**MicrobeJ, a tool for high throughput bacterial cell detection and quantitative analysis**

Supplementary Information

**Adrien Ducret[1]*, Ellen M. Quardokus, and Yves V. Brun***

1. Current address: Bases Moléculaires et Structurales des Systèmes Infectieux, IBCP, Université Lyon 1, CNRS, UMR 5086, 7 passage du Vercors, 69367 Lyon Cedex 07, France

***corresponding authors:**

ybrun@indiana.edu, 812-855-8860

adrien.ducret@ibcp.fr, +33 (0)4-72-72-26-79

Includes:

Supplementary Results

Supplementary Figures 1–5

Supplementary Videos 1–5

Supplementary Table 1

Supplementary References

**SUPPLEMENTARY RESULTS**

A detailed user's guide and video tutorials with examples are available on a dedicated website (http://www.indiana.edu/~microbej/).

**Bacterial Cell detection/segmentation**

By default, MicrobeJ can detect particles with various morphologies such as spherical-shaped, oval-shaped, rod-shaped, filament-shaped, branched-shaped, and irregularly-shaped bacterial cells. Cell detection can be achieved from either phase contrast or fluorescence images by one of the 21 thresholding or local thresholding methods implemented in ImageJ[1] (Huang, Intermodes, IsoData, Li, MaxEntropy, Mean, MinError, Minimum, Moments, Otsu, Percentile, RenyiEntropy, Shanbhag, Triangle, Yen, Local Default, Local MidGrey, Local Bernsen, Local Mean, Local Median, Local Niblack). The user can optionally display or hide the thresholded values on the active image to facilitate selecting the most appropriate method during the optimization process. When needed, the image can be pre-processed to aid the image binarization process using either pre-defined filters (subtract background, median, mean, minimum, maximum, Gaussian blur, or Laplacian of Gaussian) or any set of methods accessible through the ImageJ macro code.

To convert binary segmented images into bacterial cells contours, MicrobeJ has four built-in shape conversion methods, referred to as 'basic', 'smoothed', 'medial axis' and 'fit-shape'. Each method provides a different level of resolution that can accommodate a variety of specific experimental needs. In the 'basic' mode, object boundaries are defined by the pixel-based particle contours produced by the ImageJ edge-detection algorithm[1]. Although extremely effective with a processing time for cell identification down to 5 ms/particle (Windows 10 with Intel Core i5-4200U CPU, 1.6GHz, 8 Gb, 4 threads), pixel-based contours offer limited information related to the particle's morphology. In the 'smoothed' mode, the pixel-based particle contours are refined into sub-pixel resolution contours using a set of subdivision/interpolation algorithms. Crude polygons are first subdivided using a variant of the Catmull–Clark algorithm[2] and then simplified using a Reumann-Witkam algorithm[3]. The simplified polygons are then interpolated using a spline function and then smoothed using a linear smoother or a Butterworth filter using the specified cutoff frequency. The smoothed polygons are eventually enlarged or reduced using edge extrusion algorithms. While conventional thresholding techniques and edge-detection algorithms are extremely effective, they might be limited for applications requiring extensive contour sophistication. To accommodate these specific situations, an edge correction algorithm based on a Sobel–Feldman edge detector was implemented. The Sobel–Feldman edge detector finds edges by highlighting sharp changes in intensity in the input image. The result of the Sobel-Feldman operator is a 2-dimensional map of the gradient of the input image. The edge correction algorithm uses the smoothed version of the cell contours as initial guesses in an iterative edge optimization procedure. For each vertex of the cell contour, this method optimizes the lateral position of the edge by looking for local maxima of the normalized gradient magnitude from the input image.

To validate the contour detection and the edge correction algorithms, we generated 2000 phase-contrast images of virtual rod-shaped cells and determined the accuracy of their measurement by MicrobeJ. Virtual cells were simulated as rods of different lengths (4±2µm) and widths (0.75±0.2µm), with 1.2:1 ellipsoidal end-caps (major axis:minor axis), random orientations and random angularities (Supplementary Fig. 1a). 2D phase-contrast images were simulated with microlith[4] as previously described[5]. Each resulting image was then pixilated by downsizing using bilinear interpolation. Gaussian noise with a mean of zero and standard deviation of 10 was added to simulate camera noise. The accuracy of detection by MicrobeJ was determined by measuring the distance from each simulated contour vertex to the detected cell contour (Supplementary Fig. 1b). We found that the average

localization error is about 16.34 nm (0.25 pixels) for the standard method and 10.30 nm (0.16 pixels) for the edge correction algorithm. These localization errors are consistent with those reported by other cell sub-pixel contour detection methods[5,6]. To test the accuracy of cell measurements, we compared the measurements of cell length and width with the theoretical lengths and widths (Supplementary Fig. 1c-f). We found that the difference between the theoretical and measured length or width had standard deviations of 8.77 nm (0.2%) and 6.17 nm (0.8%) respectively, which are consistent with those reported by other cell sub-pixel contour detection methods[5,6]. To test the reproducibility of the contour detection, we measured the variation of the cell area, length and width of 10 *M. xanthus* motile cells during cell movement (Supplementary Fig. 1g). Cells were imaged on an agar pad by light microscopy using a 100x 1.4 NA phase oil objective, every 30 sec for 60 min and then analyzed with MicrobeJ (Supplementary Fig. 1g). By using the default detection parameters, we found that the distributions of the variation between the average area, length or width, and the value measured at each time point show a standard deviation of 19 nm², 13 nm, and 6 nm respectively (Supplementary Fig. 1h-j). Finally, to test the precision of MicrobeJ measurements, we imaged beads (F8823; Fluospheres) by light microscopy using a 60x 1.4 NA phase oil objective with a 1.5x magnifier (Supplementary Fig. 1k). According to the manufacturer specifications, beads are about 1 µm in diameter with a coefficient of variation of about 20.8% (±0.0480 µm) as determined by electron microscopy. When analyzed using MicrobeJ, beads were found to yield a distribution of diameters with mean 1.009 µm and standard deviation 0.036 µm (Supplementary Fig. 1l).

Medial axes, also called topological skeletons, provide geometrical and topological properties of the particle shape, such as its length, width, sinuosity, curvature and angularity, and establish the spatial information relative to the sub-cellular coordinate system. In the 'medial-axis' mode, medial axes are produced from the medial axis transform (MAT) of the smoothed particle contours and converted into planar embedded graphs to manage branching phenotypes. Medial axes are then simplified using a Radial Distance algorithm at the specified resolution. In the 'fit-shape' mode, an interpolated contour is generated using the particle attributes according to the specified Shape Mode (Rod-Shaped, Crescent-shaped, Circle and Ellipse). Although more complex, the 'medial axis' and 'fit-shape' modes have a processing time for cell identification close to 9 ms/particle (Windows 10 with Intel Core i5-4200U CPU, 1.6GHz, 8 Gb, 4 threads) which is about 135 times faster than with MicrobeTracker[7] and about 50 times faster than with Oufti[5].

MicrobeJ introduces a vetting system to remove inappropriate or unwanted particles from the final analysis. The user can easily define a set of specific attributes, including but not limited to area, length, width, circularity, curvature, sinuosity, angularity, solidity and signal intensity, designed to evaluate the particles in the active image. Since different types of cells can be present in the same image the user can define up to 10 different sets of attributes referred to as 'Morphology'. Each particle is tested against all the active morphologies until the specific attributes are met. When a particle meets the attributes of a specific morphology, options associated with this morphology are used to generate the final bacterial cell object. If not, the particle is rejected. MicrobeJ permits direct user interaction with the active image(s) during the detection process to optimize the detection parameters. Users can systematically change and test the settings, which may vary from experiment to experiment, prior to applying them to a full set of images for final analysis. To help define the rejection criteria, the values of the specific attributes are displayed adjacent to every particle detected on the active image.

When cells are in tight contact with each other and are detected as one single particle during the image binarization process, each individual cell can be segmented using a set of different methods. Each method consists of user-defined macros or a combination of specific filters including but not limited to Laplacian of Gaussian[8], Shape Index Map[9], Standard Watershed[10] , and Irregular Features[11]. Each method is performed on the pixels located inside the boundary of the rejected particle and returns a binary

image. For each method, the algorithm searches for the best method by selecting the one that maximizes the number of particles that meet the specified morphology attributes. As a consequence, the best-candidate segmentation algorithm is more effective when the attributes for each morphology are properly defined.

**Features detection**

Features are objects such as constrictions, septa, sections or patches that can be determined from the cell morphology, topology or fluorescence intensity. 'Constrictions' are detected using the width profile measured along the medial axis of the particle. Constrictions are located along the width profile using a local maxima algorithm. 'Septa' are detected using the intensity profile of pixels along the medial axis of the particle. Septa are located along the intensity profile using a local maxima algorithm. 'Sections' are detected using the width, the curvature or the intensity profile measured along the medial axis of the particle. 'Sections' delineates transversal sections of the cell when the width, the curvature or the intensity values measured along the medial axis of the cell are inside the user-defined range of values. 'Patches' are detected using the local curvature of the boundary of the particle or intensity profile of pixels measured along the boundary of the particle. The sub-cellular localization and the localization type such as polar or mid-cell are automatically computed for each feature (see 'Sub-cellular localization' below).

**Fluorescent Foci/Filament detection**

Foci and Filament detection can be achieved from either phase contrast or fluorescence images using the local maxima detection algorithm implemented in ImageJ[1]. To help the local maxima detection process, the user can interactively adjust the tolerance value, also called the minimum prominence value, on the active image. When needed, the image can be pre-processed to facilitate the detection of local maxima using pre-defined filters (median, mean, median, minimum, maximum or Gaussian blur).

To convert local maxima into particles, MicrobeJ has four built-in conversion methods, referred to as 'point', 'foci', 'bacteria' and 'filament'. Each method provides a different level of resolution that can accommodate specific experimental needs. In the 'point' mode, the particles are defined only by the position of local maxima in the image. Although extremely effective with a processing time for 'point' identification down to 0.02 ms/particle (Windows 10 with Intel Core i5-4200U CPU, 1.6GHz, 8 Gb, 4 threads), this conversion method offers only a limited number of shape properties. In the 'foci' mode, particle centers are not only defined by the position of the local maxima but also by the contiguous area surrounding the local maxima. The contiguous area is defined with the condition that the unsigned sigma distance ($z = \left|\frac{I_{pixel} - \mu_{background}}{\sigma_{background}}\right|$ of all pixel values is higher than the user-specified z-score. The background signal is measured with the reciprocal binary mask used to detect the local maxima. In the 'bacteria' mode, the contiguous area is converted into a bacterial cell object using one of the four built-in shape conversion methods described above (Bacterial Cell detection). In the 'Filament' mode, contiguous areas are converted into medial axes using the medial axis transform (MAT).

When high-precision localization is required, the position of each local maxima can be determined at the sub-pixel resolution using a 2D Gaussian fitting algorithm. When fluorescent foci are associated with bacterial cells, the sub-cellular localization and the localization type such as polar or mid-cell are automatically computed for each child particle (see 'Sub-cellular localization').

**Sub-cellular localization**

Features, Foci, and Filament can be linked hierarchically with the specified Particles (Bacteria or other Foci). Associated Features or Foci are qualified as child particles and the specified particles as parent particles. The specific location, distances and localization of the child particles relative to their respective parents are measured and recorded in the Results table. The number of child particles is measured for each parent particle and recorded in the Results table. To accommodate diverse experimental needs, MicrobeJ provides two types of association referred to as 'Inside' or 'Outside'. In the 'Inside' mode, Features or Foci that are spatially inside or within the boundaries of the specified particles can be associated. This mode can be used in situations where Fluorescent Foci, Filament or Features represent cellular components that are located inside or at the surface of the cell (protein clusters, plasmids, chromosomal loci, a patch of specific sugar, etc). In the 'Outside' mode, Feature or Foci that are spatially outside the boundaries of the specified Particles can be associated. This mode can be used in situations where Fluorescent Foci, Filament or Features represent cellular components that are located spatially outside or in close proximity to the cell (flagella, pili, or stalk).

To accommodate diverse applications requiring sub-cellular localization, MicrobeJ provides four types of relative coordinate system: Cartesian, orthogonal, normalized, and cylindrical (Supplementary Fig. 3a).

In the 'Cartesian' coordinate system, the relative coordinates of the child particle are defined by the distances from both the X and Y axes of the image, expressed as signed distances from the origin.

In the 'orthogonal' coordinate system, the relative coordinates of the child particle are defined as the position of the perpendicular projection of the child particle center onto the medial axis of the parent particle (X-coordinate) and the perpendicular distance between the child particle and the perpendicular projection (Y-coordinate), expressed as signed distances form the origin. By convention, the X-axis is oriented toward the first pole and the Y-axis is oriented toward the first side. If the child particle is assumed to form close to the parent particle surface, the Z-coordinate is defined using the relation $z = |y| \times \tan\varphi$ with $\varphi = \cos^{-1}(\frac{|y|}{r_c})$ where $y$ is the Y-coordinate and $r_c$ is the radius of the parent particle.

The 'normalized' coordinate system corresponds to the 'orthogonal' coordinate system with coordinates that are expressed as signed and **normalized** distances form the origin.

In the 'Cartesian', 'orthogonal' and 'Normalized' relative coordinate system, the origin can be set at the center, the first pole or the second pole of the parent particle.

In the 'cylindrical' coordinate system, the relative coordinates of the child particle are defined as the radial distance r, the angular position θ and the longitudinal position p. If the child particle is assumed to form close to the parent particle surface, the relative coordinates of the child particle are also defined by the elevation angle φ, the revolution angle β, and the arc length L. When the child particle is located in a polar region, the radial distance r is defined as the distance between the particle center and the closest end of the 'short medial axis' of the parent particle (cf. the segment of the medial axis between the origins of the two poles), the angular position θ is defined as the angle formed between the particle center and the axis of the closest pole, and the longitudinal position p is defined as the signed distance measured along the short medial axis of the parent particle from the parent particle center. The angular position θ is expressed in radians. The radial distance r is normalized by the radius of the cell. The longitudinal position p is normalized by the length of the short medial axis. The longitudinal position p is equal to 1 when the child particle is located in the pole 1 region and is equal to -1 when the child particle is located in the pole 2 region. The angular position θ is positive when the child particle is located on the side 1 region and negative when the child particle is located on the side 2. When the child particle is located in the 'short medial axis' region, the radial distance r is defined as the perpendicular distance between the child particle center and the 'short medial axis' of the parent particle, the angular position θ

is defined as the angle formed between the particle center and the 'short medial axis', and the longitudinal position p is defined as the signed distance along the short medial axis of the parent particle from the parent particle center. The angular position θ is expressed in radians. The radial distance r is normalized by the radius of the cell. The longitudinal position p is normalized by the length of the short medial axis. Note that the angular position θ is equal to $\frac{\pi}{2}$ when the child particle is located on the side 1 region and is equal to $\frac{-\pi}{2}$ when the child particle is located on the side 2. The elevation angle φ is defined using the relation $\varphi = \cos^{-1}(r)$ where $r$ is the normalized radial distance. The elevation angle φ is expressed in radians. The elevation angle φ is positive when the child particle is located on the upper side of the parent particle and negative when the child particle is located on the lower side of the parent particle. The revolution angle β is defined as the angle formed between the particle center and the transversal axis of the parent particle. The revolution angle β is expressed in radians [0,2π]. The arc length L is defined using the relation $L = \beta \times r_c$ where β is the revolution angle and $r_c$ is the radius of the parent particle.

**Tracking**

Bacterial cells, Features, Foci and Filaments can be tracked over time using an optimized nearest-neighbor algorithm. For each particle at a specific frame, the algorithm searches for particles on the next frame matching the specified conditions (distance, angle, acceleration, area, intensity). For each retained pair of particles, the algorithm computes a correlation score using the root mean squared of the specified ratios (distance, angle, acceleration, area, intensity). The algorithm then determines the most probable pairs by selecting the ones maximizing the correlation score. When associated with Bacterial cells, Features, Foci, and Filaments can be tracked using their relative coordinates.

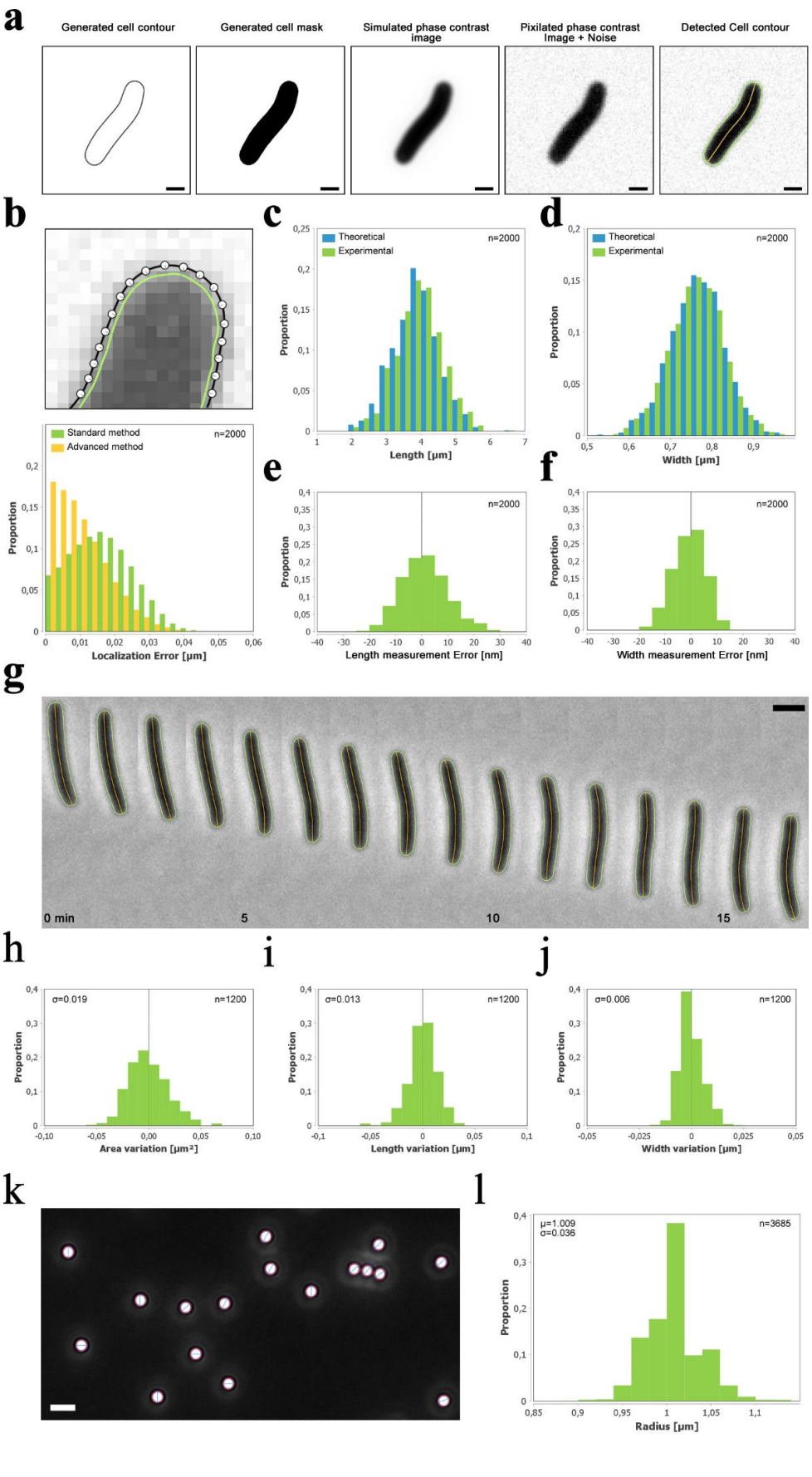
MicrobeJ provides two tracking modes, referred to as 'Basic' or 'Advanced'. In the 'basic' mode, only the distance between two frames, area and intensity are measured and used to compute the correlation score. In the 'Advanced' mode, the algorithm tracks the particles assuming they are actively moving: the distance, area, intensity, angle and acceleration are measured and used to compute the correlation score.

Particle tracks can be optionally analyzed for specific information, including but not limited to motion descriptors, events, and phases detection that can be computed and recorded in the Results table. Motion descriptors include the distance, velocity, confinement ratio, distance from origin, type of displacement (*moving*, *diffusing* and *not moving*), and the mean squared displacement. A fit can be automatically performed on the Mean Squared Displacement to compute the extrapolated value of the velocity, the diffusion coefficient and the confinement factor. For the events or phase detection, particle tracks are analyzed for specific events or specific shift in the velocity, the type of displacement, the direction and any user-defined property that could occur along the different positions over time. Each event consists of specific criteria designed to evaluate the particle properties over the time course. When the particle meets the criteria of a specific event, the timing is recorded in the results table.

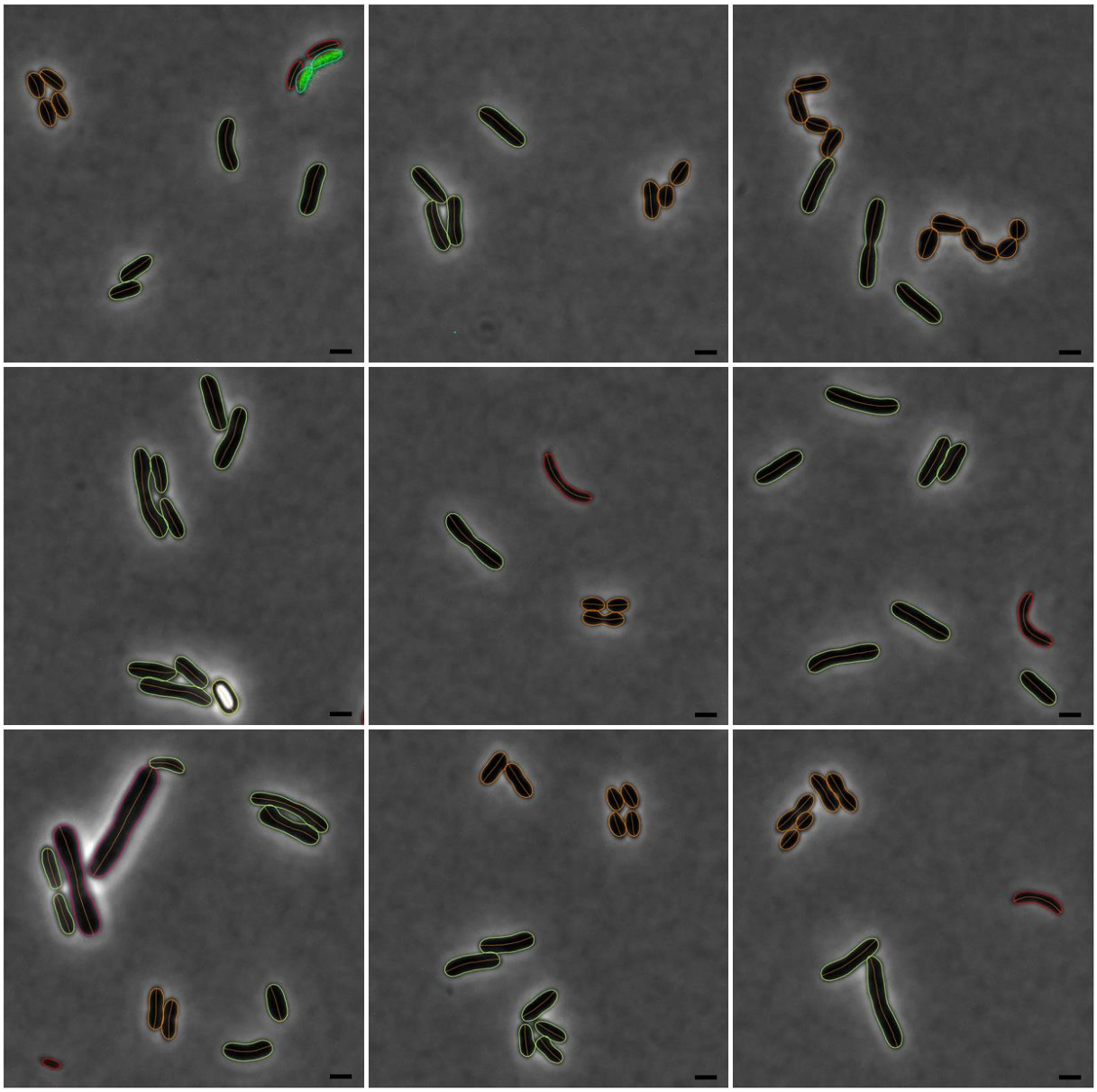
**Batch-processing**

To ensure high efficiency and better memory management, MicrobeJ supports multithreading and Virtual stack images. Virtual stack images are disk resident images, as opposed to Random-access memory (RAM) resident images, and allow analysis of image files exceeding the RAM capacity. To illustrate the high efficiency of MicrobeJ, we imaged 15 different strains of C. crescentus cells, by phase contrast and fluorescence microscopy every 5s for 3h in triplicate. Each experiment generated a 4.2 Gb stack of images with about 40,000 bacterial cells and 20,000 fluorescent foci that needed to be detected, analyzed, associated and tracked over time. Using a conventional PC (Windows 7 with Intel Xeon (R)

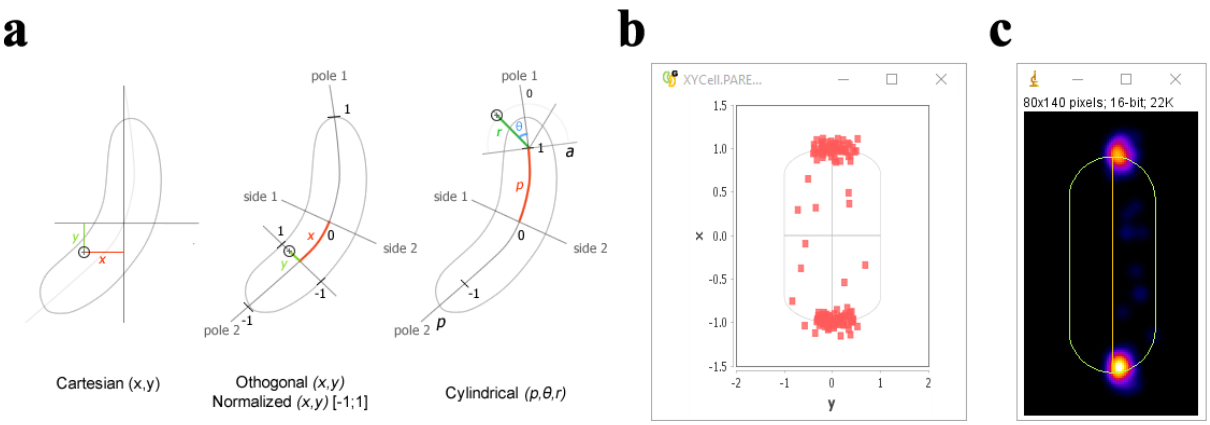
processor, 3.3 GHz, 16 Gb, 4 threads), MicrobeJ was able to process a total of 189 Gb of images, about 1.8 106 bacterial cells and 0.9 106 fluorescent foci, in about 10h, i.e., ~50 cells per second.


a
Generated cell contour
Generated cell mask
Simulated phase contrast image
Pixilated phase contrast Image + Noise
Detected Cell contour
b
Standard method
Advanced method
n=2000
Proportion
Localization Error [µm]
c
Theoretical
Experimental
n=2000
Proportion
Length [µm]
d
Theoretical
Experimental
n=2000
Proportion
Width [µm]
e
n=2000
Proportion
Length measurement Error [nm]
f
n=2000
Proportion
Width measurement Error [nm]
g
0 min
5
10
15
h
σ=0.019
n=1200
Proportion
Area variation [µm²]
i
σ=0.013
n=1200
Proportion
Length variation [µm]
j
σ=0.006
n=1200
Proportion
Width variation [µm]
k
l
µ=1.009
σ=0.036
n=3685
Proportion
Radius [µm]

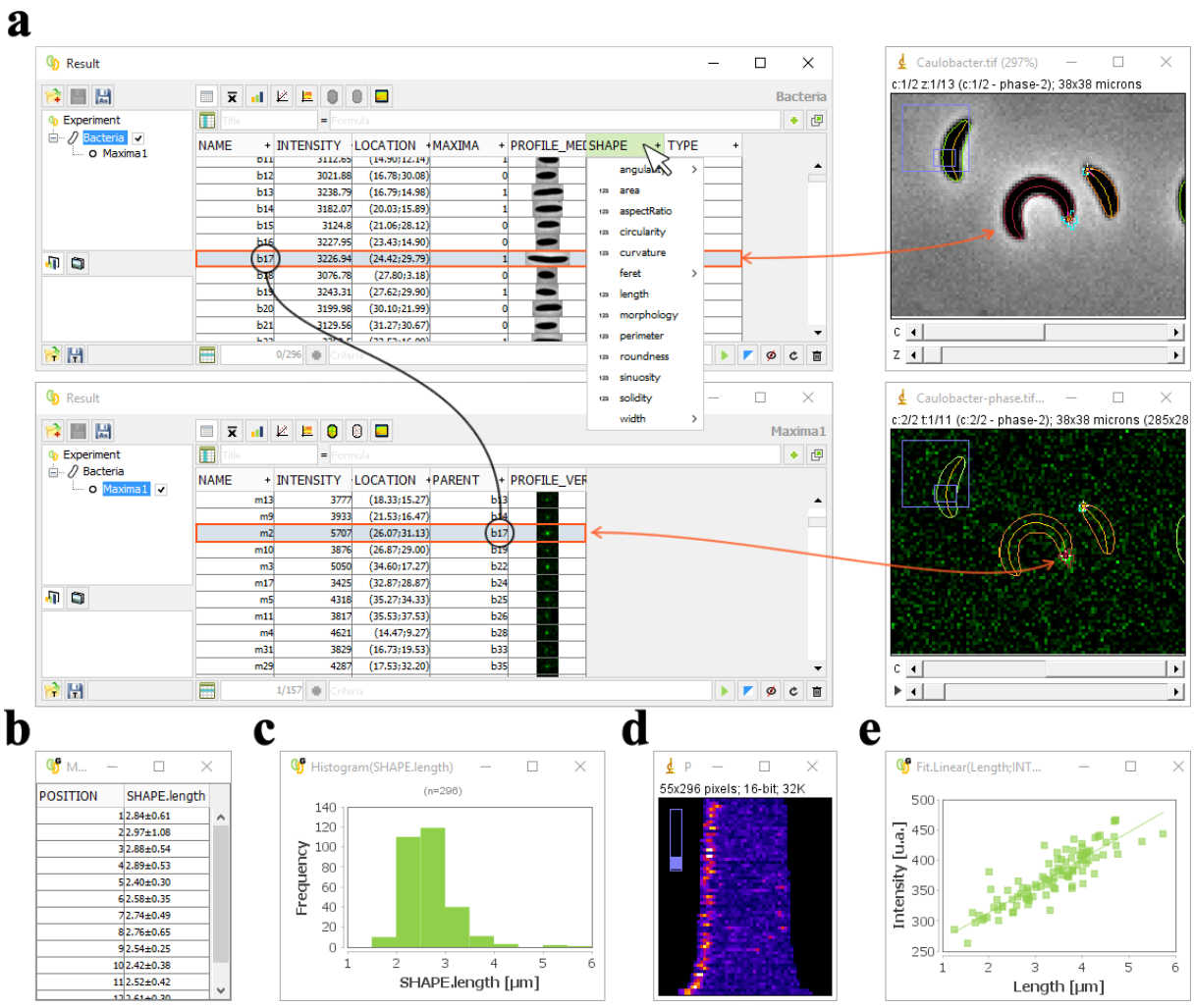
**Supplementary Figure 1: MicrobeJ measurements are accurate and precise.** (a) Virtual cells contours were generated as rods of different lengths and widths, with ellipsoidal end-caps, random orientations and random angularities (n=2,000). The cell contours were then used to generate cell masks and simulate 2D phase-contrast images. Simulated images were pixilated by down-sizing (scaling factor of 0.2) and Gaussian noise was added to simulate camera noise. The resulting images were then analyzed using MicrobeJ. The scale bar represents 1 µm. The accuracy was determined by measuring the distance from each simulated contour vertex to the detected cell contour using the standard method (green bars) or with the edge correction algorithm (yellow bars) (b). The accuracy of cell measurements was determined by comparing the theoretical length (c: blue bars) and width (d: blue bars) with the length and the width measured by MicrobeJ (green bars). The distributions of the difference between the theoretical length or width, and the length or the width measured by MicrobeJ are shown respectively in (e) and (f). The reproducibility of the contour determination was determined by measuring the variation of the cell area, length and width of 1,200 *M. xanthus* motile cells during cell movement (g). Scale bar = 1 µm. The distributions of the variation between the average area, length or width, and the value measured at each time points are shown respectively in (h), (i) and (j). (k) The precision of MicrobeJ measurements was determined by measuring the diameter of 3,685 beads of known dimensions (1±0.0480). The distribution of bead diameters as measured by MicrobeJ is shown in (l). The diameters are measured in microns. Scale bar = 2 µm.



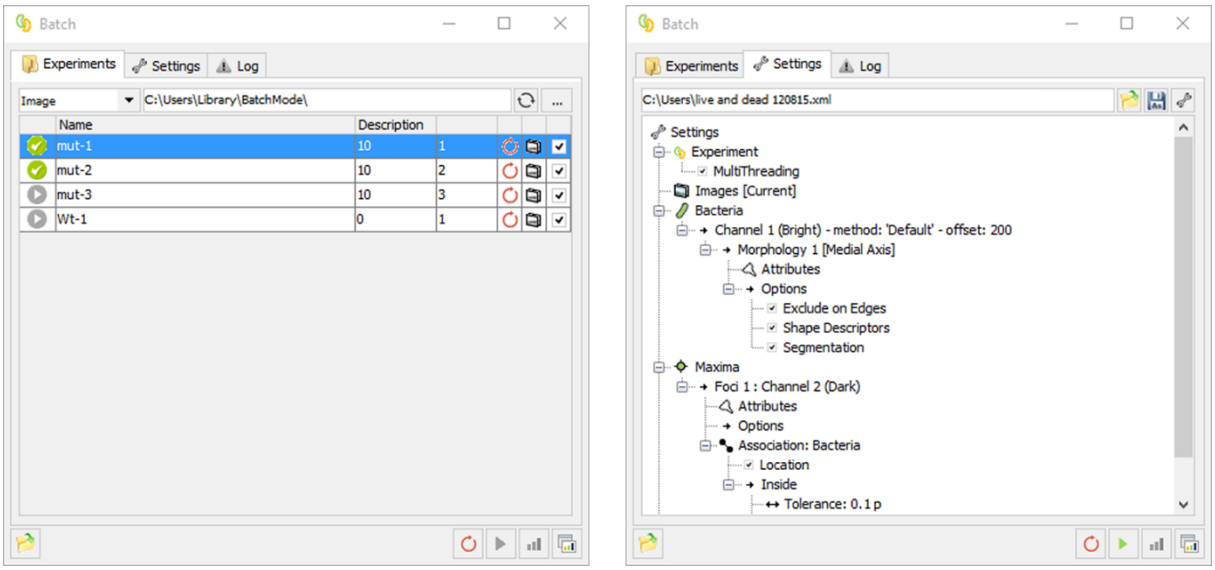
**Supplementary Figure 2: MicrobeJ can define cellular types based on morphological and signal properties.** In this experiment, *C. crescentus* CB15 cells (red contours), *C. crescentus* cells expressing a cytoplasmic GFP (*mini-Tn7(Gm)gfp*) (cyan contours), *B. subtilis* vegetative cells (magenta contours), *B. subtilis* spores (yellow contours), *Lactococcus lactis* cells (orange contours) and *E. coli* MG1655 cells (green contours) were mixed together and imaged on an agar pad (Scale bar = 1 µm). Cellular types were defined using the morphological and signal properties based on the specified criteria. See Fig. 2c for criteria used to differentiate each type.




**Supplementary Figure 3: The different types of relative coordinate systems**. When maxima, filaments or features are associated with bacteria, their sub-cellular localization is measured and recorded in the results using four relative coordinate systems: Cartesian, orthogonal, normalized, and cylindrical (a). Since maxima are associated with bacteria, the results table also provides sub-cellular representation tools such as localization charts (b) and heat-maps (c).




**Supplementary Figure 4: Presentation of the results interface of MicrobeJ.** (a) The MicrobeJ Results GUI: This interface contains information about the experiment, the detected particles, and complementary data based on user-selected options. Each row is an individual particle detected in the image and is dynamically linked to the image. If the user clicks on one row (outlined in red), it will highlight the corresponding bacterium or fluorescent focus detected on the image (outlined in red) and, likewise, if the user selects a bacterium or fluorescent focus on the image (outlined in red), it will highlight the corresponding row of the list (outlined in red). Graphical outputs, such a XY plots or histograms are also dynamically linked to the result table and the image. This interaction permits editing of the dataset to temporarily exclude, include, or remove a particle from the list and ultimately from the final analysis. The results tables (Experiment, Bacteria, and Maxima1) are connected and arranged hierarchically. Because each maximum is associated with a specific bacterial cell, if the user hides or removes a bacterial cell in the results table, both the cell and the associated maxima will be hidden or removed in the image and from the final data analysis. While measurements can be copied into other programs for subsequent data analysis, the user can easily produce common statistical analyses (b), interactive charts (c), fluorescence profiles (d), and regression analysis (e) directly from the results interface.




**Supplementary Figure 5: The batch-processing interface of MicrobeJ**. The batch mode GUI contains several tabs that allow the user to apply settings (Settings) to a set of experiments (Experiments) and track any errors that might arise during analysis (Log). Using this interface, the user can simply open a folder containing a set of Images or a set of folders containing images to be analyzed as individual experiment. Once the experiments are detected and selected, the user can load existing settings and start the analysis. Each experiment will be analyzed subsequently until the list of selected experiment is exhausted. Once the selected experiments are analyzed, the user can either load the results in a results interface or open automatically saved csv files.

## Supplementary Videos

**Supplementary Video 1: Presentation of the MicrobeJ workflow.** An example of a simple image analysis with MicrobeJ in real-time is presented. In this example, *C. crescentus* cells expressing DivJ-GFP, which forms a single fluorescent focus located at one pole of the cell, are detected from a multidimensional stack (Hyperstack) of images. After detection parameters are optimized to segment touching cells and exclude small particles from the analysis, and the results are generated. From the results interface, a number of different outputs, such as an interactive histogram of the cell length, a demograph, a heatmap, and an interactive sub subcellular localization graph are produced and saved as a template. The template is saved and then re-used to automatically perform the same analysis on a different stack of images and generate the same outputs. Note that templates can be saved as a file or directly dragged and dropped from the results interface to the MicrobeJ main Interface.

**Supplementary Video 2: FtsZ-YFP localization in *C. crescentus*.** Sequence of phase contrast and fluorescent overlay images showing several rounds of division from a single *C. crescentus* cell expressing FtsZ-YFP. Pictures were taken every 5 min. The bacterial cell contours (green) and their corresponding medial axes (orange) are shown.

**Supplementary Video 3: Example of an advanced image analysis with MicrobeJ in real-time.** In this example, we analyzed *C. crescentus* cells expressing FtsZ-YFP by time-lapse microscopy (Video S2) and tracked the respective localization of FtsZ and the cell constriction during the cell cycle (see Figure 3 for more details).

**Supplementary Video 4: FtsZ-YFP localization in a filamentous *C. crescentus* cell.** Sequence of phase contrast images showing several rounds of division from *C. crescentus* cells expressing FtsZ-YFP and harboring division defects. Pictures were taken every 5 min. The bacterial cell contours (green) and their corresponding medial axes (orange) are shown.

**Supplementary Video 5: Example of automated and manual segmentation with MicrobeJ in real-time.** In this example, we analyzed a micro-colony of *C. crescentus* cells harboring defects in cell division by time-lapse microscopy. After we optimized attribute values and used the default set of segmentation methods, we manually separated unsegmented cells and then generated the final dataset.

## Supplementary Table 1: Comparison of MicrobeJ with ImageJ, ObjectJ, ColiInspector, BactImAS, MicrobeTracker, Oufti, PSICIC, Schnitzcells and CMEIS.

| | MicrobeJ | ImageJ/Fiji | ObjectJ | Coli-Inspector | BactImAS | MicrobeTracker | Oufti | PSICIC | Schnitzcells | CMEIS |
|---|---|---|---|---|---|---|---|---|---|---|
| **Software** | | | | | | | | | | |
| Platform | ImageJ | ImageJ | ImageJ | ImageJ/ObjectJ | Standalone/Icy/ImageJ | MATLAB | Standalone | MATLAB | MATLAB | Standalone |
| Documentation | On-line | On-line | On-line | On-line | On-line | On-line | On-line | On-line | On-line | On-line |
| Written tutorials | On-line | On-line | On-line | On-line | On-line | On-line | On-line | None | On-line | On-line |
| Video tutorials | On-line | On-line | On-line | On-line | On-line | On-line | On-line | None | On-line | |
| User Interaction | GUI | GUI | GUI | GUI | GUI | GUI + Command line | GUI + Command line | GUI + Command line | GUI + Command line | GUI |
| Computer code | Open Source | Open Source | Open Source | Open Source | Open Source | Open Source | Open Source | Open Source | Open Source | Copyright MSU |
| Save settings to file | Yes | No | Yes | Yes | No | Limited | Yes | No | No | Yes |
| **Images** | | | | | | | | | | |
| Drag and Drop | Yes | Yes | Yes | Yes | Yes | No | No | No | No | No |
| File Type | Extensive | Extensive | Extensive | Extensive | Limited | Limited | Limited | Limited | Limited | Limited |
| Interactivity | Full | Full | Full | Full | Full | Limited | Limited | Limited | Limited | Limited |
| Display | Uni-Dimensionnal | Uni-Dimensionnal | Uni-Dimensionnal | Uni-Dimensionnal | Uni-Dimesionnal | Uni-Dimesionnal | Uni-Dimesionnal | Uni-Dimesionnal | Uni-Dimesionnal | Uni-Dimesionnal |
| **Detection/Analysis** | | | | | | | | | | |
| SubPixel Resolution Outlines | Fully-Automated | Manual | Manual | Semi-automated | Semi-automated | Fully-Automated | Fully-Automated | Fully-Automated | - | - |
| Branched cell Detection | Fully-Automated | - | - | - | - | - | - | - | - | - |
| Cell segmentation | Fully-Automated | Manual | Manual | Manual | Semi-automated | Fully-Automated | Fully-Automated | Fully-Automated | Fully-Automated | - |
| SubPixel Foci Detection | Fully-Automated | Manual | Manual | Manual | Semi-automated | Fully-Automated | Fully-Automated | Fully-Automated | - | - |
| SubPixel Filament Detection | Fully-Automated | - | Manual | Manual | - | - | - | - | - | - |
| Sub-Cellular localization | Fully-Automated | - | Semi-automated | Semi-automated | Semi-automated | Fully-Automated | Fully-Automated | Fully-Automated | - | - |
| Tracking | Fully-Automated | - | - | - | Fully-Automated | Fully-Automated | Fully-Automated | Fully-Automated | Fully-Automated | - |
| Sub-Cellular Tracking | Fully-Automated | - | - | - | - | Not-Implemented* | Not-Implemented* | Not-Implemented* | - | - |
| Manual Editor | Available | Available | Available | Available | - | Limited | Available | Limited | Limited | - |
| **Precision/ Accuracy** | | | | | | | | | | |
| Contour Localization Error | 10.30 - 16.34 nm | 24.48 nm | - | - | - | - | 15 - 26 nm | - | - | - |
| Area Measurement Error | 19 nm² | - | - | - | - | - | 6.6 - 82 nm² | - | - | - |
| Length Measurement Error | 13 nm | - | - | - | - | - | - | 8 nm | - | |
| Width Measurement Error | 6 nm | - | - | - | - | - | - | - | - | |
| **Features** | | | | | | | | | | |
| Cell Morphology Parameters | Extensive | Limited | Limited | Limited | Limited | Extensive | Extensive | Limited | Limited | Extensive |
| Cellular type determination | Fully-Automated | Manual | Manual | Manual | Manual | Manual | Manual | Manual | Manual | Fully-Automated |
| Profile extraction | Fully-Automated | Manual | Fully-Automated | Fully-Automated | - | Fully-Automated | Fully-Automated | Fully-Automated | - | - |
| Septa/Constriction Detection | Fully-Automated | - | Fully-Automated | Fully-Automated | - | Fully-Automated | Fully-Automated | Fully-Automated | - | - |
| Cell Contact Detection | Fully-Automated | - | - | - | - | Not-Implemented* | Not-Implemented* | Not-Implemented* | - | - |
| Addition of new features | Limited | Limited | Available | Limited | Limited | Not-Implemented* | Not-Implemented* | Not-Implemented* | Limited | Limited |
| **Results** | | | | | | | | | | |
| Data Processing | In-Program | Limited | Limited | Limited | Limited | In-Program | In-Program | In-Program | In-Program | - |
| Interactivity | Extensive | Extensive | Extensive | Extensive | Limited | Limited | Intermediate | Intermediate | Limited | Limited |
| Graphical Representations | In-Program | Limited | In-Program | In-Program | In-Program | In-Program | In-Program | In-Program | In-Program | - |
| Addition of new Representations | Limited | Limited | Limited | Limited | Limited | In-Program | In-Program | In-Program | In-Program | Limited |
| Templates | In-Program | - | - | - | - | In-Program | In-Program | In-Program | In-Program | - |
| Storage | .res/.xml/.csv/.pn | .txt/.csv | .ojj file format | .ojj file format | Relational | MATLAB format | MATLAB format | MATLAB format | MATLAB format | .txt/.csv |
| **High-Throughput** | | | | | | | | | | |
| Batch Mode | Fully-Automated | Limited | Limited | Limited | Limited | Fully-Automated | Fully-Automated | - | Limited | - |
| Multi-threading | Available | Limited | Limited | Limited | Limited | Limited | Available | Not-Implemented | Limited | - |

*Available with a significant knowledge of programing (requires command line)

**SUPPLEMENTARY REFERENCES**


1. Schneider, C. A., Rasband, W. S. & Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **9,** 671–675 (2012).
2. Catmull, E. & Clark, J. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput.-Aided Des.* **10,** 350–355 (1978).
3. Reumann, K. & WitKam, A. P. M. in *Proceedings of International Computing Symposium* 467–472 (North- Holland Publishing Company, Amsterdam, 1974).
4. Mehta, S. B. & Oldenbourg, R. Image simulation for biological microscopy: microlith. *Biomed. Opt. Express* **5,** 1822–1838 (2014).
5. Paintdakhi, A. *et al.* Oufti: An integrated software package for high-accuracy, high-throughput quantitative microscopy analysis. *Mol. Microbiol.* (2015). doi:10.1111/mmi.13264
6. Guberman, J. M., Fay, A., Dworkin, J., Wingreen, N. S. & Gitai, Z. PSICIC: Noise and Asymmetry in Bacterial Division Revealed by Computational Image Analysis at Sub-Pixel Resolution. *PLOS Comput Biol* **4,** e1000233 (2008).
7. Sliusarenko, O., Heinritz, J., Emonet, T. & Jacobs-Wagner, C. High-throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics. *Mol. Microbiol.* **80,** 612–627 (2011).
8. Mexican Hat Filter. Available at: http://rsb.info.nih.gov/ij/plugins/mexican-hat/. (Accessed: 5th February 2016)
9. Shape Index Map. Available at: http://fiji.sc/Shape_Index_Map. (Accessed: 5th February 2016)
10. Vincent, L. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE PAMI* **13,** 583–598
11. Watershed Irregular Features. Available at: http://fiji.sc/BioVoxxel_Toolbox#Watershed_Irregular_Features. (Accessed: 5th February 2016)