2016-11. Outline of segmentation procedure with improved tracking and new cell axes calculations.
Raw data consists of 512 by 512 images, with one fluorescence and one brightfield image per field of view, each accompanied by a blank for background subtraction (same exposure time, no illumination). These four images are acquired every 5-6 min for each field of view, with the number of fields of view chosen for continuous acquisition (~ 38).
insert sample images here.

SEGMENTATION

Initially, all images are segmented independent of order. For each background-subtracted fluorescence image, the algorithm proceeds in the following steps: 1) find regions of interest, i.e. the channels trapping each cell line; 2) within each region of interest, threshold and perform morphological operations to isolate cells from the background; 3) find cell boundaries and separate any artificially fused cells; and 4) remove objects smaller than 30 pixels in area. The remaining objects are passed to the tracking algorithm after all images have been segmented.

An order-independent approach was used to increase computational speed: on MP's current system, images are segmented at ~ 40 fps. Corrections to the initial segmentation are performed under the tracking algorithm.

Background subtraction. Simple background subtraction is performed by subtracting the respective blank (same exposure, no illumination) from each fluorescence image. Any negative values are set to 0.

Channel detection. The region of interest in image is defined by the channels trapping each line of cells. The x- and y-cross sections of intensity are used to detect these regions (Fig. 1a). For speed, the image is first subsampled at every 5th pixel.

Denote the corresponding intensity matrix I such that the element I_{ij} is the intensity of the pixel in the i^{th} row and j^{th} column. Note that x- and y-profiles were analysed differently because channel edge detection requires different levels of sensitivity depending on the dimension. Specifically, in x we need only find one high intensity region that approximates the height of all channels while in y we need multiple regions of relatively high intensity, allowing for cell lines of markedly different levels of fluorescence. The x-cross section is defined by the maximum intensity of each column of pixels. For the j^{th} column:

$$I_x(j) = \max\{I_{ij} | i = 1, 2, ..., n\}$$

Note since images are subsampled, $n = 103$. The y-cross section is defined by the mean intensity of each row of pixels. For the i^{th} row:

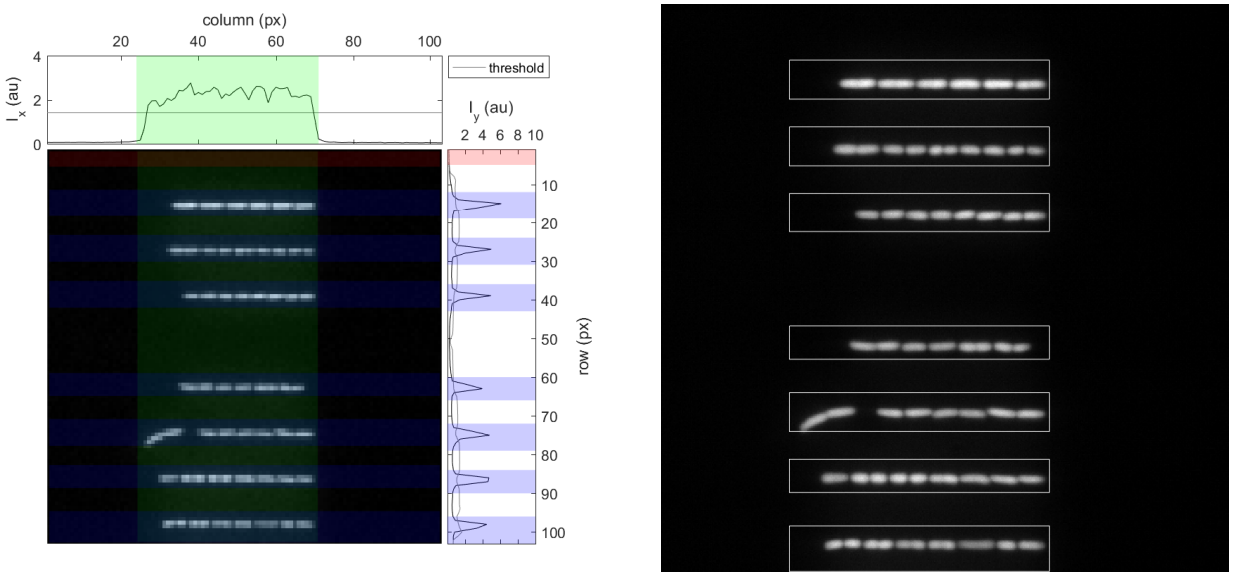
$$I_y(i) = \frac{1}{n} \sum_{j=1}^{n} I_{ij}$$

Each profile was then thresholded. For x, I used Otsu's threshold for the values in $\{I_x(j) | j = 1, 2, ..., n\}$ (Fig. 1a, top panel, grey line). A more localized method is needed for y: since channels are relatively narrow and separated by larger dark bands, I calculate the sliding average of I_y, using a 21 pixel window. The result is an oversmoothed I_y (Fig. 1a, right panel, grey line). Subtracting this from the original I_y leaves only high-intensity peaks corresponding to each channel.

The edges of each high intensity region are then rescaled to their appropriate positions in the original image and expanded on all sides by a 10 pixel cushion. Finally, each region is checked for sufficiently high intensity pixels and discarded if fewer than 10 pixels are found (Fig 1a, red). 'High' intensity in this case entails pixel intensity ≥ 500 a.u., or SNR $\gtrsim 6{:}1$. Note that for the 30 dimmest cells of the dimmest strain (P5ter3, replicate 20160221), a single cell has 31-190 (mean 93) pixels whose intensity is greater than 500 a.u. The average pixel intensity for these dim cells is 896 a.u. (SNR $\approx 10{:}1$). The final regions of interest are shown in Fig. 1b.

Cell detection. Each region of interest is processed independently. The following procedure is outlined in Fig. 2. First, intensity values are smoothed using a 2D median filter and the image is then resized (3x) using bilinear interpolation. The greyscale image is then thresholded using a modified Otsu's method (see Fig. 3) and morphologically dilated with a 3 pixel radius disk. For each object that passes the threshold, the curvature of its boundary is then calculated. Points of negative curvature are used to separate cells that have been artificially fused into one object (see Fig. 4). Finally, any objects that are smaller than 30 pixels in area are removed (see Fig. insert small area fig).

This approach based on boundary curvature turns the problem of segmentation from 2D to 1D, and minimizes the use of several morphological operations often required for e.g. watershed-based separation. Rather



(a) (b)

Figure 1: (a) Channel detection by filtering the x- and y-intensity profiles of the subsampled image.The x-profile defines the height for all channels in a given image (green), while y defines channel width (blue). Regions that pass the filter but do not contain any high intensity objects are removed from further analysis (red). (b) Channels detected after intensity filtration, shown on original 512 by 512 image.

than smoothing cell edges by morphological opening and/or closing, images are instead slightly overthresholded then dilated in order to retain the regions of sharp negative curvature anticipated at the fusion points between two cells. The resulting segmentation is comparable to that produced by edge detection filters (e.g. taking the Laplacian of the gaussian), but avoids many of the time-consuming morphological operations (see Fig. 5).

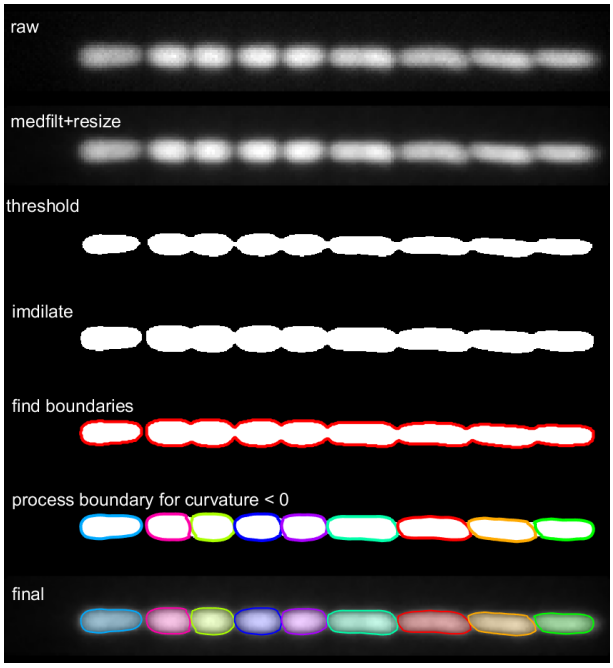


Figure 2: Cell segmentation procedure. Sample corresponds to the 6th channel from the top in Fig 1b.

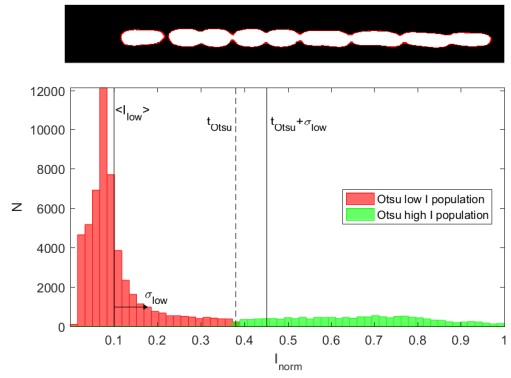


Figure 3: Overthresholding procedure. The Otsu threshold t_{Otsu} for the image is calculated using MATLAB's `graythresh`, separating pixels into 'high' (green) and 'low' (red) intensity populations. This threshold is then raised by the standard deviation of the low intensity population. The top panel shows the difference between the modified (white) and original (red) Otsu thresholds.

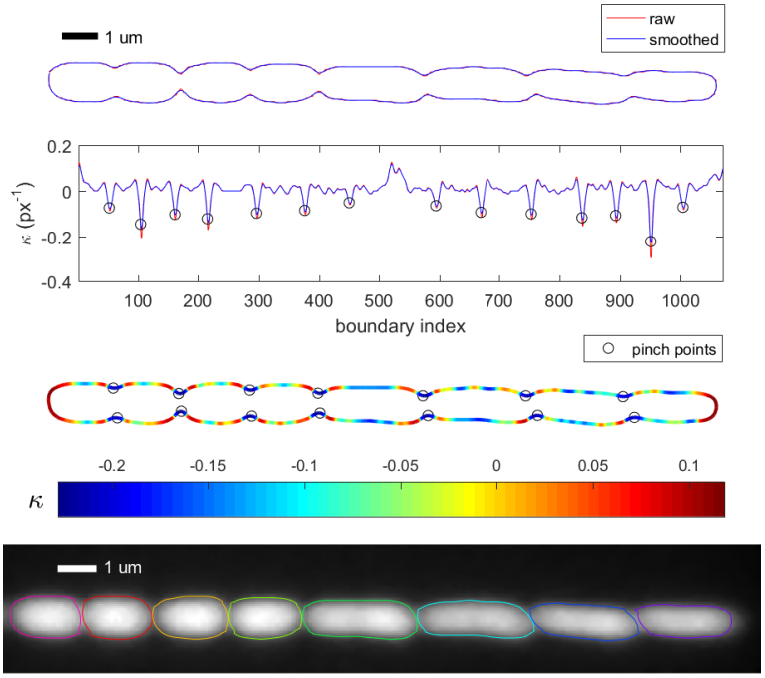
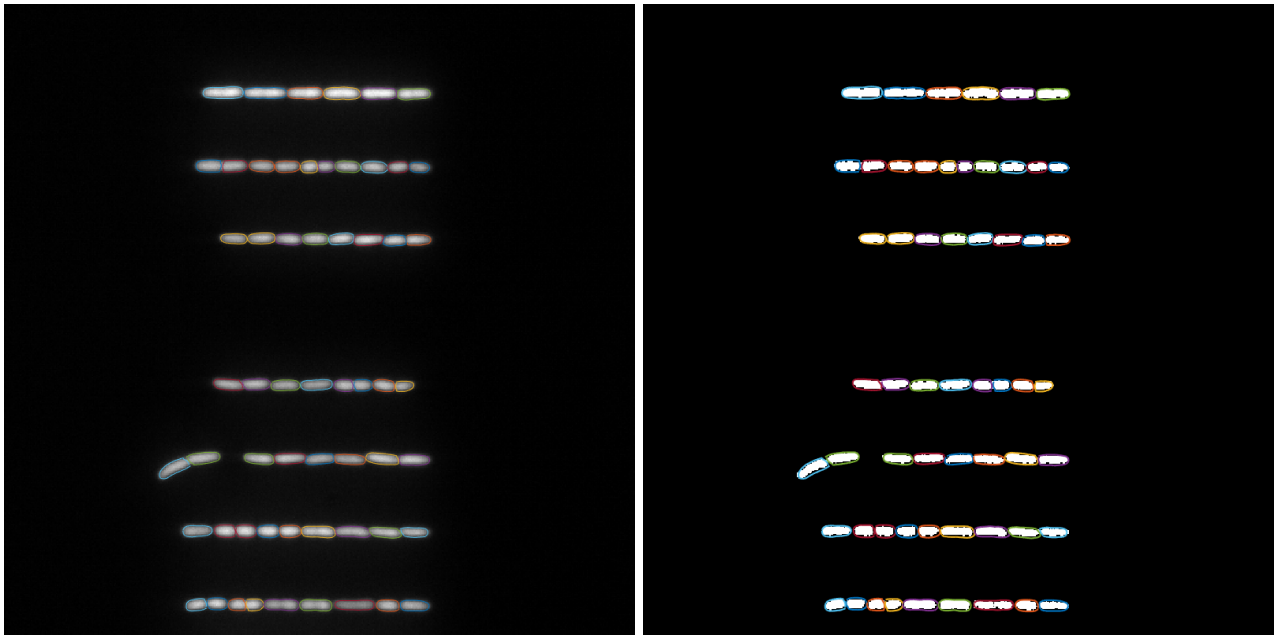


Figure 4: Boundary processing to separate fused cells. Ordered boundary points are smoothed using a 3rd degree polynomial over a 9-index window or ∼3 pixels in original image (top panel). Curvature (κ) is calculated from these boundary points and smoothed using a sliding window average of 5 indices to locate the local minima of basins of negative curvature, 'pinch points' (open black circles, second and third panels). Pinch points are then joined to their the nearest neighbours by a third degree polynomial such that all boundary points form an enclosed loop (bottom panel).



(a) (b)

Figure 5: Comparing current segmentation method to Laplacian Segmented boundaries of all cells overlayed on (a) the original fluorescence image and (b) regions found by filtering the Laplacian of the gaussian of the original fluorescence image. The regions in (b) were produced by calculating the Laplacian of the gaussian using MATLAB's built-in `fspecial` and `imfilt` and thresholding for regions where the Laplacian is negative. Noise in the background was removed by filtering objects fewer than 20 pixels. 'Holes' in objects were filled using `imfill` and spurious dark pixels along the edge of cells were filled using with `bwmorph`, set to 'majority.'

LINEAGE TRACKING AND CELL METRICS

Figure 6: Lineage identification by matching microchannels after image registration.

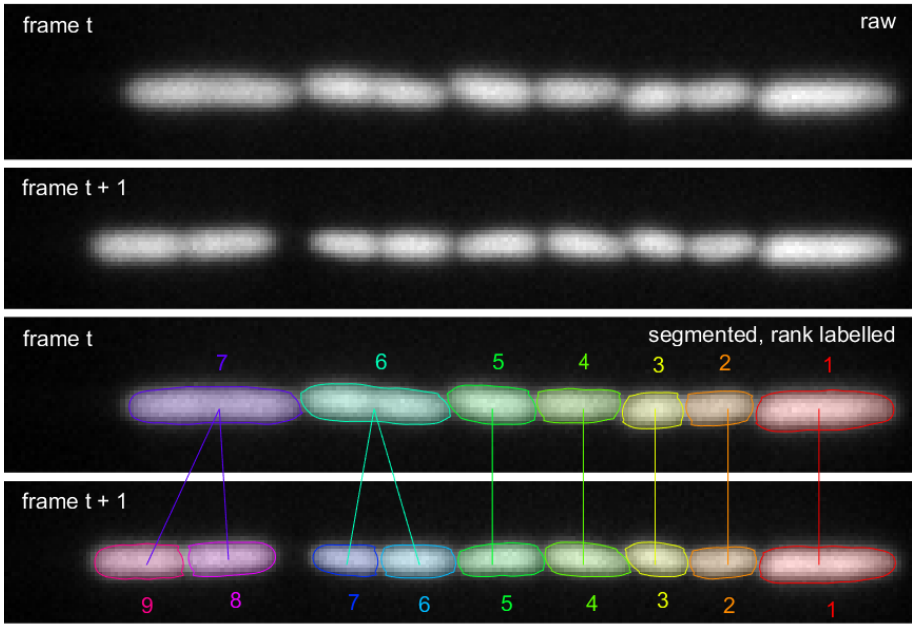


Figure 7: Cell tracking according to rank conservation with division events flagged by sudden decreases in cell size.

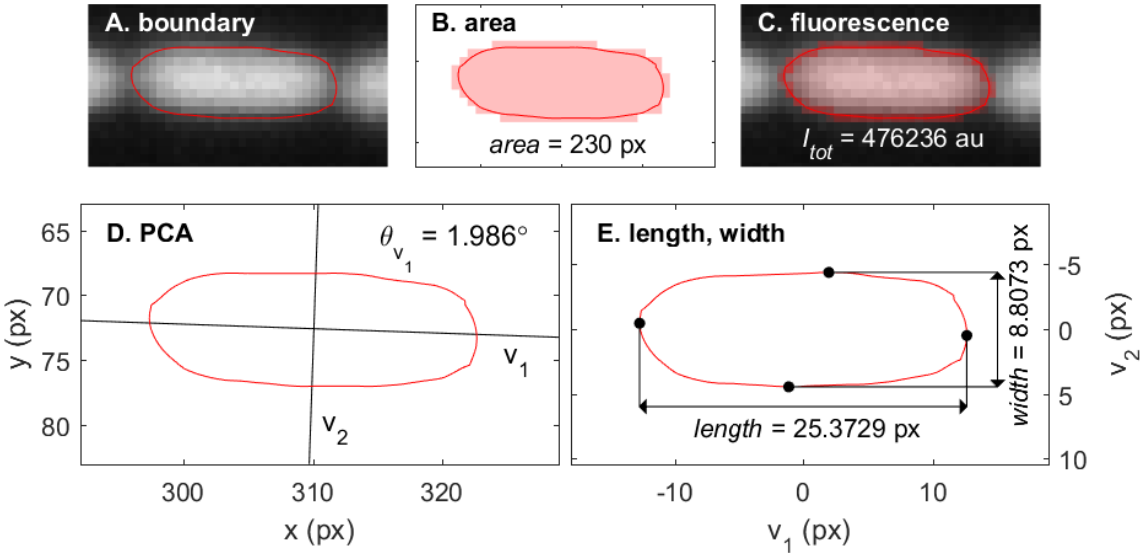


Figure 8: Cell measurements made following image segmentation.