nature protocols

PROTOCOL

https://doi.org/10.1038/s41596-019-0216-9

High-throughput detection and tracking of cells and intracellular spots in mother machine experiments

Jean Ollion[1*], Marina Elez[1,2] and Lydia Robert[1,3]

The analysis of bacteria at the single-cell level is essential to characterization of processes in which cellular heterogeneity plays an important role. BACMMAN (bacteria mother machine analysis) is a software allowing fast and reliable automated image analysis of high-throughput 2D or 3D time-series images from experiments using the 'mother machine', a very popular microfluidic device allowing biological processes in bacteria to be investigated at the single-cell level. Here, we describe how to use some of the BACMMAN features, including (i) segmentation and tracking of bacteria and intracellular fluorescent spots, (ii) visualization and editing of the results, (iii) configuration of the image-processing pipeline for different datasets and (iv) BACMMAN coupling to data analysis software for visualization and analysis of data subsets with specific properties. Among software specifically dedicated to the analysis of mother machine data, only BACMMAN allows segmentation and tracking of both bacteria and intracellular spots. For a single position, single channel with 1,000 frames (2-GB dataset), image processing takes ~6 min on a regular computer. Numerous implemented algorithms, easy configuration and high modularity ensure wide applicability of the BACMMAN software.

Introduction

A large part of our current understanding of cellular biology has been gained through the study of phenotypes at the population level. Although invaluable tools, population-averaging methods mask cell-to-cell differences and are therefore poorly suited to characterization of processes in which cellular heterogeneity plays an important role. During the past decades, single-cell characterization has become a focus of research in many different fields of biology[1–3]. The transition from population to single-cell analysis has been catalyzed by the expansion of technologies such as microscopy and flow cytometry. In particular, fluorescence time-lapse microscopy associated with gene reporter constructs has been extensively used to investigate cellular processes non-invasively and dynamically in single cells[4]. To obtain precise and reproducible measurements, time-lapse imaging is now being increasingly used in combination with microfluidic devices that allow a precise spatiotemporal control of the environment and high-throughput data collection from single cells[5].

One useful microfluidic device for single-cell studies of rod-shaped bacteria is the so-called 'mother machine', developed by Wang et al. in 2010 (ref. [6]). It consists of thousands of parallel dead-end microchannels in which cells grow in single file. Cells can grow and divide inside the microchannels for hundreds of generations, allowing the capture of high-throughput data on more than 10^5 individual cells per experiment. Mother machine devices are being increasingly used for bacterial single-cell studies to address various topics, such as cell growth[6], cell cycle control[7,8] and gene expression and regulation[9–11]. By using the mother machine and a fluorescent marker for nascent mutations created by DNA replication errors, we recently developed a new single-cell approach for studying mutagenesis and evolution in bacteria[12].

Long-term imaging of cells growing in the mother machine produces a large amount of data, typically several hundred gigabytes worth of images per experiment. Therefore, data management, image processing and analysis very rapidly become rate-limiting steps and require the implementation of a fast and reliable solution for automated image analysis.

[1]Laboratoire Jean Perrin, UMR 8237 Sorbonne Universités, UPMC Université Paris, Paris, France. [2]Institute of Systems and Synthetic Biology, UMR 8030, CNRS, Commissariat à l'Energie Atomique et aux Energies Alternatives, Genopole, Université d'Evry Val-d'Essonne, Université Paris Saclay, Evry, France. [3]Micalis Institute, Institut National de la Recherche Agronomique, AgroParisTech, Université Paris-Saclay, Jouy-en-Josas, France.
*e-mail: jean.ollion@polytechnique.org

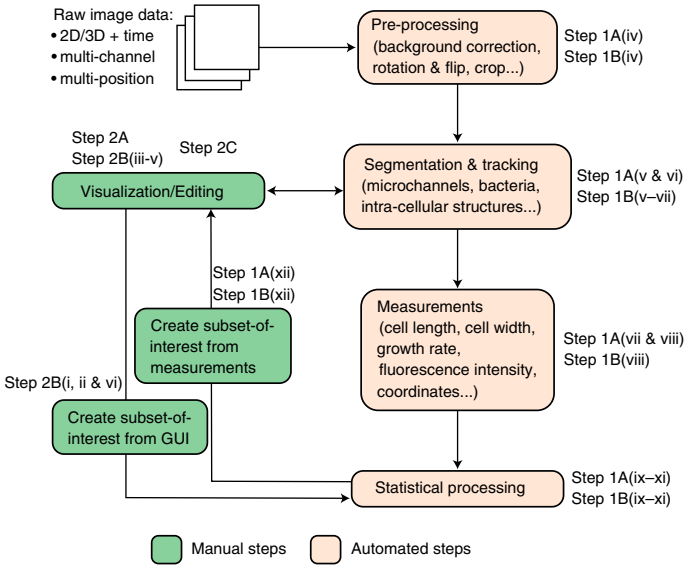

Fig. 1 | Processing pipeline. BACMMAN analysis pipeline couples automated image analysis, statistical analysis and manual visualization/editing. Raw image data are composed of time series of 2D/3D images with one or several channels. For each position, images are pre-processed, and then the different object classes are processed and measurements are performed. Through the statistical analysis of measurements, subsets of segmented objects of interest can be defined and specifically browsed and edited in the GUI, which substantially speeds up the visualization/editing part of the analysis. Subsets of interest can also be modified through the GUI.

To address this challenge, we recently developed software dedicated to mother machine data analysis (Fig. 1; ref. [12]). This software, which we call BACMMAN, allows the processing of large datasets rapidly and with high accuracy. BACMMAN has the ability to detect, segment and track bacteria in microchannels from either phase-contrast or fluorescence images and also to detect and track intracellular fluorescent spots. It also allows easy visualization and manual editing of image processing. This makes it a very complete tool.

Development of BACMANN

BACMMAN was initially developed to process data from our experiments, in which we follow mutagenesis and evolution in *Escherichia coli* growing in mother machine chips (see examples of these datasets at github.com/jeanollion/bacmman/wiki/Example-Datasets)[13]. Briefly, to follow the mutation occurrences at the single-cell level, we developed the mutation visualization (MV) approach, which is based on time-lapse imaging, use of a mother machine and expression of the fluorescently labeled mismatch repair protein MutL[14]. The latter makes foci on such DNA sites and thus allows visualization of mutations as bright fluorescent spots inside the cells. Microfluidic mutation accumulation (μMA), also based on time-lapse imaging and use of a mother machine, allows the following of fitness evolution during mutation accumulation and characterization of the distribution of fitness effects of spontaneous mutations. A detailed description of these two approaches and instructions on how to perform them can be found in an accompanying protocol[15].

BACMMAN was designed to also work on datasets other than MV and μMA. To facilitate the adaptation of BACMMAN algorithms to data from other labs, we developed specific test procedures that allow easy parameter optimization for all the different steps of pre-processing, segmentation and tracking. We also created detailed documentation for the different algorithms and functions of BACMMAN, which is displayed as pop-ups and in a specific help panel on the graphical interface. In addition, the highly modular structure of BACMMAN makes it very versatile, as new modules can be developed and easily plugged in. It is a Java-based software compatible with Linux, Windows and Mac OS and can be very easily installed through Fiji[16]. It has a user-friendly graphical user interface (GUI; Fig. 2) and can also be run from the command line to be used on a remote server.

Overview of the protocol

Here, we present the software and give a precise protocol to install and run it. To help the user discover the software and all its functionalities, we have made available two small datasets of typical

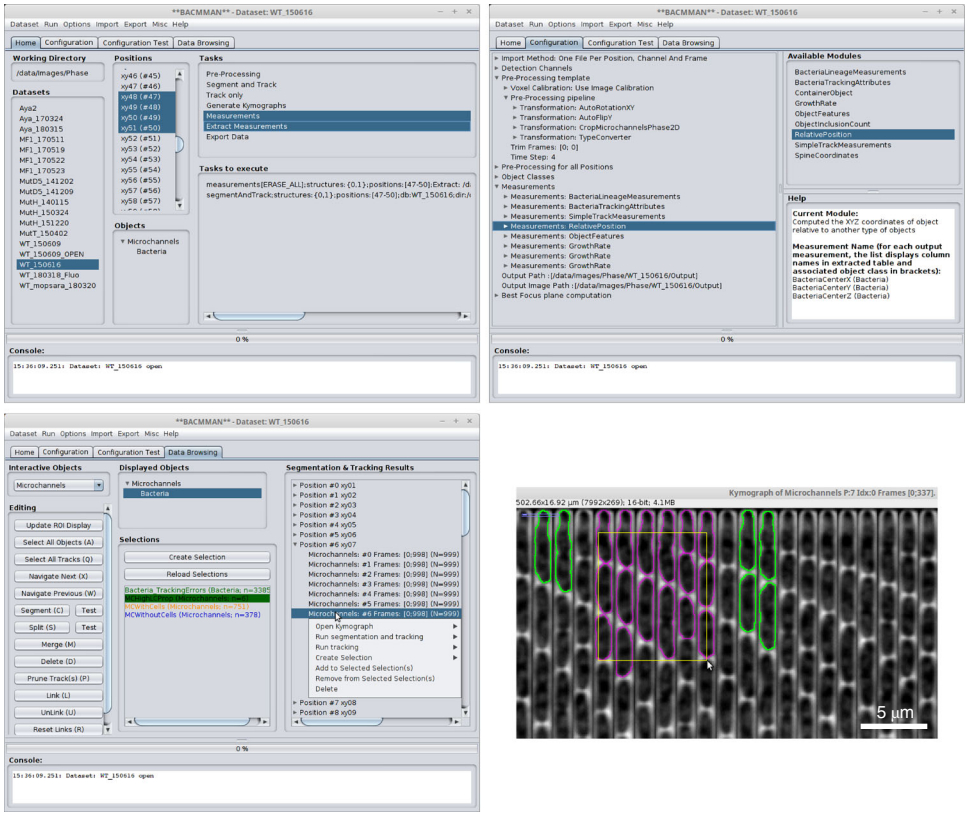

Fig. 2 | GUI. The BACMMAN GUI is composed of three panels. The Home panel (upper left) allows selection of the dataset and running of processing tasks on the different positions. The Configuration panel (upper right) allows configuration of the dataset structure (number of input channels, number of object classes), as well as the pre-processing and processing pipelines. The Data Browsing panel (lower left) allows visualization and editing of the segmentation and tracking results by displaying interactive kymographs (lower right panel) on which segmented objects can be directly selected and modified (pink contours indicate selected objects). Subsets of objects can also be displayed on those kymographs (green contours).

MV and µMA data[13]. Users can first install and run BACMMAN on these examples (Steps 1A and 1B) before using their own data. During image analysis with BACMMAN, numerous transformations are applied to the input images. All these transformations can be set and configured in BACMMAN, which generates a specific configuration file. In Step 1A or 1B, the configuration file is provided with the images and can be used without modification. By contrast, when using their own data, users may have to optimize the configuration, as illustrated in Step 1C. In a more advanced procedure, the user can also completely configure the image analysis de novo, as illustrated in Step 1D. Nevertheless, simple configuration changes such as those described in Step 1C should be sufficient most of the time to analyze experiments similar to those presented in Robert et al.[12]. For all the procedures, approximate timing is indicated, corresponding to the time it takes for an experienced user. More time may be required when using BACMMAN for the first time. In particular, learning how to adapt the configuration to a new dataset can take several hours to several days, depending on the dataset and on the user's image analysis skills.

Analysis pipeline

For BACMMAN, input data are required that consist of a time series of multichannel 2D or 3D (*z*-stack) images of several microscopy view-fields or 'positions' (a glossary of the specific terms used in BACMMAN is provided in Supplementary Table 2). From the images, BACMMAN allows detection of several classes of objects, such as the microchannels in which the cells grow, the bacteria or the fluorescently labeled DNA replication errors. The user defines the channels in which the objects must be detected, namely bright-field or fluorescence channels, and the specific processing pipeline that should be applied to each object class. For an example of this, consider dataset 1

('Equipment' section). This dataset consists of time series of phase-contrast images. It requires the definition of two object classes, microchannels and bacteria, which are detected in the same channel (bright-field) and processed through different pipelines.

The first step of the analysis pipeline, namely the pre-processing, includes a procedure to remove banding noise from the fluorescence images and a rotation that can eventually be combined with a flipping transformation to align the microchannels along the *y* axis with their closed end at the top. Images are cropped to avoid storing useless data. The next analysis step, i.e., processing, consists of pre-filtering, segmentation/tracking and post-filtering. Details of the algorithms used for the example datasets are provided in the Supplementary Notes ('Image Analysis' section) and Supplementary Figs. 1–3. The result of the processing step consists of segmented objects connected through their lineage information. Following image processing, measurements are computed and extracted as tabular data, in a format that is classically used in statistical analysis software such as pandas (Python) or R.

For result visualization and editing, BACMMAN generates kymographs in which images from successive frames are displayed next to one another, as shown in Fig. 3. Kymographs are navigable using shortcuts (Step 2A), and they are interactive, i.e., displayed segmented objects can be selected and the relevant information is then displayed next to the object. Kymographs allow easy manual editing of segmentation and tracking (Step 2C), thanks to a number of implemented functions that can delete, split or merge existing objects; create new objects; and create or delete tracking links between objects. When the quality of the images is good enough, our processing algorithms produce few segmentation and tracking errors (see 'Performance and comparison with other software' for statistics). Visualizing the whole dataset to correct such rare segmentation or tracking errors can be very inefficient and time consuming for high-throughput experiments. We therefore implemented the ability to detect subsets of objects with specific characteristics, i.e., 'selections', through the analysis of measurements using software such as R or pandas. These selections are then simply imported into BACMMAN and displayed on kymographs for visualization and editing. Examples of scripts to create selections are provided with our software ('Equipment' section). To illustrate the benefit of using selections, consider our work in Robert et al.[12], in which rare, slowly growing cells strongly affect the results. Using R, we created a selection corresponding to slowly growing cells, displayed this selection on kymographs in BACMMAN for visual inspection, and corrected any segmentation or tracking errors. Selections can thus be used to ensure error-free processing for a particularly interesting subset of the data. They can also be used to detect cells with abnormal characteristics that are likely to correspond to segmentation or tracking errors. They can therefore reduce, by several orders of magnitude, the time required for manual editing. More generally, selections are a useful tool for facilitating the understanding of specific behavior observed in rare cells, which can be selected and visualized easily.

Applications of the software

BACMMAN was initially developed to process data from μMA and MV experiments, as described in Robert et al.[12]. More specifically, it was designed to allow segmentation and tracking of *E. coli* cells growing inside the microchannels of the mother machine, imaged in either phase contrast or fluorescence, as well as segmentation and tracking of intracellular fluorescent spots corresponding to non-repaired DNA replication errors, i.e., emerging mutations. BACMMAN can process images from mother machine chips, i.e., with microchannels that are closed on one end and in which cells grow in single file, with different microchannel widths, lengths and spacing. If additional structures are visible in phase contrast—for instance, if the microchannels are fabricated with several layers such as those mentioned in ref. [17]—further development may be required. No specific imaging conditions are required, but the efficiency of the analysis will strongly depend on the quality of the images; in particular, blurry, out-of-focus images should be avoided. For cell tracking in time-lapse experiments, the time interval should be small enough to have >4 images per bacterial generation. There is no strong requirement regarding cell shape. We successfully used BACMMAN to analyze images of wild-type rod-shaped *E. coli* cells as well as images of mutants with slightly altered morphologies, such as round or Y-shaped cells. For detection and tracking of intracellular fluorescent objects, the current algorithms analyze spot-like objects, i.e., with a fluorescence profile exhibiting a single local maximum, with no requirement regarding the size of the objects. For spot tracking, the algorithm can handle gaps, i.e., when a spot is not visible on one or several successive frames, the algorithm can reconnect the last image before the gap to the first image after the gap. Options can be activated in the

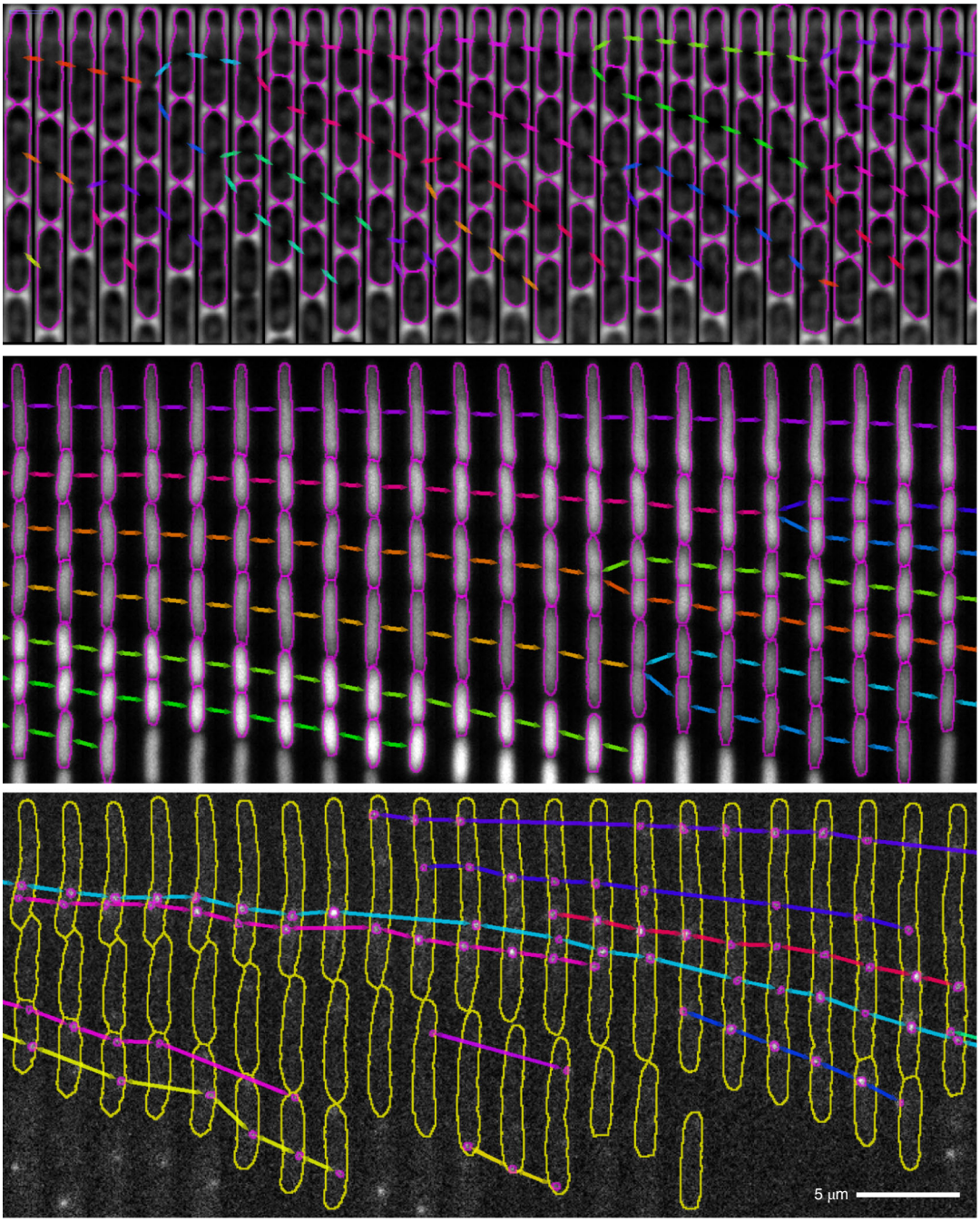

Fig. 3 | Segmentation and tracking of bacteria and spots in BACMMAN. Kymographs showing segmentation and tracking for bacteria imaged in phase-contrast (top) or fluorescence (middle) microscopy and for intracellular fluorescent spots (bottom). Contours of segmented objects are displayed in pink, and colored lines indicate tracking links.

configuration of BACMMAN to allow/forbid one spot to split into two spots and/or two spots to merge into one spot. Spot splitting may be useful for tracking of, for instance, DNA loci, which can duplicate during DNA replication.

The requirements for input images are detailed in the 'Equipment' part of the Materials section.

BACMMAN has a modular design. Importantly, this modularity makes it very versatile and allows the easy development of new processing pipelines for other related applications, through the creation and insertion of new modules. The modules of the present BACMMAN version are optimized for μMA and MV experiments, such as those provided in the datasets 1 and 2 (ref. [13]) but can be combined and configured to adapt the processing to other types of experiments, as illustrated in

Step 1C and 1D. To test the applicability of BACMMAN to other datasets, we collected datasets from six other labs (including the publicly available datasets provided with MoMA (https://github.com/fjug/MoMA) and molyso (https://github.com/modsim/molyso) software). We were able to analyze these datasets by changing only a few parameters in the configurations provided with our datasets 1 and 2 (ref. [13]). Most of the modifications corresponded to the input image format, the size of the microchannels, and, for fluorescent bacteria, the dynamic range of the signal (the corresponding modification is described in Step 1C). In the case of the molyso dataset, modifications were made at the cropping step of the pre-processing in order to remove a signal located at the closed end of microchannels that was perturbing microchannel detection. Notably, the processing pipeline for segmentation and tracking of bacteria used in dataset 1 (ref. [13]) could be used on this dataset without any modification. In the case of the MoMA dataset, the visual aspect of the bacteria differs substantially from our images, which we used to develop the segmentation algorithm. Consequently, we had to make more changes in the configuration to analyze the MoMA dataset, but we could still obtain satisfying results (no errors made on the whole dataset) without developing any new modules. A tutorial describing how to adapt the configuration to those two datasets can be found on the BACMMAN wiki (github.com/jeanollion/bacmman/wiki). Importantly, we implemented specific test procedures in BACMMAN to facilitate the configuration; these are explained in Step 1D. More specifically, all pre-processing and processing modules can be tested independently on a chosen subset of the data. The results of relevant intermediate steps are displayed to help in tuning the parameters.

Moreover, new modules can be developed and easily plugged in to extend the range of applications, for instance, to analyze experiments with other cell types and/or other microfluidic chips with different microchannel geometry. For instance, a modification of the mother machine in which the microchannels have two open ends[18] would require only a new tracking algorithm; all other modules can be used, including the segmentation module. The development of modules is further described in the developer documentation of the BACMMAN wiki. These features make BACMMAN a very flexible tool; however, to use it on a dataset that differs from the example datasets, some optimization and adjustments, as well as the development of new modules, may be necessary.

Performance and comparison with other software

Many software programs have been developed to analyze the 2D growth of cells, such as the growth of bacterial microcolonies on agarose pads[19–21]. However, it is hard to adapt them to obtain a fast and efficient analysis of cells growing in a mother machine. Only two software programs are specifically dedicated to mother machine data analysis: MoMA and molyso[11,22].

MoMA and molyso perform 1D segmentation of bacteria along the microchannel axis. However, when the width of the bacterial cells is smaller than the width of the microchannels, the cells can be inclined and two poles of adjacent bacteria can overlap along the microchannel axis, as illustrated in Fig. 4a (black arrow). In this case, the 1D-segmentation strategy can lead to many segmentation errors and bias the cell size measurements. This limitation is particularly important when the microchannel width cannot be perfectly adjusted to the size of the bacteria, for instance, in experiments in which a switch between rich and poor growth media is applied to the bacterium *E. coli*. In addition, *E. coli* cells are usually rod shaped but can be slightly bent, and some mutants or some environmental conditions may lead to unusual shapes. In such cases, the 1D strategy does not allow precise segmentation. In contrast to the 1D-segmentation strategy of molyso and MoMA, BACMMAN uses a 2D segmentation based on watershed algorithms. Segmentation in BACMMAN is therefore more robust to imperfect alignment of cells in microchannels, cell curvature and cell-shape abnormalities. In addition, it allows better estimation of cell morphological traits such as cell width.

A gap in a microchannel, i.e., a space with no bacteria, can be caused by either (i) cell lysis or (ii) motion of the cells in the channels. These two scenarios are difficult to distinguish because some cells can go out of the channel between two frames, so a loss of biomass does not necessarily indicate cell lysis. The MoMA and molyso tracking algorithms mainly rely on cell size and cell motion between frames, whereas tracking in BACMMAN is based on cell size and rank (i.e., relative position of the cell in the channel). As a result, our algorithm is not affected by cell motion within the microchannels or gaps between cells, but it cannot take cell lysis into account. By contrast, the MoMA and molyso algorithms may handle cell lysis better but are less robust to cell motion. Note that, if necessary, other tracking algorithms, such as those found in MoMA or molyso, can easily be plugged into BACMMAN.

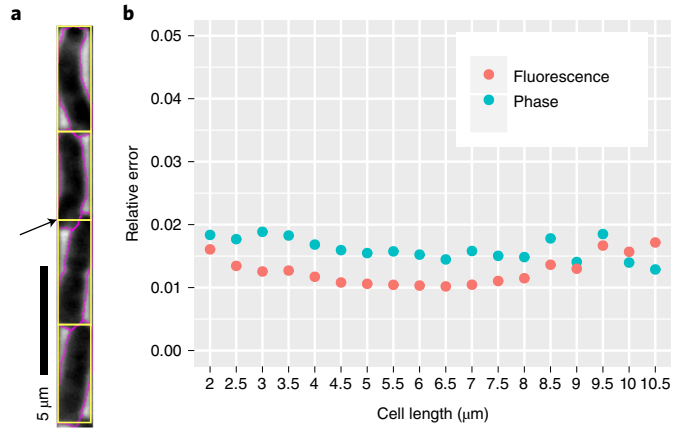

Fig. 4 | Segmentation performances. Segmentation performances of BACMMAN. **a**, Example of two overlapping cells in the microchannel axis (indicated by the arrow), illustrating the difference between 1D segmentation (yellow bounding boxes) and 2D segmentation (pink contours). **b**, Estimation of the measurement error for cell size. For each single cell, we performed a linear regression of the log-length vs. time and computed the residual errors. To obtain the relative error for cell length, we took the exponentials of these residuals and computed their coefficients of variation (see 'Precision of cell size measurement' in the Supplementary Notes for mathematical explanations), which we plotted as a function of cell length. Relative errors were computed for the analysis of both fluorescence (red; 750,424 length measurements for 126,717 cells) and phase-contrast (blue; 551,600 measurements, 103,400 cells) images. Image processing was performed automatically, with no manual editing, and the data were then filtered as explained in Robert et al.[12] to remove cells with aberrant growth rates due to cell death and tracking errors.

MoMA, but not molyso, allows manual editing of segmentation and lineages, but it does not include the ability to select a subset of data to be edited or visualized based on object measurements. To our knowledge, our software is the only one to offer such a coupling between statistical analysis of the data and image visualization and processing editing. Moreover, correction of errors in BACMMAN is very fast because it is done directly on interactive kymographs using shortcuts (for dataset 1 (ref. [13]), correction of the 30 errors takes ~2 min).

MoMA and molyso have limited extensibility and can perform segmentation and tracking of bacteria only in phase-contrast images, whereas BACMMAN can also perform segmentation and tracking of fluorescent spots within bacteria. BACMMAN can also perform segmentation and tracking of bacteria in fluorescence images, using an algorithm that is very robust to cell-to-cell variability in fluorescence levels, as exemplified in dataset 3 (ref. [13]).

To assess the performance of our cell segmentation algorithms, we first followed an approach similar to the one used by Kaiser et al.[11]. More specifically, we estimated the precision of our cell size measurement by analyzing the growth of single cells. Cell growth is exponential, and measurement errors can therefore be estimated from the residual errors in the regression of the log-size versus time (see Supplementary Notes, 'Precision of cell size measurement' section). As shown in Fig. 4b, the relative error in cell length is independent of cell length and is ~1–2% for both phase-contrast and fluorescence images. This analysis allows comparison of BACMMAN with the recently developed software MoMA, for which 2–3% relative error was estimated using this method. Although this analysis should reveal the segmentation errors that are made independently at different frames, it may not reveal a systematic bias. Therefore, we performed another analysis, this one comparing the results of segmentation performed either on phase-contrast or on fluorescence images. These two procedures use different segmentation algorithms and different input images, so systematic biases are likely to be different in the two procedures. We used fluorescence and phase-contrast images of the same cells and focused on cell length to estimate segmentation accuracy. Estimated cell length can change (typically a few percentage points) when segmentation parameters are varied. When varying the relevant parameters within a reasonable range, we found that the relative difference between the results of phase-contrast and fluorescence segmentation was always <7%. Although this analysis does not give a precise estimation of accuracy, it suggests that any systematic bias in cell length estimation should be small.

Another important aspect of the performance is the number of errors generated during segmentation and/or tracking and the time required to correct these errors. Our tracking algorithm is able to correct some segmentation errors and thus the overall procedure generates very few errors

when the quality of the images is good enough. For dataset 1 (ref. [13]), no errors are made in more than 99.8% of the 10,418 generations, with most errors clustered when cells exhibit transient morphological abnormalities. It took 11 min for an experienced user to visualize and correct the whole dataset. In dataset 2 (ref. [13]), no errors were made in more than 99.9% of the 931 generations, and visualization and editing took ~3 min.

Materials

Equipment

Data files

▲ **CRITICAL** The format of input images should be supported by BioFormats (http://www.openmicroscopy.org/bio-formats/) (such as .nd2, .tiff, .dv). BACMMAN can use either a single file containing all the images, or multiple files, with each file containing either the information for one channel and one position, or the information for one channel, one position and one frame. For a demonstration of different procedures that can be used with BACMMAN, three different datasets, and their associated configuration files and analysis scripts are available from github.com/jeanollion/bacmman/wiki/Example-Datasets [13]. All files should be unzipped.

- Dataset 1 can be used to test Step 1A. It consists of a time series of 985 images corresponding to one position (field of view) imaged through one channel (bright field).
- Dataset 2 can be used to test Step 1B and 2C. It consists of a time series of 500 images, corresponding to one position and two fluorescence channels.
- Dataset 3 can be used to test Step 1C. It consists of a time series of 556 images, corresponding to one position and one fluorescence channel.

Software

- BACMMAN (https://github.com/jeanollion/bacmman)
- Fiji (https://imagej.net/Fiji/Downloads)
- Java v.8 or later (https://www.java.com/en/download/) or OpenJDK v.8 or later (https://openjdk.java.net/install/)
- Git (https://git-scm.com/downloads)
- Maven v.3 (https://maven.apache.org/)

Computer equipment

- Any Mac, Linux or Windows computer with a 64-bit version of the operating system should be able to run our software. The amount of random access memory (RAM) allocated to Fiji/Java must be large enough to load the images corresponding to a whole position converted in 32-bit float. For the dataset 1, one position corresponds to 2.2 GB in 16-bit, thus the RAM should be at least 4.4 GB.

Equipment setup

Installation under FIJI

To install under FIJI, follow the instructions below:

1. Download and install FIJI 64-bit version (https://imagej.net/Fiji/Downloads).
2. Choose *Help* > *Update…*
3. Perform any needed updates by clicking on *Apply changes*.
4. Restart FIJI.
5. Choose *Help* > *Update* again. Repeat Steps 3 and 4 until the message 'Your Fiji is up to date!' is displayed.
6. Re-run the updater, click *Advanced Mode*, and then click *Manage update sites*. Tick the checkboxes next to *BACMMAN* and *ImageScience*.
7. Click Close and Apply Changes; when prompted, restart FIJI.
8. From the plugin menu, run the *BACMMAN* > *BACteria in Mother Machine ANalyzer* command.

Installation as a stand-alone application

▲ **CRITICAL** Requires Git and Maven v.3 and Java v.8 or later or OpenJDK v.8 or later.

Run the following BASH commands (replace <VERSION> with the actual version):

```
git clone https://github.com/jeanollion/bacmman.git
cd bacmman
mvn clean dependency:copy-dependencies package
```

```
cd bacmman-ij1_/target
java -cp dependency/*:bacmman-ij1_-<VERSION>.jar bacmman.binding.IJ1
```

Running in headless mode

To run in headless mode, follow the instructions below:

1 Generate tasks from the *Home* tab of the GUI by opening a dataset, selecting positions, object classes and tasks to run and right-clicking in the *Tasks to execute* panel and choosing *Add current task to task list*.

2 Repeat step 1 with other tasks, if necessary.

3 Save the tasks list to a file by right-clicking in the *Tasks to execute* panel and choosing *Save to file* to save all tasks in a single file in order to run them successively. Alternatively, you can choose *Split and Save to files* to generate several files, each one containing all the tasks of a given position. This allows one to run them in parallel because they are independent.

4 Run the tasks in headless mode: after installing as a stand-alone application (see above), run the following command from the bacmman-headless/target directory: (replace <VERSION> with the actual version and <taskfile.json> with the path to a .json task file generated in Step 3):

```
java -cp dependency/*:bacmman-headless-<VERSION>.jar bacmman.ui.ProcessTasks <taskfile.json>
```

Procedure

1 Follow option A to analyze a phase-contrast mother machine time-lapse experiment; follow option B to analyze a mother machine time-lapse experiment with fluorescent cells and intracellular spots; follow option C to modify an existing configuration to analyze a dataset similar to dataset 1 or 2; follow option D to create a de novo configuration to analyze a dataset that differs substantially from the provided example datasets.

(A) **Analysis of a phase-contrast mother machine time-lapse experiment** ● Timing **2–5 min for Step 1A(i–iii); 15–20 min for Step 1A(iv–viii); 10–15 min for Step 1A(ix–xvi)**

(i) *Create a dataset and import images*. Start BACMMAN ('Equipment setup'). First, create an empty folder. To define this folder as the working directory for BACMMAN, click on the *Home* tab, then right-click on the *working directory* field and choose the empty folder or enter its path manually.
▲ CRITICAL STEP The working directory is the folder that will contain subfolders corresponding to the different datasets to be analyzed. All these subfolders will be listed in the *Datasets* list.

(ii) From the *Dataset* menu, choose *New dataset from template*, select the configuration file ending with '_dataset1.json' ('Equipment' section) that corresponds to dataset 1 and enter 'dataset1' as the name of the dataset. The name of the dataset appears in the list.
▲ CRITICAL STEP Creating a new dataset results in the creation of a folder with the same name located in the path defined in Step 1A(i). In this folder, all data related to this new dataset will be stored (the configuration file, pre-processed images, segmentation and lineage results, and measurements). If you quit BACMMAN, at the next BACMMAN startup, this dataset will not be opened automatically. To access the dataset at next startup, select this dataset in the *Datasets* list and choose *Open Dataset* from the *Dataset* menu.

(iii) From the *Run* menu, choose *Import/re-link Images* and select the input folder (here, the folder containing the images of dataset 1). A new position will appear in the *Positions* list.
▲ CRITICAL STEP When images are imported, their paths are automatically saved in the .json configuration file located in the dataset folder. If the location of the images changes, they can be relinked using the *Import/re-link Images* option in the *Run* menu.
▲ CRITICAL STEP After importing/re-linking input images, input images can be visualized by right-clicking on positions in the *Positions* list and selecting *Open Input Images*.
? TROUBLESHOOTING

(iv) *Run image-processing steps*. Select one position in the *Position* list. From the *Tasks* list, select *Pre-Processing*, and from the *Run* menu, choose *Run Selected Tasks*.
▲ CRITICAL STEP In this step and Step 1A(v–viii), we detail the different processing steps one by one. All these steps can be run at once by selecting *Pre-processing*, *Segment and Track*, *Measurements*, *Extract Measurements* in the *Tasks* list, all the objects in the *Objects*

list and all the positions in the *Positions* list. See the 'Running in headless mode' paragraph of the Materials section for processing in command-line mode.

▲ **CRITICAL STEP** Pre-processed images are stored in the *Output* folder of the dataset folder; they can be visualized by right-clicking on the position in the *Positions* list and by choosing *Open PreProcessed Images*.

▲ **CRITICAL STEP** Before performing pre-processing, you can test it by running it on a smaller subset of data. For this, see Step 1D(xiii).

? TROUBLESHOOTING

(v) In the *Objects* list, select the *Microchannels* object class and in the *Tasks* list, select *Segment and Track*. From the *Run* menu, choose *Run Selected Tasks*. Visualize the results by following Step 2A(i–v).

▲ **CRITICAL STEP** Segmentation and tracking results are automatically stored in files located in the *Output* folder of the dataset. You can test a processing pipeline by running it on a smaller subset of data; see Step 1D(xxii).

? TROUBLESHOOTING

(vi) Repeat Step 1A(v), selecting *Bacteria* instead of *Microchannels*. Visualize the results by following Step 2A(i–v) and, if necessary, edit segmentation and tracking errors as explained in Step 1C.

▲ **CRITICAL STEP** When processing several object classes separately, such as those proposed in Step 1A(v and vi), one should always respect their order as given in the *Objects* list, because the processing for one object class can depend on those of the previous ones (e.g., always process *Microchannels* before *Bacteria*).

▲ **CRITICAL STEP** Processing can also be run for bacteria in a single microchannel by selecting and right-clicking on the microchannel in the *Segmentation & Tracking Results* panel of the *Data Browsing* tab and choosing *Run segmentation and tracking > Bacteria*.

? TROUBLESHOOTING

(vii) Select all positions from the *Positions* list, select *Measurements* from the *Tasks* list and choose *Run Selected Tasks* from the *Run* menu.

▲ **CRITICAL STEP** To visualize measurements from the GUI, see Step 2A(v).

(viii) Select all positions from the *Positions* list, select *Extract Measurements* from the *Tasks* list and choose *Run Selected Tasks* from the *Run* menu.

▲ **CRITICAL STEP** Measurements will be extracted as tabular data in a semicolon-separated (.csv) file. One file will be created for each object class selected in the *Objects* panel; the files will be created in the folder of the dataset, with a name ending with the index of the object class in the dataset (the index is 0 for microchannels and 1 for bacteria).

(ix) *Perform statistical analysis*. Open the analysis script 1 ('Equipment' section) and run the block 1 under R (or Python).

▲ **CRITICAL STEP** The analysis script is separated into various parts, called blocks here.

(x) Edit the *folder* variable in block 2 of the script, set the actual path of the folder and then run block 2 to import the measurements.

(xi) Run block 3 to draw a histogram of growth rates.

(xii) Run block 4 to generate a selection of long cells (filaments).

(xiii) From the *Data browsing* tab, click on the *Reload Selections* button in the *Selections* panel.

(xiv) See Step 2B(iii and iv) to display and navigate within the selection.

(xv) See Step 2C(i) to edit segmentation and tracking errors.

▲ **CRITICAL STEP** After editing, re-run Step 1A(vii and viii) and (x) to correct the measurements and update the output data file.

(xvi) Run block 5 to generate a plot of mother cell length through time.

(B) **Analysis of a mother machine time-lapse experiment with fluorescent cells and intracellular spots ● Timing 2–5 min for Step 1B(i–iii); 15–20 min for Step 1B(iv–viii); 10–15 min for Step 1B(ix–xiii)**

▲ **CRITICAL** A screencast for this procedure is available at the BACMMAN wiki (github.com/jeanollion/bacmman/wiki/Example-Datasets).

(i) *Create a dataset and import images*. Follow Step 1A(i)

(ii) From the *Dataset* menu, choose *New dataset from template*, select the configuration file of dataset 2 ('Equipment' section) and enter 'dataset2' as the name of the dataset.

(iii) From the *Run* menu, choose *Import/re-link Images* and select the folder containing the images of dataset 2.

(iv) *Run image-processing steps*. Follow Step 1A(iv).
? TROUBLESHOOTING

(v) Follow Step 1A(v).
? TROUBLESHOOTING

(vi) Follow Step 1A(vi).
? TROUBLESHOOTING

(vii) Follow Step 1A(v), selecting the object class *Spot* instead of *Microchannel*.
▲ **CRITICAL STEP** To display the contour of bacteria while visualizing DNA replication errors (fluorescent spots), use the command to create a selection of bacteria, described in Step 2B(iv).
? TROUBLESHOOTING

(viii) Follow Step 1A(vii and viii).
▲ **CRITICAL STEP** You will sometimes need/wish to check and manually edit segmentation and tracking results. For this, follow the instructions given in Step 2A and 2C(i).

(ix) *Perform statistical analysis*. Open the analysis script 2 ('Equipment' section) and run block 1 under R (or Python).
▲ **CRITICAL STEP** The analysis script is separated into various parts, called blocks here.

(x) Edit the *folder* variable in block 2 of the script and set the actual path of the folder; then run block 2 to import the measurements.

(xi) Run block 3 to display a histogram of fluorescence spot residence times.

(xii) Run block 4 to generate a selection of long tracks.

(xiii) Follow Step 1A(xiii–xv).

(C) **Adaption of an existing configuration to another dataset** ● **Timing 5 min for Step 1C (i–iii); 5 min for Step 1C(iv); 5 min for Step 1C(v and vi)**
▲ **CRITICAL** A tutorial describing how to adapt an existing configuration to datasets from other labs is available at the BACMMAN wiki (github.com/jeanollion/bacmman/wiki/How-to-Adapt-Configuration).

(i) *Create a dataset and import images*. Follow Step 1A(i).

(ii) From the *Dataset* menu, choose *New dataset from template*, select the configuration file of dataset 3 ('Equipment' section) and enter 'dataset3' as the name of the dataset.

(iii) From the *Run* menu, choose *Import/re-link Images* and select the folder containing the images of dataset 3.
▲ **CRITICAL STEP** When importing a new image format, always check that it was correctly imported by opening it and checking the dimensions of the image (Ctrl + Shift + P). In particular, if the *T* and *Z* dimensions are swapped, this can be fixed by adjusting the *Swap T & Z dimensions* parameter of the import method and re-importing the images through the *Import/re-link images* option.

(iv) *Run image-processing steps*. Follow Steps 1A(iv–vi). Some segmentation and tracking errors can be seen, in particular, in the microchannels #3 track.

(v) *Edit configuration*. In dataset 3, the fluorescence exhibits a much higher average level and a much higher cell-to-cell variability as compared to dataset 2. In consequence, image processing must be adapted to improve segmentation of bacteria. In the *Configuration Test* tab, select *Processing* as *Step* and *Bacteria* as *Object Class*. In the lower part of the tab, unfold sequentially the following nested nodes *Processing Pipeline: SegmentAndTrack*, *Tracker:BacteriaClosedMicrochannelTrackerLocalCorrections, Segmentation Algorithm:BacterialFluo* and Foreground Selection Method:EDGE FUSION and set the value of the parameter *Background Edge Fusion Threshold* to 4 by right-clicking on the current value.
▲ **CRITICAL STEP** Refer to Step 1D(xxii) to test segmentation in order to determine the value of the parameter for a different dataset.
▲ **CRITICAL STEP** Always save the configuration changes (*Dataset* > *Save Configuration changes*) before processing.

(vi) Follow Step 1A(vi) to re-run segmentation and tracking of bacteria, and 2A(i) to visualize the new segmentation and tracking results.

(D) **Creating and configuring a new dataset** ● **Timing 5 min for Step 1D(i–vii); 10–60 min for Step 1D(viii–xv); 10–60 min for Step 1D(xvi–xxii) ;10 min for Step 1D(xxiii–xxv)**
▲ **CRITICAL** If users want to analyze a time-lapse dataset that differs from those described in Step 1A and 1B, they can either adapt the configuration files provided here for datasets 1 and 2

or create a configuration file de novo. Defining de novo pre-processing and processing pipeline requires image analysis skills.

▲ **CRITICAL** In the *Configuration* and *Configuration Test* tabs, all the parameters are hierarchically organized and displayed as a tree in which sub-parameters can be accessed by unfolding the parameter. The values of the parameters can be set by right-clicking on the parameters. The red font indicates inconsistencies in the configuration. The parameters displayed in bold are the most critical for analysis optimization. When available, pop-ups giving additional information on parameters are automatically displayed on mouse-over.

▲ **CRITICAL** All the configuration tests (see the *Configuration Test* tabs) can be performed in two different modes, a 'simplified' mode for inexperienced users, in which only the most important parameters can be changed and non-technical help is provided, and an 'advanced' mode, in which all the parameters can be accessed and more technical descriptions of the algorithms are provided. The mode can be changed in the *Test Mode* sub-panel of the *Test Controls* panel.

(i) *Create a new dataset*. Follow Step 1A(i).

(ii) From the *Dataset* menu, choose *New* and enter the name of the dataset.

(iii) First, set the channels by right-clicking on *Detection Channel* in the *Configuration* tab and choosing *Add Element*.

(iv) Right-click on the newly added channel and set a name.
▲ **CRITICAL STEP** Depending on the format of the input dataset, the user might need to indicate a keyword that will allow finding of the image files corresponding to the appropriate channel; refer to the *Help* panel for the *Channel Keyword* parameter.

(v) Repeat Step 1D(iii) and 1D(iv) for each channel of the input dataset.
▲ **CRITICAL STEP** The number of channels set in the configuration must match the number of channels of the input data; otherwise, the images will not be imported.

(vi) Second, set the import image method by right-clicking on *Import Method*. Here you should choose one of the three different available methods, depending on your dataset: one file containing all images, one file per channel and per position or one file per channel, position and frame. For the two last methods, additional configuration is needed; refer to the *Help* panel for the *Import Method* Parameter.

(vii) Follow Step 1A(iii) to import images.
▲ **CRITICAL STEP** When importing a new image format, always check that it was correctly imported by opening it and checking the dimensions of the image (Ctrl + Shift + P). In particular if the *T* and *Z* dimensions are inverted, this can be fixed by adjusting the *Swap T & Z dimensions* parameter of the import method and re-importing the images through the *Import/re-link images* option.

(viii) *Configure the pre-processing pipeline*. In the *Configuration* tab, unfold the *Pre-Processing for all Positions* node, then the first node that corresponds to the first position (called *First position node* below), and then the *Pre-processing* node of that position.

(ix) By default, voxel calibration is read from the image metadata. You can check that the images are correctly calibrated by opening the images by right-clicking on the *First position node*, choosing *Open Input Images* and checking the calibration in the image properties (Ctrl + Shift + P). If the calibration is incorrect, set it by right-clicking on *Voxel Calibration*, choosing *Custom Calibration*, and unfolding the *Voxel Calibration node* and configuring it.

(x) To edit pre-processing, select the *Configuration Test* tab, select *Pre-Processing* as *Step* in the *Test Controls* panel, and select the position on which pre-processing should be tested in the *Position* sub-panel. Then, right-click on *Pre-Processing pipeline* in the *Configuration* panel below and choose *Add Element*.

(xi) Right-click on the newly created *Transformation node*, set a transformation (i.e., element of the pre-processing pipeline) from the *Modules* menu and configure it, referring to the help (*Help* panel or pop-up menu) of the transformation and of its parameters.

(xii) Repeat Steps 1D(x and xi) to add as many transformations as needed.

(xiii) To test a transformation, choose the test mode (simplified or advanced) in the *Test Mode* sub-panel of the *Test Controls* panel, right-click on the transformation and

choose *Test Transformation*. Images of the current position before and after the transformation, as well as images or graphs corresponding to intermediate steps, will automatically open.

▲ **CRITICAL STEP** To save time, transformations can be tested on a subset of frames. To do so, modify the *Frame Range* parameter in the *Test Controls* panel.

? TROUBLESHOOTING

(xiv) When the transformations are set and configured for one position, they can be copied to other positions. To do so, click on the *Copy to all positions* button.

▲ **CRITICAL STEP** To have this configuration set by default on newly imported positions, click on *Copy to template button*.

(xv) Perform pre-processing by following Step 1A(iv).

(xvi) *Configure object classes and processing*. From the *Configuration* tab, right-click on *Object Classes* and choose *Add Element*.

(xvii) Set the name of the object class by right-clicking and unfold it.

(xviii) Set the detection channel associated to the object class by right-clicking on *Detection Channel* in the node of the object class that you just created.

(xix) If necessary, set the *Parent* and the *Segmentation Parent*. Some object classes can be located within others on the images; for instance, bacteria are inside microchannels and fluorescent spots are inside bacteria. Image processing takes this into account by segmenting and tracking a given object within an object of another class (called *Segmentation Parent* and *Parent*, respectively) For instance, in dataset 2, the *Parent class* of *Bacteria* and *Spots* is *Microchannels*, the *Segmentation parent* of *Bacteria* is *Microchannels* and the *Segmentation parent* of *Spots* is *Bacteria*.

(xx) In the *Configuration Test* tab, select *Processing* as *Step* in the *Test Controls* panel. For each object class, select the object class in the *Test Controls* panel, set the *Processing Pipeline* and configure it. For instance, when performing segmentation and tracking jointly, choose *SegmentAndTrack* as pipeline and set the tracking algorithm and eventually pre-filters and post-filters.

(xxi) Perform segmentation and tracking for each object class, sequentially following Step 1A(v).

(xxii) To test a processing step (segmentation, tracking, pre-filtering or post-filtering), from the *Configuration Test* tab, select *Processing as Step* in the *Test Controls* panel and select the object class in the *Test Controls* panel; right-click on the operation to be tested and choose a *Test* option. This will display interactive kymographs of the operation results and eventually images corresponding to intermediate steps to *Help* panel in the *Configuration* module. To know how to configure the parameters using intermediate images, refer to the *Help* panel for each parameter displayed in the *Help* panel. Depending on the module, additional data may be displayed by right-clicking the menu on the resulting kymograph (refer to the module documentation displayed in the *Help* panel).

▲ **CRITICAL STEP** To save time, processing operations can be tested on a subset of frames. To do so, modify the *Frame Range* in the *Test Controls* panel.

? TROUBLESHOOTING

(xxiii) *Configure the measurements*. In the *Configuration* tab, right-click on *Measurements* and choose *Add Element*.

(xxiv) Right-click on *Measurement*, choose a module and configure it.

(xxv) Repeat Step 1D(xxi and xxii) as many times as needed.

2 Follow option A to visualize segmentation and tracking results; follow option B to use selections to visualize or analyze only a part of the dataset; follow option C to edit segmentation and tracking results.

(A) **Visualization of segmentation and tracking results** ● **Timing ~1–5 min**

▲ **CRITICAL** The visualization and editing GUI is designed to be used with shortcuts for faster manipulation. The shortcuts depend on the keyboard layout, which should first be set in the *Help* menu (*Help* > *Shortcut preset*). All shortcuts can be displayed and printed from this menu. In this section, we will refer to command names instead of shortcuts.

(i) From the *Segmentation & Tracking Results* panel of the *Data Browsing* tab, click on position #0: all segmented microchannels will be listed. To visualize the kymograph of a microchannel, right-click on the microchannel, select *Open Kymograph*, and choose the object class to visualize it.

▲ CRITICAL STEP The opened image is usually too big to be integrally displayed on the screen. The first mouse wheel scroll will set the zoom to 100%, then it will allow the user to navigate through the kymograph in the time axis. Use Shift + mouse wheel for faster navigation through the kymograph.

▲ CRITICAL STEP To visualize microchannel tracking, kymographs of the whole position can be displayed by right-clicking on a position element of the *Segmentation & Tracking Results* panel.

(ii) To change the interactive object class, i.e., segmented objects displayed on the kymographs, select the object class from the list in the *Interactive Objects* panel.

▲ CRITICAL STEP Only the objects that have already been segmented can be displayed.

(iii) To select and visualize all segmented objects, click on the corresponding button in the *Editing* panel of the *Data browsing* tab or use the associated shortcut.

▲ CRITICAL STEP To select only a subset of the objects, create a selection on the image by using any ImageJ selection tool (such as a rectangle selection or freehand selection) or by clicking on an object. Objects can be added or removed from the current selection by pressing Shift.

(iv) To select and visualize all tracks, click on the corresponding button in the *Editing* panel of the *Data browsing* tab or use the associated shortcut.

▲ CRITICAL STEP To select only one track, use the shortcut indicated in the *Shortcut* table *Toggle display object/track* and click on one object of the track.

(v) Right-click on the selected object(s) to display the associated information and measurements (if already computed).

(vi) To open the next or previous kymograph, use the corresponding shortcut from the *Shortcut* > *Navigation / Display* menu.

(B) **Creation, display and navigation of selections** ● Timing ~1–5 min

(i) Create a selection by pressing the *Create selection* button in the *Selections* panel of the *Data Browsing* tab.

▲ CRITICAL STEP All the selections of the current dataset are automatically saved in the database and will appear in the list of the *Selections* panel.

(ii) To add or remove objects from this selection, select them on a kymograph, right-click on the selection and choose *add* or *remove* objects. The number of objects in the selection and the index of their object class are indicated next to the selection name.

▲ CRITICAL STEP Addition or removal of elements is accessible through shortcuts and will be performed only on selections set as *active*. To allow such actions to be simultaneously applied to several selections, active selections can be grouped. Two groups of active selections, group 0 and group 1, can be used. To add a selection to an active group, right-click on a selection and enable *Active selection group*. Shortcuts for each active selection group are listed in the *Shortcuts* > *Selections* menu.

(iii) To display objects and tracks from a selection, right-click on the selection and select *Display Objects* and *Display Tracks*, respectively.

▲ CRITICAL STEP Alternatively, object display can be activated/deactivated through shortcuts on active selections listed in *Shortcuts* > *Selection* (see CRITICAL STEP note for Step 2B(ii)).

(iv) To navigate a selection, right-click on the selection and select *Enable Navigation*. The commands *Navigate Next* and *Navigate Previous* (see *Shortcuts* > *Navigation/Display*) will center the kymograph view on the next (or previous) objects of the selection. If no objects are present in the current kymograph after (or before) the current view, executing the command twice will open the next (or previous) kymograph containing objects of the selection.

(v) To simultaneously visualize the cells and the intracellular spots, a selection containing all cells in a microchannel can be displayed on the *Spots* kymograph. To automatically generate such a selection, select the microchannel in the *Segmentation & Tracking Results* panel; from the right-click menu, choose *Create Selection* > *Bacteria*. The selection will be automatically displayed. This command can also be performed on several microchannels at the same time, or on a whole position.

(vi) To export selections in a tabular format that can be imported in R or pandas, select the appropriate selection(s) in the *Selections* panel and run the *Extract Selections* command from the *Run* menu and choose a folder.

(C) **Editing segmentation and tracking** ● **Timing 5–15 min for Step 2C(i); 1–2 min for Step 2C (ii–vii)**

▲ **CRITICAL** A screencast for this procedure is available at the BACMMAN wiki (github.com/jeanollion/bacmman/wiki/Example-Datasets).

▲ **CRITICAL** All editing commands and associated shortcuts are listed in the *shortcuts list* from the *Help* menu.

(i) *Setup*. Open a kymograph and set the interactive object class as in Step 2A(ii). The currently implemented editing commands are as follows:

Command	Description
Merge	Merges the selected objects frame by frame; i.e., only the objects from the same frame can be merged, so when objects belonging to different frames are selected they are only merged with the selected objects of the same frame
Manual Split	Split the object(s) according to the line the user draws, using ImageJ's *Freehand selection* tool
Split	Splits the selected object(s) in two when possible, using the split algorithm set for the interactive object class
Segment	Creates objects from points that the user draws, using the manual segmentation algorithm set for the interactive object class. Use the *Switch to object creation tool* command (*Shortcuts* > *Object/Lineage editing*) to draw points on the kymograph. The same command switches back to the rectangle selection tool.
Delete	Deletes the selected objects
Delete all after	Delete all objects of the kymograph after the frame of the first selected object
Prune	Delete the selected object and all its progeny
Link	Create a link between selected objects. If several scenarios are available, the solution that globally minimizes the motion of objects between frames will be chosen
Unlink	Removes all links between all selected objects
Reset Links	Removes all links involving any of the selected objects
Create Track	Creates new track(s), starting from selected object(s)

▲ **CRITICAL STEP** When running the split or segment algorithms, a delay of a few seconds may occur for some processing pipelines.

▲ **CRITICAL STEP** All modifications are automatically stored in the database and are displayed on the image as additional arrows on links or pointing modified objects.

▲ **CRITICAL STEP** Editing segmentation usually breaks the lineage. So after editing segmentation, also edit the lineage of the modified objects. In addition, when editing segmentation of microchannels, processing of bacteria and spots should be re-run. Likewise, when editing segmentation of bacteria, processing of spots should be re-run.

▲ **CRITICAL STEP** When an object 'A' is linked to two objects 'B' and 'C' in a following frame, if the link between A and B is removed, the tracks of A and C will be merged. They can be split again if necessary using the *Create Track* command.

▲ **CRITICAL STEP** Measurements are not automatically updated after editing segmentation or lineage; they should be performed and extracted again on each position where editing has been done.

(ii) *Example of manual editing on dataset 2*. Open dataset 2 and go to the data browsing tab.

(iii) Open the kymograph of the first microchannel track (Step 2A(i–iv)): between frames 173 and 183; the two first bacteria are merged by error.

(iv) To correct the segmentation errors, select the bacteria that are mistakenly merged from frames 173 to 183 and execute the split command (Step 2C(i))

(v) Correct the lineage information, using the *Link/Unlink* commands (Step 2C(i)).

(vi) Re-run the segmentation of spots, which depends on the segmentation of bacteria, in this microchannel by right-clicking on the *Microchannel* > *Run Segmentation and Tracking* > *Spots*.

(vii) Follow Step 1A(viii) to update measurements.

Troubleshooting

Troubleshooting advice can be found in Table 1.

Table 1 | Troubleshooting table

Step	Problem	Possible reason	Solution
1A(iii)	No positions are created	Incorrect number of channels in the configuration	Set the right number of channels in the configuration; see Step 1D(iii)
		Import method is not adapted to the dataset	Set the import method corresponding to the input images; see Step 1D(vi)
1A(iv) 1B(iv)	Pre-processing is not performed and an 'Image not found' error is displayed	Location of input images has changed	Click on *Import/Re-link images* from the *Run* menu to update the links to input images
	Pre-processing is not performed and an 'Out Of Memory Error' is displayed	Not enough memory available to perform pre-processing	Increase allocated memory (see *Help* for FIJI), or perform pre-processing on a subset of input images by setting the parameters *Trim Frame Start/Stop* in the *Configuration* tab (*Positions > Position Name > Pre-Processing*)
	Closed ends of the microchannels are not located at the top of the image	*AutoFlipY* transformation is not configured properly	Configure the transformation (Step 1D(viii–xiv)). Either change from *AutoFlipY* to *Flip*, or configure *AutoFlipY* according to its documentation
	Pre-processed images are not cropped properly	*CropMicroChannels* transformation is not configured properly	Configure the transformation *CropMicroChannelsPhase2D* or *CropMicroChannelsFluo2D* in the pre-processing pipeline (Step 1D(viii–xiv)) according to its documentation
1A(v & vi) 1B(v–vii)	Errors in segmentation and/or tracking	Processing pipeline is not configured properly	Configure the segmenter, tracker, prefilters and postfilters according to their documentation (see Step 1D(xxii) for information on testing processing modules)
1D(xiii & xxii)	*Frame Range* in the *Configuration Test* tab cannot be changed	*Z* and *T* dimensions were swapped during image import	Modify the *Swap T & Z dimension* parameter, and click on *Import/Re-link image*s to re-run the command

Timing

Step 1A, analysis of a phase-contrast mother machine time-lapse experiment: 27–40 min
Step 1B, analysis of a mother machine time-lapse experiment with fluorescent cells and intracellular spots: 27–40 min
Step 1C, adaption of an existing configuration to another dataset: 15 min
Step 1D, creating and configuring a new dataset: 35 min–2 h 15 min
Step 2A, visualization of segmentation and tracking results: 1–5 min
Step 2B, creation, display and navigation of selections: 1–5 min
Step 2C, editing segmentation and tracking: 6–17 min

Anticipated results

BACMMAN offers a solution for fast and reliable analysis of high-throughput data from video-microscopy experiments in which bacteria grow in a mother machine microfluidic chip. It can be used to follow cells in phase contrast or in fluorescence, to quantify single-cell fluorescence as well as to detect and follow intracellular fluorescent spots. Therefore, it should be useful in the study of many biological processes. After analysis with BACMMAN, pre-processed images are saved and can be stored in place of the raw images for a gain of space. BACMMAN can perform several user-defined measurements related to cell morphology, growth and fluorescence intensity and to intracellular fluorescent spot characteristics. The principal measurements are listed in Supplementary Table 1. BACMMAN allows an easy interplay between data analysis with R or Python and image analysis, facilitating the study of cell subpopulations. All measurements can be performed on either all objects, i.e., cells or spots, or on a subset of objects, through the 'selection' procedure. Measurements are saved in tables that can be directly imported into software such as R, Python or MatLab for data analysis. In these tables, each object—for instance, a bacterium or a spot detected at a given frame—can correspond to different measurements and is identified by a unique barcode'. For instance, a cell

segmented at a given frame is uniquely identified by a series of numbers corresponding to the index of the frame, the index of the microchannel in which the cell is found, and the location of the cell in the microchannel at this frame. This kind of identifying number allows easy manipulation with data analysis software such as R or Python. A cell or a spot can therefore be easily followed through time, with all its measurements, such as size and fluorescence intensity. The tables also contain a code indicating the position of the cells in the lineage. Therefore cell measurements can be followed along a specific cell lineage, such as the 'old pole' lineage, in which the mother cell, abutting the channel's dead end, is followed at division. The different measurements can also be studied separately for each microchannel. Therefore the results obtained with BACMMAN allows all kinds of analyses, with several measurements that can be followed on different subsets of objects. Some examples of analysis and a description of the different variables in the tables can be found in the R script provided with dataset 1.

Reporting Summary

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

Data availability

The example datasets used in this protocol are available at the zenodo.org repository (https://doi.org/10.5281/zenodo.3243467).

Code availability

Source code for BACMMAN is available at github.com/jeanollion/bacmman under GNU General Public License v3.0.

References

1. Elowitz, M. B., Levine, A. J., Siggia, E. D. & Swain, P. S. Stochastic gene expression in a single cell. *Science* **297**, 1183–1186 (2002).
2. Davie, K. et al. A single-cell transcriptome atlas of the aging Drosophila brain. *Cell* **174**, 982–998.e20 (2018).
3. Navin, N. et al. Tumour evolution inferred by single-cell sequencing. *Nature* **472**, 90–94 (2011).
4. Muzzey, D. & van Oudenaarden, A. Quantitative time-lapse fluorescence microscopy in single cells. *Annu. Rev. Cell Dev.* **25**, 301–327 (2009).
5. Mehling, M. & Tay, S. Microfluidic cell culture. *Curr. Opin. Biotechnol.* 25, (95–102 (2014).
6. Wang, P. et al. Robust growth of *Escherichia coli*. *Curr. Biol.* **20**, 1099–1103 (2010).
7. Taheri-Araghi, S. et al. Cell-size control and homeostasis in bacteria. *Curr. Biol.* **25**, 385–391 (2015).
8. Tanouchi, Y. et al. A noisy linear map underlies oscillations in cell size and gene expression in bacteria. *Nature* **523**, 357–360 (2015).
9. Norman, T. M., Lord, N. D., Paulsson, J. & Losick, R. Memory and modularity in cell-fate decision making. *Nature* **503**, 481–486 (2013).
10. Brenner, N. et al. Single-cell protein dynamics reproduce universal fluctuations in cell populations. *Eur. Phys. J.* **38**, 102 (2015).
11. Kaiser, M. et al. Monitoring single-cell gene regulation under dynamically controllable conditions with integrated microfluidics and software. *Nat. Commun.* **9**, 212 (2018).
12. Robert, L. et al. Mutation dynamics and fitness effects followed in single cells. *Science* **359**, 1283–1286 (2018).
13. J. Ollion, M. Elez, & L. Robert. Example datasets for BACMMAN software. https://doi.org/10.5281/zenodo.3243467 (2019).
14. Elez, M. et al. Seeing mutations in living cells. *Curr. Biol.* **20**, 1432–1437 (2010).
15. Robert, L., Ollion, J. & Elez, M. Real-time visualization of mutations and their fitness effects in single bacteria. *Nat. Protoc.* https://doi.org/10.1038/s41596-019-0215-x (2019).
16. Schindelin, J. et al. Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012).
17. Russell, J. R., Cabeen, M. T., Wiggins, P. A., Paulsson, J. & Losick, R. Noise in a phosphorelay drives stochastic entry into sporulation in Bacillus subtilis. *EMBO J.* **36**, 2856–2869 (2017).
18. Long, Z. et al. Microfluidic chemostat for measuring single cell dynamics in bacteria. *Lab Chip* **13**, 947–954 (2013).
19. Young, J. W. et al. Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy. *Nat. Protoc.* **7**, 80–88 (2012).
20. Mekterović, I., Mekterović, D. & Maglica, Z. BactImAS: a platform for processing and analysis of bacterial time-lapse microscopy movies. *BMC Bioinforma.* **15**, 251 (2014).

21. 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**, 612–627 (2011).
22. Sachs, C. C. et al. Image-based single cell profiling: high-throughput processing of mother machine experiments. *PLoS ONE* **11**, e0163453 (2016).

Acknowledgements
We acknowledge J. Elf (Uppsala University), S. Uphoff (Oxford University), H. Salman (Pittsburgh University) and anonymous reviewer 4 for kindly providing datasets from their labs that allowed us to test the applicability of our software. This work was funded by the Agence Nationale de Recherche (grant ANR-14-CE09-0015-01 to M.E.) and by the city of Paris (program Emergences 2018 to M.E.).

Author contributions
All authors wrote the article and contributed to tests and documentation. J.O. developed the software; L.R. and J.O. analyzed the software performance; L.R. generated dataset 1; and M.E. generated datasets 2 and 3.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information is available for this paper at https://doi.org/10.1038/s41596-019-0216-9.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to J.O.

Peer review information *Nature Protocols* thanks Thomas Julou, Christian C. Sach and other anonymous reviewer(s) for their contribution to the peer review of this work.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 27 July 2018; Accepted: 21 June 2019;
Published online: 25 September 2019

Related links
Key references using this protocol
Robert, L. et al. *Science* **359**, 1283–1286 (2018): http://science.sciencemag.org/content/359/6381/1283
Robert, L., Ollion, J. & Elez, M. *Nat. Protoc* (2019): https://doi.org/10.1038/s41596-019-0215-x.

Key data used in this protocol
Robert, L. et al. *Science* **359**, 1283–1286 (2018): http://science.sciencemag.org/content/359/6381/1283

nature research

Corresponding author(s): Jean Ollion

Last updated by author(s): Mar 27, 2019

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☒	☐	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	images were collected using Softworks (commercial software) for µMA experiments and Nis (commercial software) for MV experiments
Data analysis	images were analyzed using BACMMAN (open source software described in this publication), analysis results were analyzed using R (open source software)

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

example datasets are available on zenodoo server (doi:10.1234/bacmman.images)

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	entire µMA and MV experiments from Robert et al. Science 2018
Data exclusions	no data was excluded prior to analysis
Replication	software was used to analyze at least 3 independent µMA and MV experiments from Robert et al. Science 2018, as well as data from 5 other labs. Two of these datasets are publically available, so called Molyso and MoMa datasets from the groups of Dietrich Kohlheyer (Forschungszentrum Julich GmbH) and Eric van Nimwegen (University of Basel), respectively. Three other datasets come from the groups of Johan Elf (Uppsala University), Stephan Uphoff (Oxford University), Hanna Salman (Pittsburgh University))
Randomization	this publication describes a software, thus there is no life science study design
Blinding	this publication describes a software, thus there is no life science study design

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging