SUPPLEMENTARY INFORMATION FOR PANLILIO *ET AL.*

ACCUMULATION AND RESET OF A CONSTITUTIVE PROTEIN TRIGGERING DIVISION EXPLAINS DYNAMICS OF CELL SIZE CONTROL ACROSS NUTRIENT UPSHIFTS.

SUPPLEMENTARY FIGURES

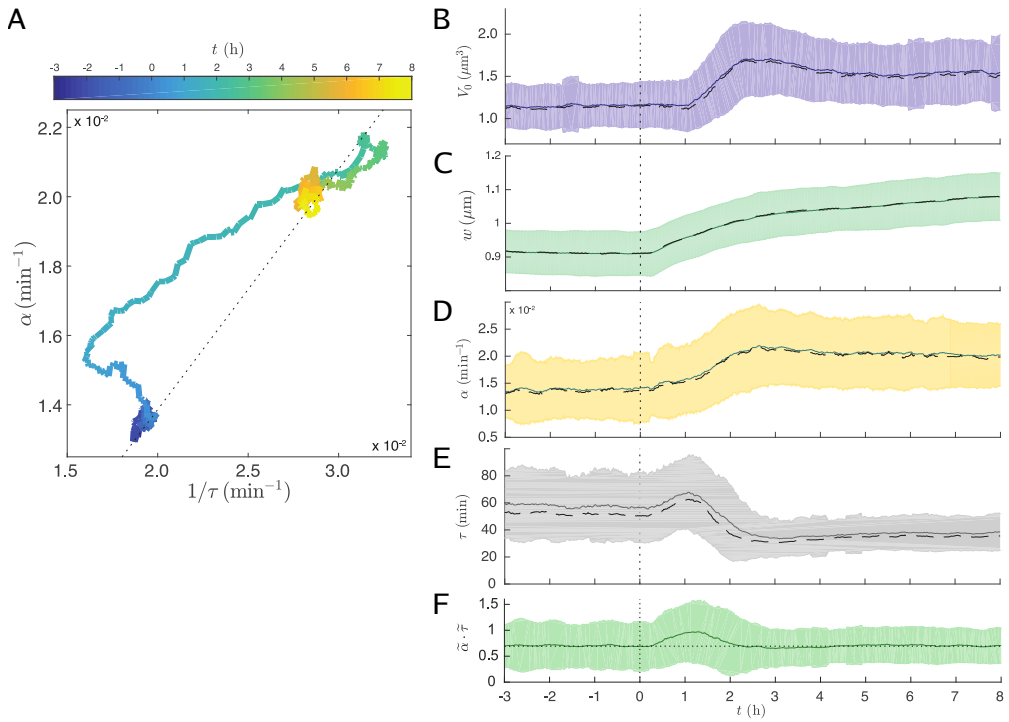

FIG. S1 Complex dynamics of growth and division processes across nutrient shifts. A) Time trace in the (mean) $1/\langle\tau\rangle$-$\langle\alpha_{\text{inst}}\rangle$ diagram. These two quantities reach fixed points where they are proportional to each other (by a factor of log(2), dotted line) in steady-growth conditions, but they deviate forming a complex pattern during the shift, due to the different time scales involved. B) Mean birth volume initially overshoots its target increased value in rich media. C) Mean cell width relaxes slowly in the new condition. D) Mean growth rate as a function of time into the experiment. E) Mean interdivision time as a functon of time in the experiment. Note that this quantity has an early increasing trend, opposite to the target of the new condition (i.e., cells whose target is to divide faster, initially divide with a slower pace) increasing trend). F) The product of median growth rate and median interdivision time deviates from a constant value during the shift ("non adiabatic" transition). Media shift time is indicated at vertical dotted line. Unless otherwise stated, solid line indicates observable mean, dashed line indicates median, and shaded regions indicate SD for all time series.

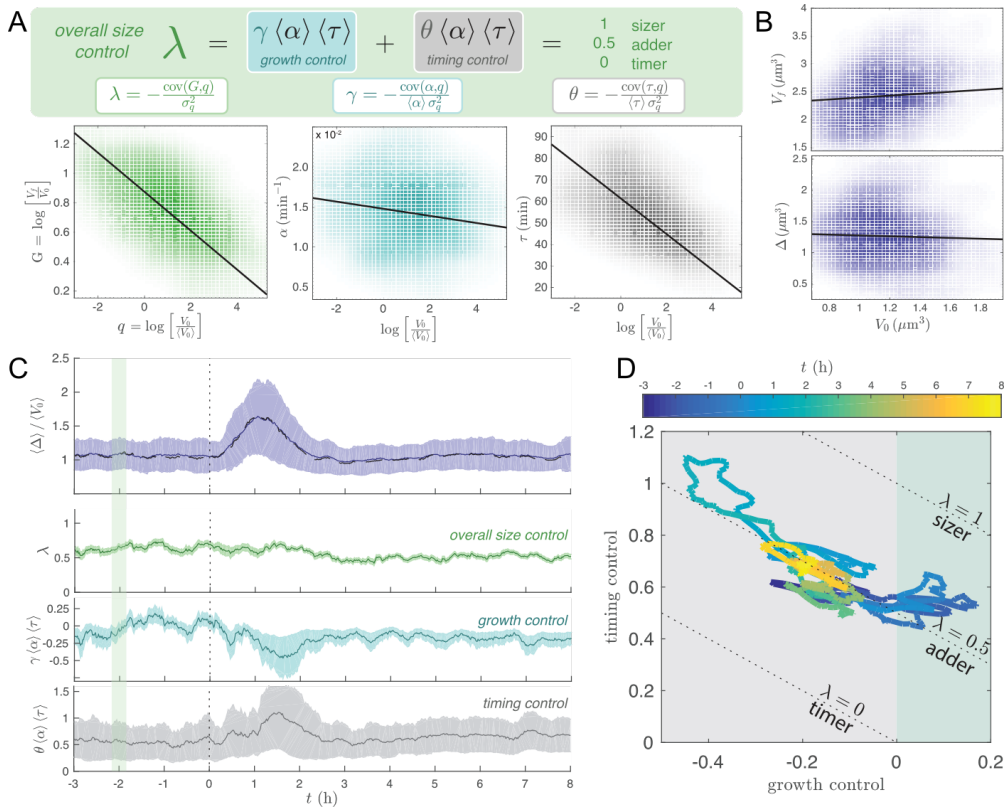

FIG. S2 Near-adder behaviour throughout the shift is preserved through compensation of division timing control and growth-rate control despite complex dynamics between growth and division processes. Media shift indicated at dotted vertical line. A) The size-growth plot measuring multiplicative growth vs logarithmic initial size quantifies size corrections with the slope λ (=1 for sizer, 0 for timer, 1/2 for adder) and can be split into inter-division time and growht-rate contributions (1). B) Equivalent to these slopes, one can measure the slope of the added size or final size *vs* initial size (1; 2). C and D) Dynamics of size correction across the shift. The slope of the size-growth plot λ remains constant, and timing/growth control variables compensate to maintain near-adder correlations.

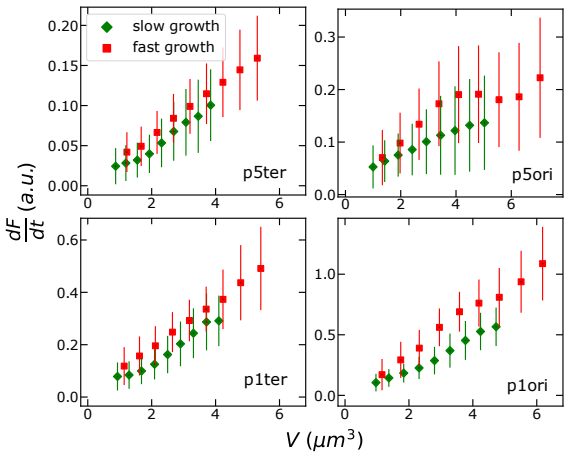

FIG. S3 GFP production rate from the ribosomal and constitutive promoters used in this study is proportional to cell volume. The data shown here come from measurements of GFP expression from the P5 and P1 and promoters in different chromosomal locations (different panels) in slow (M9 + 0.4% glucose, green diamonds) and fast (M9 + 0.4% glucose + 0.5% casamino acids, red squares) steady growth conditions. Points on the y axis are averages of discrete derivatives of fluorescence-versus-time tracks $dF(t)/dt$, correlated to measured volume (x axis). Error bars indicate the standard deviations on binned averages.

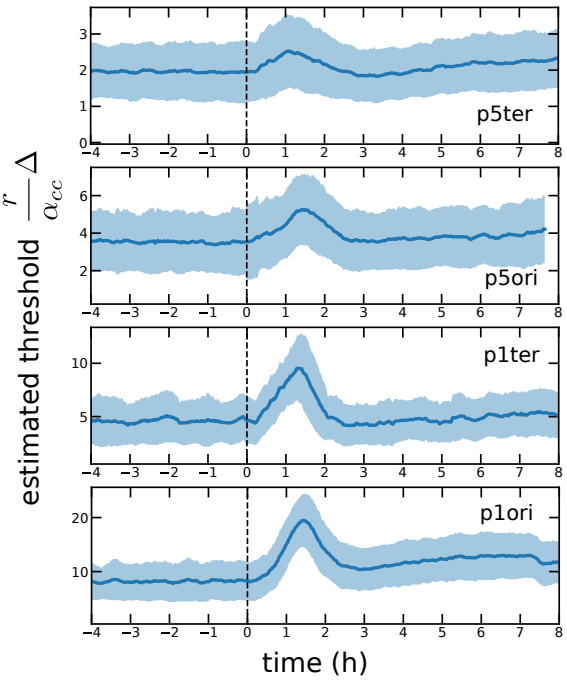

FIG. S4 The estimated threshold for an accumulator model is similar across conditions for all tested promoters. Assuming the accumulator model described in the main text, the plot evaluates a running average of the quantity $\frac{r}{\alpha_{\rm cc}}\Delta$, which, at steady state, corresponds to the theoretical prediction of the threshold value N^* of the accumulator model (see Eq. (2) of the main text). The data are compatible with a threshold that remains roughly constant across the two conditions. The production rate $r = 1/VdF/dt$ is averaged over cell cycles, and the growth rate $\alpha_{\rm cc}$ comes from exponential fits of the volume *vs* time data of the same cell cycle. Both variables are associated to the division time of a cell.

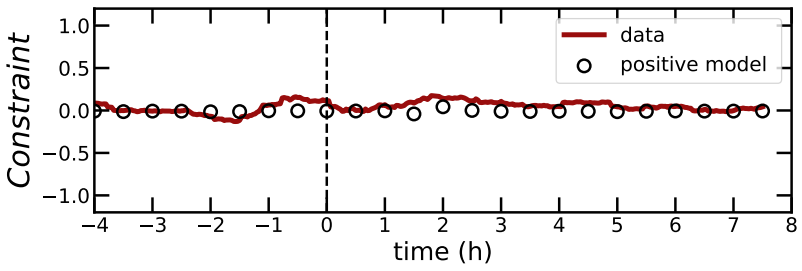

FIG. S5 Both data and model fulfill the expected constraint $\langle e^{G(1-\lambda)}/(1+\zeta)\rangle = 1$, where $G = \log(V_f/V_0)$, λ is the slope of the size-growth plot and ζ is the slope of the adder plot. This constraint holds theoretically beyond steady state, and the data confirm this prediction. The constraint can be derived from the definition $\zeta = d\Delta/dV_0$ noting that, if $q = \log(V_0)$, by chain rule $d\Delta/dq = (d\Delta/dV_0)(dV_0/dq) = \zeta e^q$, and equally, since $\Delta = V_f - V_0$, $d\Delta/dq = e^q e^G(1+\lambda)$. The verification of this constraint provides a useful consistency check for our analysis.

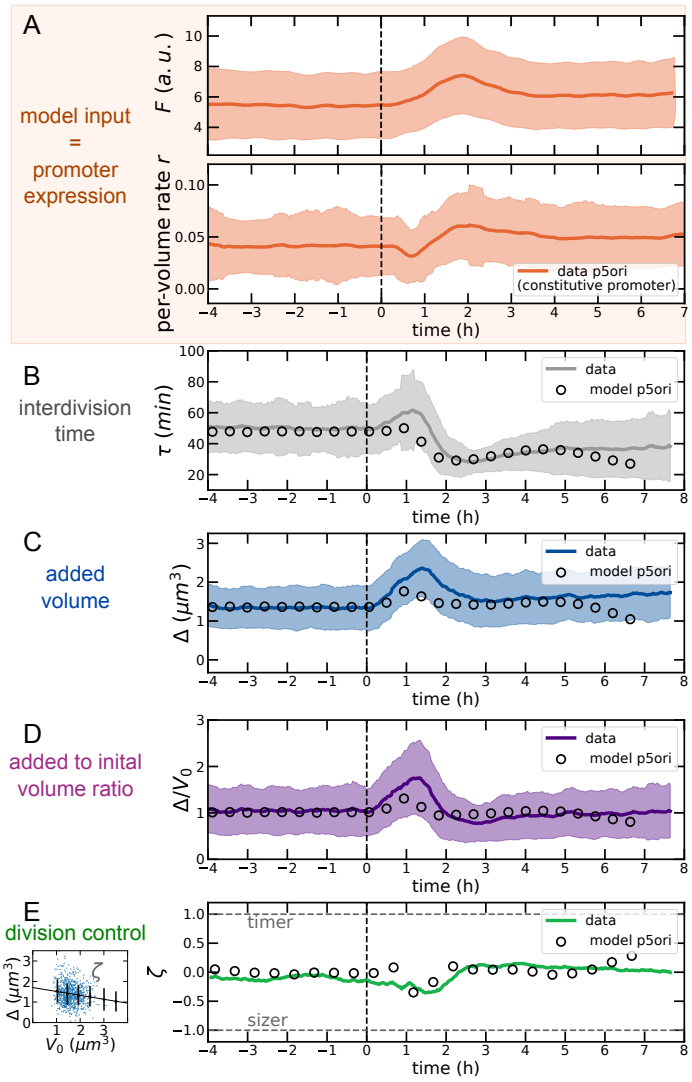

FIG. S6 Results of the divisor protein model using as input the measured volume-specific production rate r from the *P5 constitutive promoter* inserted close to the *replication origin*. All panels as in Figure 4 of the main text.

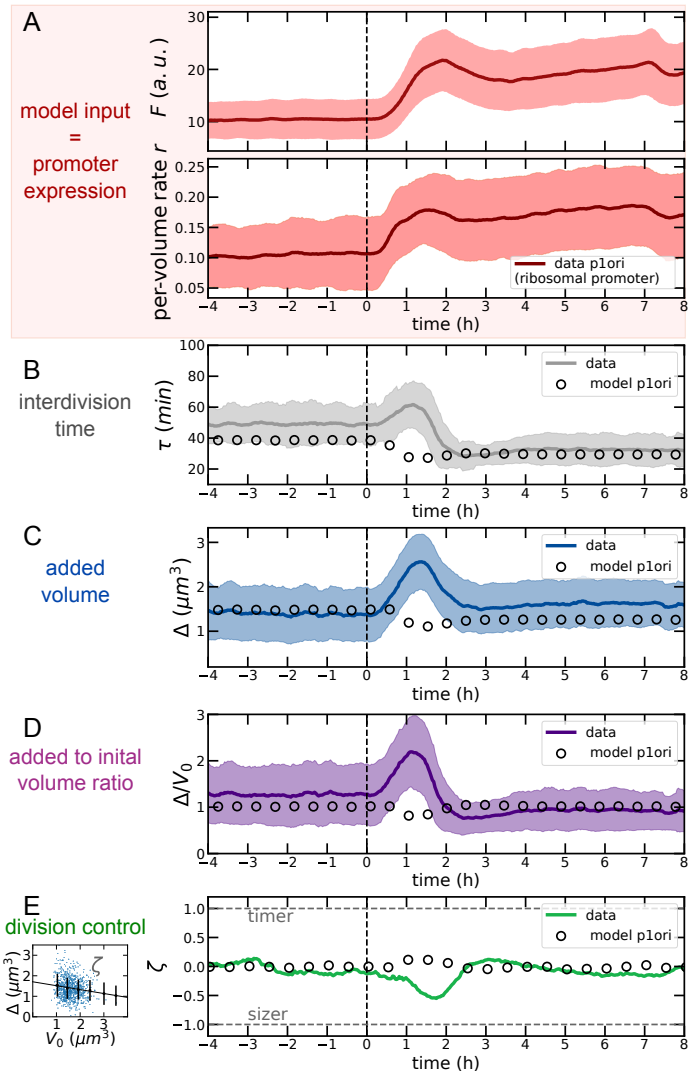

FIG. S7 Results of the divisor protein model using as input the measured volume-specific production rate r from the *P1 ribosomal promoter* inserted close to the *replication origin*. All panels as in Figure 4 of the main text.

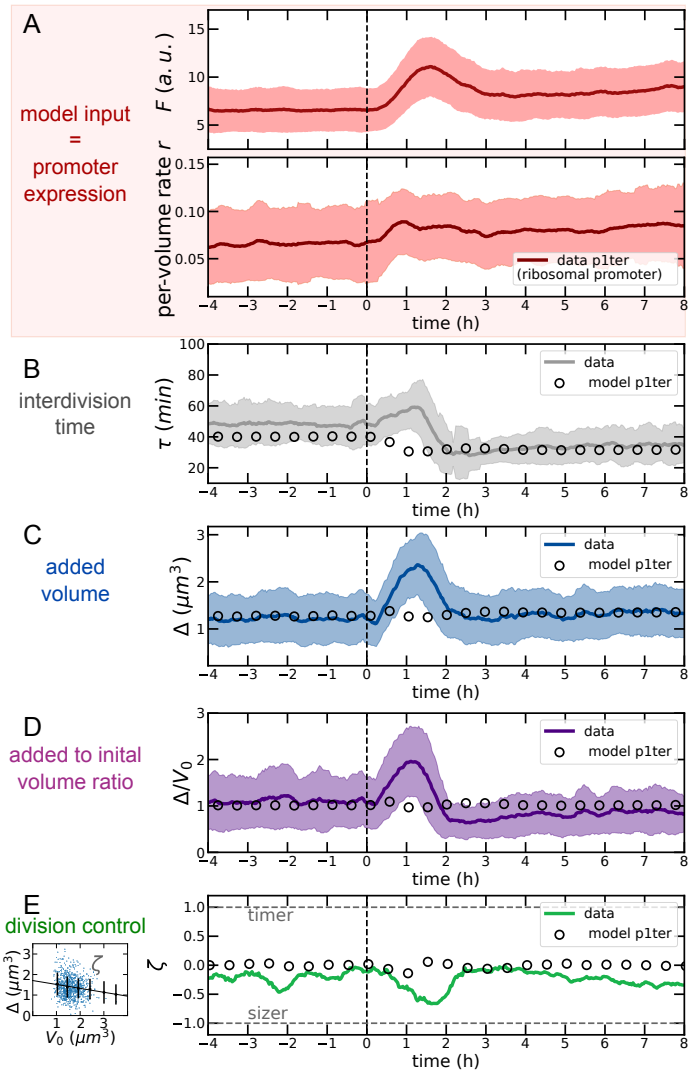

FIG. S8 Results of the divisor protein model using as input the measured volume-specific production rate r from the *P1 ribosomal promoter* inserted close to the *replication terminus*. All panels as in Figure 4 of the main text.

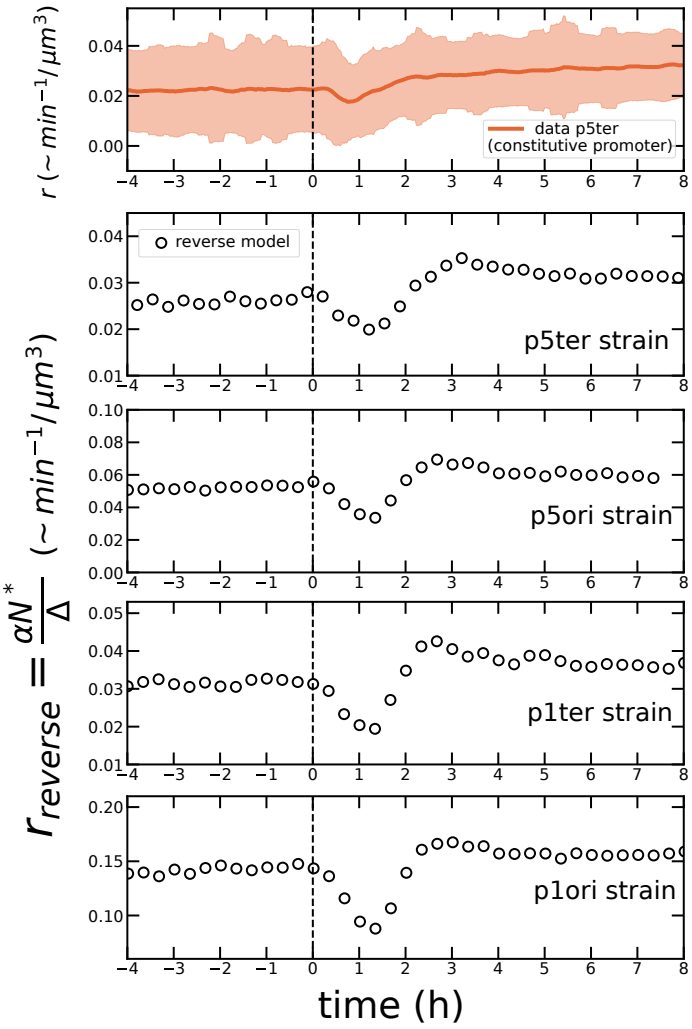

FIG. S9 The theoretical volume-specific production rate of the adder molecule derived from data on growth rate and added size transiently decreases after the shift, before reaching a higher plateau. The top panel is the measured volume-specific rate of GFP from the p5ter promoter shown in Fig. 4 of the main text. The other plots are binned averages of $r_{\text{reverse}}(t) = \frac{\alpha_{\text{cc}} N^*}{\Delta(t)}$, shown in the different panels to be robust across shift experiments with strains carrying different promoters. The reverse argument assumes that N^* is constant after the shift. In these plots, N^* is set to the value of 2 (in arbitrary units, taken from Supplementary Figure S4) before and after the shift, in order to make the model match the data also quantitatively. The average uses experimental values of $\Delta = V_f - V_0$ and of the growth rate α_{cc} comes from exponential fits of the volume *vs* time data of each cell cycle, and the variables are associated to the division time of a cell.

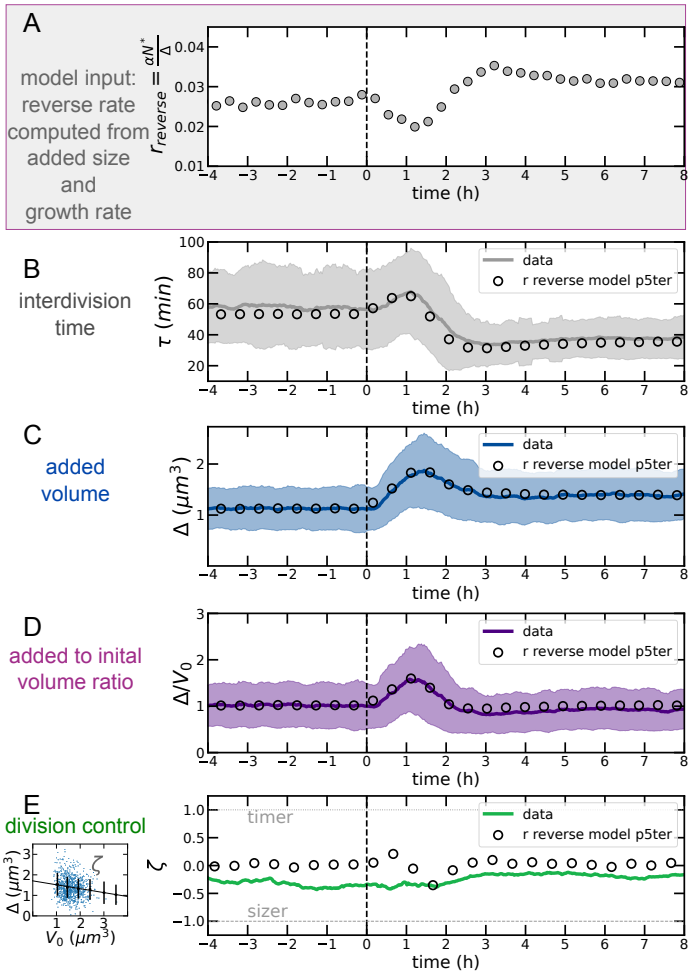

FIG. S10 Results of the divisor protein model using the reverse production rate $r_{\text{reverse}}(t)$ inferred as described in Fig. S9 using the strain carrying the P5 constitutive promoter close to the replication terminus. All panels as in Figure 4 of the main text.

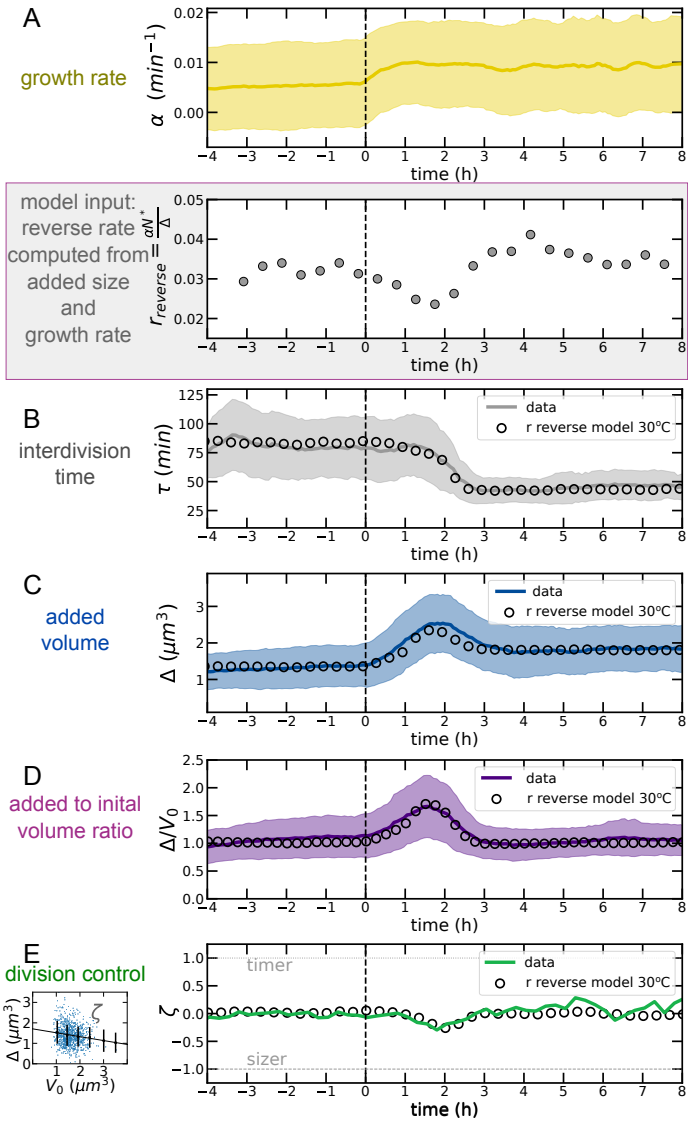

FIG. S11 Results of the divisor protein model using the reverse production rate $r_{\text{reverse}}(t)$ for an experiment at 30^{o}C using the strain carrying the P1 ribosomal promoter close to the replication origin. All panels as in Figure 4 of the main text, except for panel A, which also includes the growth rate for this different experiment.

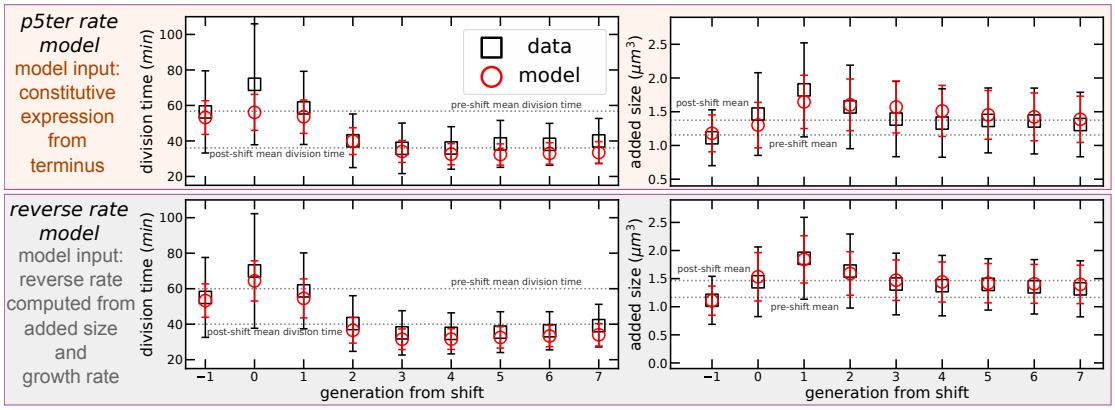

FIG. S12 Cell-division behavior by generation from/to the nutrient shift. The generation index is set to 0 for the cells that see the shift during their cell cycle. The plots compare data (circles) with the forward model (top) using P5ter promoter production (see Fig. 4 in the main text) and with the reverse model (bottom, see Fig. S9 andS10). Both interdivision time (left panels) and added size (right panels) show overshoots (equivalent to the ones seen in Fig. 2 of the main text) that are reproduced by the model. Error bars are standard deviations

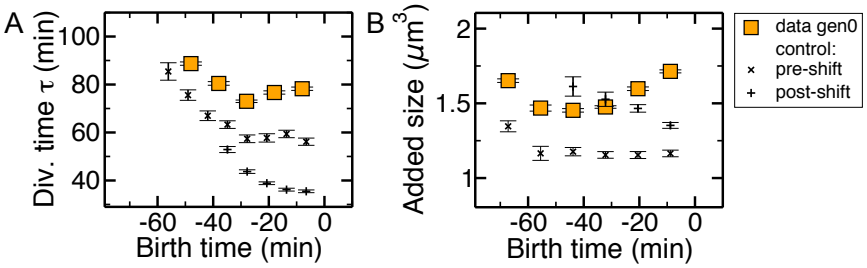

FIG. S13 The behavior of the generation seeing the shifts differs from the pre- and post-shift background in a cell-cycle dependent way. The plots report mean interdivision time (A) and added size (B) as a function of birth time (time 0 is the shift) for "generation 0" cells, which see the nutrient shift during their cell cycle. Control data report the same quantity for cells in the steady pre-shift (crosses) and post-shift condition, with respect to an arbitrary reference time. Error bars are standard errors.

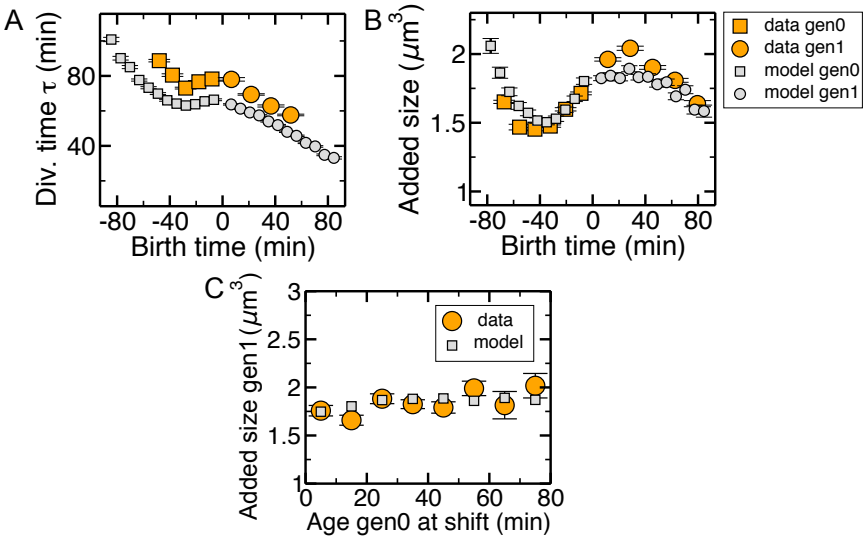

FIG. S14 The behavior of the generation seeing the shift and the one after is reproduced qualitatively by the accumulator model. The top panels report interdivision time (A) and added size (B) for for "generation 0" (orange squares) cells, which see the nutrient shift during their cell cycle and for their daughters, "generation 1" cells (orange circles). Grey squares and circels show the equivalent results for the reverse model. C) The added size of generation 1 cells does not depend on the cell-cycle time at which their mothers saw the shift, in both model and data. Error bars are standard errors.

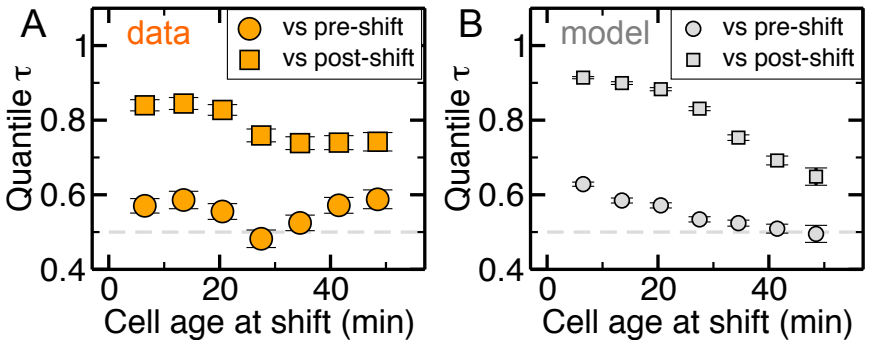

FIG. S15 The cell-cycle pattern in division delays differs between model and data, suggesting that the checkpoint is also related to additional processes not described by the model, such as chromosome replication and segregation. The plots report the mean quantile of the interdivision time for "generation 0" cells, which see the nutrient shift during their cell cycle, as a function of the time (from birth) when they see the shift. The y axis quantifies the position of the cell cycle time of a cell compared to the cells of its same age that do not see the shift. A quantile of 0.5 means that the cell behaves as pre- or post-shift cells. In the model, cells that see the nutrient shift late in their cell cycle do not modify their cycle duration compared to the pre-shift conditions, while in the data these cells delay their division. Error bars are standard errors.

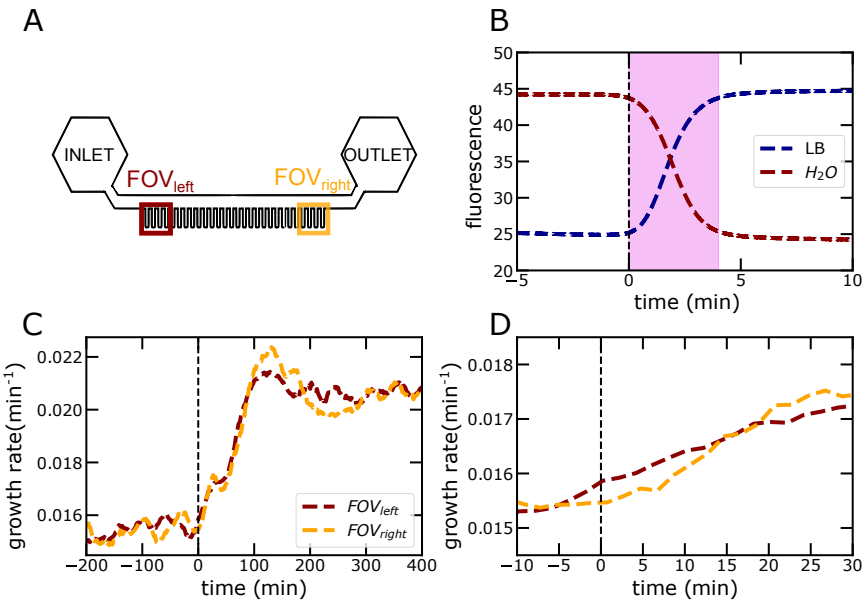

FIG. S16 Test of filling time and delays in nutrient availability within the microfluidic device. A) Scheme of the device with the outermost fields of view labelled in dark red and orange. B) Experiment loading the device with fluorescent LB medium and monitoring mean fluorescence within the channels. Side channels start seeing the new medum immediately, and completely fill in about 4 minutes (purple shaded area) once the medium enters the main channel ($t = 0$). C) The delay between growth rate changes in the experiment averaged in the outermost fields of view in the device is small. D) Zoom of panel C showing that the delay is about 5 minutes.

SUPPLEMENTARY INFORMATION ON THE FRAMEWORK FROM CADART *ET AL.* 2018.

This appendix describes the framework from refs. (1; 2) used in Supplementary Fig. S2. This analysis assumes a random exponential growth rate that may be dependent on a cell's initial size. The quantity $G = \log(V_f/V_0)$ is the overall multiplicative growth. Size control here is quantified as the linear correction to deviations from the mean logarithmic size (i.e., smaller-born cells grow more than larger-born ones). Three key parameters indicating the degree of size control fall from the procedure (1; 2). The overall size control is $\lambda = -\frac{\mathrm{cov}(G,q)}{\sigma_q^2}$, where $q = \log V_0$. The value of λ signals the size control strategy in use. Specifically, $\lambda = 1$ corresponds to a sizer (constant target size); $\lambda = 0$ represents no control; and $\lambda = 0.5$ denotes an adder. The growth and timing components of size control are, respectively, $\gamma = -\frac{\mathrm{cov}(\alpha_{\mathrm{inst}},q)}{\langle\alpha_{\mathrm{inst}}\rangle\sigma_q}$ and $\theta = -\frac{\mathrm{cov}(\tau,q)}{\langle\tau\rangle\sigma_q}$. A positive value for either γ or θ indicates a homeostatic effect while negative values mark a negative contribution to size control. for example, for positive θ, smaller-born cells grow for longer periods of time, and for positive γ they grow at a faster rate. Viceversa, for negative γ, growth rate contributes to variability in the initial size. The three parameters yield a quantitative method to determine the contribution of timing and growth-rate corrections to size homeostasis. They are mathematically related through the expression

$$\lambda = \langle\alpha_{\mathrm{inst}}\rangle \langle\tau\rangle (\gamma + \theta) \ ,$$

which we find verified in the data (Fig. S2C). At steady state, this analysis confirms that *E. coli* division control is adder-like and that this strategy is realized primarily through timing control, while fluctuations in growth rate have little influence (Fig. S2A). This is consistent with previous analyses (1; 2).

Departure from the steady state occurs 20 min after the shift, when the average added volume exceeds the average birth volume. This ratio is maximized at 1.1 h after the upshift at 1.64 : 1, equilibrating back to 1 : 1 after about 4 h (Fig. S2C). The shape of this response is largely expected because we have an approximately closed system in the sense that nascent cell lineages are not added to the device after the initial incubation and trapping, hence individual birth volumes cannot increase without an increase of added volume in the previous cell cycle. This result is not trivial though: it is conceivable that a perturbation could briefly arrest cells that were smaller than average at birth, prompting larger cells to overtake the dividing population and thereby increasing birth volume without a preceding increase in added volume. Despite the necessary break from the steady state $\langle\Delta\rangle / \langle V_0\rangle = 1$, the overall strength of size control is nearly constant during the upshift as evidenced by λ, which exhibits only a moderate monotonic decrease from 0.638 ± 0.046 in slow growth media to 0.516 ± 0.033 in fast media (Supplementary Fig. S2CD). Thus, although there is a change in the overall added volume, cells demonstrate near-adder behaviour that is uninterrupted by the environmental shift.

This persistence is effected by complementary dynamics between the timing- and growth-related components of size control. The nutritional upshift induces a brief but dramatic decrease in the normalized growth-related contribution $\langle \alpha_{\text{inst}} \rangle \langle \tau \rangle \gamma$, which pulses downwards from a negligible -0.019 ± 0.101 in the first steady state to a minimum of -0.466 ± 0.298 after 1.6 h in the second media (Supplementary Fig. S2CD). Growth-related control then returns to a comparatively small equilibrium value of -0.176 ± 0.090. Control from timing, $\langle \alpha_{\text{inst}} \rangle \langle \tau \rangle \theta$, responds in an opposing manner, increasing considerably from 0.551 ± 0.335 preshift to a 1.107 ± 0.679 maximum 1.5 h into the upshift (Supplementary Fig. S2CD). It too equilibrates after about 3 h, settling to 0.665 ± 0.310. Therefore, although growth and timing processes are thrust out of their respective equilibria by the media shift, the balanced interplay of their contributions to cell size ultimately conserves near-adder behaviour in this dynamic environment (Supplementary Fig. S2D). It should be noted that $\lambda = 0.5$ strictly coincides with a perfect adder provided it is at steady state $\langle \Delta \rangle / \langle V_0 \rangle = 1$. Generally, the birth size-independent addition of an arbitrary volume during the cell cycle that defines an adder can present a range of λ values: e.g. if the added volume is between 0.5 and 2 times the average birth volume, λ would be in the range [0.33,0.67]. Nevertheless, *E. coli* is well within the adder regime at all time points.

SUPPLEMENTARY INFORMATION ON THE *BUGPIPE* SEGMENTATION/-TRACKING ALGORITHM

Bugpipe is the custom-coded MATLAB-based package that was used to segment, track, and analyse the resultant cell data presented in this work. The following contains a summary of the algorithms developed. The package and its full documentation is available at https://github.com/panlilio/bugpipe.

RAW IMAGES USED

Raw data consisted of 16-bit images at 512 x 512 pixel resolution. Three images were taken per field of view, per time point: one fluorescence; one brightfield; and one dark frame for background subtraction, i.e. an image acquired under the same exposure time but without illumination (Fig. S17). Images were obtained every 5-6 min for each field of view, with the number of fields chosen for continuous acquisition. This translated to a total of 35-40 fields of view, each spanning up to 8 microchannels. The chip was aligned such that the microchannels run approximately parallel to the image x-axis.

Fluorescence images were used for segmentation here because of the associated high signal-to-noise ratio. In principle, high quality phase contrast images can be successfully substituted with minor changes to the segmentation procedure. Brightfield images were used largely for manual inspection when required.

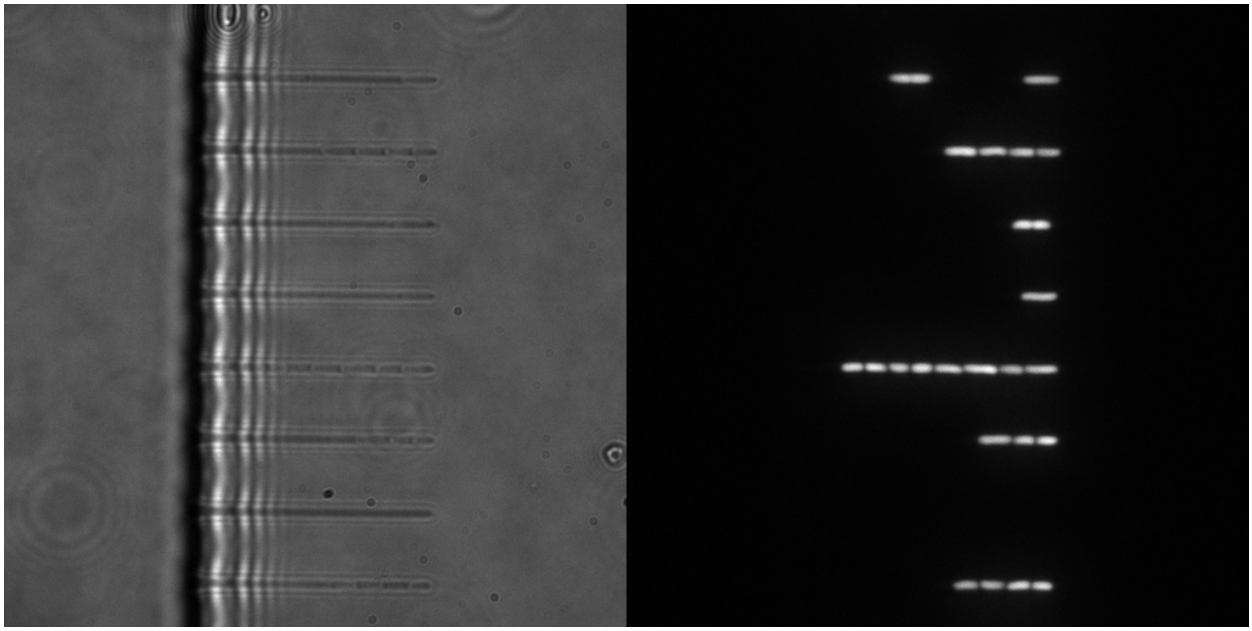

FIG. S17 Sample brightfield (left) and fluorescence images (right).

BUGPIPE: IMAGE SEGMENTATION

Initially, all images are segmented independent of the order in which they were acquired by the function `segmentationMP`. For each background-subtracted fluorescence image the algorithm proceeds in the following steps: 1) find regions of interest, i.e. the micropistons trapping each cell line; 2) within each region of interest, threshold and perform morphological operations to isolate cells in the background; 3) find cell boundaries and separate any artifically fused cells; 4) remove artefacts based on size.

The order-independent first pass at segmentation enables parallel processing of all frames and thereby imparts a drastic decrease in the overall characterization time. Corrections to the initial segmentation are performed under the tracking algorithm. All programming was done in MATLAB with built-in functions indicated by (†).

Background subtraction

Simple background subtraction is performed by subtracting the respective dark frame from each fluorescent image. Any resulting negative intensity values are set to 0.

Microchannel detection

The regions of interest (ROI) in each image are defined by the microchannels trapping each line of cells. The function `getChannelsMP` uses x- and y-cross sections of pixel intensity to detect these regions by thresholding: namely, the Otsu threshold is used along x (parallel to direction of growth, Fig. S18A, green) and a sliding average background subtraction is implemented along y (perpendicular to growth, Fig. S18A, blue). The final regions of interest are shown in Fig. S18B. The typical microchannel region delineated by this procedure across all fields of view and replicates was around 35 x 200 pix.

Cell segmentation

Each region of interest is processed independently. The following procedure is implemented by `segmentationMP` as outlined in Fig. S19A. First, the cropped image of the k-th microchannel I^k is passed through a 2D median filter to remove hot pixels (`medfilt2`†) and then resized using bilinear interpolation (`imresize`†, 3x magnification) to further smooth intensity values before thresholding.

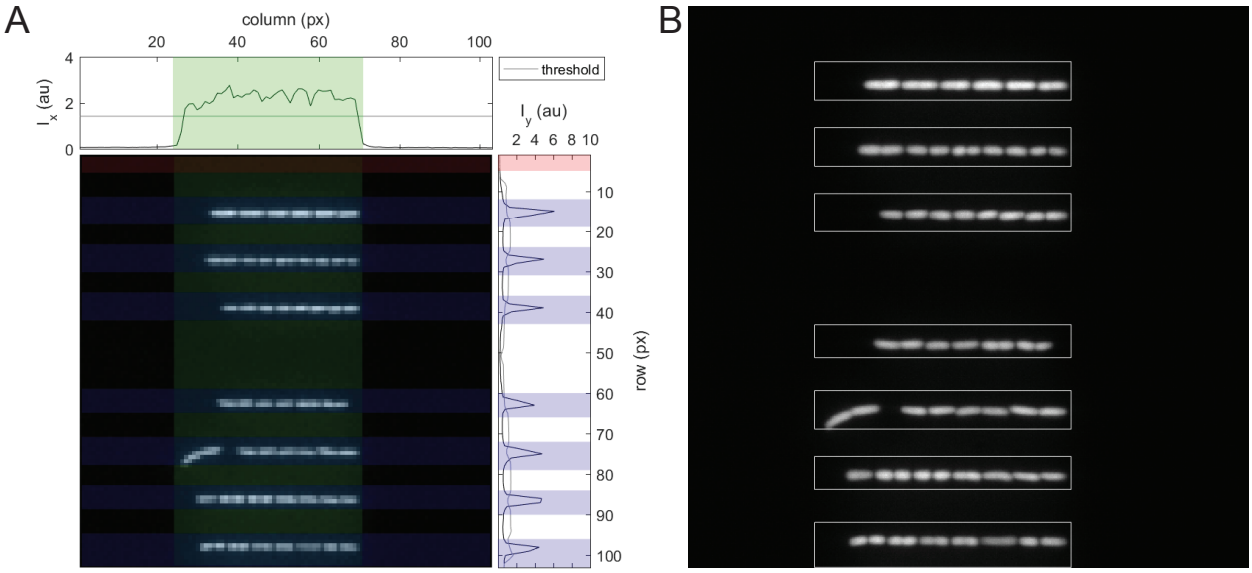

FIG. S18 Microchannel identification by thresholding image cross sections of intensity **A** Subsampled image where the x-profile ι_x defines the height of all channels in a given field of view (green) and ι_y determines the width of each microchannel (blue). Regions that pass the filter but do not contain any high intensity objects are omitted from further analysis (red). **B** Found microchannel bounding boxes overlayed on image at its original resolution.

Modified Otsu threshold

The cropped greyscale image I^k is thresholded using Otsu's method with one modification (Fig. S19B). The standard Otsu threshold (as calculated by `graythresh`†) tended to drastically undersegment the fluorescence images here: i.e. several cells or sometimes entire microchannels were grouped into a single object, often lacking a sufficiently detailed perimeter from which cells might be distinguished. To overcome this, a correction term was added equal to the standard deviation of the Otsu-determined lower intensity population. That is, if T_{Otsu} is the Otsu threshold and the set of low intensity values is $L^k = \{I^k_{ij} | I^k_{ij} < T_{\text{Otsu}}\}$, the applied threshold is

$$T_{I^k} = T_{\text{Otsu}} + \sqrt{\langle (L^k)^2 \rangle - \langle L^k \rangle^2}. \tag{S1}$$

All I^k_{ij} less than T_{I^k} are set to 0 and all pixels greater or equal to the threshold are set to 1. The result is a binarized image of black 0's and white 1's (Fig. S19A, 3rd panel down).

The thresholded image is then morphologically dilated using a 3 pixel radius disc (`imdilate`†) to compensate for the slight oversegmentation during thresholding and further smooth object edges. The boundaries of the resulting objects are then found with `bwboundaries`†.

Curvature assessment

Each object was then assessed by the function `getCellsMP` according to its ordered (x, y) boundary points, with each coordinate pair representing an edge pixel. For a single bacterium this set typically consisted of about 120 points on the magnified image. Although a large fraction of detected objects were indeed individual bacterium, the above procedures did tend to artificially join multiple cells by undersegmentation. In this case, the boundary of the joint object consistently had at least two regions of high magniture, negative curvature near the point of contact between adjacent cells (Fig. S19C).

For a single rod-shaped cell, curvature is theoretically non-negative for the entire boundary. Candidate objects for further segmentation are therefore identified as those possessing two or more regions in which the above smoothed curvature is less than -0.03 pix^{-1}. This gives an allowance for some noise in the image and in practice serves as an empirical threshold for separating emerging daughter cells during cell division.

Objects without multiple negative curvature regions were rescaled to match the original 512 x 512 pix image and assessed for area using `regionprops`[†]: to remove the rare imaging artifact, only those objects consisting of more than 30 pix ($\sim 0.034\,\mu$m^2) were accepted as individual bacterium. Note that a typical bacterium has a corresponding projected area of roughly 156 pix ($\sim 1.79\ \mu$m^2). The aforementioned candidates for further segmentation were passed to the next processing step.

Boundary processing

The local minima of negative curvature regions are each designated as a "pinch point" and, owing to the configuration of the cells, generally one pair of pinch points can be connected to separate two adjacent cells. That is, because cells tend to stack along their major axes, two pinch points appeared at the point of contact between two cells: one above and one below the axes (Fig. S19C, second and third panels down). In the case of more than two pinch points, it was neccesary to calculate each point's nearest neighbour for correct pairing. In many cases, only two bacteria are enclosed in the object and they are easily separated using a pair of pinch points.

For example, consider the case of two merged bacteria. Line segments were demarcated by pinch points. Each segment was then closed by connecting its first and last points. This connecting line was interpolated at n_c points using the first and last n_c points of each segment, assuming a third degree polynomial, where n_c was the approximate number of pixels along the straight line between the endpoints. The closed segments were then rescaled to match the original 512 x 512 pix image and passed by the same size filter to ensure that the area enclosed was greater than 30 pix. The final cells segmented for the sample channel are shown in the bottom panel of Fig. S19A.

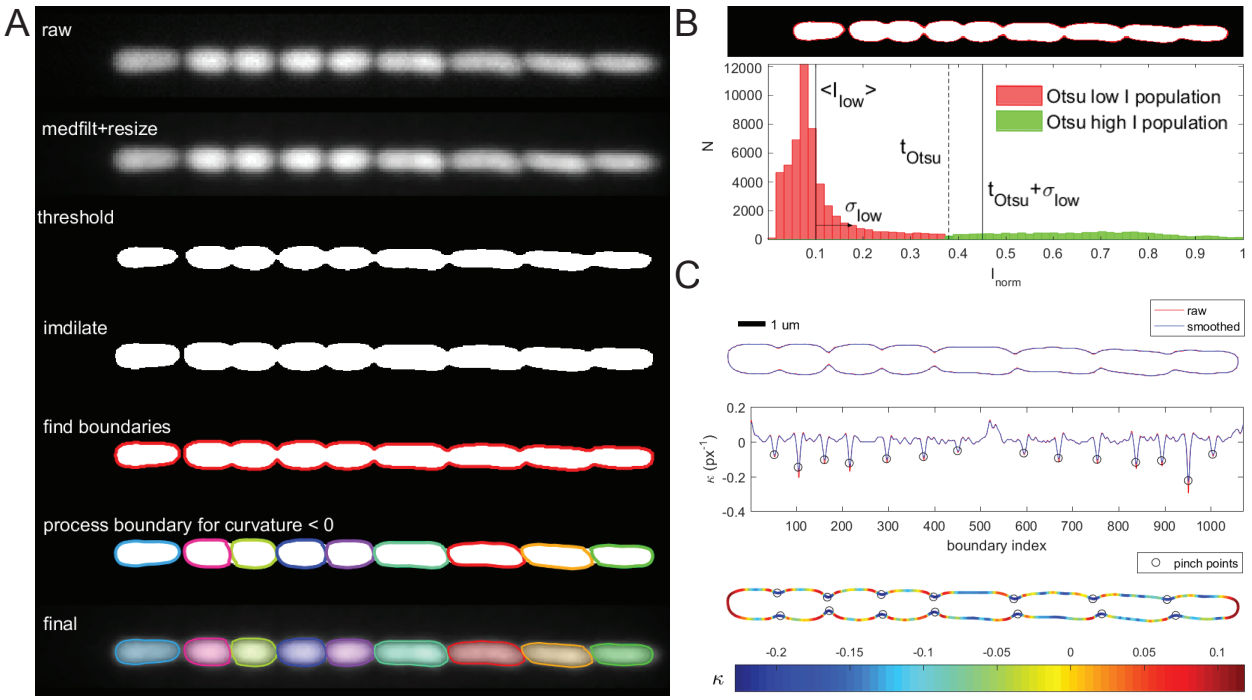

FIG. S19 Morphological operations and object boundary processing for cell segmentation within a microchannel. **A** Top to bottom: overview of the full segmentation process on a single microchannel. **B** Demonstration of the modified Otsu threshold: histogram of mean-normalized pixel intensities for the channel of interest, with the red population indicating the background (i.e. low intensity) pixels and green indicating object (high intensity) pixels according to the standard Otsu threshold (dotted vertical line). Difference between the Otsu threshold and modified threshold is shown in upper panel (red). **C** Segmentation of the large, merged object in A using boundary curvature (second panel). Boundary coloured according to local curvature (third panel) with regions of strongly negative curvature (dark blue) indicating pinch points at which to further segment cells (black circles).

This approach based on boundary curvature turns the problem of segmentation from 2D to 1D and minimizes the use of the repeated morphological operations often required for e.g. watershed-based separation. Rather 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—real, during the process of cell fission, or artifactual—between two cells. The resulting segmentation yields results comparable to that produced by edge detection filters but avoids the more time-consuming morphological operations.

BUGPIPE: CELL TRACKING

An order-dependent approach is blatantly necessary for frame-to-frame cell tracking. The time series for separate fields of view are independent however and subsequently they are processed by separate MATLAB workers (`spmd`† in the Parallel Computing Toolbox). The

tracking procedure takes advantage of the geometric constraints imposed by the mother machine's format: in particular, that in the absence of cell divisions a cell's rank within a channel is conserved between frames. That is, suppose one channel has five cells in one frame, five in the next, and there was no evidence of cell division. The first cell away from the dead-end of the channel in one frame would be the first cell away from the dead-end in the next frame, the second cell in one frame would be the second cell in the next frame, and so forth. The task was thus reduced to: 1) microchannel tracking between frames, 2) sorting cells in-channel based on distance from the dead-end, and 3) determining markers for cell division to adjust the rank-based pairing as necessary. These operations are all contained in the function `cellTrackerMP`, which takes consecutive images and the results of their respective segmentations as input.

Microchannel matching

The location of each microchannel's bounding box is passed through the segmentation variables associated with each frame. Pairwise-distance calculations are perfomed for channel tracking because of the small channel numbers: there are always fewer than 9 microchannels under the reported magnification and camera conditions. This approach assumes sufficiently low optical drift, specifically that any relative movement between consecutive frames is less than the half-distance between microchannels: a criterion that was easily met by the previously described microscopy system.

Example matched channels are shown in Fig. S20.

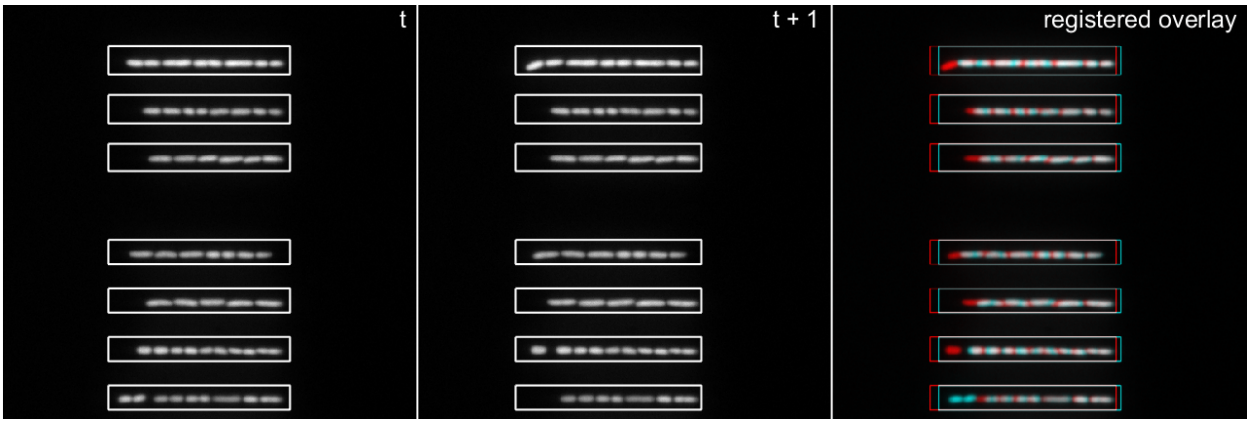

FIG. S20 Microchannel detection and pairing between frames. Bounding boxes identified for the microchannels in two consecutive grayscale images (left, centre). Sample image registration of the two frames under translational and rotational transformations using `imregister`†(right).

Rank-based cell identification and flagging of division events

After locating microchannels in consecutive frames, all cells in the frame of interest f are ranked within their respective channels: 1 being closest to the dead-end. The x-component of each cell's centroid is sufficient for ranking due to the orientation of the device. For the first frame in the time series, all cells are given naïve labels with no cells designated as daughters of a division. Cell rank, labelling, and microchannel index were then stored and passed on to the next tracking iteration using the next frame.

For all $f > 1$, a rank-based comparison of cell area between frames f and $f - 1$ was performed to detect division events. For two channels paired across consecutive frames, denote r_0 and r_1 as rank-based indices for the cells/ranks of interest in frames $f - 1$ and f respectively. Let $A(r_0)$ and $A(r_1)$ be the areas of the cells of rank r_0 and r_1 in their associated frames. The procedure begins by checking the first cells in the channel: $r_0 = r_1 = 1$. The proportional change in area ϕ_A is then calculated:

$$\phi_A(r_1, r_0) = \frac{A(r_1)}{A(r_0)}. \tag{S2}$$

The conditions of the experiments presented here do not anticipate any decreases in cell area unattributable to cell division: under stable nutritional environments, both growth media used are capable of sustaining long-term, balanced growth of all assayed strains. A lower empirical threshold $\phi_A < 0.7$ was established to mark significant area decreases as cell divisions. Note that since healthy divisions are roughly symmetric in *E. coli*, the next cell along the microchannel should also be relatively small and thus $\phi_A(r_1+1, r_0)$ should also fall below this threshold. The labelling procedure is demonstrated in Fig. S21); the algorithm is explicitly described in bugpipe documentation.

CHARACTERIZATION OF SEGMENTED CELLS

Fluorescence images and the cell boundaries retrieved from segmentation are used to measure cell size, shape, promoter expression, and the time derivatives of these quantities. As a rod-shaped bacteria, the geometry of an *E. coli* cell can be approximated to a cylinder with hemispherical caps. As shown, the cells grow in the mother machine with their long axis parallel to the plane of observation so that their length and width can be ascertained in addition to their cross sectional area and fluorescence intensity.

The function `getCellPropsMP`, called by `cellTrackerMP`, takes boundary coordinates and a fluorescence intensity matrix as input to determine single cell properties using methods summarized in Table S1 (see also Fig. S22)

Alternative ways to estimate cell volume and the overall geometry of the cell were explored(3); the spherocylindrical model provided the greatest consistency with the vol-

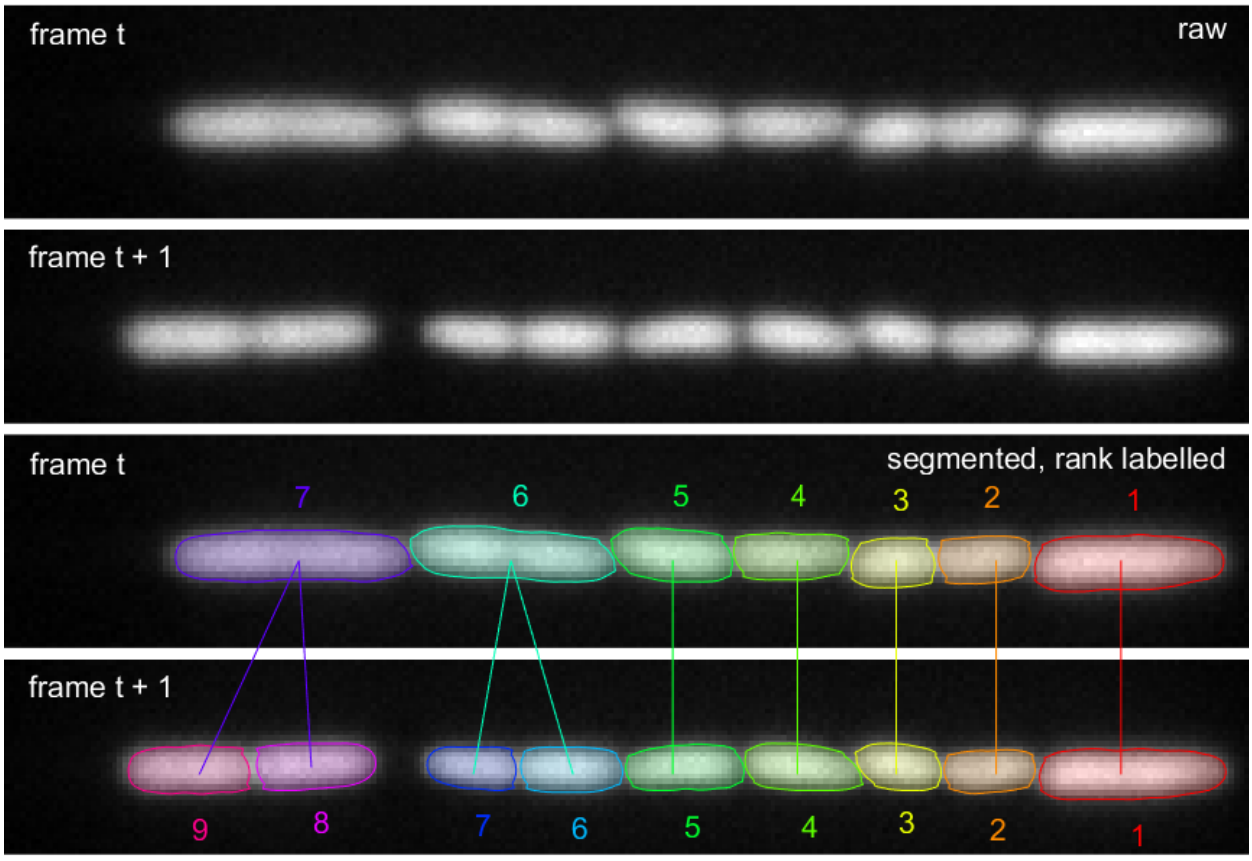

FIG. S21 Rank-based comparison of cell area to track cells and division events. A given microchannel is matched between consecutive frames (top two panels). Segmented cells are ranked according to centre of mass distance from the dead-end and changes in area are used to detect division events (bottom two panels). Here, the sixth ranked cell shows a dramatic decrease in area between frames. Finding the area of the next cell in line to be sufficiently small, the procedure determines that a division has indeed occurred and that the sixth and seventh ranked cells in frame $t + 1$ are the daughters of the sixth ranked cell in frame t.

ume estimated by the rotation of the boundary contour. During the shift, tapered cells may form due to cell width dynamics (4; 5). In our experiment, these tapered cells are not easily visible by eye, due to the limited resolution (Fig. S23A). However, we could quantify their contribution by looking at the differences between measured area from segmented perimeter and area computed by width and length measurements assuming a spherocylinder (Fig. S23BC). Quantitatively, this phenomenon is small in our data (about 2%), and does not affect any of our conclusions.

BUGPIPE: SCRIPT IMPLEMENTATION AND DATA MANAGEMENT

Application of the segmentation and tracking procedures is quite straightforward, as demonstrated in the subsection below. There was, however, an additional computational

TABLE S1 Summary of key single cell measurements

quantity	notation	description
area	A	area enclosed by boundary points, including edge pixels
length	L	greatest pairwise distance between boundary points along the first principal axis (found by `pca`†)
width	w	width assuming the projected area is a rectangle with semi-circular caps of radius $\frac{w}{2}$, i.e. $w = \frac{2[L-\sqrt{L^2-(4-\pi)A}]}{4-\pi}$
surface area	S	$S = \pi L w$
volume	V	$V = \frac{4}{3}\pi\left(\frac{w}{2}\right)^3 + \pi(L-w)\left(\frac{w}{2}\right)^2$
volumetric growth rate	α_{inst}	instantaneous rate assuming exponential growth: $\alpha_{\text{inst}} = \frac{1}{V}\frac{dV}{dt}$
surface synthesis rate	β	instantaneous surface area growth rate assuming it is proportional to volume, i.e. $\beta = \frac{1}{V}\frac{dS}{dt}$
cell cycle growth rate	α_{cc}	overall volumetric growth rate assuming exponential growth, determined from the slope of the linear fitting of $\log(V)$ vs t for all points of the cell cycle
interdivision time	τ	time between birth and division, estimated as the time between a cell's first and last frames plus the time between consecutive frames
promoter/gene expression	F	total fluorescence intensity of the segmented pixels corresponding to A
protein concentration	C	concentration of fluorescent protein within the cell: $C = F/V$

challenge following image processing and analysis: the high throughput and high spatiotemporal resolution nature of acquisition necessitated the development of a fast and flexible means of data manipulation. Custom MATLAB classes were written to handle and analyse these large, non-uniform datasets.

Sample script and directory organization

The following MATLAB script can be taken as a review of the image processing steps described in previous sections and should act as a guide for future users of this package. Note that the script can be easily modified to decrease processing times further by taking advantage of MATLAB's Parallel Computing Toolbox. Indeed, parallel loops (`spmd`† and `parfor`†) were used to process all images pertaining to the data presented in this work.

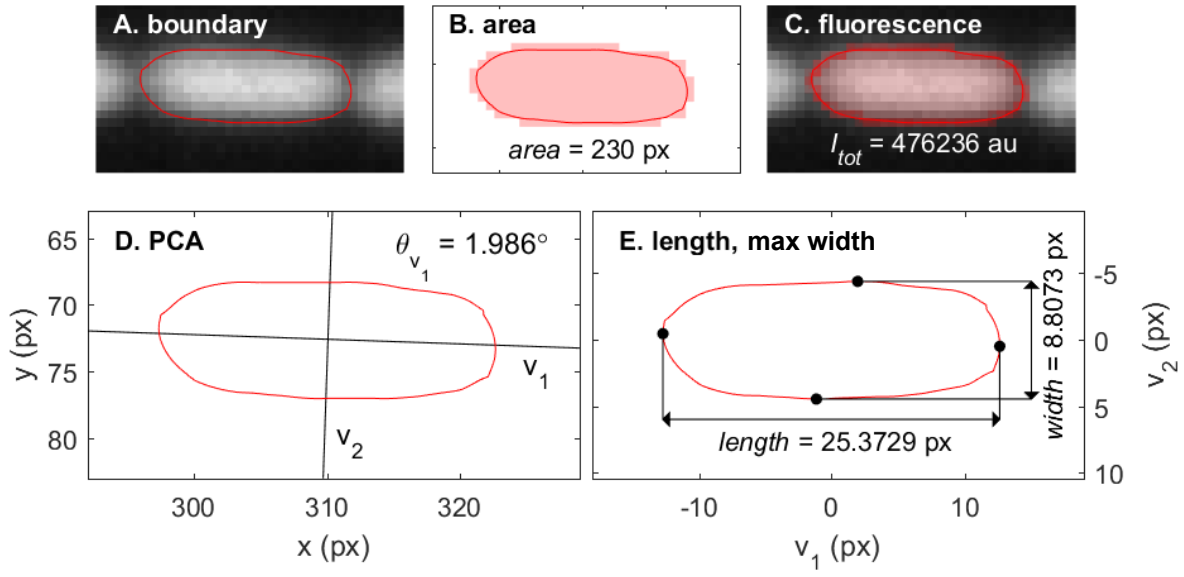

FIG. S22 Sample cell measurements retrieved from segmentation of a single cell. **A** Boundary coordinates found overlayed on the original grayscale image. **B** Total cross sectional area calculated as the area enclosed by the boundary, including edge pixels. **C** Total fluorescence intensity associated with the enclosed area. **D** Principal component analysis on boundary to determine axis of orientation. **E** Cell presented in principal axis space, where length is the maximum distance between boundary points along the first axis. Along the second axis, maximum width also shown.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      PREAMBLE: PATHS, PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Retrieve a list of all available datasets
D = allDataBookDirsMP;

%Select a specific replicate, get a list of all directory paths, and
    delete any existing folders
D = D.dirList{26};
D = getDirsMP_SSD(D);
clearDirsMP(D,{'mat2Dir','lineDir','textDir'});

%Get the orientation of the device, store in tracking parameters
dataInfo = lookupDataMP(D.baseDir);
P = segmentationParamsMP(dataInfo.ORIENTATION);

%Retrieve list of PNG images, including any dark frames
L = imListMP(D.dataDir);
```

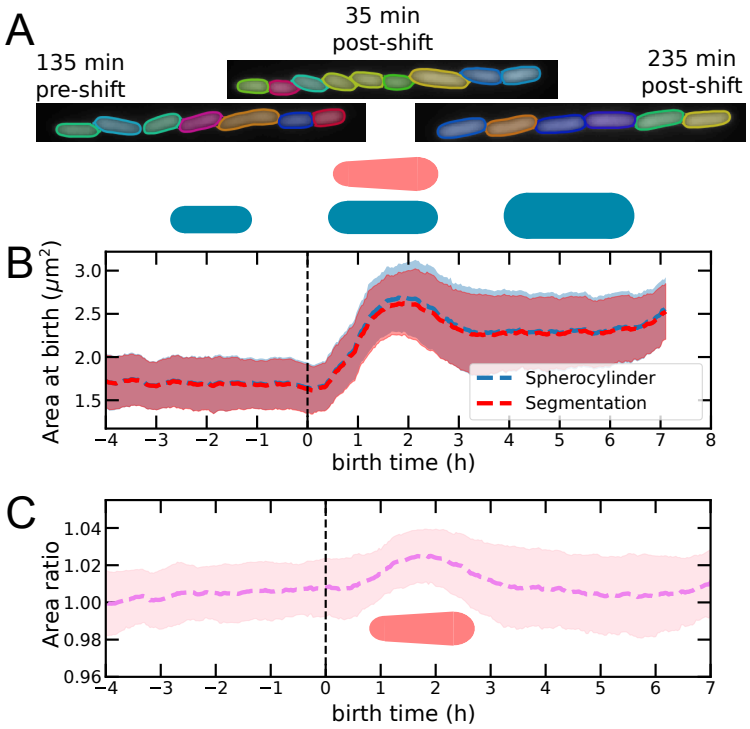

FIG. S23 Tapered cells determine minor deviations in estimated volume. **A** Tapered cells are not easily visible by eye at the resolution of our experiments. **B** Comparison of area at birth computed from width and length assuming a spherocylinder, and area at birth determined by perimeter segmentation shows a different during the shift, likely due to the existence of tapered cells (5). **C** The ratio of assumed to measured area deviates of at most 2% during the shift. The same analysis shows smaller deviations (1-1.5%) if all cell areas are considered and not only area at birth.

```
19
20 %Directory paths for different stages of processing
21 dataDir = D.dataDir;
22 matDir = D.matDir;
23 mat2Dir = D.mat2Dir;
24 lineDir = D.lineDir;
25
26 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27 %     SEGMENTATION (FOV AND TIME-INDEPENDENT)
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29
30 %Iterate through the entire list of images, order and fov-independent
31 for k = 1:numel(L.imList)
32
33     %Read in image and blank/dark frame
```

```
    I = imread(fullfile(dataDir,L.imList{k}));
    if ~isempty(L.blankList{k})
        Iblank = imread(fullfile(dataDir,L.blankList{k}));
    else
        Iblank = zeros(512);
    end

    %Background subtraction
    I = double(I) - double(Iblank);
    I(I<0) = 0;

    %Segment
    [CELLS,CHANNELS] = segmentationMP(I,P);

    %Record time image was acquired by camera
    CELLS.tAcq = L.timeCreated(k);

    %Save variables
    parsave1(fullfile(matDir,sprintf('fov%.2d_t%.4d',...
        L.imListFOVT(k,1),L.imListFOVT(k,2))),CELLS,CHANNELS)

    kDone(k) = true;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     TRACKING (FOV-INDEPENDENT)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Retrieve range of fields of view, times/frames
fovRange = L.fovRange;
tRange = L.tRange;

%Orientation of device
deadEnd = P.deadEnd;

%Iterate through each field of view
for f = 1:numel(L.fovRange)
    fov = fovRange(f);
```

```
72
73      %Initialize structure for segmentation results and zero matrix for
        intensity matrix
74      C0 = {};
75      I0 = zeros(512);
76
77      %Iterate through frames
78      for t = tRange
79
80          %Load the current segmented data
81          mat1 = fullfile(matDir,sprintf('fov%.2d_t%.4d.mat',fov,t));
82          if ~exist(mat1,'file'),
83              fprintf('\tCONTINUE: No MAT found. fov = %d, t = %d \r', fov,t
        )
84              continue
85          end
86          C1 = load(mat1);
87
88          %Skip any dropped frames
89          if isempty(C1.CELLS.boundary) || isempty(C1.CHANNELS)
90              fprintf('\tCONTINUE: No cells segmented. fov = %d, t = %d \r',
        fov,t)
91              continue
92          end
93
94          %Read in the current image, including blank/dark frame
95          [imname,blankname] = L.getfilenamesFOVT(fov,t,1); %#ok<PFBNS>
96          I1 = imread(imname);
97          if isempty(blankname)
98              Iblank = zeros(size(I1));
99          else
100             Iblank = imread(blankname);
101         end
102
103         %Background subtraction
104         I1 = double(I1) - double(Iblank);
105         I1(I1<0) = 0;
106
```

```
        %Track cells
        [C2,~] = cellTrackerMP(I1,I0,C1,C0,t,fov,lineDir,deadEnd);

        %Save tracking result variables
        CELLS = C2.CELLS;
        CHANNELS = C2.CHANNELS;
        parsave1(fullfile(mat2Dir,sprintf('fov%.2d_t%.4d',fov,t)),CELLS,
    CHANNELS)

        %Update for the next iteration but skip empty frames from tracking
     consideration
        if ~isempty(CELLS.boundary)
            C0 = C2;
            I0 = I1;
        end
    end
end
```

Directory structure

Variables are saved at each stage of the procedure, with the final processing step being a conversion from MAT format to text file by the function `mat2txtMP`. The user can delete intermediary variables and directories where needed. A list of the output variables in their associated directories as laid out in the example script are presented in Table S2, with intermediary outputs indicated by (*). More detailed information on each variable is present in the relevant function's description.

Data handling classes

Under the acquisition settings descibed in the main text, about 2,000 cells were imaged every 5 minutes in a typical sample. For the usual 18 h experiment, this translates to over 400,000 cell snapshots and 20,000 fully tracked cell cycles per replicate. Two handling classes were developed to manipulate this data effectively: `dataMP2`, which handles specific replicates; and `dataLabMP`, which manages several replicates at once. These objects allow the user to perform calculations on and retrieve statistics from different populations: whether it be of single cell cycles, specific lineages, whole replicates, or multiple datasets. The complete documentation, including example analyses, can be found on https://github.com/panlilio/bugpipe.

TABLE S2 Organizational structure of segmentation results

directory	description	file variables
\Data_png	image files, at least two files per field of view per time point, one illuminated and one dark frame	—
\Data_mat*	initial segmentation as output from `segmentMP`	`CELLS`, `CHANNELS`
\Data_mat2*	labelled output of previous directory, i.e. matched channels and tracked cells	`CELLS`, `CHANNELS`
\Data_line*	lineage information and tracked single cell data files containing e.g. frames present, pixel intensity range, parent cell identity, area, etc.	`cellData`; `cellLine`, `cellLineFamilyMap`, `cellLineFrames`
\Data_txt	text file containing flattened single cell data, i.e. the initial and final/pre-division values of, e.g. acquisition time, length, width, and fluorescence, from each cell cycle	—
\Data_txt\lineage	text files with complete cell cycle data, with one file per tracked lineage	—
\Data_flat	MATs containing matrices of flattened summary data and cell cycle/lineage data, shaped identically to \Data_txt files	`A` (cell cycle), `B` (summary)

Applied data filters

Following the highly automated cell segmentation and tracking procedures, there were a few common data filters applied for subsequent analyses. Specifically, we required that only cells meeting the following criteria were considered:

1. The entire life cycle was observed, i.e. both parent and daughter(s) were at least partially tracked so that a division event was flagged both directly before and after the cell cycle.

2. The division event terminating the cell cycle of interest must split the mother into near-symmetrical daughters, specifically each daughter occupying 40-60% of the mother's final cross sectional area. This step aims to eliminate filamentous cells and segmentation artifacts.

3. The observed exponential growth rate corresponds to a doubling time that is greater than 10 minutes but finite. Note that the physiological lower limit is 15-20 min.

Note that the last filter is largely unnecessary for the vast majority of datasets presented in this work. For example, in replicate 20160325, less than 0.3% (104 out of 37,570) of segmented cells lie outside of this reasonable growth rate interval.

SUPPLEMENTARY REFERENCES

[1] C. Cadart, S. Monnier, J. Grilli, P. J. Sáez, N. Srivastava, R. Attia, E. Terriac, B. Baum, M. Cosentino Lagomarsino, and M. Piel. Size control in mammalian cells involves modulation of both growth rate and cell cycle duration. *Nature communications*, 9:3275, Aug. 2018.

[2] J. Grilli, C. Cadart, G. Micali, M. Osella, and M. Cosentino Lagomarsino. The Empirical Fluctuation Pattern of E. coli Division Control. *Frontiers in Microbiology*, 9(July):1–10, 2018.

[3] M. Panlilio. *Non-equilibrium growth laws: growth and gene expression dynamics in Escherichia coli under a nutritional upshift*. PhD thesis, University of Cambridge, 2021. Available at https://www.repository.cam.ac.uk/.

[4] C. L. Woldringh, N. B. Grover, R. F. Rosenberger, and A. Zaritsky. Dimensional rearrangement of rod-shaped bacteria following nutritional shift-up. ii. experiments with escherichia coli b/r. *Journal of theoretical biology*, 86:441–454, Oct. 1980.

[5] A. Zaritsky and C. L. Woldringh. Chromosome replication, cell growth, division and shape: a personal perspective. *Frontiers in microbiology*, 6:756, 2015.