iSBatch: a batch-processing platform for data analysis and exploration of live-cell single-molecule microscopy images and other hierarchical datasets

Journal:	*Molecular BioSystems*
Manuscript ID:	MB-MET-05-2015-000321.R1
Article Type:	Method
Date Submitted by the Author:	10-Jul-2015

Complete List of Authors:	Armini Caldas, Victor Emanoel; University of Groningen , Zernike Institute for Advanced Materials Punter, Christiaan; University of Groningen, Zernike Institute for Advanced Materials Ghodke, Harshad; University of Groningen, Zernike Institute for Advanced Materials Robinson, Andrew; University of Groningen, Zernike Institute for Advanced Materials van Oijen, Antoine; University of Groningen, Zernike Institute for Advanced Materials

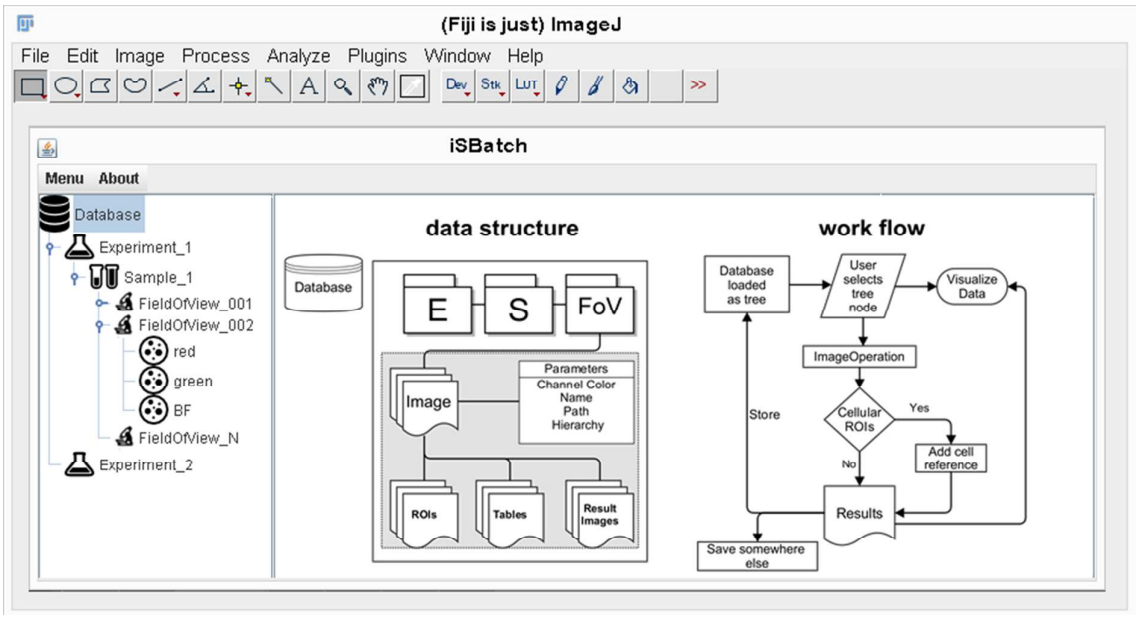
Table of contents accompanying

iSBatch: a batch-processing platform for data analysis and exploration of live-cell single-molecule microscopy images and other hierarchical datasets

Victor E. A. Caldas, Christiaan M. Punter, Harshad Ghodke, Andrew Robinson and Antoine M. van Oijen

---

iSBatch: an ImageJ plugin for `fast evaluation of analysis pipelines and visual exploration of datasets.`



iSBatch: a batch-processing platform for data analysis and exploration of live-cell single-molecule microscopy images and other hierarchical datasets

Victor E. A. Caldas[1], Christiaan M. Punter[1], Harshad Ghodke[1], Andrew Robinson[1] and Antoine M. van Oijen[1*]

[1]Zernike Institute for Advanced Materials, Centre for Synthetic Biology, University of Groningen, The Netherlands

*Corresponding author. Current address: School of Chemistry, University of Wollongong, Wollongong NSW 2522, Australia.
Phone: +61-(2)4221 4780. Fax: +61-(2)4221 4287. Email: vanoijen@uow.edu.au

Running title: Batch processing tool for single-molecule single-cell images

Key words: single-molecule microscopy, live-cell imaging, fluorescence, batch processing

Abstract

Recent technical advances have made it possible to visualize single molecules inside live cells. Microscopes with single-molecule sensitivity enable the imaging of low-abundance proteins, allowing for a quantitative characterization of molecular properties. Such data sets contain information on a wide spectrum of important molecular properties, with different aspects highlighted in different imaging strategies. The time-lapsed acquisition of images provides information on protein dynamics over long time scales, giving insight into expression dynamics and localization properties. Rapid burst imaging reveals properties of individual molecules in real-time, informing on their diffusion characteristics, binding dynamics and stoichiometries within complexes. This richness of information, however, adds significant complexity to analysis protocols. In general, large datasets of images must be collected and processed in order to produce statistically robust results and identify rare events. More importantly, as live-cell single-molecule measurements remain on the cutting edge of imaging, few protocols for analysis have been established and thus analysis strategies often need to be explored for each individual scenario. Existing analysis packages are geared towards either single-cell imaging data or *in vitro* single-molecule data and typically operate with highly specific algorithms developed for particular situations. Our tool, iSBatch, instead allows users to exploit the inherent flexibility of the popular open-source package ImageJ, providing a hierarchical framework in which existing plugins or custom macros may be executed over entire datasets or portions thereof. This strategy affords users freedom to explore new analysis protocols within large imaging datasets, while maintaining hierarchical relationships between experiments, samples, fields of view, cells, and individual molecules.

Introduction

Fluorescence microscopy has played an enormously important role in our understanding of biology. By tagging molecules of interest with fluorescent proteins, the dynamics of many cellular systems have been observed within live cells. However, many important cellular processes are carried out by proteins that are expressed at very low levels and are therefore undetectable using standard fluorescence microscopes[1,2]. Proteins that replicate and repair chromosomes in bacteria, for example, are often expressed at a level of less than 100 molecules per cell[3]. The recent development of fluorescence microscopes with single-molecule sensitivity is allowing us to peer into this world for the first time.

In addition to extending the sensitivity of established wide-field microscopy techniques, single-molecule microscopes allow rapid image sequences to be recorded that reveal the movements of individual molecules. Single-molecule microscopes can be used to record wide-field video-rate movies, with exposure times of 10–100 ms for individual images. On this timescale, fluorescent signals from molecules that diffuse freely within the cytosol of a bacterial cell or within the organelles of eukaryotic cells, blur out over the accessible volume in the cell or organelle due to rapid diffusion rates ($D \sim 1–10\ \mu m^2/s$ [4–6]). On the other hand, molecules that bind relatively static structures, such as chromosomal DNA, exhibit a much smaller diffusion constant and thus present as static foci of diffraction-limited size (~ 300 nm). Similarly, molecules that diffuse slowly, such as proteins associated with cell membranes, present discrete foci that move along the cell periphery. Movements of such single-molecule foci can be tracked in order to observe events that lead to a change in diffusion rate, for example, binding of molecules to DNA or other large structures. At the same time, intensities of foci in conjunction with photobleaching can be tracked in order to measure the number of fluorescent molecules giving rise to each focus, allowing the compositions of molecular complexes to be determined[7].

These extra layers of information provide fresh insight into the behaviors of molecules within cells, but also pose a problem for the scientists who study them: in order to obtain sufficient statistics to generalize observations, data must be recorded for many molecules, within many cells. Single-molecule imaging requires the use of high-magnification, high-numerical aperture objectives[6], limiting the size of the field-of-view and thus the number of cells that can be observed simultaneously. Typically, to discern statistically significant outcomes, hundreds of images must be recorded for a particular a live-cell single-molecule sample. That sample may contain hundreds of fields, potentially containing hundreds of time-points, up to thousands of cells of which each contain a handful of foci. Furthermore, it is often desirable to collect images in two or more fluorescence colors in order to correlate the behaviors of multiple types of molecules, as well as bright-field or phase-contrast images to define cell boundaries. These data are highly hierarchical in nature and efficient analysis is only possible when the hierarchical relationships between the different levels in the data are maintained during analysis.

A software package for single-molecule analysis in live-cells should meet four basic conditions. Firstly, it should allow for hierarchical classification of images and regions-of-interest (ROIs): samples contain fields of view (images), fields of view contain ROIs that capture individual cells (cell ROIs), and cells contain ROIs that define single-molecule foci (focus ROIs). Secondly, it should allow for analysis over both long and short time scales, resulting in the generation of different data structures: in time-lapse datasets there is one cell ROI per time point, whereas in rapid-imaging mode each cell ROI is typically used to analyze fluorescence signals over many time-points (**Fig. 1**). Thirdly, and most importantly, a package for live-cell single-molecule analysis should be highly flexible and allow for exploration of new analysis techniques. Finally, the source code used in the package be made available to users so that researchers can fully understand the algorithms they use[8].

Sophisticated packages for both cell analysis and single-molecule analysis are currently available, however none meet all of the requirements listed above[9]. Commercial packages typically offer out-

of-the-box solutions to a particular set of problems, often involve high licensing fees and utilize undisclosed source code, limiting the users' ability to adapt the software or to add their own customized code. CellProfiler[10] (and its extension CellProfiler Analyst[11]) is a free open-source package with a robust set of algorithms for analysis of 2D images. CellProfiler excels at automated assignment of cellular phenotypes, as well as identification of sub-cellular particles. However, with its focus on high-throughput screening data, the package provides little support for time-resolved studies. MicrobeTracker[12] allows users to conveniently assign outlines for microbial cells within time-lapse datasets and provides some support for characterization of foci. It is, however, not suitable for analysis of rapid-imaging data and is not geared towards exploration of new analysis methods. In addition, while MicrobeTracker itself is free, it runs within an environment that requires a paid licence (Matlab). Single-molecule packages such as the Mosaic Suite[13], as well as plugin collections, such as GDSC ImageJ Plugins[14] offer a myriad of analysis methods for single-molecule image processing, but are intended for *in vitro* analysis and thus lack the hierarchical classification systems that are required for the analysis of data derived from cellular systems. A significant advantage of these packages, however, is that they are extensions of the popular image-analysis platform ImageJ[15,16], which is extremely flexible, supported by a strong user community and a wealth of user-written extensions. Unfortunately, ImageJ is geared towards working with individual files, making hierarchical analysis strategies difficult to implement.

Flexible software that links analysis routines used in single-molecule imaging with those used in live-cell imaging is required for researchers to keep up with the rapid development of new imaging techniques. Ideally, one would be able to utilize ImageJ to develop code for new analysis routines, whilst being able to easily accommodate data structures that are large, hierarchical and multi-dimensional.

We present a free open-source ImageJ plugin, iSBatch, which allows users to use batch processing to treat files within hierarchical datasets in a straightforward manner. Routines built into ImageJ[15],

downloadable plugins and even user-written macros can be executed across any level of the dataset hierarchy. This strategy dramatically simplifies the often cumbersome tasks of scripting and data management, allowing users to run scripts over their entire datasets or portions thereof. Our tool complements existing single-cell and single-molecule analysis packages by allowing cell and focus ROIs generated in single-cell packages to be applied across hierarchical time-lapse and rapid-imaging datasets, with complete flexibility in choice of analysis methods.

Results and Discussion

iSBatch is straightforward to use, platform independent, and requires only ImageJ and Java Virtual Machine, which are freely available. iSBatch provides an interface to explore data in hierarchical datasets. Its graphical user interface (GUI) provides an intuitive means for controlling the operations and manipulating datasets of any size. iSBatch incorporates a powerful adapter for the ImageJ macro interpreter, allowing users to implement existing or newly written macros within the data hierarchy. Data is stored in an SQL database and displayed in a tree format for manipulation (**Fig. 2a**). The database format assists in managing the transfer and back-up of large imaging datasets, which may contain hundreds or even thousands of images and can be prone to errors when handled manually [17]. A file named 'iSBatch.zip', which contains the plugin, its source code and user manual, is included in the online Supplementary Material. To help to illustrate the concepts in the following sections of this report, we also include an example dataset containing three Experiments in the Supplementary Material.

Data Structure and Graphical User Interface (GUI)

The fundamental unit of iSBatch is the image itself. Each image belongs to a Field of View, representing the region of the sample that was imaged by the microscope. A collection of Fields of View is called a Sample, and a collection of Samples is called an Experiment. This hierarchy is assigned to each image by placing hierarchy parameters alongside the image within an image object. Image objects may contain an unlimited number of additional parameters. Within iSBatch, image

objects contain information on the nature of the image, for example identifiers for color channels, metadata generated during operations, such as peak tables and image projections, as well as ROIs that designate the positions of cells and foci. A dedicated dialog guides the import of imaging data and assures compatibility with iSBatch. There is no specific requirement for file name structure, however we suggest the inclusion of a useful identifier for the imaging channel (e.g. 514.tif, BF.tif, GFP.tif).

The general workflow within iSBatch is straightforward (**Fig. 2b**). In short, the user selects which subset needs to be processed, chooses the operation to be performed and indicates either to save results and images to disc or keep it in the database. The graphic user interface is divided into subpanels containing the navigation tree, file lists, buttons to run built-in functions or custom macros and a log panel (**Fig. 2c**). The GUI also has buttons to add images to the data structure, as well as cell ROIs generated in ImageJ or in MicrobeTracker[18].

We have included several operations commonly used in single-molecule analysis within iSBatch, such as functions to correct images for uneven illumination, find and fit peaks inside or outside of cells, and basic peak table operations. These operations will be explored in detail in the form of case studies in the sections below.

Case studies

To demonstrate the applicability of our iSBatch software we present here a case in which the *custom macro interpreter* was applied to a dataset, as well as two detailed case studies based on the most common types of data generated by single-molecule single-cell measurements: rapid-acquisition movies and time-lapse series. We imaged *Escherichia coli* cells in which two different subunits of the replisome were tagged with fluorescent proteins at their carboxy-termini; the ϵ subunit (DnaQ gene) is tagged with red mKate2 (DnaQ-mKate2) and the τ subunit (DnaX gene) is tagged with yellow YPet (DnaX-YPet). The *E. coli* replisomes contain ten different proteins, each at different copy numbers, including up to three molecules of τ (a component of the clamp loader complex) and three

molecules of ϵ (proof-reading exonuclease)[3]. Replisome proteins are of particular interest for single-molecule studies[3,19] both because of their biological role of importance (replisomes duplicate the genome prior to cell division)[20] and because the replisomal proteins are present at extremely low levels within cells. A single *E. coli* cell produces only about 100 molecules of τ per cell and ~250 molecules of ϵ[3].

The example data is comprised of a single database containing three experiments, labeled RA_DnaX-YPet, RA_DnaQ-mKate2 and TimeLapse). RA_DnaX-YPet and RA_DnaQ-mKate2 are Rapid Acquisition (RA) experiments (500 times 34 ms) that each contain three samples recorded at different excitation laser powers. Each of these samples contains 10 fields of view. TimeLapse contains just one sample and 10 fields of view (50 ms every 20 min, repeated for 400 minutes). RA_DnaX-YPet includes 134 cell selections, RA_DnaQ-mKate2 contains 107 and TimeLapse contains 10 fully tracked cells. iSBatch assumes that, if no cell ROIs are provided, the entire image is selected. This scenario is applicable to analyses that do not rely on cell outlines, such as reconstruction of super-resolution images by PALM [21,22] or STORM [23], or even to the analysis of *in vitro* single-molecule data.

When loaded into iSBatch, our datasets appear in the operation panel (**Fig 2c**). Selecting a node within one of the datasets allows image-processing operations to be executed across all images falling under that node. For example, when the user selects the node RA_DnaQ in the tree and the operation *flatten*, iSBatch guides the user through the steps required for image flattening and correction for the unevenness of the beam profile (more details found in the User Manual – Supplementary Materials) within selected images in the RA_DnaQ experiment. Next, iSBatch assumes that operations will be performed on the resulting flattened images as will be shown in the following sections.

Custom macro interpreter

The ImageJ support to macros is a powerful tool to execute a sequence of operations in an image. Traditionally, in order to apply basic ImageJ functions across portions of a dataset, the user has to

write sequences of steps and functions to navigate through the folders, to identity the required files, and to save the results. Even small changes in the folder or file structure prevent the code from running properly and troubleshooting becomes a daunting task. iSBatch, via its *custom macro interpreter,* provides the necessary tools to automatize these steps (**Fig. 3**).

Within our rapid acquisition data, for instance, stacks exported from the microscope contain dark frames at the beginning of the image series, resulting from a small delay before the opening of the laser shutter. The *custom macro interpreter* can be easily used to trim stacks in order to remove these frames. There are two possibilities of implementation: an experienced user may just write a macro to trim one image and them use it within the *custom macro interpreter*; or could take advantage of ImageJ *Macro Recorder* – a panel that stores all commands performed by the user while processing an image– and then simply paste the sequence of steps into the iSBatch *custom macro interpreter.* The user then can analyse the images further in a statistical package, like *R*[24].

Rapid-Acquisition Analysis

Rapid-Acquisition experiments usually result in a stack of fluorescence images, containing hundreds or thousands of individual frames, acquired at rapid frame rates (typically continuous series of frames, 10-100 ms duration each, with a total duration of seconds), as well as a bright-field image enabling the identification of cell boundaries in cases of low fluorescence signals. This type of imaging allows the behaviors of individual molecules to be monitored in real time. It is typically used to count molecules within foci, to count the total number of molecules in cells, to measure diffusive behavior and to observe binding kinetics[1,3,7].

In our datasets, DnaX-YPet and DnaQ-mKate2 frequently are associated with DNA-bound replisomes, and as a result form immobile foci on the imaging timescale (34 ms). We used iSBatch to detect foci and measure their integrated intensities using the *peak fitter* operation in a selected node (**Fig. 4a**). The built-in *peak fitter* fits each peak to a Gaussian profile using least-squares fitting. It takes into account sources of noise, such as general background noise, and uses a non-symmetric 2D Gaussian,

so peaks can be later filtered based on their symmetry[25]. The properties of foci in single-molecule single-cell measurements can vary between experiments, depending on the brightness of the fluorophore and the amount of background fluorescence arising from cellular auto fluorescence. It is therefore desirable to be able to explore parameters such as peak-detection thresholds for individual samples. iSBatch automatically stores peak tables generated from the *peak fitter* module, appending the results with the values of key parameters used. In this way, the user can explore different parameters and plot the resulting peaks lists in an external plotting or statistical analysis package, for example *GNU Octave*[26] or *R*[24]. In our example data, we see that for both fluorescent species the intensities of peaks increase with higher excitation power, as expected (**Fig. 4b**).

Foci containing fewer than about 10 molecules show step-wise photobleaching behavior that can be used to quantify the number of fluorescent molecules within each focus [5,27]. Using iSBatch, trajectories of intensity *versus* time can be generated for foci using the *traces* module. This can be done in two different ways. One option is to produce an average projection of each image stack, assign focus ROIs in the projected image using *peak finder* and measure the integrated intensity under each ROI for each frame of the stack. The second option is to use *peak fitter* to measure foci throughout the entire stack of a 'Field of View' and then use *track* to identify foci falling within a small, user-defined search radius of a focus that appeared in the first frame and produce a time-ordered list of their intensities. As expected, traces for DnaQ-mKate2 show step-wise photobleaching behavior (**Fig. 5**). Intensity levels within traces can be automatically assigned using the *changepoint analysis* (**Fig. 5c, red lines**). This algorithm estimates the time point at which the statistical properties of a sequence change, e.g. photobleaching causing a discrete jump in intensity followed by a period of constant intensity[28,29].

As well as quantifying the number of molecules in each focus, the single-molecule intensity determined within the *change-point* module can be used to determine the total number of molecules in each cell. For this, it is necessary to have ROIs defining the cell boundaries. These can

be generated in ImageJ or imported from MicrobeTracker using the module *MicrobeTracker I/O*. In iSBatch, the total fluorescence signal originating from a cell as it photobleaches can be measured by applying the *cell intensity* operation to a batch (**Fig. 6**). Comparing the three DnaQ-mKate2 samples within the RA_DnaQ experiment (**Fig 6a**), we observe that DnaQ-mKate2 photobleaches faster at higher laser excitation intensities, as expected (**Fig. 6b**). Comparing the OD1 samples between the RA_DnaX and RA_DnaQ experiments (**Fig. 6c)**, we observed that YPet photobleaches faster than mKate2 (**Fig. 6d**), as expected [30,31]. Using the *cellular concentration* operation*,* the amplitudes of these decays (representing the total fluorescence of the cell) is divided by the intensity of a single molecule in order to obtain the number of molecules in that cell and the cellular concentration. For DnaX-YPet and DnaQ-mKate2 we measure 110 ± 35 and 95 ± 22 molecules per cell respectively. Based on the mean volume of cells as measured from bright field images (4.6 ± 0.9 fL), these values correspond to concentrations of approximately 23 and 20 nM for DnaX-YPet and DnaQ-mKate2 respectively.

Rapid-acquisition imaging can also be used to measure the movements of molecules. Single-particle tracking can be used to measure the diffusional motions of molecules. In iSBatch this operation is implemented in the *tracking* module. Here foci within the tables generated by *peak fitter* are assigned to trajectories if they fall within a set distance on consecutive frames and, optionally, are within the same cell ROI (**Fig. 7a**). These trajectories can be used to build step-size distributions or mean-square displacement plots that allow for measurement of properties such as diffusion coefficients. For DnaQ-mKate2, which present long-lived trackable foci, we observe two populations: one with low diffusion coefficients corresponding to molecules bound to DNA, and one with higher diffusion coefficients corresponding to freely-diffusing molecules[5] (**Fig. 7b**).

Time-Lapse Analysis

Time-lapse datasets consist of image stacks containing equal numbers of bright-field images and fluorescence images, with individual frames corresponding to measurements at periodically sampled

time-points. Time-lapse measurements can be used to monitor temporal changes in the expression level of a protein, the number of foci within cells, or the localization of proteins within cells. With the availability of automated microscopes, we can monitor hundreds cells in several fields of view over a period of minutes to days[32].

Our example dataset, TimeLapse, contains images of cells expressing DnaX-YPet and DnaQ-mKate2 recorded over 400 minutes. Using the module *cell intensity*, we measured the levels of each fluorescent protein for ten cells over time. The levels of DnaX and DnaQ remain relatively constant throughout the measurement (**Fig. 8a**). Using the number of foci detected by *peak finder* or *peak fitter*, we quantified the number of DnaX-YPet and DnaQ-mKate2 foci observed over time. As expected, cells periodically changed between zero, one, two and occasionally three foci (**Fig. 8b**). Because we imaged in time-lapse mode, movie sequences of individual cells could be synchronized to the beginning of the cell cycle. This analysis shows that after division, daughter cells contain one foci on average, then the increases to two foci later in the cell cycle (**Fig. 8c**). If cell ROIs have been imported from MicrobeTracker, it is possible to produce maps of focus locations within cells using the *location maps* module. MicrobeTracker ROIs consist of high-resolution meshes, allowing the relative positions of foci to be mapped to their relative cellular coordinates. For DnaX-YPet and DnaQ-mKate2 cells, one focus was present from 0 to 40 min after birth (**Fig. 8c**). This focus was located close to the mid-cell position (**Fig. 8d**). In contrast, 60 to 120 min after division, cells exhibited two foci (**Fig 8c**). These foci were more evenly distributed through the entire cell (**Figure 8d**).

Materials and Methods

Implementation

Software

iSBatch is a Java 1.6-based plugin for ImageJ[15] (version 1.49d) or its distribution Fiji[33]. iSBatch is designed for quick evaluation of analysis pipelines and visual exploration of datasets. It is distributed under an open open-source[34] license (GNU General Public License, version 3). iSBatch handles the data in a hierarchical fashion based on a source folder containing all data and little guidance provided by the user. Due to memory limitations when handling large datasets, iSBatch alleviates memory overload by loading only the minimum set of images required for a process. Garbage collection is done after each cycle so effective memory limitations are imposed by the amount of memory available in the system and not by the size of the database.

The software is designed for rapid exploration of large datasets and it includes an internal SQLite database (http://sqljet.com/) for convenience. All files related to the iSBatch platform, including source codes and API for developers can be accessed directly from the plugin website (https://github.com/SingleMolecule/iSBatch).

General workflow

In the following subsections, we describe the general workflow and how to use the plugin for accessing basic cellular information. iSBatch guides the user in the initial configuration steps to proper categorization of the input data.

Processing and Exploring Data - Custom functions

iSBatch couples its hierarchical data structure management to an extended version of ImageJ's macro interpreter. The user can record the executed operations, e.g. using ImageJ's built in macro recorder, and simply copy and paste the code in iSBatch interpreter. After selecting the desired parameters, the results are displayed, allowing the user to quickly check the results.

Built-in functions

Data preprocessing involves image operations as well. Image Flattening is available and follows the equation

$$FlatImage = \frac{RawImage - BackgroundImage - CameraDarkCount}{BackgroundImage - CameraDarkCount} * ImageRange$$

were ImageRange depends on the image type (8-, 16- or 32-bit), the CameraDarkCount can be provided either as a constant or an image; BackgroundImage, if not available, can be generated from all images acquired. Generating the Background image may lead to biased correction if saturated peaks or high intensity regions are found for long time in the movies. A Gaussian filter with a default value of four pixels is applied to reduce the influence of bright spots.

Ideally, the background should be an image taken in the same conditions of the experiment prior to have the sample in the Field Of View.

To allow for fast and accurate detection of peaks, we implemented the fluoroBancrof algorithm[35]. This algorithm localizes peaks with sub-diffraction limit accuracy without the need of numerical fitting[36]. All the results will be available in human-readable format like comma-separated-values (csv).

Acquiring peak tables from the images configures a starting point of a whole new section of analysis of single molecule data. Change point analysis is used to assign steps to single-molecule traces and infer stoichiometry of molecules. Cellular ROIS can be either added manually or imported from MicrobeTracker. In the later, a detailed subdivision of each cell with meshes is available. Therefore, is possible to localize every peak in relation to the mesh and assign relative positions. With the cellular parameters, such as cell length, width, area, can be obtained from the imported ROIs and an artificial cell is created for the peaks to be inserted.

Image Acquisition

Cell Culture

Derivatives of *E. coli* K12 MG1655 carrying a chromosomal C-terminal fusions[37] containing DnaX-YPet and DnaQ-mKate2 were grown overnight in M9 Minimal medium supplemented with Glycerol 2% and 10mM thiamine hydrochloride; Cell cultures were diluted to 1:100 and grown from 4 hours at 37°C at 1100 rpm prior to the start of the imaging experiment.

Image Acquisition

The images were taken on a home-built single-molecule fluorescence microscope consisting of a fully-automated inverted microscope body (Olympus IX-81) with excitation light provided by 514 nm and 568 nm Sapphire lasers (Coherent) and equipped with a 1.49 NA 100x objective and a 512 × 512 pixel EM-CCD camera (C9100-13, Hamamatsu). For imaging we used flow cells derivatized with 3-aminopropyl triethoxy silane (APTES, Sigma) and kept the flow at 10 µl/min.

The datasets are described as follows: 1) Rapid acquisition of DnaX-YPet and DnaQ-mKate2 each containing 10 Fields of View. A Field of View comprises a reference bright field image and a fluorescence movie (500 frames each with 34ms interval between acquisitions under different laser intensities); 2) Time Lapse acquisition of DnaX-YPet and DnaQ-mKate2 containing 10 fields of View containing a bright field and two fluorescent images of 50 ms for each fluorescent protein. The cycle time is 20 minutes and the experiment was carried out for 400 minutes. Datasets are available as supplementary materials S1 and S2;

Conclusion

We present here a fully open-source and community-driven ImageJ plugin for single-molecule analysis focused on hierarchical data obtained from live-cell single-molecule experiments. The plugin facilitates data exploration and bookkeeping of datasets with large number of images in multiple colour channels, including basic pipelines and support for custom macros. We present case studies that illustrate the ability to carry out analysis in a structured way, minimizing the burden of code development. With this in mind, we envision that the user will be able to place a larger focus on

exploration of biological phenomena and new analysis routines. The development of open-source analysis tools such as the ones presented here allows for a community-based sharing and development[38] of the platforms required to analyse experiments that increasingly grow in complexity and data richness. Software documentation is included within the Supplementary Material. The source code is available for download at https://github.com/SingleMolecule/iSBatch.

Acknowledgements

We would like to thank M. Cox and E.A. Wood for providing the *E. coli* strain used in this work. The authors would like to acknowledge funding from the Netherlands Organization for Scientific Research (NWO; Vici 680-47-607) and the European Research Council (ERC Starting 281098). The authors alone are responsible for the content and writing of the paper.

Conflict of interest

The authors declare that they have no conflicts of interest concerning this article.

References

1. Xia, T., Li, N. & Fang, X. Single-molecule fluorescence imaging in living cells. *Annu. Rev. Phys. Chem.* **64,** 459–80 (2013).

2. Rigler, R. Fluorescence and single molecule analysis in cell biology. *Biochem. Biophys. Res. Commun.* **396,** 170–175 (2010).

3. Reyes-Lamothe, R., Sherratt, D. J. & Leake, M. C. Stoichiometry and architecture of active DNA replication machinery in Escherichia coli. *Science* **328,** 498–501 (2010).

4. Mashanov, G. I. Single molecule dynamics in a virtual cell: a three-dimensional model that produces simulated fluorescence video-imaging data. *J. R. Soc. Interface* **11,** (2014).

5. Yu, J., Xiao, J., Ren, X., Lao, K. & Xie, X. S. Probing gene expression in live cells, one protein molecule at a time. *Science* **311,** 1600–1603 (2006).

6. Haas, B. L., Matson, J. S., DiRita, V. J. & Biteen, J. S. Imaging Live Cells at the Nanometer-Scale with Single-Molecule Microscopy: Obstacles and Achievements in Experiment Optimization for Microbiology. *Molecules* **19,** 12116–12149 (2014).

7. Lenn, T. & Leake, M. C. Experimental approaches for addressing fundamental biological questions in living, functioning cells with single molecule precision. *Open Biol.* **2,** 120090 (2012).

8. Cardona, A. & Tomancak, P. Current challenges in open-source bioimage informatics. *Nat. Methods* **9,** 661–5 (2012).

9. Eliceiri, K. W. *et al.* Biological imaging software tools. *Nat. Methods* **9,** 697–710 (2012).

10. Kamentsky, L. *et al.* Improved structure, function and compatibility for cellprofiler: Modular high-throughput image analysis software. *Bioinformatics* **27,** 1179–1180 (2011).

11. Jones, T. R. *et al.* CellProfiler Analyst: data exploration and analysis software for complex image-based screens. *BMC Bioinformatics* **9,** 482 (2008).

12. Sliusarenko, O., Heinritz, J., Emonet, T. & Jacobs-Wagner, C. High-throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics. *Mol. Microbiol.* **80,** 612–27 (2011).

13. Shivanandan, A., Radenovic, A. & Sbalzarini, I. F. MosaicIA: an ImageJ/Fiji plugin for spatial pattern and interaction analysis. *BMC Bioinformatics* **14,** 349 (2013).

14. Herbert, A. D., Carr, A. M. & Hoffmann, E. FindFoci: A Focus Detection Algorithm with Automated Parameter Training That Closely Matches Human Assignments, Reduces Human Inconsistencies and Increases Speed of Analysis. *PLoS One* **9,** e114749 (2014).

15. Abràmoff, M. D., Magalhães, P. J. & Ram, S. J. Image processing with ImageJ Part II. *Biophotonics Int.* **11,** 36–43 (2005).

16. Schneider, C. a, Rasband, W. S. & Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **9,** 671–675 (2012).

17. Wollman, R. & Stuurman, N. High throughput microscopy: from raw images to discoveries. *J. Cell Sci.* **120,** 3715–22 (2007).

18. Sliusarenko. Microbetracker software. *Mol. Microbiol.* **80,** 612–627 (2012).

19. Stratmann, S. a & van Oijen, a M. DNA replication at the single-molecule level. *Chem. Soc. Rev.* **43,** 1201–20 (2014).

20. Robinson, A. & van Oijen, A. M. Bacterial replication, transcription and translation: mechanistic insights from single-molecule biochemical studies. *Nat. Rev. Microbiol.* **11,** 303–15 (2013).

21. Betzig, E. *et al.* Imaging intracellular fluorescent proteins at nanometer resolution. *Science* **313,** 1642–5 (2006).

22. Henriques, R., Griffiths, C., Hesper Rego, E. & Mhlanga, M. M. PALM and STORM: unlocking live-cell super-resolution. *Biopolymers* **95,** 322–31 (2011).

23. Zhu, L., Zhang, W., Elnatan, D. & Huang, B. Faster STORM using compressed sensing. *Nat. Methods* **9,** 721–3 (2012).

24. Team R Core Development. R: a language and environment for statistical computing | GBIF.ORG. *R Foundation for Statistical Computing* (2013). at <http://www.r-project.org/>

25. Thompson, R. E., Larson, D. R. & Webb, W. W. Precise nanometer localization analysis for individual fluorescent probes. *Biophys. J.* **82,** 2775–83 (2002).

26. Hauberg, J. W. E. and D. B. and S. *GNU Octave version 3.0.1 manual: a high-level interactive language for numerical computations*. (CreateSpace Independent Publishing Platform, 2009). at <http://www.gnu.org/software/octave/doc/interpreter>

27. Leake, M. C. *et al.* Stoichiometry and turnover in single, functioning membrane protein complexes. *Nature* **443,** 355–8 (2006).

28. Eckley, R. K. and I. A. changepoint: An R Package for Changepoint Analysiso Title. *J. Stat. Softw.* **58,** (2014).

29. Watkins, L. P. & Yang, H. Detection of intensity change points in time-resolved single-molecule measurements. *J. Phys. Chem. B* **109,** 617–28 (2005).

30. Shcherbo, D. *et al.* Far-red fluorescent tags for protein imaging in living tissues. *Biochem. J.* **418,** 567–574 (2009).

31. Shaner, N. C., Steinbach, P. A. & Tsien, R. Y. A guide to choosing fluorescent proteins. *Nat. Methods* **2,** 905–909 (2005).

32. Muzzey, D. & van Oudenaarden, A. Quantitative time-lapse fluorescence microscopy in single cells. *Annu. Rev. Cell Dev. Biol.* **25,** 301–27 (2009).

33. Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9,** 676–82 (2012).

34. Prlić, A. & Procter, J. B. Ten Simple Rules for the Open Development of Scientific Software. *PLoS Comput. Biol.* **8,** 8–10 (2012).

35. Hedde, P. N., Fuchs, J., Oswald, F., Wiedenmann, J. & Nienhaus, G. U. Online image analysis software for photoactivation localization microscopy. *Nat. Methods* **6,** 689–690 (2009).

36. Andersson, S. B. Localization of a fluorescent source without numerical fitting. *Opt. Express* **16,** 18714–18724 (2008).

37. Datsenko, K. A. & Wanner, B. L. One-step inactivation of chromosomal genes in Escherichia coli K-12 using PCR products. *Proc. Natl. Acad. Sci. U. S. A.* **97,** 6640–5 (2000).

38. Carpenter, A. E., Kamentsky, L. & Eliceiri, K. W. A call for bioimaging software usability. *Nature Methods* **9,** 666–670 (2012).

Figure Captions

Figure 1 Schematic design of a single-cell, single-molecule experiment. Panel A – Structure of a time-lapse experiment. Each time point shows a bright-field (BF) image and its corresponding

fluorescence channel (in this example 568-nm excitation). The intervals are on the time scale of minutes. Panel B – Monitoring of cell fluorescence intensity and its relation to total observable protein concentration and protein number per cell throughout the experiment. Panel C – Structure of a rapid-acquisition experiment. A single bright-field image is taken prior to subsequent rapid image acquisition in the fluorescence channel (in this case 568-nm excitation). Panel D – Simulated data of binding dynamics of a molecule.

Figure 2 iSBatch Structure. Panel A – Schematic representation of data structure (Experiment – E, Sample – S, Field of View – FoV) and its connections. Panel B – Logic structure of the algorithm; Panel C – User interface including ImageJ main panel (upper part) and iSBatch interface with the main commands.

Figure 3 Custom Macro runner. iSBatch contains a custom macro runner that support syntax-highlighting for creating, running and editing existing ImageJ macros and plugin commands from the MacroRecorder.

Figure 4 Built-in Peak Fitting Operation. Panel A – Selected node highlighting the 'Experiment' level. Panel B – Distribution of detected peak intensities within different 'Samples' in the same selected 'Experiment' node for DnaQ-mKate2.

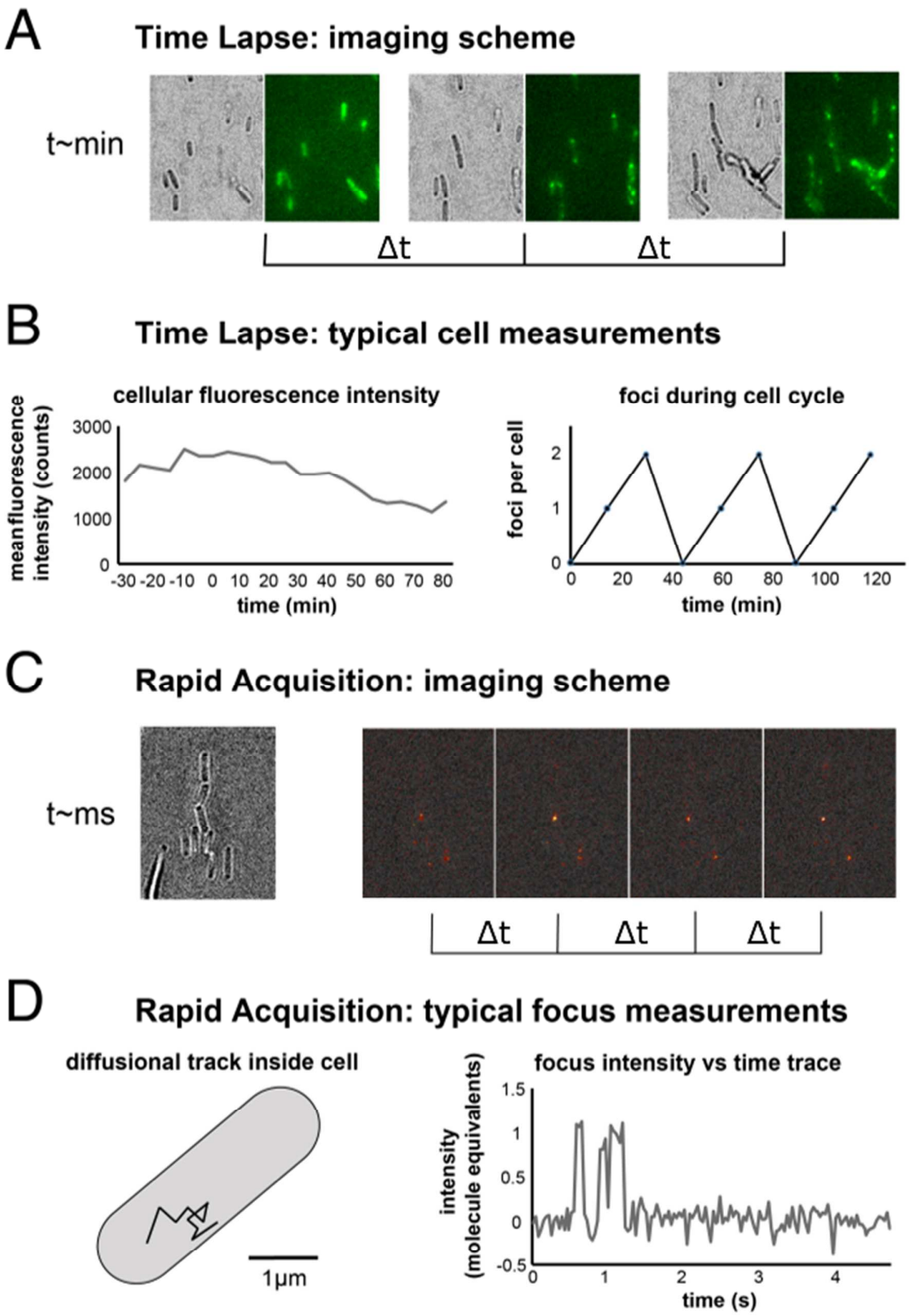
Figure 5 Step-wise photobleaching. Panel A – Selected node highlighting a 'Field of View' level. Panel B – Selected cell within a 'Field of View' with the boundaries assigned in yellow and a selection box in red. Panel C – Representative photobleaching trace of a detected focus. Red traces represent the detected steps by change-point analysis algorithm.

Figure 6 Cellular fluorescence obtained by Rapid Acquisition. Panel A – Selected node highlighting a 'Experiment' level Panel B – Cellular fluorescence photobleaching dependent on laser intensity for DnaQ-mKate2. Panel C – Selected node highlighting two 'Samples' selected within different experiments. Panel D – Comparison of photobleaching properties of YPet and mKate2 when excited with same laser intensity (180 W/cm^2).

Figure 7 Particle tracking within cells. Panel A – DnaQ-mKate2t particles tracked inside a live E. coli cell. Blue: Confined track indicating protein bound to DNA. Panel B – Analysis of all detected focus tracks within a 'Sample' level, e.g. DnaQ-mKate2 acquired at 180 W/cm^2. Left panel shows the peak lifetime distribution and right panel the calculated diffusion coefficient for the same population.

Figure 8 Built-in Time-Lapse analysis. Panel A – Fluorescence cell intensity over time for DnaX-YPet and DnaQ-mKate2. Panel B – Number of long-lived immobile peaks per cell, i.e. foci. Panel C – Data synchronization considering cell division times. Time zero is the first frame after cell division; cell division time is 100 – 120 min. Panel D – Location maps. A projection of detected peaks in an artificial, normalized cell. Left: projected cells with one detected focus, distributed towards the *centre* of the cell; Right: projected cells with two detected foci, distributed towards the ¼ and ¾ of the cell.

Figure 1.



500

501

Figure 3~~1~~ Schematic design of a single-cell, single-molecule experiment. Panel A – Structure of a time-lapse experiment. Each time point shows a bright-field (BF) image and its corresponding fluorescence channel (in this example 568-nm excitation). The intervals are on the time scale of minutes. Panel B – Exemplified cell fluorescence intensity and its relation to total observable protein concentration and protein number per cell throughout the experiment. Panel C – Structure of a rapid-acquisition experiment. A single bright-field image is taken prior to subsequent rapid image acquisition in the fluorescence channel (in this case 568-nm excitation). Panel D – Simulated data of binding dynamics of a molecule.

Figure 2.

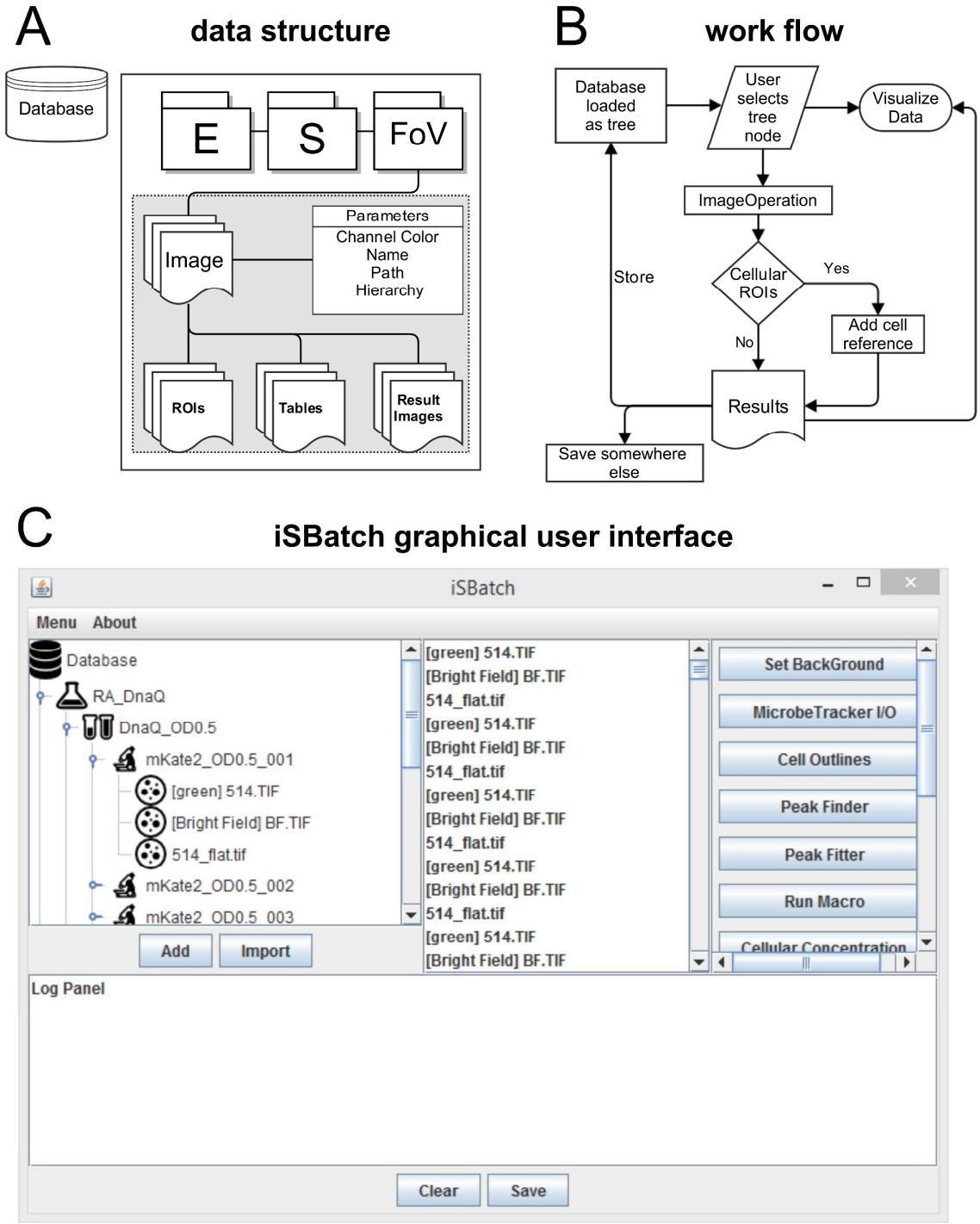


Figure 4~~2~~ iSBatch Structure. Panel A – Schematic representation of data structure (Experiment – E, Sample – S, Field of View – FoV) and its connections. Panel B – Logic structure of the algorithm; Panel C – User interface including ImageJ main panel (upper part) and iSBatch interface with the main commands.

Figure 3.

iSBatch *custom macro* panel

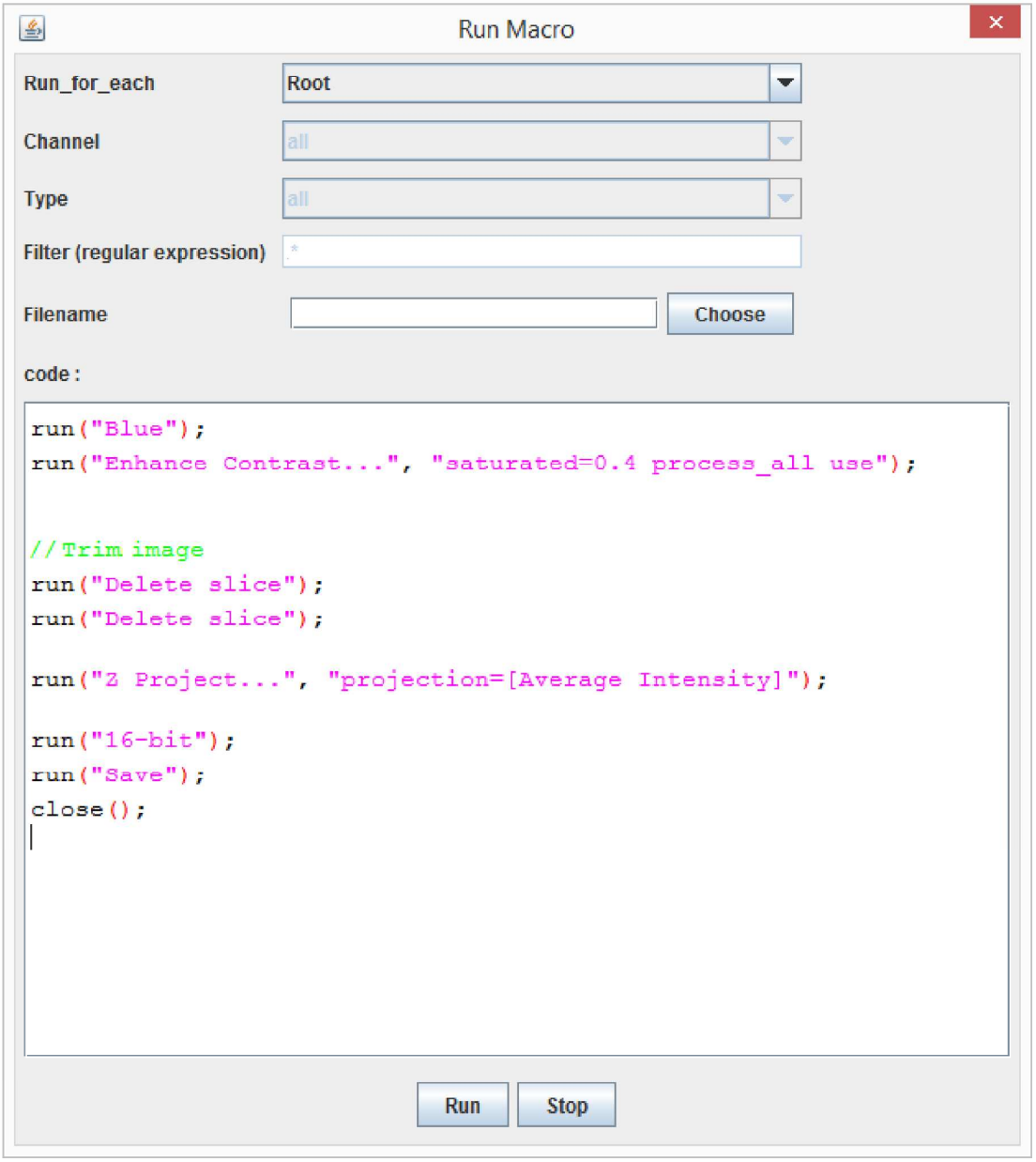


Figure 3 Custom Macro runner. iSBatch contains a custom macro runner that support syntax-highlighting for creating, running and editing existing ImageJ macros and plugin commands from the MacroRecorder.

Figure 4.

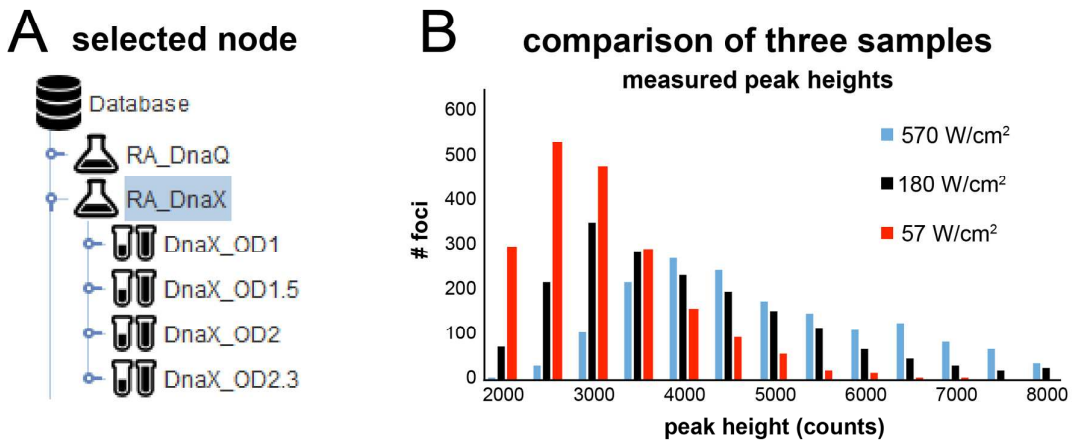


Figure 4 Built-in Peak Fitting Operation. Panel A – Selected node highlighting the 'Experiment' level. Panel B – Distribution of detected peak intensities within different 'Samples' in the same selected 'Experiment' node for DnaQ-mKate2.

Figure 5.

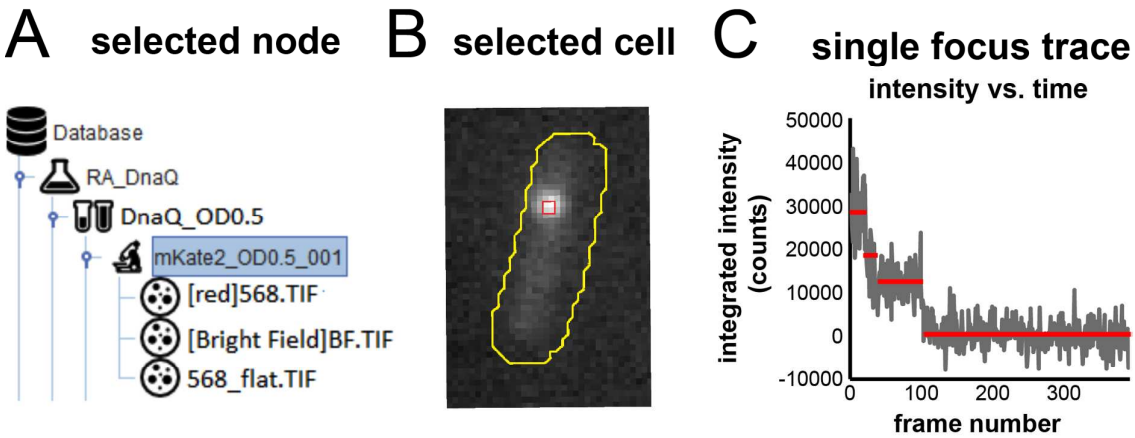


Figure 5 Step-wise photobleaching. Panel A – Selected node highlighting a 'Field of View' level. Panel B – Selected cell within a 'Field of View' with the boundaries assigned in yellow and a selection box in red. Panel C – Representative photobleaching trace of a detected focus. Red traces represent the detected steps by change-point analysis algorithm.

Figure 6.

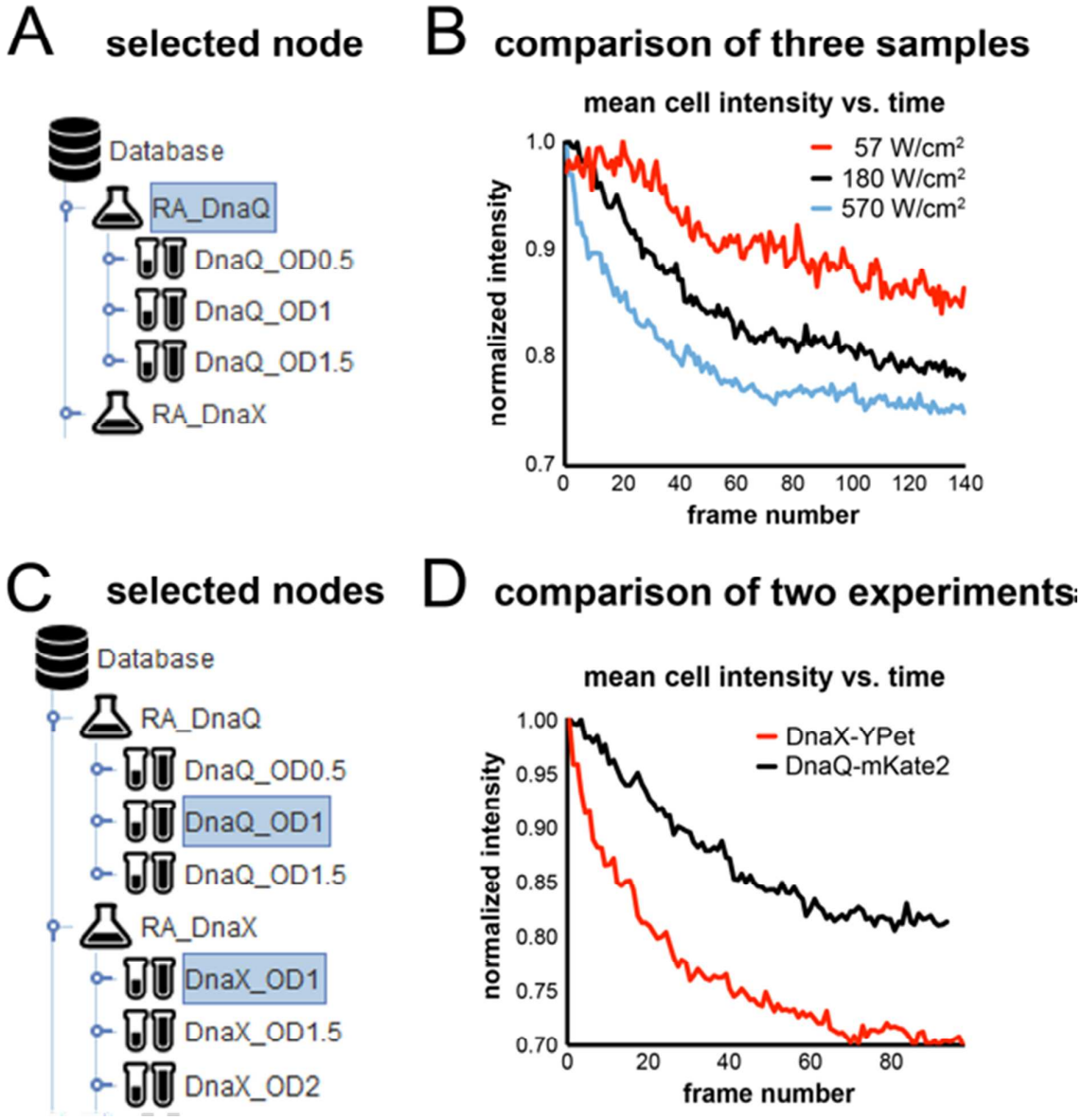


Figure 6 Cellular fluorescence obtained by Rapid Acquisition. Panel A – Selected node highlighting a 'Experiment' level Panel B – Dependence of cellular fluorescence photobleaching on laser intensity for DnaQ-mKate2. Panel C – Selected node highlighting two 'Samples' selected within different experiments. Panel D – Comparison of photobleaching properties of YPet and mKate2 when excited with same laser intensity (180 W/cm²).

Figure 7.

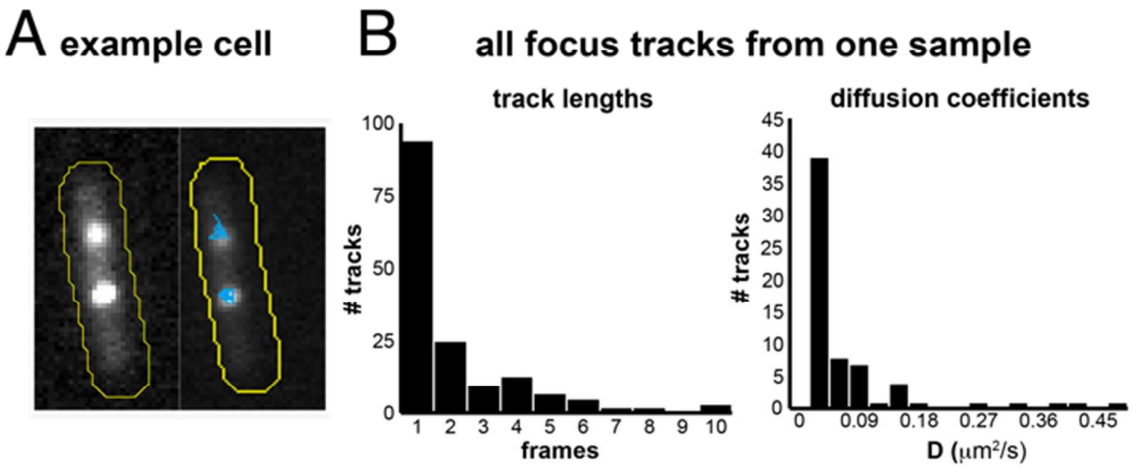


Figure 7 Particle tracking within cells. Panel A – DnaQ-mKate2 particles tracked inside a live *E. coli* cell (fluorescence on the left, tracked positions on the right). The confined nature of the track indicates protein bound to DNA. Panel B – Analysis of all detected focus tracks within a 'Sample' level, e.g. DnaQ-mKate2 acquired at 180 W/cm². Left panel shows the peak lifetime distribution and right panel the calculated diffusion coefficient for the same population.

Figure 8.

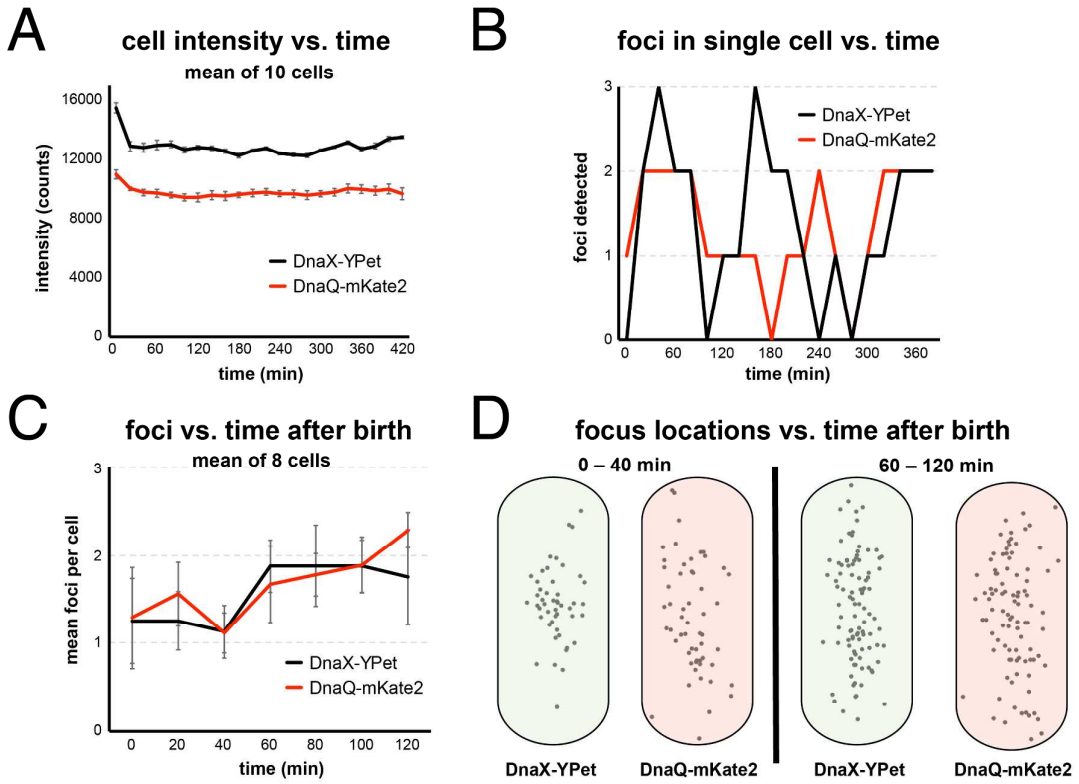


Figure 8 Built-in Time-Lapse analysis. Panel A – Fluorescence cell intensity over time for DnaX-YPet and DnaQ-mKate2. Panel B – Number of long-lived immobile peaks per cell, i.e. foci. Panel C – Data synchronization considering cell division times. Time zero is the first frame after cell division; cell division time is 100 – 120 min. Panel D – Location maps. A projection of detected peaks in an artificial, normalized cell. Left: projected cells with one detected focus, distributed towards the centre of the cell; Right: projected cells with two detected foci, distributed more towards the ¼ and ¾ positions in the cell.