High-Throughput Detection and Tracking of Cells and Intracellular Spots in Mother Machine Experiments

Jean Ollion, Marina Elez, Lydia Robert

High-Throughput Detection and Tracking of Cells and Intracellular Spots in Mother Machine Experiments

Authors: Jean Ollion,[1*] Marina Elez,[1,2] Lydia Robert [1,3]
Affiliations: 1 Laboratoire Jean Perrin, UMR 8237 Sorbonne Universités, UPMC Université Paris 06, 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
*Corresponding author: jean.ollion@polytechnique.org

Abstract

The analysis of bacteria at the single cell level is essential to characterize 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 1000 frames (2GB dataset) image processing takes about 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 last 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 have been extensively used to investigate cellular processes non-invasively and dynamically in single cells [4]. In order to obtain precise and reproducible measurements, time-lapse imaging is now being increasingly

used in combination with microfluidic devices that allow a precise spatio-temporal 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 [6]. It consists of thousands of parallel dead-end microchannels where 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, addressing various topics such as cell growth [6], cell cycle control[7], [8] or 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 to study mutagenesis and evolution in bacteria [12].
Long-term imaging of cells growing in the mother machine produces a large amount of data, typically several hundreds of Gb of images per experiment. Therefore data managing, image processing and analysis very rapidly becomes a rate-limiting step and requires the implementation of a fast and reliable solution for automated image analysis.
To address this challenge, we recently developed a software dedicated to mother machine data analysis [12]. This software, that we call BACMMAN (BACteria Mother Machine ANalysis), 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 MV approach (for Mutation Visualisation), that is based on time-lapse imaging, mother machine and expression of fluorescently labelled Mismatch Repair protein MutL [14]. The latter makes foci on such DNA sites and thus allows visualizing mutations as bright fluorescent spots inside the cells. µMA (for microfluidic Mutation Accumulation) also based on on time-lapse imaging and mother machine allows following fitness evolution during mutation accumulation and characterizing the distribution of fitness effects of spontaneous mutations. Detailed description of these two approaches and instructions on how perform them can be found in an accompanying protocol [15]
BACMMAN was designed to also work on datasets other than MV and µMA. In order to facilitate the adaptation of BACMMAN algorithms to data from other labs, we developed specific test procedures allowing easy parameter optimization for all the different steps of preprocessing, segmentation and tracking. We also created a detailed documentation for the different algorithms and functions of BACMMAN that 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 very easily installed through Fiji [16]. It has a user-friendly Graphical User Interface (GUI) and can also be run on 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. In order to help

the user discover the software and all its functionalities we have made available two small datasets of typical MV and µMA data [13]. The 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 steps 1A or 1B, the configuration file is provided with the images and can be used without modifications. In 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 described in step 1C should most of the time be sufficient to analyze experiments similar to those presented in Robert el 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 is required that consists 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 S2). 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 have to 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 provided here (see EQUIPMENT). 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 which 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 and tracking, and post-filtering. Details of the algorithms used for the example datasets are provided in Supplementary Notes (section *Image Analysis)* and Supplementary Figures 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 where images from successive frames are displayed next to one another, as shown in Figure 2. Kymographs are navigable using shortcuts (see 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 (See 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 methods” 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 possibility to detect subset 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 in BACMMAN and displayed on kymographs for visualization and editing. Examples of scripts to create selections are provided with our software (See EQUIPMENT). In order to illustrate the benefit of using selections, consider our work in Robert et al. 2018 [12], where 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 error. 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 to facilitate the understanding of specific behaviour 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. 2018 [12]. More specifically, it was designed to allow **segmentation and tracking of *E. coli* cells growing inside the microchannels of the mother machine, imaged either in phase-contrast or in 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 where 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 in [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 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 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 provided in the datasets 1 and 2 [13], but can be combined and configured to adapt the processing to other types of experiments, as illustrated in steps 1C and 1D. In order to test the applicability of BACMMAN to other datasets, we collected datasets from six other labs (including the publicly available datasets provided with MoMA and Molyso software). We were able to analyze these datasets by only changing a few parameters in the configurations provided with our datasets 1 and 2 [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 [13] could be used on this dataset without any modification. In the case of the MoMA dataset, the aspect of the bacteria differs significantly 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 BACMMAN's wiki (github.com/jeanollion/bacmman/wiki). Importantly, we implemented specific test procedures in BACMMAN to facilitate the configuration, which 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 tuning the parameters.
Moreover, new modules can be developed and easily plugged-in in order 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 where the microchannels have two open ends [18] would only require 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 BACMMAN's wiki. Those features make BACMMAN a very flexible tool, however, in order to use it on a dataset that differs from the example datasets, some optimisation and adjustments as well as the development of new modules may be necessary.

Performance and Comparison with other software

Many software 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 the mother machine. Only two software are specifically dedicated to mother machine data analysis, named 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 Figure 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 where 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 also 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 either by 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. Tracking algorithms of MoMA and molyso 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. In contrast, the algorithms of MoMA and molyso 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 in BACMMAN.

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

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

In order 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**, section *Precision of cell size measurement*). As shown in figure 4b, the relative error in cell length is independent of cell length and is around 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, 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 percent) when segmentation parameters are varied. When varying the relevant parameters in a reasonable range, we found that the relative difference between the results of phase contrast and fluorescence segmentation was always smaller than 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 images is good enough. For dataset 1 [13], no errors are made in more than 99.8% of the 10418 generations, with most errors clustered when cells exhibit transient morphological abnormalities. It took 11 minutes to an experienced user to visualize and correct the whole dataset. In dataset 2 [13], no errors were made in more than 99.9% of the 931 generations and visualization and editing took about 3 minutes.

Materials

EQUIPMENT

Data files: The format of input images should be supported by BioFormats (openmicroscopy.org/bio-formats/) (such as .nd2, .tiff, .dv etc...). 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 BACMMAN's different procedures, three different datasets, associated configuration files and analysis scripts are available at: 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 steps 1B and 2C. It consists of a time series of 500 images, corresponding to one position and 2 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.

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 has to 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 4.4Gb.

EQUIPMENT SETUP

Installation under FIJI: TIMING 10-15 min

1. Download and install FIJI 64-bit version (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 3-4 until the message "Your Fiji is up to date!" is displayed.
6. Re-run the updater, click Advanced Mode 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: TIMING 2-5 min CRITICAL Requires Git and Maven 3 and Java 8 or further. Run the following bash commands (replace <VERSION> by the actual version):

1. git clone https://github.com/jeanollion/bacmman.git
2. cd bacmman
3. mvn clean dependency:copy-dependencies package
4. cd bacmman-ij1_/target
5. java -cp dependency/*:bacmman-ij1_-<VERSION>.jar bacmman.binding.IJ1

Run in headless mode: TIMING 2-15 min

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* in order to generate several files, each one containing all the tasks of a given position. This allows 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 bacmman-headless/target directory: (replace <VERSION> by the actual version and <taskfile.json> by the path to a .json task file generated in 3):
   java -cp dependency/*:bacmman-headless-<VERSION>.jar
   bacmman.ui.ProcessTasks <taskfile.json>

Procedures

1) Follow option 1A to analyse a phase contrast mother-machine time-lapse experiment; follow option 1B to analyse a mother-machine time-lapse experiment with fluorescent cells and intracellular spots; follow option 1C to modify an existing configuration in order to analyse a dataset similar to datasets 1 or 2; follow option 1D to create a configuration de novo in order to analyse a dataset that differs

substantially from the provided example datasets.

1A. Analysis of a phase contrast mother-machine time-lapse experiment TIMING 2-5 min for i-iii ; 15-20 min for iv-viii ; 10-15 min for ix-xvi**.**

(i) **Create a dataset and import images.** Start BACMMAN (see EQUIPMENT SETUP). First create an empty folder. To define this folder as 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" (see EQUIPMENT) that corresponds to dataset 1 and enter "dataset1" as 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 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 next BACMMAN startup, this dataset will not be opened automatically again. To access the dataset at next startup, select this dataset it 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.
TROUBLESHOOTING
CRITICAL STEP When images are imported, their path is 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* command.
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.*

(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*.
TROUBLESHOOTING
CRITICAL STEP In this step and (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 Batch processing box 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).

(v) Select in the *Objects* list the *Microchannels* object class and in the *Tasks* list *Segment and Track*. From the *Run* menu choose *Run Selected Tasks*. Visualize the results following steps 2A(i)-(v). TROUBLESHOOTING
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).

(vi) Repeat step (v), selecting “Bacteria” instead of “Microchannels”. Visualize the results following steps 2A(i)-(v), and if necessary edit segmentation and tracking errors as explained in procedure 1C. TROUBLESHOOTING
CRITICAL STEP: When processing several object classes separately, such as proposed in steps (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 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.*

(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, and 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 semicolons separated .csv file. One file will be created for each object class selected in the *Objects* panel, 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 (See EQUIPMENT) 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 the block 2 of the script and set the actual path of the folder, then run the 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 *Reload Selections* button in the *Selections* panel.

(xiv) See step 2B(iii)-(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 steps 1A(vii)-(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

1B. Analysis of a mother-machine time-lapse experiment with fluorescent cells and intracellular spots TIMING 2-5 min for (i)-(iii) ; 15-20 min for (iv)-(viii) ; 10-15 min for (iX-Xiii).

CRITICAL: A screencast for this procedure is available on BACMMAN's wiki (github.com/jeanollion/bacmman/wik/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 (see EQUIPMENT) and enter "dataset2" as name of the dataset.

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

(iv) Run image processing steps. Follow steps 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* TROUBLESHOOTING

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).

(viii) Follow steps 1A(vii)-(viii)

CRITICAL STEP: You will sometimes need/wish to check and manually edit segmentation and tracking results. For this follow the instructions given in 2A and 2C(i).

(ix) Perform statistical Analysis. Open the analysis script 2 (See EQUIPMENT) 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 the block 2 of the script and set the actual path of the folder, then run the block 2 to import the measurements.

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

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

(xiii) Follow steps 1A(xiii)-(xv)

2C. Adaption of an existing configuration to another dataset TIMING 5 min for steps i-iii ; 5 min for step iv ; 5 min for step v-vi

CRITICAL : a tutorial describing how to adapt an existing configuration to datasets from other

labs is available on BACMMAN's 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 (see EQUIPMENT) and enter "dataset3" as name of the dataset.

(iii) From the *Run* menu choose *Import/re-link Images* and select the folder containing 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 T and Z dimension are swapped, this can be fixed by setting the parameter named "Swap T & Z dimensions" of the import method and re-importing the images through the command *Import/re-link images*.

(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 the dataset 3, the fluorescence exhibits a much higher average level and a much higher cell-to-cell variability compared to the dataset 2. In consequence, image processing has to 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, select Processing Pipeline: *SegmentAndTrack* > Tracker:*BacteriaClosedMicrochannelTrackerLocalCorrections* > Segmentation Algorithm:*BacterialFluo* and set the value of the parameter *Foreground Selection Method:EDGE FUSION* > *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 (menu *Dataset* > *Save Configuration changes*) before processing.

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

2D. Creating and configuring a new dataset TIMING 5 min for i-vii ; 10-60 min for xiii-xv ; 10-60 min for xvi-xxii ;10 min for xxiii-xxv.

CRITICAL If users want to analyse a time-lapse dataset that differs from those described in 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, where only the most important parameters can be changed and non technical help is provided, and an “advanced” mode where 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 choose *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 the keyword allowing finding the image files corresponding to the appropriate channel; refer to the help of the *Channel Keyword* parameter.

(v) Repeat steps (iii) and (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 won’t be imported.

(vi) Second, set the import image method by right-clicking on *Import Method.* You should choose here one of 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 of 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 T and Z dimension are inverted, this can be fixed by setting the parameter named “Swap T & Z dimensions” of the import method and re-importing the images through the command *Import/re-link images*.

(viii) **Configure the preprocessing 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), 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 calibration is incorrect, set it by right-clicking on *Voxel Calibration,* choosing *Custom Calibration*, 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 (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. In order to do so, modify the Frame Range 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 *Copyto template button*.

(xv) Perform pre-processing 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 through right-click, 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 respectively *Segmentation Parent* and *Parent*) 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 in module configuration. In order to know how to

configure the parameters using intermediate images, refer to the help of each parameter displayed in the help panel. Depending on the module, additional data may be displayed on right-click 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. In order 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 the Measurement, choose a module and configure it.

(xxv) Repeat steps (xxi) and (xxii) as many times as needed.

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

2A. Visualization of segmentation and tracking results TIMING ~1-5min

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 the 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.
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 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 through 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 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.

2B. Creation, display and navigation of selections TIMING ~1-5min

(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 is indicated next to the selection name.
CRITICAL STEP: Addition or removal of elements are accessible through shortcuts and will be performed only on selections set as *active.* In order 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 menu *Shortcuts* > *Selections*.

(iii) To display objects (respectively Tracks) of a selection, right-click on the selection and select *Display Objects* (respectively *Display Tracks*) CRITICAL STEP Alternatively, object display can be activated / deactivated through shortcuts on active selections listed in *Shortcuts* > *Selection* (see CRITICAL STEP of step (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 (respectively previous) objects of the selection. If no objects are present in the current kymograph after (respectively before) the current view, executing the command twice will open the next (respectively previous) kymograph containing objects of the selection.

(v) In order 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.

2C. Editing segmentation and tracking TIMING 5-15 min for step 1 ; 1-2 min for steps ii-vii

CRITICAL : A screencast for this procedure is available on BACMMAN's 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) Open a kymograph and set the interactive object class as in step 2A(ii). The currently implemented editing commands are:

Command	**Description**
Merge	Merges the selected objects frame-by-frame, i.e. only the objects from the same frame can be merged but merging can be done simultaneously for objects in different frames.
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 scenarii 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. Additionally, 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 are 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 the dataset2 and go to the data browsing tab

(iii) Open the kymograph of the first microchannel track (see steps 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 (see step 2C(i))

(v) Correct the lineage information using link/un-link commands (see 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

See Table 1 for troubleshooting guidance.

Table 1 : Troubleshooting guidance.

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)
1D(xiii) 1D(xxii)	*Frame Range* in the *Configuration Test* tab cannot	Z and Time dimensions were swapped during image import	Modify the *Swap T & Z dimension* parameter, and re-run the command *Import/Re-link images.*

	be changed		
*1*A(iv) 1B(iv)	Preprocessing is not performed and *Image not found* error is displayed	Localisation of Input images has changed	Run the command *Import/Re-link images* from the *Run* menu to update the links to input images.
*1*A(iv) 1B(iv)	Preprocessing is not performed and *Out Of Memory Error* is displayed	Not enough memory available to perform preprocessing.	Increase allocated memory (see Help of Fiji), or perform preprocessing on a subset of input images by setting the parameters *Trim Frame Start/Stop* in the configuration tab (*Positions>*"Position Name">*Pre-Processing*)
1A(iv) 1B(iv)	Closed-end of microchannels are not located at the top of the image	AutoFlipY transformation is not configured properly	Configure the transformation (see steps 1D(viii)-(xiv)). Either change AutoFlip for Flip, or configure AutoFlipY according to its documentation
1A(iv) 1B(iv)	Preprocessed images are not cropped properly	CropMicroChannels transformation is not configured properly	Configure the transformation CropMicroChannelsPhase2D or CropMicroChannelsFluo2D in the pre-processing pipeline (see steps 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 testing processing modules)

Anticipated Results

BACMMAN offers a solution for fast and reliable analysis of high-throughput data from video-microscopy experiments where bacteria grow in the 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 to study 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 or fluorescence intensity and to intracellular fluorescent spot characteristics. The principal measurements are listed in Supplementary table S1. 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 either on 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 softwares 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 respectively 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 softwares such as R or Python. A cell or a spot can therefore be easily followed through time, with all its measurements, such as size, or fluorescence intensity. The tables contain also 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 where always 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 kind 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 dataset1.

Acknowledgements
We acknowledge Johan Elf (Uppsala University), Stephan Uphoff (Oxford University), Hanna Salman (Pittsburgh University) as well as anonymous reviewer 4 for kindly providing datasets from their labs that allowed us to test the applicability of our software. Funding: This work was funded by the Agence Nationale de Recherche (grant ANR-14-CE09-0015-01 to ME) and by the city of Paris (program Emergences 2018 to ME). Author Contribution: All authors wrote the article and contributed to tests and documentation, JO developed the software, LR and JO analyzed software performances, LR generated dataset 1, ME generated datasets 2 and 3. Competing financial and non-financial interests: The authors declare that they have no competing financial and non-financial interests. Data availability: The example datasets used in this protocol are available a the zenodo.org repository (doi: 10.5281/zenodo.3243467). Code Availability: source code of bacmman is available at github.com/jeanollion/bacmman under GPL-3.0 licence.

Figures

Figure 1: Processing Pipeline

Legend: BACMMAN analysis pipeline couples automated image analysis, statistical analysis and manual visualisation/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, 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 significantly speeds up the visualisation/editing part of the analysis. Subsets of interest can also be modified through the GUI.

Figure 2: Segmentation and Tracking of bacteria and spots in BACMMAN

Legend: Kymographs showing segmentation and tracking for bacteria imaged in phase contrast (top) or fluorescence (middle), and for intracellular fluorescent spots (bottom). Contours of segmented objects are displayed in pink and colored arrows indicate tracking links.

Figure 3: GUI

Legend: The Graphical User Interface (GUI) of BACMMAN is composed of three panels. The home panel (upper left) allows selecting the dataset and running processing tasks on the different positions. The configuration panel (upper right) allows configuring 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 visualizing and editing 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).

Figure 4: Segmentation performances

Legend: 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) or 2D-segmentation (pink contours). b: Estimation of the measurement error on cell size. For each single cell, we performed a linear regression of the log-length vs time and we computed the residual errors. In order to obtain the relative error on cell length, we took the exponential of these residuals and computed their coefficient of variation (see Supplementary Notes *Precision of cell size measurement* for mathematical explanations), which we plotted as a function of cell length. Relative errors were computed for the analysis of both fluorescence (red; 750424 length measurements on 126717 cells) and phase contrast (blue; 551600 measurements, 103400 cells) images. Image processing was performed automatically, with no manual editing, and the data was then filtered as explained in Robert et al.[12] in order to remove cells with aberrant growth rates due to cell death and tracking errors.

Supplementary Figures

Figure S1: Segmentation of fluorescent bacteria

Legend: A: input image, illustrating fluorescence heterogeneity within and between cells. B: Image of edges computed with a sigma transform of the gaussian smoothed image. Along the cell contours, fluorescence intensities are maximal at the interface between cells and background, and minimal at the boundary between cells. C: regions obtained by a watershed algorithm computed on image B. For each region, the median value of the fluorescence intensity within this region in image A is displayed through a color code (see color bar below).

D: Interfaces between the different regions shown in C. For each interface, the color indicates the value of the first decile of the distribution of pixel intensity in image B at this interface. E: Maximum Eigenvalue of the Hessian transform (MEH). MEH is maximal at the interfaces between cells. F: Regions obtained by watershed algorithm computed on E within the foreground region defined using image C . For each region, the color indicates the median intensity of image A within this region. G: mean intensity of image E divided by mean intensity of image A at the interfaces between foreground regions displayed in F. H Final result of the segmentation process.

Figure S2: Spine coordinate system for fluorescent spot tracking

Legende: Red fluorescence images of bacteria in a microchannel at 3 successive frames: F, F+1 and F+2. Magnified images show the spine coordinate system, with the segmented contour of the cell in yellow, the spine in pink and the spine radial directions in blue. The spine coordinates of a spot (red arrow) at frame F appear in red. The same spot is observed at frame F+1 and F+2 (green arrows). At frames F+1 and F+2, the projection of the spot from frame F is indicated with a red arrow and its projected coordinates appear in red. The distance between the location of the spot at frame F and its location at frame F+1 (or between frames F+1 and F+2) is the distance between the green and red arrow.

Figure S3: Bacteria Segmentation in phase contrast images

Legend: A: Input image. B: Pre-filtered image. C: Image of edges computed with a sigma transform on the gaussian smoothed transform of B. D: partitioning obtained by a watershed algorithm computed on C. For each region, the color indicates the median intensity within this region in image B (see color bar below). This shows that a simple thresholding can remove most of the background regions, except small border artefacts that can be removed by filtering out thin objects. E: Maximum eigenvalue of the hessian transform. F: Partitioning obtained by a watershed algorithm computed on E within the foreground region defined from image D. For each region, the color indicates the median intensity within this region in image B. G: mean value of E divided by mean value of B at the interface between foreground regions displayed in F. H: Result of segmentation.

Supplementary Tables

Table S1: Measurements

The following table describes the main measurements computed by BACMMAN. “Module” is the name of the measurement module, and “Column” indicates the column name of the measurement in the exported csv file. The column names are indicated in the popup help window of the module. Note that a user can sometimes change the column name in the configuration of the module (for instance add a suffix).

Table S2 Glossary

Supplementary Notes

References:

[1] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, “Stochastic gene expression in a single cell”, *Science*, vol. 297, no. 5584. American Association for the Advancement of Science, pp. 1183–1186, 2002.

[2] K. Davie *et al.*, “A single-cell transcriptome atlas of the aging Drosophila brain”, *Cell*. Elsevier, 2018.

[3] N. Navin *et al.*, “Tumour evolution inferred by single-cell sequencing”, *Nature*, vol. 472, no. 7341. Nature Publishing Group, p. 90, 2011.

[4] D. Muzzey and A. van Oudenaarden, “Quantitative time-lapse fluorescence microscopy in single cells”, *Annual Review of Cell and Developmental*, vol. 25. Annual Reviews, pp. 301–327, 2009.

[5] M. Mehling and S. Tay, “Microfluidic cell culture”, *Current opinion in Biotechnology*, vol. 25. Elsevier, pp. 95–102, 2014.

[6] P. Wang *et al.*, “Robust growth of Escherichia coli”, *Current biology*, vol. 20, no. 12. pp. 1099–1103, 2010.

[7] S. Taheri-Araghi *et al.*, “Cell-size control and homeostasis in bacteria”, *Current Biology*, vol. 25, no. 3. Elsevier, pp. 385–391, 2015.

[8] Y. Tanouchi *et al.*, “A noisy linear map underlies oscillations in cell size and gene expression in bacteria”, *Nature*, vol. 523, no. 7560. Nature Publishing Group, p. 357, 2015.

[9] T. M. Norman, N. D. Lord, J. Paulsson, and R. Losick, “Memory and modularity in cell-fate decision making”, *Nature*, vol. 503, no. 7477. Nature Publishing Group, p. 481, 2013.

[10] N. Brenner, E. Braun, A. Yoney, L. Susman, J. Rotella, and H. Salman, “Single-cell protein dynamics reproduce universal fluctuations in cell populations”, *The European Physical Journal E*, vol. 38, no. 9. Springer, p. 102, 2015.

[11] M. Kaiser *et al.*, “Monitoring single-cell gene regulation under dynamically controllable conditions with integrated microfluidics and software”, *Nature communications*, vol. 9, no. 1. Nature Publishing Group, p. 212, 2018.

[12] L. Robert, J. Ollion, J. Robert, X. Song, I. Matic, and M. Elez, “Mutation dynamics and fitness effects followed in single cells”, *Science*, vol. 359, no. 6381. American Association for the Advancement of Science, pp. 1283–1286, 2018.

[13] J. Ollion, M. Elez, and L. Robert, “Example datasets for BACMMAN software”, *https://doi.org/10.5281/zenodo.3243467*, 2019.

[14] M. Elez, A. W. Murray, L.-J. Bi, X.-E. Zhang, I. Matic, and M. Radman, “Seeing mutations in living cells”, *Current Biology*, vol. 20, no. 16. pp. 1432–1437, 2010.

[15] L. Robert, J. Ollion, and M. Elez, “Real-time visualization of mutations and their fitness effects in single bacteria”, *Nature protocols*. 2019.

[16] J. Schindelin *et al.*, “Fiji: an open-source platform for biological-image analysis”, *Nature methods*, vol. 9, no. 7. p. 676, 2012.

[17] J. R. Russell, M. T. Cabeen, P. A. Wiggins, J. Paulsson, and R. Losick, “Noise in a phosphorelay drives stochastic entry into sporulation in Bacillus subtilis”, *The EMBO journal*, vol. 36, no. 19, pp. 2856–2869, Oct. 2017.

[18] Z. Long *et al.*, “Microfluidic chemostat for measuring single cell dynamics in bacteria”, *Lab on a Chip*, vol. 13, no. 5. pp. 947–954, 2013.

[19] J. W. Young *et al.*, “Measuring single-cell gene expression dynamics in bacteria using

fluorescence time-lapse microscopy", *Nature protocols*, vol. 7, no. 1. Nature Publishing Group, p. 80, 2012.
[20] I. Mekterović, D. Mekterović, and others, "BactImAS: a platform for processing and analysis of bacterial time-lapse microscopy movies", *BMC bioinformatics*, vol. 15, no. 1. BioMed Central, p. 251, 2014.
[21] O. Sliusarenko, J. Heinritz, T. Emonet, and C. Jacobs-Wagner, "High-throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics", *Molecular microbiology*, vol. 80, no. 3. Wiley Online Library, pp. 612–627, 2011.
[22] C. C. Sachs *et al.*, "Image-based single cell profiling: High-throughput processing of mother machine experiments", *PloS one*, vol. 11, no. 9. p. e0163453, 2016.

Figure 1 Ollion

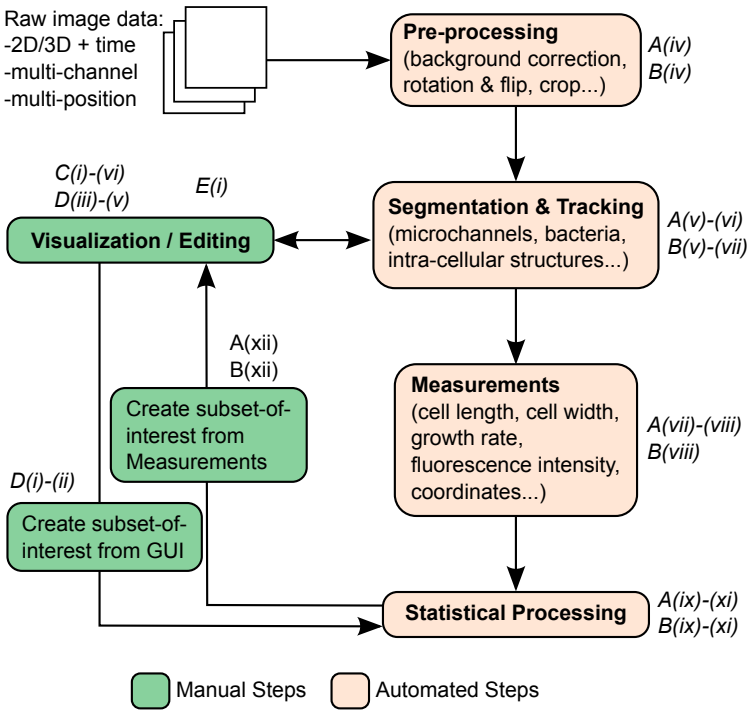


Figure 2 Ollion

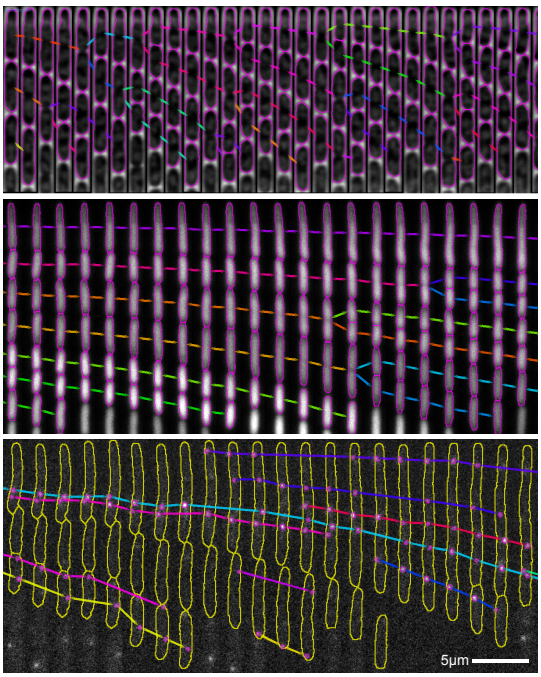
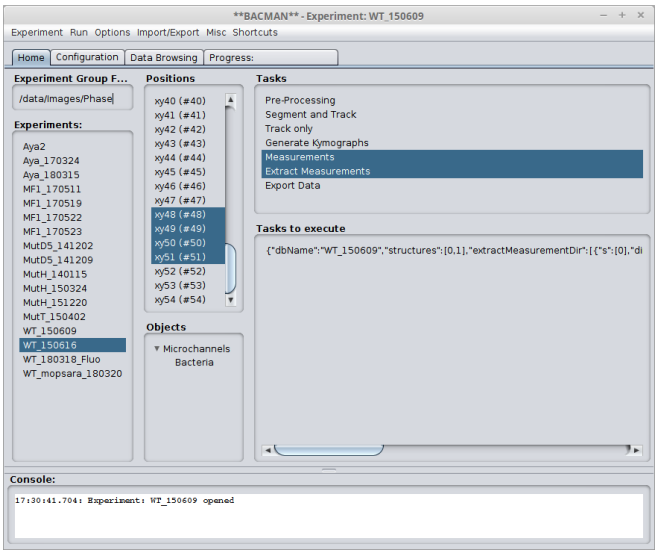
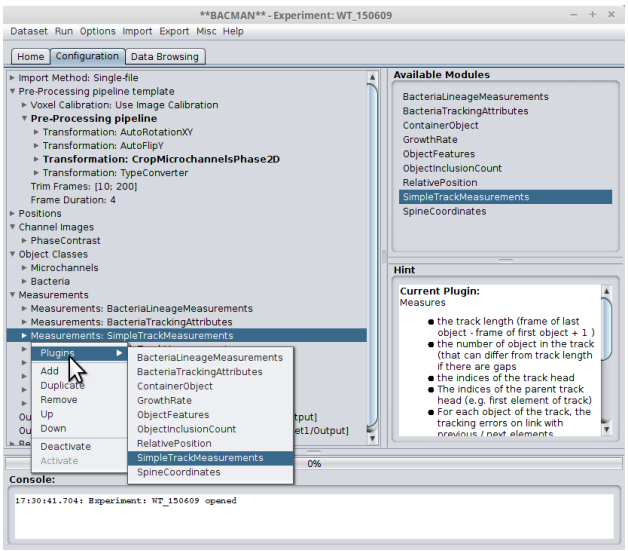
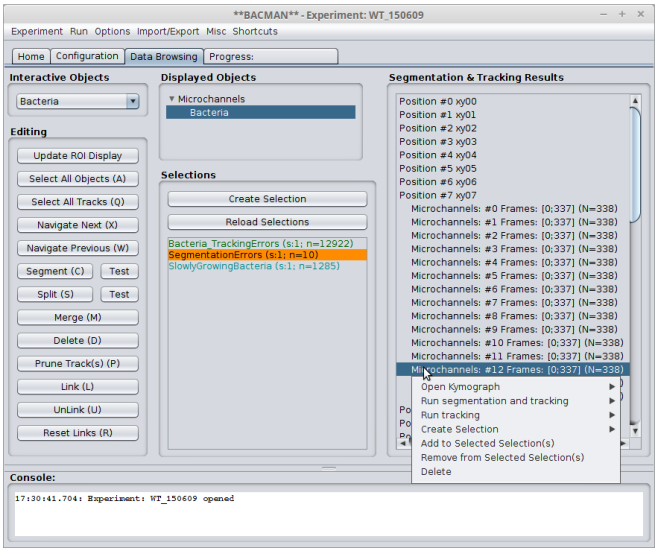


Figure 3 Ollion







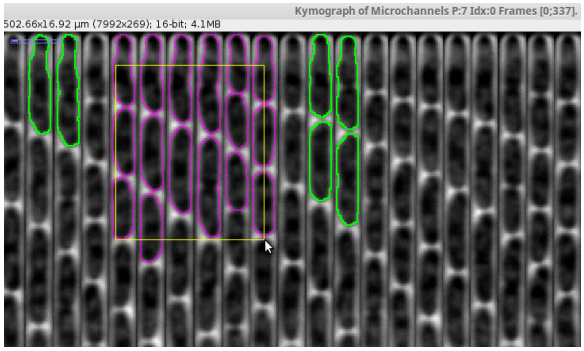


Figure 4 Ollion

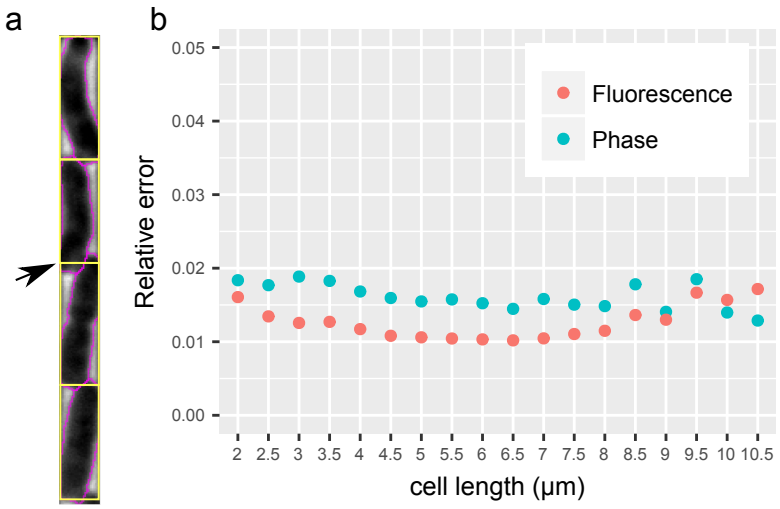


Figure S1 Ollion

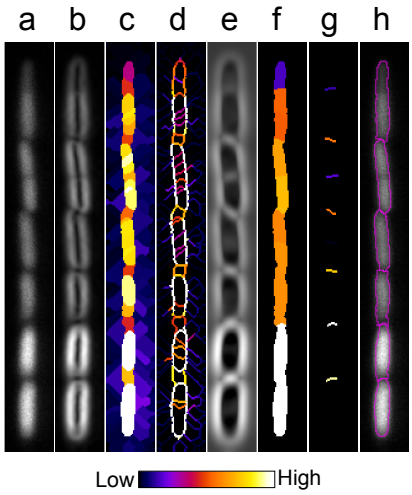


Figure S2 Ollion

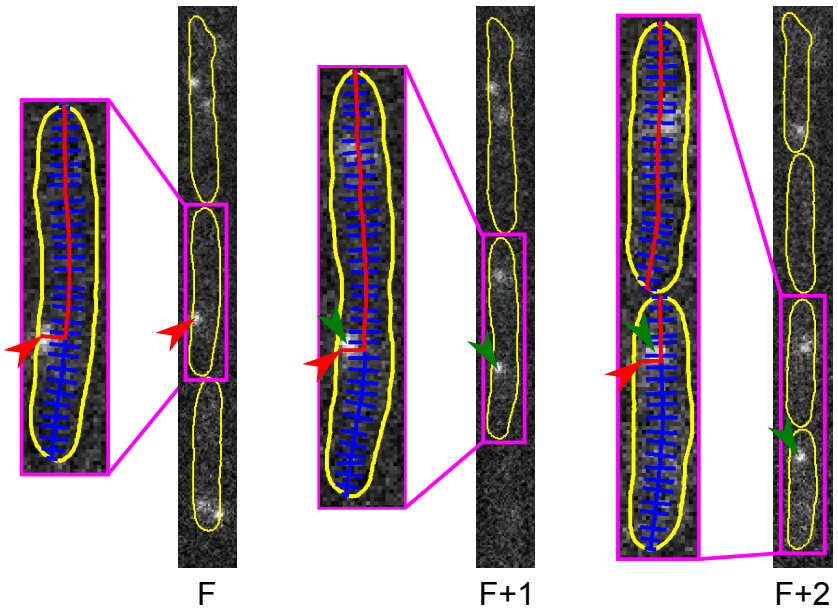
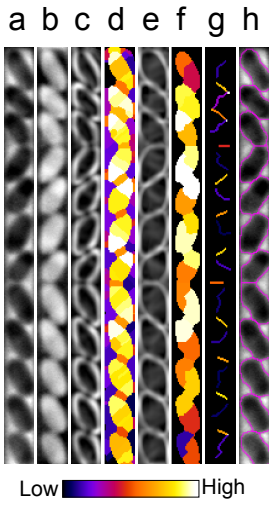


Figure S3 Ollion



Supplementary Notes for High-Throughput Detection and Tracking of Cells and Intracellular Spots in Mother Machine Experiments

Authors: Jean Ollion, Marina Elez, Lydia Robert

Software

BACMMAN was developed in Java language and based on several Java-based libraries ([1], [2]). It can either be run within the FIJI environment[2] or as a stand-alone program (see main text). It is distributed under licence GPL-3.0 and source code is available at github.com/jeanollion/bacmman.

Image Analysis

BACMMAN can be used to segment and track bacteria either in phase contrast or in fluorescence, and to segment and track intracellular fluorescent spots. Here we will explain the different segmentation and tracking strategies implemented in the software, beginning with the case where bacteria (and spots) are imaged in fluorescence, and then describing the segmentation and tracking of bacteria in phase contrast.

1. Fluorescence Experiments

In our Mutation Visualization experiments, presented in[3], Bacteria are visible in the Red fluorescence channel (RFP) and fluorescent spots in the Yellow fluorescence channel (YFP). For simplicity, in the following we assume bacteria are imaged in the RFP channel and spots in the YFP channels. This is not a constraint for the analysis and any fluorescence channel can be used.

1.1. Estimation of background fluorescence

In several processing steps, the background fluorescence needs to be separated from the signal of interest. We adapted a method described Panier et al.[4], which is based on the estimation of the background fluorescence standard deviation σ_{noise}. In our images, many pixel values correspond to background fluorescence and the mode of the fluorescence intensity distribution corresponds to the mean background value μ_{noise}. We assume a gaussian distribution for the background fluorescence and estimate σ_{noise} from the shape of the distribution of fluorescence intensity, truncated from above μ_{noise}.

1.2. Pre-processing

First, for each YFP or RFP image, banding noise is removed by subtracting for each pixel line the mean pixel value within the background region. Background regions are defined in

the RFP images as region of pixels with a value smaller than $\mu_{noise} + 5 \times \sigma_{noise}$. μ_{noise} and σ_{noise} are estimated on the whole time-serie of RFP images as described above. The RFP fluorescence of single-cells exhibits some heterogeneity, with rare cells reaching a fluorescence that is 5- to 20-fold higher than the average value, potentially perturbing further image processing. To solve this problem we impose a saturation threshold τ_{sat}. A typical value for cell fluorescence μ_{cell} is estimated as the median of the image fluorescence values that are above a threshold τ_{cell} , defined as $\tau_{cell} = \mu_{noise} + 10 \times \sigma_{noise}$. The saturation threshold is then defined as $\tau_{sat} = \mu_{noise} + 10 * (\mu_{cell} - \mu_{noise})$. In order to align the microchannels along the y-axis, with the dead-end at the top, both YFP and RFP images are automatically rotated using an algorithm based on the radon projection: all pixels are projected on an axis forming a given angle with the x-axis, then the variance of the values projected along the axis is computed. The axis that is perpendicular to the microchannels is the one that corresponds to the largest variance.

1.3. Segmentation and tracking of micro-channels

In fluorescence, the PDMS structures are not visible. Therefore, the microchannels are detected using the RFP fluorescence of the bacteria growing inside them, by a simple thresholding with a threshold $\tau = \mu_{noise} + 10 \times \sigma_{noise}$, with μ_{noise} and σ_{noise} computed as described in 1.1. The segmentation of the microchannels is done for each frame, then the detected microchannels are tracked over the frames based on a global minimization of microchannel motion between successive frames, thanks to the TrackMate[5] implementation of the algorithm developed by Jaqaman et al.[6]. After tracking, the widths and heights of each microchannel at different frames are equalized by replacing the value at each frame by the median value for all frames.

1.4. Detection of bacteria

Our detection strategy is divided into three steps: i) global detection of the foreground regions, i.e. the fluorescent cells, ii) separation of the cells and iii) adjustments of cell contours.

i) First, edges are detected in the images by applying a local sigma filter on the gaussian-smoothed images (see figure S1B). Sigma filter is chosen instead of gradient intensity because it does not involve derivation and is thus more robust to noise. Then the foreground signal is detected by partitioning the pixels within the segmented microchannels using a watershed transform performed on the image of edges. When fluorescence levels are very heterogeneous among or within cells, it is not possible to define one single threshold to select foreground regions, because background regions that are next to highly fluorescent cells can have a higher fluorescence level than regions located inside weakly fluorescent cells. In order to overcome this problem, we merge adjacent regions using a criterion based on the values of the edge image at the interface between the regions. More precisely, we compute the distribution of the pixel intensity at the interface, calculate its first decile and normalize it by σ_{noise} (see figure S1D). If this value is lower than a threshold then the two adjacent regions are merged. All regions that verify this criterion are merged, starting with the partitions with the lowest criterion values. Then the partitions that are in contact with the microchannel border are eliminated, leaving only foreground regions are removed.

ii) Procedure i) does not allow a good separation of bacteria because the edges computed using the sigma filter are maximal at the boundary between foreground and background but

reach a local minimum at the boundary between two bacteria (see figure S1B). On the contrary, the maximal eigenvalue of the hessian transform (MEH), which represents the curvature of the fluorescence intensity, reaches a maximal value at the boundary between two bacteria (figure S1E). Thus to separate bacteria, we apply a split-and-merge approach on the previously detected foreground. A watershed transform is first performed on the MEH within the previously detected foreground (figure S1F). When several local minima of the MEH are present in a single bacterium, it is fragmented accordingly. In order to reduce this over-segmentation, some of the adjacent fragments are then merged. We estimated the likelihood that two adjacent fragments belong to two distinct cells as the mean value of the MEH at the interface divided by the mean value of the gaussian smoothed transform at the interface (this variable is further called SC for Split Criterion, see figure S1G). Interfaces are sorted according to SC value and merged successively if the value is under a predefined threshold, called *Split threshold*. Importantly this threshold should be calibrated for each new experimental setup.

iii) Procedure i) does not allow a good detection of cell contours near the boundary between two cells. To do so, a threshold $T_{contour}$ is computed for each cell based on the pixel intensity distribution within the cell. $T_{contour}$ is defined as $T_{contour} = m - \theta * IQ$, where m is the median of the distribution, IQ is its interquartile and θ is a threshold we set to 1.25 for our example datasets. Each pixel of the cell contour is eliminated if its intensity is smaller than $T_{contour}$. In this case, the same procedure is applied to the neighboring pixels, until all pixels in the contour have an intensity higher than $T_{contour}$ (figure S1H). Importantly the threshold θ (called *Local Threshold Factor*) should be calibrated for each new experimental setup. Note that as explained above, the size of the segmented bacteria can slightly vary with the segmentation parameters, in particular the θ and *Split* thresholds. Therefore, for users who need to measure cell size precisely, a calibration with independent measurements may be necessary.

1.5. Tracking of bacteria

We developed a tracking algorithm that allows local correction of segmentation errors using previous and next frames. In the dead-end microchannels, the rank of the cells within one channel is directly linked to their position in the lineage (cells can only get out of the channel from the opened end and two cells cannot exchange position), with the old pole mother cell abutting the dead end. Tracking is performed based on the rank of the cell and on its size ratio (SR) between successive frames.

When segmentation is error-free, one cell at frame F can be either linked to one cell at frame F+1 or to two cells, if division occurred between F and F+1. In order to decide between these two scenarios, we compute the SR in both cases, where SR is defined as the sum of the sizes of the daughter cells divided by the size of the mother cell in the case of a division event. Then we choose the scenario in which the SR is the closest to its expected value, which is defined as the median of the 20 previous SR values in the cell lineage (with the exception of the first frames, for which a constant range of 0.85-1.5 is used).

However, our detection strategy can lead to rare errors when a cell is in the middle of the division process and the two daughter cells are not clearly separated yet. In this case, the two daughter cells may be identified as separated at frame F and fused at frame F+1. This segmentation error also leads to a tracking error. We implemented a procedure to automatically correct both the segmentation and tracking errors in such cases. To do so, we allow more than one cell at F to be linked to more than two cells at F+1 to optimize the SR.

When tracking errors are present, i.e. more than one cell at frame F are linked to a single cell at F+1 or a single cell at F is linked to more than two cells at F+1, different correction scenarios are compared. For instance if two cells at F are linked to one cell at F+1, the cells are either merged on previous frame(s) (until previous division) or split on next frame(s) (until next division). In some scenarios, some residual tracking errors may remain. A scenario is accepted only if it reduces the number of errors compared to the initial situation. When two scenarios lead to the same number of errors, a cost is computed for each of them, which sums the cost of each merge/split operation that is performed. Only scenarios with a cost smaller than a constant threshold are compared, and the scenario with the smallest cost is then chosen.The cost of a merge / split operation is the difference between the SC criterion and its predefined threshold (see detection of bacteria in previous section). This procedure allows correcting simultaneously the segmentation errors and the tracking errors they generate.

1.6. Detection of spots

In order to reduce the effect of cell-to-cell variability in intracellular fluorescence level, the distribution of pixel intensity inside each cell is normalized by subtracting its mean and dividing by its standard deviation. Intracellular spots are then detected using a seeded watershed algorithm on the Laplacian-of-Gaussian (LoG) transform. Seeds are determined as regional maxima in the LoG image, and filtered using a threshold on a quality parameter defined as the square root of the product of the LoG value and Gaussian value.

1.7. Tracking of spots

We use the TrackMate[5] implementation of the algorithm developed by Jaqaman et al.[6], allowing efficient simultaneous tracking of several spots and including a procedure for gap-closing, in case some spots are not detected during one or two frames. Our tracking procedure includes a correction for cell motion and a procedure to reduce false positive detection events.

Between two successive frames, a spot is subject to 3 sources of motion : 1) its intrinsic motion within the cell, 2) the motion of the cell containing the spot, when it swims within the microchannel or when it is pushed by growing cells that are closer to the dead-end 3) the growth of the cell containing the spot. In order to remove sources 2 and 3, the position of the spot is expressed in an appropriate coordinate system, described below.

We define the “spine” of the bacterium as the central line crossing it from one pole to the other. Each point of the spine is equidistant from the two closest points of the contour located on each side of the spine. The distance between the two poles along the spine is referred to as spine length, and is larger than the euclidean distance if the cell is not straight. The spine coordinate system is composed of the curvilinear distance along the spine and the distance from the spine in the direction perpendicular to the spine, referred to as radial distance.This coordinate system allows suppressing the source 2) of motion. To compute the distance between a spot (S_F) contained in a cell (C_F) at frame F and a spot (S_{F+1}) contained in a cell at frame F+1 (C_{F+1}), S_F is projected in C_{F+1} and the euclidean distance between S_{F+1} and the projection of S_F is computed. To project S_F in C_{F+1}, we assume homogeneous growth and calculate the curvilinear coordinate as the curvilinear coordinate of S_F multiplied by the ratio of spine lengths of the two cells C_F and C_{F+1}. In case of division, the spine length at F+1 is the sum of the spine lengths of the two daughter cells (see Figure S2 where a

division occurs between frames F+1 and F+2)

In order to reduce the number of false positive detection events, detected spots are divided into two categories, low quality and high quality spots, according to the quality parameter defined in the previous section (detection of spots). Low quality spots are taken into account only if they can be linked to a high quality spot during tracking. To achieve this, first all spots are linked together, then tracks containing only low quality spots are removed.

2. Phase Contrast Experiments

2.1. Pre-processing

The images are first rotated using the radon projection, as for fluorescence images. However here the detected structure is not the microchannels but the edge of the main channel (where it connects to the microchannels), which appears as a particularly bright region in phase contrast. When the projection axis is perpendicular to the edge of the main channel, i.e. parallel to the microchannels, the maximum along the projection axis reaches a maximum. After rotation, images are cropped above the bright linear region created by the edge of the main channel.

2.2. Detection and tracking of microchannels

The end of the microchannels is detected as the maximum in the y-derivative image averaged in the x-direction. To detect the microchannel sides, the image x-derivative is averaged in the y-direction and the sides are detected as a maximum followed by a minimum, these extrema being separated by a distance close to the microchannel width. Only extrema over a predefined threshold are considered.

After this procedure, microchannels are segmented as rectangles. However, some microchannels can have a rounded shape at the dead-end. In order to precisely fit the shape of each microchannel, a seeded watershed algorithm is applied on an edge image computed by applying a local sigma filter on a gaussian-smoothed image. All regions that contained local minima located within the bounds of the microchannels are then merged and form the new segmented microchannel. Segmentation is done at each frame, then the detected objects are tracked over frames, using the same algorithm as in 1.3.

2.3. Detection and tracking of Bacteria

All the steps described below are performed within the mask of the segmented microchannels. First, background is removed by applying the subtract background sliding paraboloïd algorithm distributed with ImageJ[1]. In order to have homogeneous background reduction, the algorithm is applied for each microchannel on an image formed by the concatenation of the images of all frames, one microchannel's tail being joined to the next microchannel's head. The concatenated image is also mirrored along the x-axis to avoid border effects. This image is then normalized and inverted (see Figure S3B). Thanks to the image inversion, we can use a segmentation algorithm that is similar to the one used for fluorescent cells in 1.4, except that in the foreground detection step, a different approach is used to remove background regions. In our phase-contrast images, the sides of the microchannels have an intensity close to the values inside bacteria (see Figure S3D). Thus we use both an intensity and a shape criterion to remove background regions: a region is removed either if its intensity is lower than a threshold computed with Otsu's method[7] on the whole time-lapse series, or if it is in contact with the sides of the microchannel and has a

thickness in x-direction smaller than 5 pixels. The tracking algorithm is the same as described in 1.5.

Precision of cell size measurement

As explained in the main text, we analyzed the growth of single-cells in order to assess the precision of cell size measurement. The growth of single *E. coli* cells is exponential. Therefore two possible regressions can be performed, either i) an exponential regression of cell size S vs time t ($S_i = S_0 \exp(mt_i)+e_I$, where e_I is the noise term), or ii) a linear regression of log-size L vs time t ($L_i= L_0 + \mu t_i +e_I$). Although these two regression strategies are based on the same underlying assumption of exponential growth, they correspond to two different models, where the noise term is either additive or multiplicative on cell size. We performed regressions i and ii on approximately 10^5 cells, compared the residuals and found that the variance of the residuals was independent of cell size for regression ii and not for regression i. The stability of the variance, i.e. homoscedasticity, is a central assumption in regression analysis so we used regression ii. The residuals e_I in regression ii are additive errors on the log-size and therefore in this model, the noise in cell size is multiplicative and given by $\exp(e_I)$ ($S_i= S_0 \exp(mt_i) \times \exp(e_I)$). Therefore we estimated the relative error on cell size by computing the coefficient of variation of $\exp(e_I)$. This procedure is slightly different than the one performed by Kaiser et al.[8], but the procedure in Kaiser et al. gives similar results (1-2% of relative error).

References:

[1] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, “NIH Image to ImageJ: 25 years of image analysis" *Nature methods*, vol. 9, no. 7. p. 671, 2012.

[2] J. Schindelin *et al.*, “Fiji: an open-source platform for biological-image analysis" *Nature methods*, vol. 9, no. 7. p. 676, 2012.

[3] L. Robert, J. Ollion, J. Robert, X. Song, I. Matic, and M. Elez, “Mutation dynamics and fitness effects followed in single cells" *Science*, vol. 359, no. 6381. pp. 1283–1286, 2018.

[4] T. Panier *et al.*, “Fast functional imaging of multiple brain regions in intact zebrafish larvae using selective plane illumination microscopy” *Frontiers in neural circuits*, vol. 7. p. 65, 2013.

[5] J.-Y. Tinevez *et al.*, “TrackMate: An open and extensible platform for single-particle tracking" *Methods*, vol. 115. pp. 80–90, 2017.

[6] K. Jaqaman *et al.*, “Robust single-particle tracking in live-cell time-lapse sequences" *Nature methods*, vol. 5, no. 8. p. 695, 2008.

[7] N. Otsu, “A threshold selection method from gray-level histograms" *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1. pp. 62–66, 1979.

[8] M. Kaiser *et al.*, “Monitoring single-cell gene regulation under dynamically controllable conditions with integrated microfluidics and software" *Nature communications*, vol. 9, no. 1. p. 212, 2018.

Supplementary Tables for High-Throughput Detection and Tracking of Cells and Intracellular Spots in Mother Machine Experiments

Authors: Jean Ollion, Marina Elez, Lydia Robert

Table S1: Measurements

The following table describes the main measurements computed by BACMMAN. “Module” is the name of the measurement module, and “Column” indicates the column name of the measurement in the exported csv file. The column names are indicated in the popup help window of the module. Note that a user can sometimes change the column name in the configuration of the module (for instance add a suffix).

Column	**Module**	**Description**
Position	*automatic*	name of the position
PositionIdx	*automatic*	index of the position (zero-based)
Frame	*automatic*	Index of the frame (zero-based)
Idx	*automatic*	Index (or rank) of the object (zero-based): - For a microchannel: index of the microchannel object among all segmented microchannel at this frame. - For a bacterium or a spot: index of the object among all objects inside a given microchannel at this frame. E.g: For bacteria, the Idx value of the mother cell is 0 because it is the first object in a microchannel.
Indices	*automatic*	“barcode” defining an object (unique for a given position). It contains: - for a microchannel: frame index - index of the microchannel in the frame - for a bacterium: frame index - index of the microchannel - index of the bacterium within this microchannel at this frame - for a spot: frame index -

		index of the microchannel - index of the spot within this microchannel at this frame
Time	*automatic*	Calibrated time point
ParentTrackHeadIndices	SimpleTrackMeasurements	For bacteria and spots: barcode of the first microchannel object
TrackLength	SimpleTrackMeasurements	Length of the track, i.e. number of frames where the object is tracked
TrackHeadIndices	SimpleTrackMeasurements	barcode (Indices) of the first object of the track
LineageIndex	BacteriaLineageMeasurements	Index of the cell in the lineage tree, in a given microchannel. The first letter (A, B..) is unique, then at each division “H” is added to the index of the bacterium abutting the dead-end, and “T” to the index of its sister.
PreviousDivisionFrame	BacteriaLineageMeasurements	Frame of the previous division event (frame of the first element of the track), or NA if no division was observed before
NextDivisionFrame	BacteriaLineageMeasurements	Frame of the next division (frame following the last element of the track), or NA if no division is observed after
CoordX(Y,Z)	RelativePosition	Position of the center of the object (geometrical or intensity-weighted ; absolute or relative to another object, see popup window).
GrowthRate	GrowthRate	Estimation of the growth rate of a cells, between birth and division The value is obtained by a fit on the logarithm of cell sizes and depends on the calibration of the frame interval. The estimation of the size can be configured (e.g. area, feret diameter, spine length…)
ObjectCount	ObjectInclusionCount	Number of objects of an object class B located within an object of class A. Exemple: number of spots located in a bacteria.
ContainerObject	ContainerObject	Indices of the object of class B that containsan object of class A, if

		existing, otherwise “NA”
Size	ObjectFeature>Size	Estimation of the area of the object (number of voxels)
FeretMax	ObjectFeature>FeretMax	Estimation of the length of an object by computing the maximal distance between two points of its contour
SpineLength	ObjectFeature>SpineLength	Estimation of the length of a bacterium (see supplementary material)
Thickness	ObjectFeature>Thickness	Estimation of the thickness of an object
Mean, Max, Min ...	ObjectFeature>Mean, Max, Min...	Intensity measurements within the object, the intensity channel can be configured and prefilters can be set

Table S2 Glossary

Term	**Description**
Measurement	Operation that computes one or several scalar values from segmented objects or segmented object tracks, e.g: mean of fluorescence within a segmented object, length of a bacteria, growth rate of a generation...
Module	Image processing operation that can be developed in Java and plugged into BACMMAN. For example : segmentation module, tracking module, measurement module etc...
Object class	Type of object to be detected from pre-processed images, for instance microchannels, bacteria, spots...
Position	A given microscope field of view.
Pre-processing pipeline	Succession of pre-processing modules (e.g.: rotation, cropping, noise reduction...) applied to raw input images. Resulting images are called pre-processed images.
Processing pipeline	Succession of modules (pre- and post- filtering, segmentation and tracking) applied to pre-processed images that result in segmented objects linked into tracks.
Selection	Group of segmented objects from the same object class. They can be generated from the GUI or from the statistical processing of output data (using R or Python)

Track	List of segmented objects ordered in time. Each object has exactly one previous object and one next object, except the first object that can have several previous objects (that are the last object of several other tracks) and the last object that can have several next objects (that are the first objects of several other tracks). In the case of bacteria, a track corresponds to a generation: when a division occurs, two new tracks are created and linked to the last observation of the mother cell which is the last object of the previous track.
Track head	First object of a track.