokeh (Bokeh documentation) to display key parameters of the experiment (e.g. cycle number, fluidics status, etc.) and a real-time pre-analysis of the acquired image data.

Implementation of hardware dummies

Qudi-HiM may be used on a computer without connected devices thanks to the implementation of dummy devices: mock-ups of the microscope hardware components that simulate their physical response. The dummies are especially useful during development and for the initial tests of the logic and GUIs. They were introduced in the Qudi software package and we consistently continued developing this approach, including the use of dummy tasks to simulate sequences of the Hi-M experiment for optimization

Currently supported hardware and extensions

An overview of all devices that are available in the current version of Qudi-HiM is shown in Table 2. An online resource with the most up to date supported hardware is available here. Swapping devices between experimental setups is straightforward, as long as the device has been interfaced in Qudi-HiM's hardware modules. This simply requires updating the setup configuration file to interface the new piece of hardware. Example entries are provided in the hardware modules documentation (see examples of configuration files at: https://github.com/NollmannLab/qudi-HiM/tree/master/config/custom_config).

Extending the available hardware is achieved by setting up a new hardware module and implementing an abstract interface using existing Python APIs or C libraries. Finally, the configuration file is updated to interface the new hardware component.

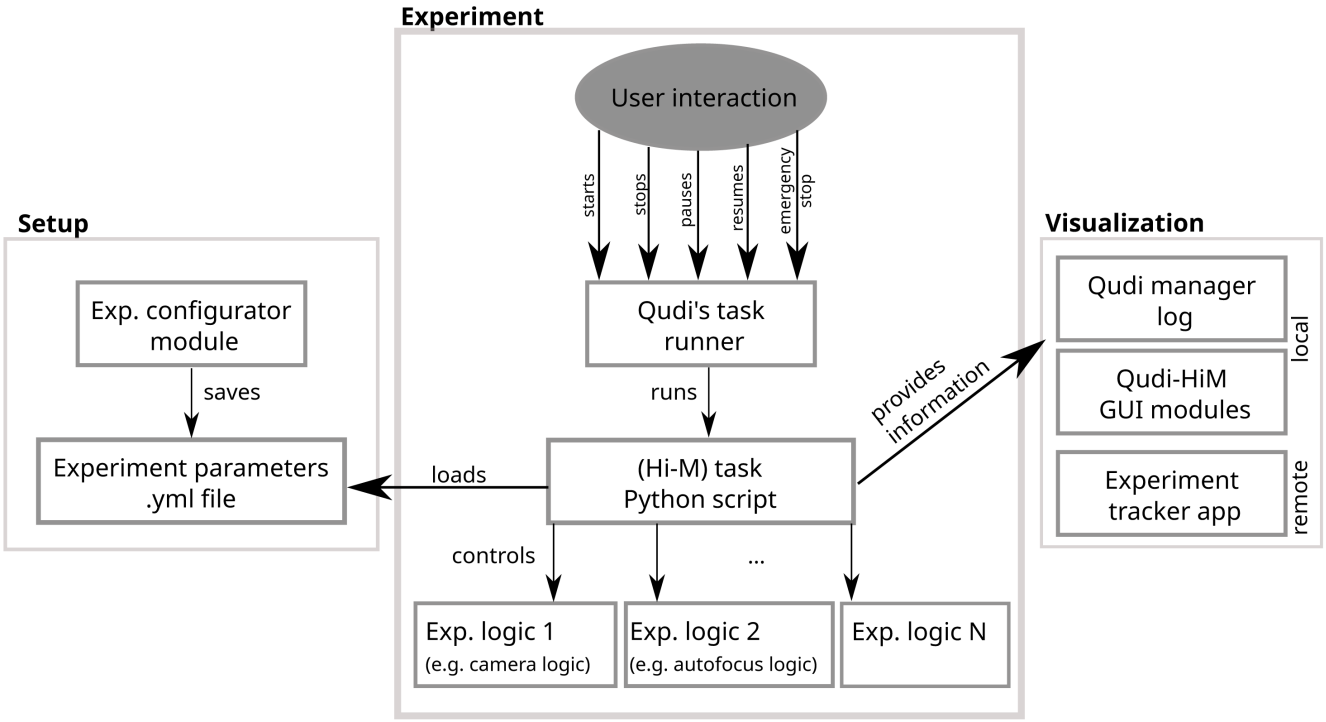


Figure 4. Scheme of the workflow used by Qudi-HiM to set up, run and control automatized experiments.

Table 2. Overview of currently available hardware in Qudi-HiM.

Hardware class	Supported devices
Camera	Andor emCCD - iXon Ultra 897 Hamamatsu sCMOS - ORCA-Flash4.0 V3 Thorlabs CMOS - DCC1545M
DAQ	National Instruments DAQ - M-Series PCIe-6259 Measurement computing DAQ - USB3104
FPGA	National Instruments RIO FPGA - PCIe-7841R
Lightsources	Lumencor Celesta Light Engine
Motorized stages	Applied Scientific Instrumentation - MS-2000 XY and XYZ Physik Instrumente Motor - Controller C-863 Mercury and C-867
Single axis piezoelectric stage for objective	Physik Instrumente - E-625 Controller & PIFOC stage Mad City Labs - NanoDrive & Nano-F-100
Filter wheels	Thorlabs - FW-102C Thorlabs - FW-103 High-Speed Motorized & APT Motor BSC201
Microfluidics	Fluigent - Flowboard FLB Fluigent Pump - MFCS-EZ Fluigent Flow Rate Sensor - Flow Unit L Hamilton Modular Valve Positioner

Use cases

A typical Hi-M experiment requires a succession of 20–60 hybridization / image acquisition / chemical bleaching cycles (Figure 5A). In each cycle, 3D multicolor image data are acquired on multiple predefined ROIs (Figure 5B). Critical steps to ensure the success of the experiment are: (1) the precise control of injected volumes at target flow rates for each step; (2) the correct and reproducible focusing of the sample for all the pre-defined ROIs (see Refs 5,10 for a detailed description of the autofocus hardware); (3) the reliable synchronization between light sources, camera and actuators controlling the sample XYZ position; (4) the storage of imaging data using a filename format encoding the cycle number and ROI, and of metadata containing the imaging and fluidics parameters. For Hi-M experiments, metadata saved with each image include sample name, imaging information such as exposure time, laser lines and intensity, autofocus parameters, sample XY position, scan step size and total Z range covered. To comply with a standardized metadata format[32] depending on the type of experiment that is performed, it is possible to add custom information by adapting the underlying software module containing the instructions of the experiment.

A Hi-M experiment starts with the injection of a DNA-specific dye to identify each single cell and fiducial imaging probe used for later image registration[5,10]. This procedure, designated by cycle 0 in Figure 5B, is performed separately from the other cycles since it requires a very specific combination of injections and incubation periods. The user first defines the adequate procedure using the injection configurator (Figure 3B): for each single step the user chooses which solution to inject, the target volume and the expected flow rate (typically 1–2 ml at 150–300 µl/min). Then, the automatic procedure is launched using a dedicated fluidics task from the Qudi Taskrunner. During this procedure, the program continuously monitors the flow-rate according to the user's requirements and adjusts the pump speed through a PID controller. When the right amount of solution is injected, the program automatically switches to the following solution or incubation period until the procedure is completed. Next, the user checks the quality of the fiducial labeling and if the cells have been properly labelled using the imaging toolbox (Figure 2A, continuous illumination with either 405 nm or 561 nm laser). Finally, the ROIs are manually selected using the ROI selector (Figure 2B), by identifying the most relevant sample positions. These ROIs positions are saved as a YAML file.

In anticipation of the successive acquisition cycles and to ensure proper reproducibility, a focus calibration is performed using the Focus toolbox GUI (Figure 2C, Figure 5C). The user selects a typical ROI, focuses on the sample and launches the calibration procedure: the program performs a quick 2 µm ramp

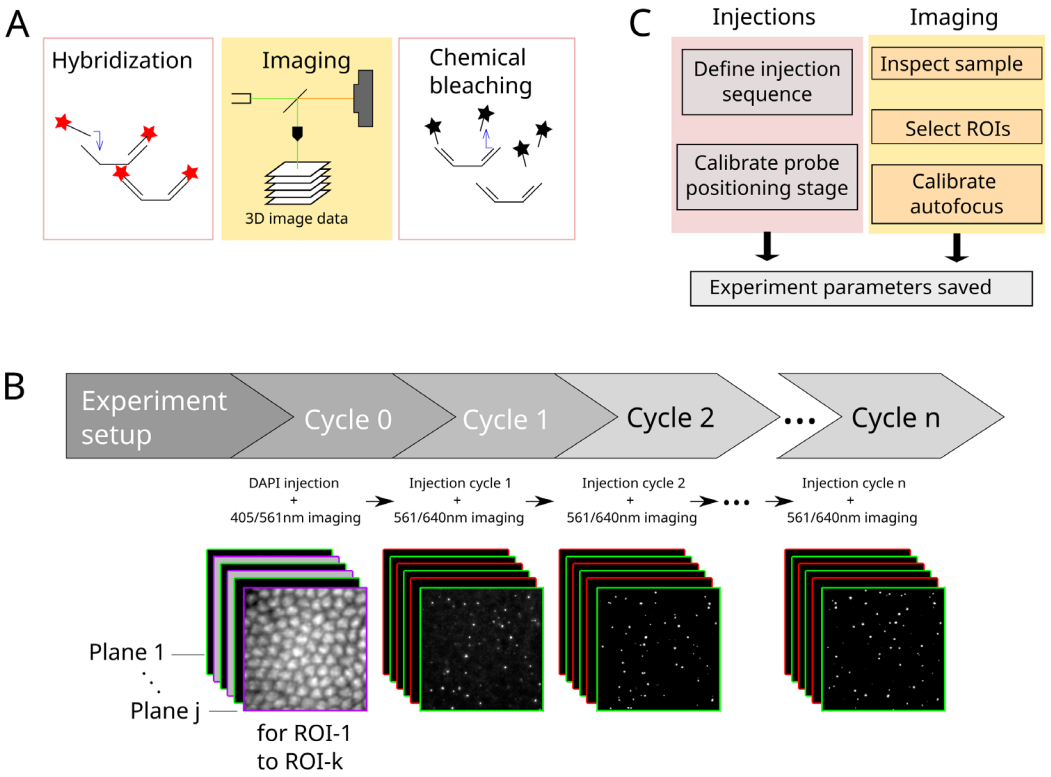


Figure 5. A. Each cycle in a Hi-M experiment consists of the hybridization with imaging probes, image data acquisition and subsequent chemical bleaching. **B**. Process flow of a typical Hi-M experiment. Multiple-color 3D images are acquired in each cycle, for multiple ROIs. From cycle 1 on, the experiment runs autonomously. **C**. Overview of steps performed during experiment preparation for subsequent automatized fluidics handling and imaging.

of the objective axial position while monitoring the position of the 785 nm laser beam thanks to a quadrant photodiode. If the calibration is valid (monotonous displacement with a precision <30 nm), the user defines the reference position of the 3D stack by adjusting the position of the objective lens in relation to an axial piezo stage. A starting position 1–2 µm below the sample focal plane is usually chosen, to prevent the acquisition of incomplete 3D data due to variability in the axial position of the biological sample.

Finally, a multi-color stack of images is acquired for each ROI using a Hi-M sub-task developed for this purpose. For this step, the user sets the imaging conditions using the experiment configurator (laser lines, laser intensity, stack size, etc.) and indicates where to find the ROI YAML file. For a standard experiment, 20–40 ROIs are selected, and for each of them a stack of 60–70 planes, each separated by 250 nm, are acquired simultaneously in two or three colors with 50 ms exposure. For each ROI, the program launches a 3D-stack acquisition by synchronizing the camera exposure with the lasers and the piezo displacements. Following each acquisition, the images are saved and the program moves to the next ROIs, in the order indicated by the user.

Following this initial cycle, the injection sequences for the Hi-M probes are defined (Figure 5C). First, buffers common to all cycles are connected to a specific valve outlet and its position is entered in the injections configurator GUI (Figure 3B). Second, cycle-specific Hi-M probes are arranged on the delivery tray of the pipetting robot by the user and their names and positions are entered into the GUI, according to their injection order. Third, the injection sequences for the hybridization and the chemical bleaching procedure are indicated: for each single step, the user chooses which solution to inject, the target volume and the expected flow rate. After each injection, an incubation time can be further added if required. Finally, the complete injection procedure is saved as a YAML file and can hence be reused for similar experiments.

The final configuration step involves setting the acquisition parameters using the experiment configurator (Figure 5B, cycles 1 to n). While the experiment is running, concurrent actions that could be triggered via the user interfaces are disabled. Both an emergency stop as well as a smooth transition to stopping or pausing the experiment after the current cycle are supported. The current status of the experiment can be tracked in real-time using the log displayed by the Qudi manager module. Relevant information (*e.g.* flow rate, ROI position, etc.) are also displayed on the GUI elements that remain accessible during an experiment.

Additionally, the experiment can be followed remotely using the status tracker application that displays the most important information of the experiment in real time as well as real-time image analysis (see section Automatized experiments above).

Conclusions

Qudi-HiM provides a range of functionalities from essential toolkits for live imaging up to fully automated experiments with a major focus on multiplexed imaging. For the latter, optical imaging is coordinated with microfluidic injections. Hence, all the necessary tools to control microfluidics and home-build microscopes were implemented, providing user-friendly GUIs to prepare and perform complex experiments. Moreover, real-time, remote tracking of the ongoing acquisition, including interactive data visualization, and real-time data analysis are enabled using a file exchange-based interface between Qudi-HiM tasks and a status tracker application.

Other Python-based solutions for general control of optical microscopes exist (Python Microscopy, Oxford Microscope-Cockpit[27], Tormenta[28], Storm-Control, MicroMator[29]). Some of these software packages provide an out-of-the-box solution for standard acquisition modes (*e.g.* single-molecule localization microscopy), while Qudi-HiM requires more programming skills to set up the desired functionalities and establish the necessary configuration files. However, this apparent limitation provides the required flexibility to adapt to complex, non-standard imaging modalities that require the design of complex acquisition pipelines and the control of conventional microscopy hardware components (cameras, lasers) as well as of other less standard devices (*e.g.* for fluid control). In this paper, we provided an example of the capabilities of Qudi-HiM to implement a full sequential-imaging acquisition modality for multiplexed DNA imaging.

Qudi-HiM was initially developed on a home-made Hi-M microscope made from independent hardware components, and later deployed in a minimal amount of time to other setups, keeping the same experimental logic and user interfaces. Therefore, hardware variability remains transparent to the end-users, facilitating both training and transfer of competences. Qudi-HiM is now used on a daily basis on two Hi-M setups and a standard single-molecule localization microscope at the Center for Structural Biology (Montpellier). It is worth noting that the initial architecture of Qudi offers high flexibility and helped us design Qudi-HiM as a versatile tool for multiplexed imaging. In particular, tasks initially developed for Hi-M imaging can easily be adapted according to user requirements, and the design of new tasks is greatly facilitated, simplifying Qudi-HiM evolution alongside research projects.

Data availability

Underlying data

Zenodo: Minimal dataset to test multiplexed DNA imaging (Hi-M) software pipelines [dataset] https://doi.org/10.5281/zenodo.6351755[33]

- scan_001_RT27_001_ROI_converted_decon_ch00.tif (Fiducial image for RT27 for ROI 001 using 561-nm laser line)
- scan_001_RT27_001_ROI_converted_decon_ch01.tif (RT27 image for ROI 001 using 641-nm laser line)
- scan_001_RT29_001_ROI_converted_decon_ch00.tif (Fiducial image for RT29 for ROI 001 using 561-nm laser line)

- scan_001_RT29_001_ROI_converted_decon_ch01.tif (RT29 image for ROI 001 using 641-nm laser line)
- scan_001_RT37_001_ROI_converted_decon_ch00.tif (Fiducial image for RT37 for ROI 001 using 561-nm laser line)
- scan_001_RT37_001_ROI_converted_decon_ch01.tif (RT37 image for ROI 001 using 641-nm laser line)
- scan_006_DAPI_001_ROI_converted_decon_ch00.tif (Fiducial image for DAPI for ROI 001 using 561-nm laser line)
- scan_006_DAPI_001_ROI_converted_decon_ch01.tif (DAPI image for ROI 001 using 405-nm laser line)
- scan_006_DAPI_001_ROI_converted_decon_ch02.tif (RNA image for ROI 001 using 488-nm laser line)

Data are available under the terms of the Creative Commons Attribution 4.0 International license (CC-BY 4.0).

Software availability

Source code available from: https://github.com/NollmannLab/qudi-HiM

Archived source code at time of publication: https://doi.org/10.5281/zenodo.6379944[30]

License: Creative Commons Attribution 4.0 International license (CC-BY 4.0)

Qudi-HiM has been added to the RRID database (record ID: SCR_022114).

Ethics and consent

Ethical approval and consent were not required.

References

1. Stelzer EHK, Strobl F, Chang BJ, *et al.*: **Light sheet fluorescence microscopy.** *Nat Rev Methods Primers.* 2021; **1**: 73.
Publisher Full Text
2. Sahl SJ, Hell SW, Jakobs S: **Fluorescence nanoscopy in cell biology.** *Nat Rev Mol Cell Biol.* 2017; **18**(11): 685–701.
PubMed Abstract | Publisher Full Text
3. Sigal YM, Zhou R, Zhuang X: **Visualizing and discovering cellular structures with super-resolution microscopy.** *Science.* 2018; **361**(6405): 880–887.
PubMed Abstract | Publisher Full Text | Free Full Text
4. Schermelleh L, Ferrand A, Huser T, *et al.*: **Super-resolution microscopy demystified.** *Nat Cell Biol.* 2019; **21**(1): 72–84.
PubMed Abstract | Publisher Full Text
5. Cardozo Gizzi AM, Espinola SM, Gurgo J, *et al.*: **Direct and simultaneous observation of transcription and chromosome architecture in single cells with Hi-M.** *Nat Protoc.* 2020; **15**(3): 840–876.
PubMed Abstract | Publisher Full Text
6. Chen KH, Boettiger AN, Moffitt JR, *et al.*: **RNA imaging. Spatially resolved, highly multiplexed RNA profiling in single cells.** *Science.* 2015; **348**(6233): aaa6090.
PubMed Abstract | Publisher Full Text | Free Full Text
7. Lubeck E, Coskun AF, Zhiyentayev T, *et al.*: **Single-cell *in situ* RNA profiling by sequential hybridization.** *Nat Methods.* 2014; **11**(4): 360–361.
PubMed Abstract | Publisher Full Text | Free Full Text
8. Wang S, Su JH, Beliveau BJ, *et al.*: **Spatial organization of chromatin domains and compartments in single chromosomes.** *Science.* 2016; **353**(6299): 598–602.
PubMed Abstract | Publisher Full Text | Free Full Text
9. Bintu B, Mateo LJ, Su JH, *et al.*: **Super-resolution chromatin tracing reveals domains and cooperative interactions in single cells.** *Science.* 2018; **362**(6413): eaau1783.
PubMed Abstract | Publisher Full Text | Free Full Text
10. Cardozo Gizzi AM, Cattoni DI, Fiche JB, *et al.*: **Microscopy-Based Chromosome Conformation Capture Enables Simultaneous Visualization of Genome Organization and Transcription in Intact Organisms.** *Mol Cell.* 2019; **74**(1): 212–222.e5.
PubMed Abstract | Publisher Full Text
11. Nir G, Farabella I, Pérez Estrada C, *et al.*: **Walking along chromosomes with super-resolution imaging, contact maps, and integrative modeling.** *PLoS Genet.* 2018; **14**(12): e1007872.
PubMed Abstract | Publisher Full Text | Free Full Text
12. Mateo LJ, Murphy SE, Hafner A, *et al.*: **Visualizing DNA folding and RNA in embryos at single-cell resolution.** *Nature.* 2019; **568**(7750): 49–54.
PubMed Abstract | Publisher Full Text | Free Full Text
13. Gut G, Herrmann MD, Pelkmans L: **Multiplexed protein maps link subcellular organization to cellular states.** *Science.* 2018; **361**(6401): eaar7042.
PubMed Abstract | Publisher Full Text
14. Su JH, Zheng P, Kinrot SS, *et al.*: **Genome-Scale Imaging of the 3D Organization and Transcriptional Activity of Chromatin.** *Cell.* 2020; **182**(6): 1641–1659.e26.
PubMed Abstract | Publisher Full Text | Free Full Text
15. Takei Y, Yun J, Zheng S, *et al.*: **Integrated spatial genomics reveals global architecture of single nuclei.** *Nature.* 2021; **590**(7845): 344–350.
PubMed Abstract | Publisher Full Text | Free Full Text
16. Rowley MJ, Corces VG: **Organizational principles of 3D genome architecture.** *Nat Rev Genet.* 2018; **19**(12): 789–800.
PubMed Abstract | Publisher Full Text | Free Full Text
17. Nollmann M, Bennabi I, Götz M, *et al.*: **The Impact of Space and Time on the Functional Output of the Genome.** *Cold Spring Harb Perspect Biol.* 2022; 14(**4**) a040378.
PubMed Abstract | Publisher Full Text | Free Full Text
18. Espinola SM, Götz M, Bellec M, *et al.*: ***Cis*-regulatory chromatin loops arise before TADs and gene activation, and are independent of cell fate during early *Drosophila* development.** *Nat Genet.* 2021; **53**(4): 477–486.
PubMed Abstract | Publisher Full Text
19. Wu YL, Tschanz A, Krupnik L, *et al.*: **Quantitative Data Analysis in Single-Molecule Localization Microscopy.** *Trends Cell Biol.* 2020; **30**(11): 837–851.
PubMed Abstract | Publisher Full Text
20. Harris CR, Millman KJ, van der Walt SJ, *et al.*: **Array programming with NumPy.** *Nature.* 2020; **585**(7825): 357–362.
PubMed Abstract | Publisher Full Text | Free Full Text
21. Berg S, Kutra D, Kroeger T, *et al.*: **ilastik: interactive machine learning for (bio)image analysis.** *Nat Methods.* 2019; **16**: 1226–1232.
Publisher Full Text
22. Schmidt U, Weigert M, Broaddus C, *et al.*: **Medical Image Computing and Computer Assisted Intervention – MICCAI.** In: (Springer International Publishing, 2018). 2018; 265–273.
Publisher Full Text
23. van der Walt S, Schönberger JL, Nunez-Iglesias J, *et al.*: **scikit-image: image processing in Python.** *PeerJ.* 2014; **2**: e453.
PubMed Abstract | Publisher Full Text | Free Full Text
24. The Astropy Collaboration, Price-Whelan AM, Sipőcz BM, *et al.*: **The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package.** *Astron J.* 2018; **156**(3): 123.
Publisher Full Text
25. The Astropy Collaboration, Robitaille TP, Tollerud EJ, *et al.*: **Astropy: A community Python package for astronomy.** *Astron Astrophys.* 2013; **558**: A33.
Publisher Full Text
26. Binder JM, Stark A, Tomek N, *et al.*: **Qudi: A modular python suite for experiment control and data processing.** *SoftwareX.* 2017; **6**: 85–90.
Publisher Full Text
27. Phillips MA, Pinto DMS, Hall N, *et al.*: **Microscope-Cockpit: Python-based bespoke microscopy for bio-medical science.** *bioRxiv.* 2021;

2021.01.18.427178.
Publisher Full Text

28. Barabas FM, Masullo LA, Stefani FD: **Note: Tormenta: An open source Python-powered control software for camera based optical microscopy.** *Rev Sci Instrum.* 2016; **87**(12): 126103.
PubMed Abstract | **Publisher Full Text**

29. Fox ZR, Fletcher S, Fraisse A, *et al.*: **MicroMator: Open and Flexible Software for Reactive Microscopy.** *bioRxiv.* 2021; 2021.03.12.435206.
Publisher Full Text

30. Barho F, Fiche JB, Nollmann M: **qudi-HiM (0.1.0).** *Zenodo.* 2022.
http://www.doi.org/10.5281/zenodo.6379944

31. Fiche JB, Cattoni DI, Diekmann N, *et al.*: **Recruitment, assembly, and molecular architecture of the SpoIIIE DNA pump revealed by superresolution microscopy.** *PLoS Biol.* 2013; **11**(5): e1001557.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

32. Montero Llopis P, Senft RA, Ross-Elliott TJ, *et al.*: **Best practices and tools for reporting reproducible fluorescence microscopy methods.** *Nat Methods.* 2021; **18**(12): 1463–1476.
PubMed Abstract | **Publisher Full Text**

33. Nollmann M, Espinola S: **Minimal dataset to test multiplexed DNA imaging (Hi-M) software pipelines (Version 1) [Data set].** *Zenodo.* 2022.
http://www.doi.org/10.5281/zenodo.6351755

Open Peer Review

Current Peer Review Status: ✔ ✔

Version 2

Reviewer Report 17 August 2022

https://doi.org/10.21956/openreseurope.16171.r29737

© **2022 Henriques R.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Ricardo Henriques**
Instituto Gulbenkian de Ciência, Oeiras, Portugal

I'm happy with the revisions and for the paper to be indexed.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 25 July 2022

https://doi.org/10.21956/openreseurope.16171.r29736

© **2022 Laine R.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Romain Laine**
MicrographiaBio, London, UK

I have now reviewed the changes made by the authors in response to my suggestions and I am pleased that everything was addressed appropriately.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 16 May 2022

https://doi.org/10.21956/openreseurope.15811.r29003

© **2022 Laine R.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Romain Laine
[1] MicrographiaBio, London, UK
[2] MicrographiaBio, London, UK

Barho et al. provide a great overview of their Qudi-HiM tool, designed to improve the integration of microscopy hardware components for automated imaging, especially that required by their HiM approach. It is a very timely and well organised piece of work describing the important modules and steps of a use-case large-scale acquisition pipeline.
The tool was built with the end-user in mind and robustness which is again very important for the community increasingly shifting towards Python-based acquisition platforms, and away from more traditional ones like MicroManager and MATLAB. It clearly uses state-of-the-art computer science tools with web interfaces, efficient Qt GUI and handles to advanced image analysis pipelines. I think that the microscopy development community is interested in understanding this extension of the author's Nature Protocol paper (Cardozo Gizzi et al., 2020).

I only have one main suggestion to the authors about this work and it's about placing the work in context. In my view, developing acquisition tools in this day and age needs to be subject to explicit standardization and reporting methods, essential for reproducibility and transparency as highlighted in the Nature Methods Focus 2021 (https://www.nature.com/collections/djiciihhjh).So, although the authors describe the tool capabilities at high level and provide the code through their Github page, I would enjoy seeing a better description of the ways in which efforts were put into standardization and reporting. For instance, how easy is it to swap one piece of hardware for another? What are the necessary steps and pre-requisite? Especially in the light of the abstract hardware device class definition.

Regarding reporting, what is reported and how can a user handle what is reported to help them understand their experiment (or failure thereof)?

And finally, discussing the advantages and disadvantages of Qudi-HiM in light of the other Python based platforms (as cited in introduction) would be very useful.

As a minor point, it would be useful to have a figure dedicated to the experiment configurator module, showing how the experimental pipeline is set up, highlighting the versatility of the acquisitions made possible here.

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: I am a researcher working on developing new microscopy methods

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 07 Jul 2022

Marcelo NOLLMANN

We provide below a point-by-point answer to the reviewer comments.

Comment 1. Barho et al. provide a great overview of their Qudi-HiM tool, designed to improve the integration of microscopy hardware components for automated imaging, especially that required by their HiM approach. It is a very timely and well organised piece of work describing the important modules and steps of a use-case large-scale acquisition pipeline. The tool was built with the end-user in mind and robustness which is again very important for the community increasingly shifting towards Python-based acquisition platforms, and away from more traditional ones like MicroManager and MATLAB. It clearly uses state-of-the-art computer science tools with web interfaces, efficient Qt GUI and handles to advanced image analysis pipelines. I think that the microscopy development community is interested in understanding this extension of the author's Nature Protocol paper (Cardozo Gizzi et al., 2020).

Response: We thank the reviewer for his comments and suggestions to improve our work. Our point by point answers are provided below.

Comment 2. I only have one main suggestion to the authors about this work and it's about placing the work in context. In my view, developing acquisition tools in this day and age needs to be subject to explicit standardization and reporting methods, essential for reproducibility and transparency as highlighted in the Nature Methods Focus 2021 (https://www.nature.com/collections/djiciihhjh).So, although the authors describe the tool

capabilities at high level and provide the code through their Github page, I would enjoy seeing a better description of the ways in which efforts were put into standardization and reporting. For instance, how easy is it to swap one piece of hardware for another? What are the necessary steps and prerequisite? Especially in the light of the abstract hardware device class definition.

Response: To address this important remark, we now include a new section in the revised text providing detailed instructions on how hardware exchange is handled in Qudi-HiM:

“Currently supported hardware and extensions: An overview of all devices that are available in the current version of Qudi-HiM is shown in Table 2. An online resource with the most up to date supported hardware is available here. Swapping devices between experimental setups is straight-forward, as long as the device has been interfaced in Qudi-HiM’s hardware modules. This simply requires updating the setup configuration file to interface the new piece of hardware. Example entries are provided in the hardware modules documentation (see examples of configuration files at: https://github.com/NollmannLab/qudi-HiM/tree/master/config/custom_config. Extending the available hardware is achieved by setting up a new hardware module and implementing an abstract interface using existing Python APIs or C libraries. Finally, the configuration file is updated to interface the new hardware component.“

Comment 3. Regarding reporting, what is reported and how can a user handle what is reported to help them understand their experiment (or failure thereof)?

Response; We thank the reviewer for pointing this out. Qudi-HiM reports the outcome of experiments in several ways. First, Qudi-HiM continuously reports information concerning its status and failures while the pipeline is executed. This log is displayed on the Qudi manager GUI and is very useful to monitor the experiment progress, as well as to debug the code (when developments are made). Second, Qudi-HiM exports the metadata of any ongoing experiment (e.g. sample name, exposure time, laser lines, intensity, autofocus parameters, sample XY position, etc) in a human readable format (.txt or .yml) along with or as header of the associated image file. Third, Qudi-HiM real-time operation can be monitored by connecting to a Bokeh server providing key experimental parameters (e.g. cycle number, flow rate) and real-time analysis of the acquired imaging data. Overall, these three reporting solutions are great tools for the user to monitor, diagnose, and debug any experiment.

We have made several changes to the text to address this issue:

a. The following information was modified in the text to clarify how the log is working: “The current status of the experiment can be tracked in real-time using the log displayed by the Qudi manager module. Relevant information (e.g. flow rate, ROI position, etc.) are also displayed on the GUI elements that remain accessible during an experiment.”

b. We extended the description of metadata exported by Qudi-HiM, and explained how the software can be adapted to include additional metadata: “[...] the storage of imaging data using a filename format encoding the cycle number and ROI, and of metadata containing

the imaging and fluidics parameters. For Hi-M experiments, metadata saved with each image include sample name, imaging information such as exposure time, laser lines and intensity, autofocus parameters, sample XY position, scan step size and total Z range covered. To comply with a standardized metadata format [29] depending on the type of experiment that is performed, it is possible to add custom information by adapting the underlying software module containing the instructions of the experiment."

c. We added a new figure (revised Figure 4) where the information logged by Qudi-HiM are explicitly mentioned.

Comment 4. And finally, discussing the advantages and disadvantages of Qudi-HiM in light of the other Python based platforms (as cited in introduction) would be very useful.

Response: To address this remark, we added a dedicated subsection to the "Impact and conclusions" section where we discuss the advantages and disadvantages of Qudi-HiM compared to other Python-based microscope platforms:

"Other Python-based solutions for general control of optical microscopes exist (Python Microscopy, Oxford Microscope-Cockpit [25], Tormenta [26], Storm-Control, MicroMator [27]). Some of these software packages provide an out-of-the-box solution for standard acquisition modes (e.g. single-molecule localization microscopy), while Qudi-HiM requires more programming skills to set up the desired functionalities and establish the necessary configuration files. However, this apparent limitation provides the required flexibility to adapt to complex, non-standard imaging modalities that require the design of complex acquisition pipelines and the control of conventional microscopy hardware components (cameras, lasers) as well as of other less standard devices (e.g. for fluid control). In this paper, we provided an example of the capabilities of Qudi-HiM to implement a full sequential-imaging acquisition modality for multiplexed DNA imaging."

Comment 5. As a minor point, it would be useful to have a figure dedicated to the experiment configurator module, showing how the experimental pipeline is set up, highlighting the versatility of the acquisitions made possible here.

Response: We have added a figure explaining in more detail the steps to set up the experiment using the experiment configurator module and the subsequent interactions between the Qudi task runner and the experiment scripts (Figure 4 in the revised manuscript). We modified and extended the text of the Automatized experiments section as follows:

[...] Each task requires a set of user parameters that needs to be easily accessible and adapted according to the sample and the experimental conditions. For this reason, an experiment configurator module for the setup of experiments is included in Qudi-HiM. Depending on the selected task, the configurator GUI displays a form that provides access to all required parameters and verifies their validity before saving the configuration (Figure 4). The configuration is stored as a YAML file and is subsequently loaded by the Python script that contains all information for the task execution. The script itself is controlled via the Task runner GUI: Here, the user selects which task to run and has the ability to run,

stop, pause or resume the task. The task script connects to all required experimental logic modules to combine the necessary high level hardware actions. Since multiplexed acquisitions can last for several days, it is convenient for the users to track the current status of their experiment. Besides visualization on the Qudi-HiM GUIs and a textual information status on the log for local observation, we also provided a file exchange-based interface from Qudi-HiM software to a status-tracker application using Bokeh (Bokeh documentation) to display key parameters of the experiment (e.g. cycle number, fluidics status, etc.) and a real-time pre-analysis of the acquired image data."

Competing Interests: No competing interests were disclosed.

Reviewer Report 03 May 2022

https://doi.org/10.21956/openreseurope.15811.r29002

© 2022 Henriques R et al. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Ricardo Henriques
[1] Instituto Gulbenkian de Ciência, Oeiras, Portugal
[2] Instituto Gulbenkian de Ciência, Oeiras, Portugal
Simao Coelho
[1] Instituto Gulbenkian de Ciência, Oeiras, Portugal
[2] Instituto Gulbenkian de Ciência, Oeiras, Portugal

Barho et al. present an open-source hardware control and software package, written in Python. The primary function is to perform basic image acquisition, some hardware control, and automation, principally for fluid handling. In practice, Qudi-HiM represents a software upgrade to an already existing package, the Hi-M.

Previously, comparable software has been made available via Gizzi et al. 2020 amongst others. Via GitHub links (in Gizzi et al. 2020), the control software (LabVIEW) and analysis software (Matlab) have already been readily available to the public. Whilst the authors are correct that LabVIEW and Matlab have costs associated with them, many institutes mitigate this by having site licences, which allow for routine access. Although LabVIEW remains the preferred software for instrumentation control, Python developers have made significant advances.

With the advent of Python, it is reasonable to include this upgrade and stands to benefit less fortunate research groups without site licences. The Python code, provided by the Nollmann lab GitHub page, has already become popular. This is a good indication that there is enthusiasm for this package.

Although the software provides the simulation of dummy devices, it does not, however, state what it is currently compatible with, nor which hardware was used here. Hardware integration is barely

mentioned. It is unclear which optomechanical components integrate the microscope used (lasers, filter wheels etc).

In the 'Use cases' the authors state: 'reproducible focusing of the sample for all the pre-defined ROIs (with ~50 nm accuracy)'. Does this not depend on the hardware in question? I suggest expanding this section to fully detail the hardware and describe current software limitations.

Altogether, the software presents a valid platform with great potential but is insufficiently described in the current format.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: My research entails the development of optical microscopy approaches, including hardware and software.

We confirm that we have read this submission and believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however we have significant reservations, as outlined above.

Author Response 07 Jul 2022

Marcelo NOLLMANN

We provide below a point by point response to reviewer comments.

Comment 1: Barho et al. present an open-source hardware control and software package, written in Python. The primary function is to perform basic image acquisition, some hardware control, and automation, principally for fluid handling. In practice, Qudi-HiM represents a software upgrade to an already existing package, the Hi-M.

Previously, comparable software has been made available via Gizzi et al. 2020 amongst others. Via GitHub links (in Gizzi et al. 2020), the control software (LabVIEW) and analysis software (Matlab) have already been readily available to the public. Whilst the authors are

correct that LabVIEW and Matlab have costs associated with them, many institutes mitigate this by having site licences, which allow for routine access. Although LabVIEW remains the preferred software for instrumentation control, Python developers have made significant advances.

With the advent of Python, it is reasonable to include this upgrade and stands to benefit less fortunate research groups without site licences. The Python code, provided by the Nollmann lab GitHub page, has already become popular. This is a good indication that there is enthusiasm for this package.

Response: We thank the reviewer for his comments and suggestions to improve our work. Our point by point answers are provided below.

Comment 2: Although the software provides the simulation of dummy devices, it does not, however, state what it is currently compatible with, nor which hardware was used here. Hardware integration is barely mentioned. It is unclear which optomechanical components integrate the microscope used (lasers, filter wheels etc).

Response: We thank the reviewer for this remark. In the documentation page for our original Github repository, we had included a section that described in detail the hardware components that are currently used for the installation of Qudi-HiM in four different microscopy setups: https://github.com/NollmannLab/qudi-HiM/blob/master/documentation/qudi-cbs%20documentation/qudi-HiM_hardware/overview_hardware_qudi-him.md. In addition, we added a new table (Table 2) with the instruments currently controllable by Qudi-HiM. These two resources, in addition to a new section describing the steps required to add new hardware components (see below), are now mentioned. The revised manuscript is as follows:

“Currently supported hardware and extensions An overview of all devices that are available in the current version of Qudi-HiM is shown in Table 2. An online resource with the most up to date supported hardware is available here. Swapping devices between experimental setups is straight-forward, as long as the device has been interfaced in Qudi-HiM’s hardware modules. This simply requires updating the setup configuration file to interface the new piece of hardware. Example entries are provided in the hardware modules documentation (see examples of configuration files here: https://github.com/NollmannLab/qudi-HiM/tree/master/config/custom_config). Extending the available hardware is achieved by setting up a new hardware module and implementing an abstract interface using existing Python APIs or C libraries. Finally, the configuration file is updated to interface the new hardware component.“

Comment 3. In the ‘Use cases’ the authors state: ‘reproducible focusing of the sample for all the pre-defined ROIs (with ~50 nm accuracy)’. Does this not depend on the hardware in question? I suggest expanding this section to fully detail the hardware and describe current software limitations.

Response: The reviewer’s comment is correct. This 50 nm precision is related to the optical design of our autofocus system as well as the instruments we are using. Qudi-HiM is only a

means to synchronize these devices to perform calibration and focus stabilization in real time. For clarity, we removed the mention of the autofocus precision from the text and added a reference to publications that already thoroughly describe the hardware components and performance of the autofocus system. The revised manuscript is as follows:

"[...] (2) the correct and reproducible focusing of the sample for all the pre-defined ROIs (see [5,10] for a detailed description of the autofocus hardware) [...]"

Comment 4. Altogether, the software presents a valid platform with great potential but is insufficiently described in the current format.

Response: We expect that the changes added to the revised manuscript would have provided enough detail to satisfy this comment of the reviewer.

Competing Interests: No competing interests were disclosed.