

# SUPPORTING INFORMATION

## ChipSeg: an automatic tool to segment bacterial and mammalian cells cultured in microfluidic devices

Irene de Cesare<sup>1,#</sup>, Criseida G. Zamora-Chimal<sup>1,2,#</sup>, Lorena Postiglione<sup>1</sup>, Mahmoud Khazim<sup>1,3</sup>, Elisa Pedone<sup>1,3</sup>, Barbara Shannon<sup>2,4</sup>, Gianfranco Fiore<sup>1,2</sup>, Giansimone Perrino<sup>5</sup>, Sara Napolitano<sup>5</sup>, Diego di Bernardo<sup>5,6</sup>, Nigel J. Savery<sup>2,4</sup>, Claire Grierson<sup>2,7</sup>, Mario di Bernardo<sup>1,2,8,+,\*</sup>, Lucia Marucci<sup>1,2,3,+,\*</sup>

<sup>1</sup> Department of Engineering Mathematics, University of Bristol, Woodland Road, Bristol BS8 1UB, UK. <sup>2</sup> BrisSynBio, Life Sciences Building, Tyndall Avenue, Bristol BS8 1TQ, UK. <sup>3</sup> School of Cellular and Molecular Medicine, University of Bristol, University Walk, Bristol BS8 1TD, UK. <sup>4</sup> School of Biochemistry, University of Bristol, University Walk, Bristol BS8 1TD, UK. <sup>5</sup> Telethon Institute of Genetic and Medicine Via Campi Flegrei 34, 80078 Pozzuoli, Italy. <sup>6</sup> Department of Chemical, Materials and Industrial Production Engineering, University of Naples Federico II, 80125 Naples, Italy. <sup>7</sup> School of Biological Sciences, University of Bristol, Tyndall Avenue, Bristol BS8 1TQ, UK. <sup>8</sup> Department of EE and ICT, University of Naples Federico II, Via Claudio 21, 80125 Naples, Italy.

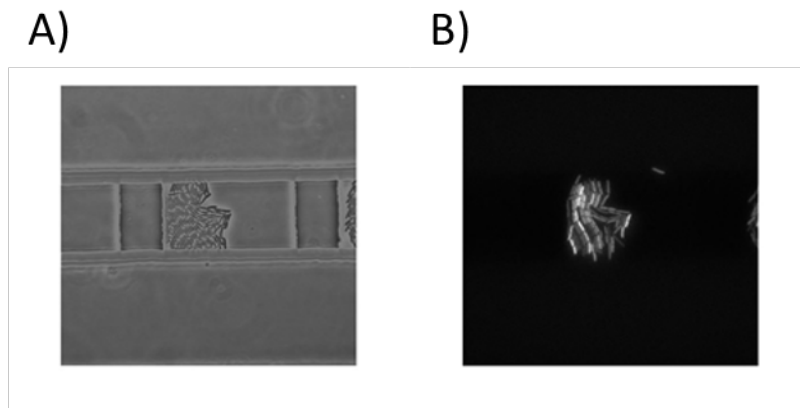
# These authors contributed equally to this work and should be considered co-first authors.

+ These authors contributed equally to this work and should be considered co-senior authors.

\* Correspondence should be addressed to M.d.B. (mario.dibernardo@unina.it) and L.M (lucia.marucci@bristol.ac.uk).

## **Bacterial cell segmentation**

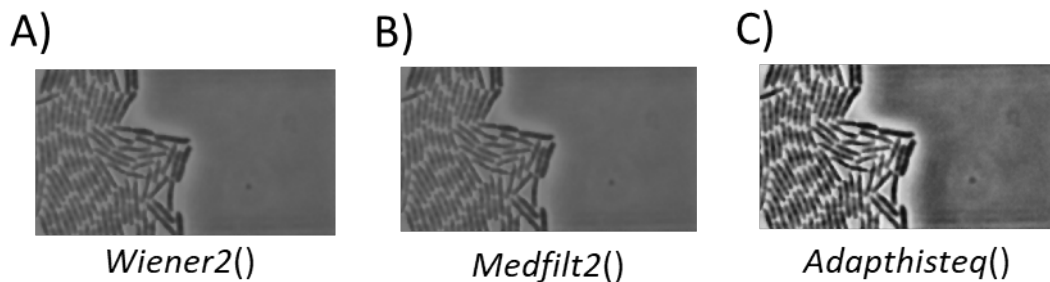
The segmentation algorithm is applied here to phase contrast images (e.g. Figure S1A) to distinguish the foreground (bacteria cells) from the background in each frame of the time-lapse experiment. A pre-processing of the raw images is generally required before running the segmentation routine, which can then segment and count individual cells in a filtered frame. The segmentation algorithm output is a mask, used to calculate the fluorescence of individual cells (raw fluorescence image as in Figure S1B). and the average fluorescence across the bacteria population. The Matlab files implementing the reported algorithm are 'segmentation\_main\_offline\_of\_video.m', 'segmentation\_GF.m', 'fluorescence\_eval\_FG\_Init.m', 'fluorescence\_eval\_GF.m' and reshapeHist.m.



**Figure S1. Two channels frames.** A) Raw phase contrast image. B) Raw green fluorescence image.

### **1. Pre-processing of the raw images**

For image pre-processing, firstly the user defines manually in the original image a crop; in so doing, the region of interest where cells are expected to grow is defined, excluding external objects (e.g. microfluidic device borders). Then, interpolation by *Interp2* is required to set the proper size of the image and to increase the resolution. Finally, filters such as *Wiener2* (2-D adaptive noise-removal filter) and *Medfilt2* (2-D median filter) are applied to reduce noise and blurriness (Figures S2A, B); the *Adapthisteq* function is used to enhance the contrast of the grayscale image (Figure 2C).



**Figure S2. Matlab filters for pre-processing.** A) Wiener2 filtering: adaptive noise-remove filter; B) Medfilt2 filtering: the function performs median filtering of a matrix in two

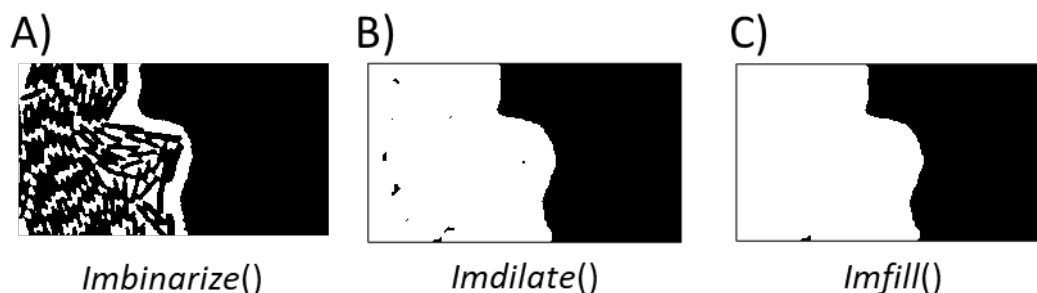
dimensions; C) *Adaphisteq* contrast saturation: the function enhances the image contrast; it operates on small data regions (tiles), rather than on the entire image.

## 2. Global and local thresholding

After pre-processing, the algorithm applies the Otsu method <sup>1</sup> to binarize the greyscale image by the *Imbinarize* function and uses morphological operators such *Imdilate* and *Imfill* to calculate the global area where cells are located (Figure S3). Local thresholding is then applied to find cell centres and edges to segment individual cells by the *Niblack* and *Averagefilter* user-defined functions (Figure S4).

### a. Global thresholding

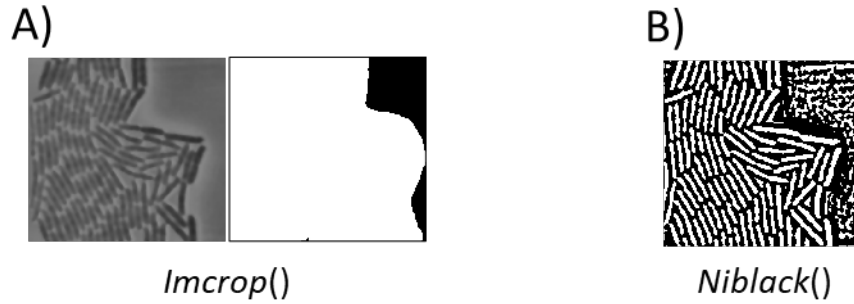
A first step for the computation of a mask is given by the identification of the “global area” where cells can be distinguished from the background. This rough result is achieved using three simple functions: *Imbinarize*, that calls the Otsu method and generates a binary image; *Imdilate* and *Imfill*, which dilate the objects in the image and then fill the internal gaps, respectively (Figure S3).



**Figure S3. Global thresholding functions.** A) The *Imbinarize* function binarizes the grayscale image by thresholding (Otsu method). The threshold is automatically defined as global, unless a different method is specified. B) The *Imdilate* function dilates the image referring to the shape of a structuring element object (STREL). C) The *Imfill* function performs a filling operation on background pixels of the input binary image specified.

### b. Local Thresholding

Local thresholding is needed to differentiate individual cells in a binary mask. First, an automatic crop is applied to select the zone of the image containing cells (Figure S4A). The size for the crop is chosen referring to the mask size, previously calculated by global thresholding. To apply a local thresholding routine, we use two user-defined functions (*Niblack* and *Averagefilter*) to segment individual objects; the *Averagefilter* function is nested inside the *Niblack* call (Figure S4B).

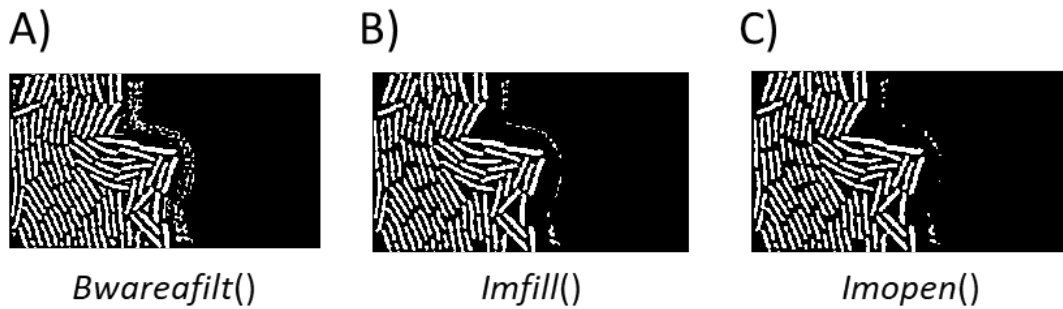


**Figure S4. Local thresholding functions.** A) The *Imcrop* function crops an image; if no area is specified, the command can create an interactive window to select an area to be cropped.

B) *Niblack* and *Averagefilter* user-defined functions: these nested functions, based on the *Niblack* algorithm in <sup>2</sup>, divide foreground and background using local information about the image.

### 3. Segmentation refinement and evaluation

After local thresholding, we refine the segmentation results to get rid of possible artefacts of the segmentation process. Firstly, small objects which are out of the individual cells size range are removed. This is done by a selective Matlab function – *Bwareafilt* - that keeps the *n* largest objects according to user-defined criteria (Figure S5A). Then, the morphological operators *Imfill* and *Imopen* are called to smooth the edges and fix small objects which are likely to be artefacts of the segmentation algorithm (Figures S5B and C). The resulting local thresholding mask is shown (Figure S5C); there are still few small objects which are removed later by a user-defined function based on fluorescence calculation (see below).

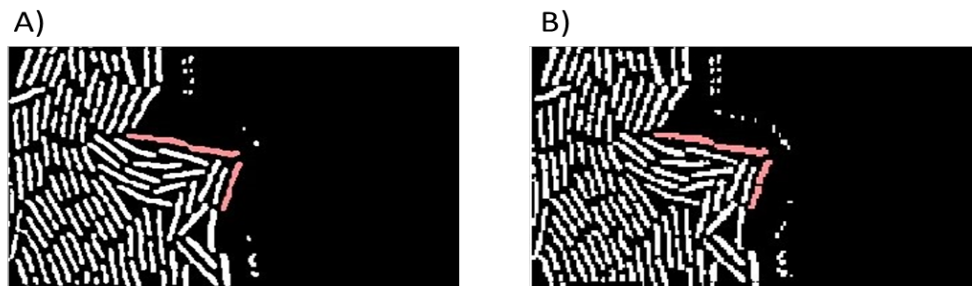


**Figure S5. Segmentation refinement functions and mask resulting from local thresholding.**

A) The *Bwareafilt* function extracts objects from a binary image by size. A value for the size must be specified. B) *Imfill* function, as in Figure S3C. C) *Imopen* function: the morphological operation is an erosion followed by a dilation, using the same structuring element (STREL) for both operations. This operation allows a further removal of some small objects.

A further refinement routine is carried out by the user-defined function *segmentationRefinement*. This method aims at removing artefacts due to the fact that individual cells, if too close, can get segmented as a single object. Firstly, the algorithm chooses the objects which have an area twice bigger than the average area calculated over all the objects

present in the mask. These larger objects are further segmented until identifying individual cells (Figure S6).



**Figure S6. Long objects refinement** A) The image shows the original mask before the refinement process. Long objects, highlighted in red, are detected by the algorithm. B) The bigger objects are divided by a line, drawn calculating the convex points representing edges where two cells touch.

A final refinement of the mask is made by deleting the remaining small objects. The average cell area evaluated in the previous step is used again to identify objects with an area at least 50% smaller than the average. Through the user-defined function *SmallObjectsRemoval*, those objects are deleted from the mask, which is then resized to the original crop size defined in the pre-processing stage (Figure S7).



**Figure S7. Small objects removal.** The image shows the last process of small objects removal (user-defined function) to delete the spots around the segmented cells. This is also the final mask used to calculate the fluorescence.

#### 4. Fluorescence calculation and segmentation evaluation

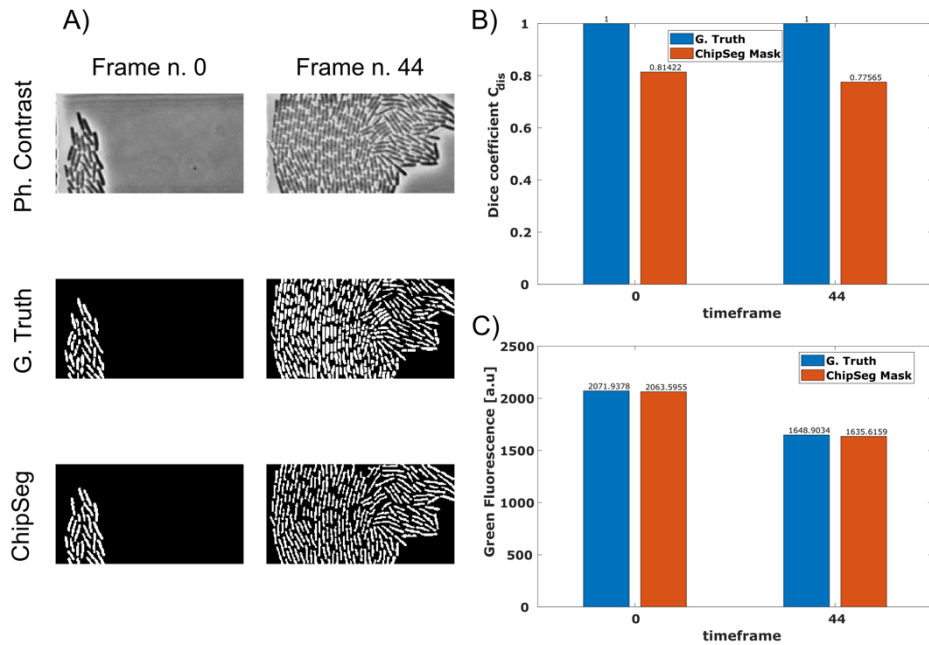
The final mask (Figure S7) is overlapped to the green field image to calculate the fluorescence using the custom function *fluorescence\_eval.m*. This is computed as the sum of all pixels in the segmented area minus the background fluorescence value. The background is calculated as the mean of the pixels acquired cropping an area without cells from the green field channel. A threshold fluorescence value (using *fluorescence\_eval\_Init.m*) is computed to exclude cells over a fluorescence threshold set by the user: in this way possible outliers (e.g. mutants resulting from phototoxicity effects) are removed. The threshold is calculated as the highest fluorescence in a set of images. The final average fluorescence value is then computed as the mean of the fluorescence exhibited by all the objects in the final mask after removing those over the threshold. The algorithm outputs at each sampling time of the experiment the average fluorescence across the cell population (Figure 1C, Supporting Movie 1), its statistical

moments and the fluorescence histogram. The algorithm outputs also the cell number by counting the numbers of individual objects <sup>3</sup>.

To assess the quality of segmentation, we compared our computed masks with the ground truth images (i.e., images annotated by an expert, using the Image Labeler toolbox in Matlab and manually drawing the cell's interior). As in <sup>4</sup>, the Dice coefficient metric was used; it represents the ratio of overlapped region between the segmented and the ground truth regions, and is defined as

$$C_{dice} = \frac{2 |M \cap GT|}{|M| + |GT|}$$

where M is the segmented mask, GT the ground truth and  $||$  corresponds to a measure of the area. Results in S8 for two frames with partially full/full microfluidic chamber show an agreement with the ground truth  $> 75\%$  for both images; as expected, the performance is slightly worst with increased device density. The fluorescence quantification was comparable using either of the masks (Figure S8C).



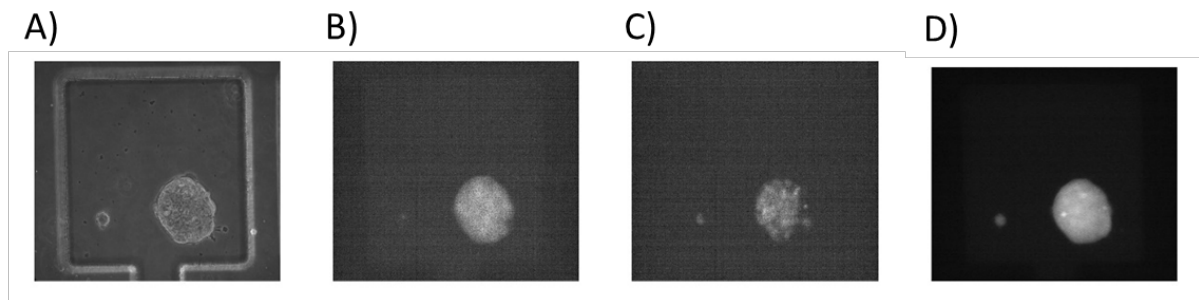
**Figure S8. Metrics calculation.** A) Phase contrast, and corresponding ground truth and ChipSeg-computed masks at two specific time points of the experiment in Supplementary Movie 1. B) Dice coefficient for ground truth and ChipSeg-computed masks. C) GFP quantification for ground truth and ChipSeg-computed masks.

## **Mammalian cell segmentation**

The segmentation algorithm for mammalian cells, similarly to the bacteria one, can be applied to different types of images to distinguish the foreground from the background. The algorithm can identify cellular marks, compute fluorescence and track cells over time. The steps and parameters have been optimised for our experimental settings (see below); the user might need to refine parameters or could possibly skip some pre-processing steps depending on the acquired raw images and the cells used. The Matlab files implementing the reported steps are 'Main\_Offline.m', 'Offline\_function.m', 'Mask\_n\_Track.m' and 'fluorescence\_evaluation.m'.

### **1. Selection of a channel for the segmentation process**

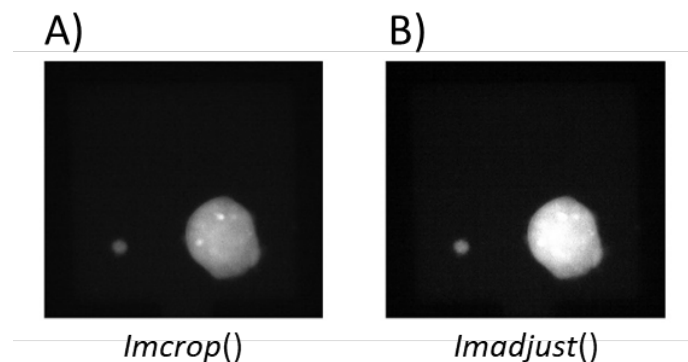
The algorithm applied to the first exemplar experiment shown here (Figures S9-S11, Supporting Movie 2) uses an Otsu method to identify masks and quantify the fluorescent reporter expression in a dome-shaped population of Rex1-GFPd2 mESCs. We applied the Otsu method to either the phase contrast images, or images collected using a Blue channel to capture a blue dye incorporated into cells (Figures S9A, D). Blue dye images were used for segmentation given the superior contrast in the scale of grey as compared to phase contrast images; furthermore, if based on dye images, the algorithm avoids including in the foreground non-fluorescent objects which are not cells (e.g. edges of the microfluidic device). In what follows, we show only the segmentation process in case dye images are used; still, the same routine could be used for phase contrast frames.



**Figure S9. Raw images, Rex1-GFPd2 mESCs.** A) Phase contrast image. B) Green fluorescence image. C) H2B- Nuclei tag image. D) Blue dye image.

### **2. Pre-processing of the raw images**

As for bacterial cells, at the start of the time-lapse a crop is manually made to define the region of interest from the initial picture frame (Figure S10A); the same crop is then used throughout the time-lapse. At the start of the experiment, an additional crop is required to identify a region without cells, which will be later used to calculate the background fluorescence. Then, the cropped image with cells is filtered to adjust the colour intensity values; 1% of data is saturated at low and high intensities by the *Imadjust* MATLAB function (Figure S10B).

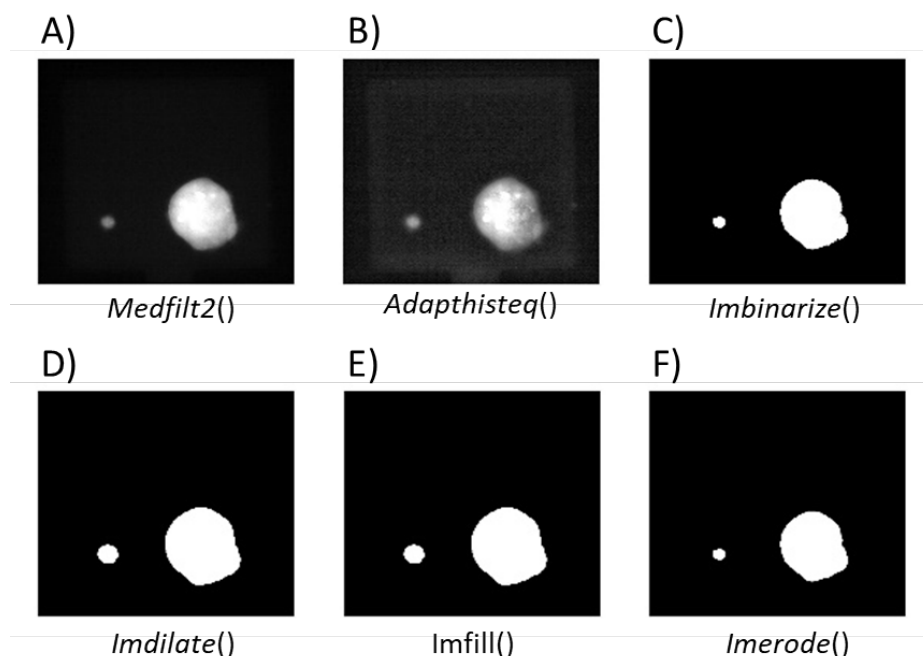


**Figure S10. Pre-processing functions.** A) *Imcrop* function: the original raw image is manually cropped from the first timeframe. The crop function is the same described in Figure S4A. B) *Imadjust* function operates a saturation of 1% of data at low and high intensities.

### 3. Global thresholding and single-cell tracking

#### a. Global Thresholding

Before and after applying the Otsu algorithm (*Imbinarize* function), some filtering actions might be required to improve the image and mask quality, as for bacterial cells (Figure S11). Of note, parameters in these functions (e.g. the structuring element object within the *Imerode* function) need to be optimised by trial and error, depending of the images; if segmenting images acquired under different settings, some of these functions might be redundant.



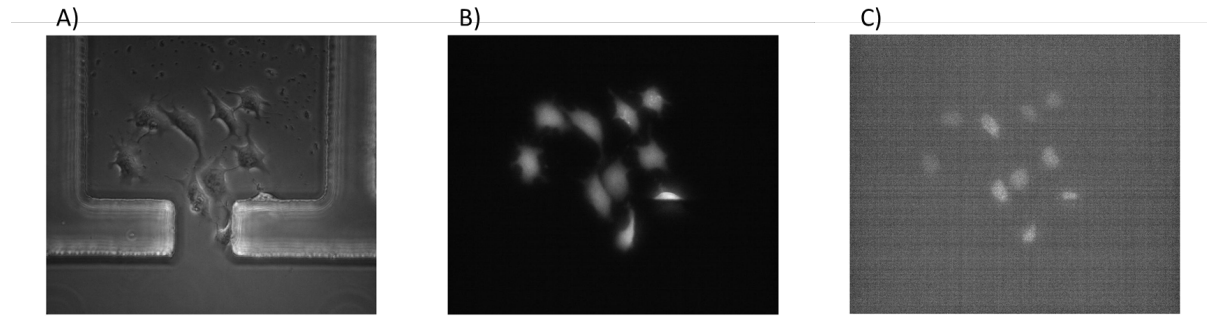
**Figure S11. Global thresholding and filters.** A) *Medfilt2*\* filter: as described in Figure S2B, this function performs a median filtering of a matrix. B) *Adapthisteq*\* function: as described in Figure S2C, enhances the contrast of images. C) *Imbinarize* function: as described in Figure S3A. D) *Imdilate*\* function: as described in Figure S3B. E) *Imfill*\* function: described



in Figure S3C. F) *Imerode\** function: erodes images referring to the shape of a structuring element object (STREL). \* Depending on the raw images, some of these functions might be redundant.

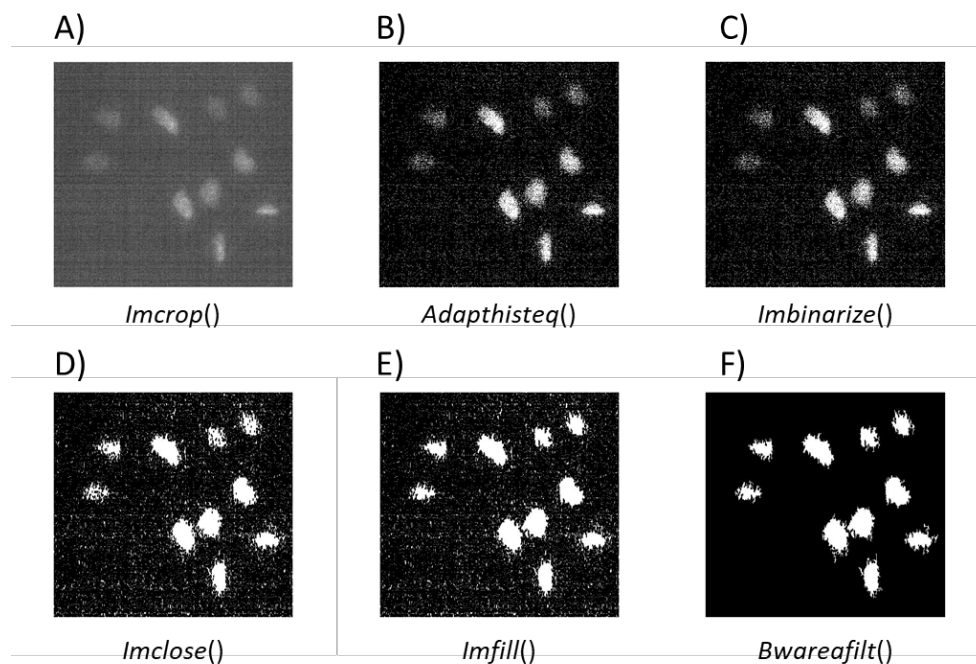
### b. Single-cell tracking

Single-cell tracking is another feature embedded in our code, always relying on the Otsu method for mask recognition. Here we show an application to a time-lapse where dual-reporter mESC images of three channels were collected (Figure S12); to clearly identify individual cells we used an H2B-tag reporter for nuclei (Figure S12C).



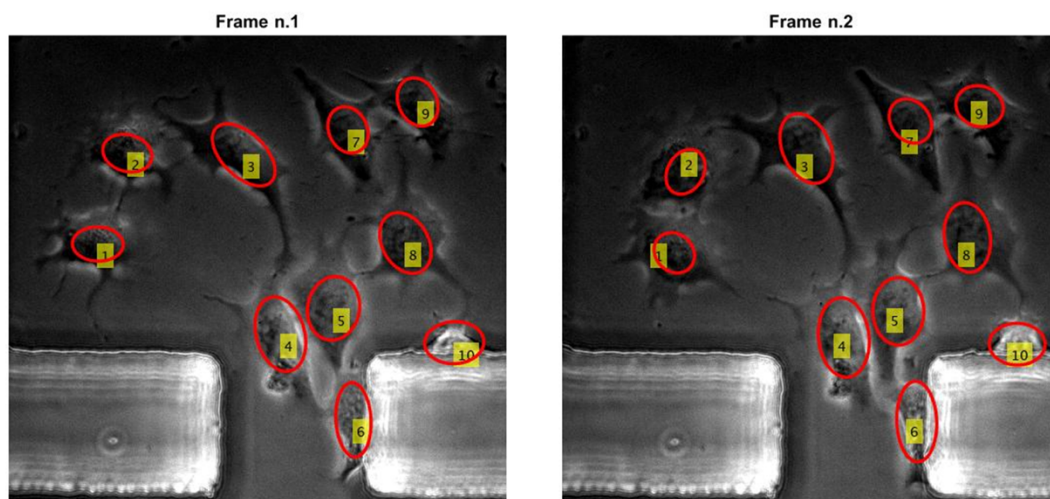
**Figure S12. Raw images of dual-reporter mESCs used in single-cell tracking experiment.** A) Phase contrast image. B) mCherry fluorescence image. C) H2B nuclear tag image.

The tracking code firstly computes a rough mask detecting nuclei intensity by applying filters and thresholding functions (Figure S13), as defined within the user-defined function *mask\_n\_Track*. Then, the mask is used to identify and label connected objects (i.e. individual cells). We employed the mCherry fluorescent image (expressed both in the nucleus and in the cytosol, Figure S12B) to calculate the average cell area; using the *Bwareafilt* function, objects whose area is less than the 5% of the average are removed (Figure S13F). This step might be neglected if the nuclei images have a higher contrast as compared to the background than in our exemplar experiment.



**Figure S13. Filtering functions.** A) *Imcrop* function: the original raw image is manually cropped in the first timeframe. The crop function is the same described in Figure S4A. B) *Adapthisteq* function: as described in Figure S2C. C) *Imbinarize* function: as described in Figure S3A. D) *Imclose* function: the morphological open operation is a dilation followed by an erosion, using the same structuring element (STREL) for both operations. E) *Imfill\** function, as described in Figure S3C. F) *Bwareafilt* function: as described in Figure S5A.

Using the Matlab function *Bwlabel*, the code then associates to each object a label, that is going to be maintained throughout the whole time-lapse experiment. For each object, a centroid is evaluated using the Matlab function *Regionprops*; this function also computes the major and minor axis of the ellipses defining the dimensions, their orientation and the area of cells nuclei. The algorithm then performs tracking by using the custom functions *MaskGenerator* and *Tracking\_fun* (the latter encompassing *SaveStructEllipse*, *TraceOldNew* and *UpdateOBJS* functions). *MaskGenerator* generates multiple masks, one for each cell detected, overlapping the rough mask previously computed with ellipses data generated by *Regionprops*. Then, from the second timeframe onwards, the centroids among contiguous timeframes are compared. Each object (i.e. cell) is saved by *SaveStructEllipse* along with the ellipse associated data and computed masks, and a minimum distance problem is solved by the *TraceOldNew* function to identify an object in two consecutive frames. The label associated to each cell is kept or swapped with another cell during the experiment according to the *updateOBJS* method; a cell label is removed only in case the cell goes out of focus or leaves the cropped region. Some parameters of the routine (e.g. the maximum nuclei dimension) need to be fixed by the user given the acquisition settings and specific cell morphology. In Figure S14 we show two consecutive frames and the relative tracking of individual cells.



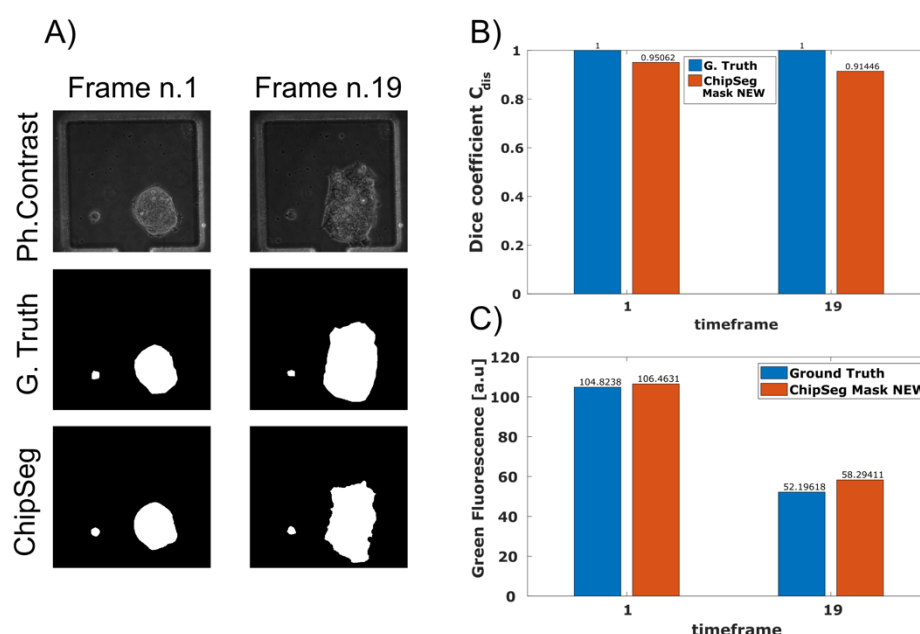
**Figure S14. Cell tracking.** The tracking of individual cells using the functions *Mask\_n\_Track* and *Tracking\_fun* in two consecutive frames is shown.

## 5. Fluorescence calculation and segmentation evaluation

Once a mask is available, it can be applied to the frame of interest to evaluate the associated fluorescence value at cell population (Figure 1D, Supporting Movie 2) or single-cell (Figure 1E, Supporting Movie 3) level; an average background fluorescence is subtracted to cell

fluorescence. The background intensity value can be either computed by cropping an empty area of the frame or by defining the reversed logical mask. In the latter case, the background is then obtained as it would be for cells fluorescence, i.e. by applying the mask to a frame and averaging over the non-covered area. When a fluorescent tag is inserted into mammalian cell nuclei, a nuclei normalization can also be performed within fluorescence computation.

The performance of ChipSeg using global thresholding for mammalian cells was compared to ground truth, manually annotated images using the Dice coefficient, as for bacterial cells (see above). Results on two timeframes of the time-lapse in Supplementary movie 2 show > 90% overlap of the ChipSeg-computed vs manually annotated masks (Figure S15B), and comparable fluorescence quantification (Figure S15 C).



**Figure S15. Metrics calculation.** A) Phase contrast and corresponding ground truth and ChipSeg-computed masks at two specific time points of the experiment in Supplementary Movie 2. B) Dice coefficient for ground truth and ChipSeg-computed masks. C) GFP quantification for ground truth and ChipSeg-computed masks.

## References

1. Otsu, N., A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics* **1979**, 9 (1), 62-66.
2. Niblack, W., *An introduction to Digital Image Processing*. Prentice-Hall: 1986.
3. Shannon, B.; Zamora-Chimal, C. G.; Postiglione, L.; Salzano, D.; Grierson, C. S.; Marucci, L.; Savery, N. J.; di Bernardo, M., In Vivo Feedback Control of an Antithetic Molecular-Titration Motif in Escherichia coli Using Microfluidics. *ACS Synth Biol* **2020**, 9 (10), 2617-2624.

4. Feng, Y.; Kawrakow, I.; Olsen, J.; Parikh, P. J.; Noel, C.; Wooten, O.; Du, D.; Mutic, S.; Hu, Y., A comparative study of automatic image segmentation algorithms for target tracking in MR-IGRT. *J Appl Clin Med Phys* **2016**, *17* (2), 441-460.