

A label-free method to track individuals and lineages of budding cells

Julian M. J. Pietsch¹, Alán F. Muñoz^{1,*}, Diane-Yayra A. Adjavon^{1,*}, Iseabail Farquhar¹, Ivan B. N. Clark¹, and Peter S. Swain¹

¹SynthSys–Synthetic and Systems Biology and School of Biological Sciences, University of Edinburgh, Edinburgh EH9 3BF, United Kingdom

*These authors contributed equally to this work.

Abstract

Much of biochemical regulation ultimately controls growth rate, particularly in microbes. Although time-lapse microscopy visualises cells, determining their growth rates is challenging because cells often overlap in images, particularly for those that divide asymmetrically, like *Saccharomyces cerevisiae*. Here we present the Birth Annotator for Budding Yeast (BABY), an algorithm to determine single-cell growth rates from label-free images. Using a convolutional neural network, BABY resolves overlaps through separating cells by size and assigns buds to mothers by identifying bud necks. BABY uses machine learning to track cells and determine lineages, estimates growth rates as the rate of change of volumes, and identifies cytokinesis by how growth varies. Using BABY and a microfluidic device, we show that bud growth is first size- then timer-controlled, that the nuclear concentration of Sfp1, a regulator of ribosome biogenesis, varies before the growth rate does, and that growth rate can be used for real-time control. Growth rate and fitness are strongly correlated, and BABY should therefore generate much biological insight.

Introduction

For microbes, growth rate correlates strongly with fitness [1]. Cells increase growth rates through balancing their synthesis of ribosomes with their intake of nutrients [2–4] and target a particular size through coordinating growth with division [5–8]. Multicellular organisms, too, not only coordinate growth over time but also in space to both size and position cells correctly [9].

To understand how growth rate is regulated, studying single cells is often most informative [10]. Time-lapse microscopy, particularly with microfluidic technology to control the extracellular environment [11, 12], has been pivotal, allowing, for example, studies of the cell-cycle machinery [7], of the control of cell size [13–15], of antibiotic effects [16, 17], of the response to stress [18–20], of feedback between growth and metabolism [21], and of ageing [22].

For cells that bud, like *Saccharomyces cerevisiae*, estimating an instantaneous growth rate for individual cells is challenging. *S. cerevisiae* grows by forming a bud that increases in size while the volume of the rest of the cell remains relatively unchanged. Although growth rate is typically reported as the rate of change of volume [13, 15, 23–26], which approximates a cell’s increase in mass, these estimates rely on solving multiple computational challenges: accurately determining the outlines of cells – particularly buds – in an image, extrapolating these outlines to volumes, tracking cells over time, assigning buds to the appropriate mother cells, and identifying births. Growth rates for budding yeast are therefore often only reported for isolated cells using low-throughput and semi-automated methods [13, 26]. In contrast, for rod-shaped cells that divide symmetrically, like *Escherichia coli*, the growth rate can be found more simply, as the rate of change of a cell’s length [21].

A particular difficulty is identifying cell boundaries because neighbouring cells in images often overlap – yeast, like other microbes, grows in colonies. Although software that automatically identifies and tracks cells in bright-field and phase-contrast images is well established [27–31] – with deep learning now improving both accuracy and speed [32–34], only a few algorithms allow for overlaps [30, 33]. For example, the convolutional neural network U-net [35], a workhorse in biomedical image processing, performs semantic segmentation, lumping adjacent objects in the same category under one label, and instances of individual cells must be found using post-processing. Even then different instances typically cannot overlap [32, 34]. Other deep-learning approaches, like Mask-RCNN [36] and extended U-nets like StarDist [37], can identify overlapping instances in principle, but typically do not, by either implementation [37] or the labelling of the training data [33]. Furthermore, assigning lineages and births is often performed manually [13, 23] or through fluorescent markers [15, 25], but such markers use an imaging channel.

Here we describe the Birth Annotator for Budding Yeast (BABY), a complete pipeline to robustly and accurately determine single-cell growth rates from label-free images of budding yeast. In developing BABY, we solved multiple

49 image-processing challenges generated by cells dividing asymmetrically. BABY resolves overlapping instances – buds,
50 particularly small ones, usually overlap with their mothers and neighbouring cells – by extending the U-net architec-
51 ture with custom training targets and post processing. It tracks cells between time points with a machine-learning
52 algorithm, which is able to resolve any large movements of cells from one image to the next, and assigns lineages with
53 another informed by the U-net. These innovations substantially improve performance. BABY produces high-fidelity
54 time series of the volumes of both mother cells and buds and so the instantaneous growth rates of single cells at fine
55 temporal resolution.

56 Using BABY, we find that concurrent behaviour in the growth rates of the mother cell and its bud enables us to
57 identify cytokinesis from bright-field images and so accurately estimate birth times. We see a peak in growth rate
58 during the S/G2/M phase of the cell cycle and show that this peak marks a transition in the bud’s growth from
59 being sizer- to timer-controlled. Studying Sfp1, a regulator of ribosome synthesis, we observe correlated fluctuations
60 between this regulator’s nuclear concentration and growth rate. Finally, we demonstrate that BABY is sufficiently
61 fast to enable real-time control, running an experiment where changes in the extracellular medium are triggered when
62 the growth of the cells being imaged crosses a pre-determined threshold.

63 Results

64 Segmenting overlapping cells using a multi-target convolutional neural network

65 To estimate single-cell growth rates from time-lapse microscopy images, accurately identifying and segmenting cells is
66 essential. Poorly defined outlines, missed time points, and mistakenly joined cells all substantially affect how reported
67 volumes vary with time.

68 Correctly segmenting asymmetrically dividing cells, such as budding yeast, is particularly challenging because the
69 imbalance in the size of the mother and bud gives each distinct appearances and behaviours. Even when constrained in
70 a microfluidic device, small buds imaged in a single Z section may appear to overlap with their mother and neighbouring
71 cells (Fig. 1a & 1b). Even if cells can be semantically separated, then the area of either the bud or the neighbouring
72 cells is often underestimated and the bud may even be undetected (Figure 1—figure supplement 1). Buds also move
73 more in the Z-axis compared to mother cells, changing how they appear in bright-field images (Fig. 1c). Depending
74 on the focal plane, a bud may be difficult to detect even by eye.

75 Our BABY algorithm, however, has high reliability (Fig. 1e). We leverage the relative simplicity of the U-net [35],
76 but extend this semantic approach to identify overlapping cells by combining custom training targets with a post-
77 processing algorithm to segment instances.

78 To identify potentially overlapping cells from the semantic output of a convolutional neural network (CNN) like the
79 U-net (Fig. 2a), we noted that the frequency and extent of overlap varies with cell size (Appendix 3 Fig. 2). Therefore
80 by dividing training masks – filled cell outlines – into categories of different sizes, we could define training images
81 for each size category that reduced whether cells overlap. We further decreased overlaps by applying and optimising
82 a morphological erosion to the training masks (Fig. 2b; Appendix 3 Figs. 2 & 1). With these data, we trained a
83 multi-target CNN whose output then predominantly comprised separated interiors of cells (Fig. 2c).

84 Using a four-layer U-net, we achieved high accuracies for predicting these targets early in training and with
85 relatively few training images (588 trap images – 1813 annotated cells in total; Appendix 3 Fig. 3). Using standard
86 image processing, we then readily identified cell instances and applied a morphological dilation appropriate for each
87 size category to recover the training masks (Appendix 3 Fig. 4).

88 To alleviate any loss in resolution from the morphological erosion used to generate the training masks, we incorpo-
89 rated the training masks’ edge pixels as an additional set of training targets for each size category (Fig. 2d). Like the
90 StarDist [37] and DISCO algorithms [30], we parameterise masks using a radial representation: it naturally describes
91 the elliptical shape of a yeast cell but also specifies any star-convex shape, including non-convex pinched ones. Specif-
92 ically, we fit radial splines with 4-8 rays depending on the cell’s size to a re-weighted version of the edge pixels and use
93 outlines estimated from the CNN targets of the cells’ interiors as initial guesses (Appendix 3 Fig. 5). The resulting
94 masks improve the intersection-over-union score on test images (Figure 3—figure supplement 1). Importantly, our
95 method successfully detects and segments buds that overlap with adjoining mother and daughter cells (Fig. 1b & 1c).

96 Other features further improve performance. We developed a powerful graphical user interface (GUI) to label and
97 annotate overlapping cells with radial splines (Appendix 2 Fig. 1). The GUI allows concurrent viewing of the same
98 cell across both Z-sections and time and so enables errors in segmentation and tracking errors to be rapidly identified
99 and fixed. We therefore focused on challenging cases, iteratively cycling between training and curating in what was
100 effectively active learning. For training, we developed scripts that optimise the hyper-parameters, and so no specialist
101 knowledge is required to change to other imaging modalities and cell types. Finally, although BABY can predict
102 overlapping outlines from a single 2D image, performance is improved by using multiple Z-sections (Figure 3—figure
103 supplement 1).

104 The algorithm is highly accurate (Fig. 3a). For larger cell sizes, BABY performs comparably with both our

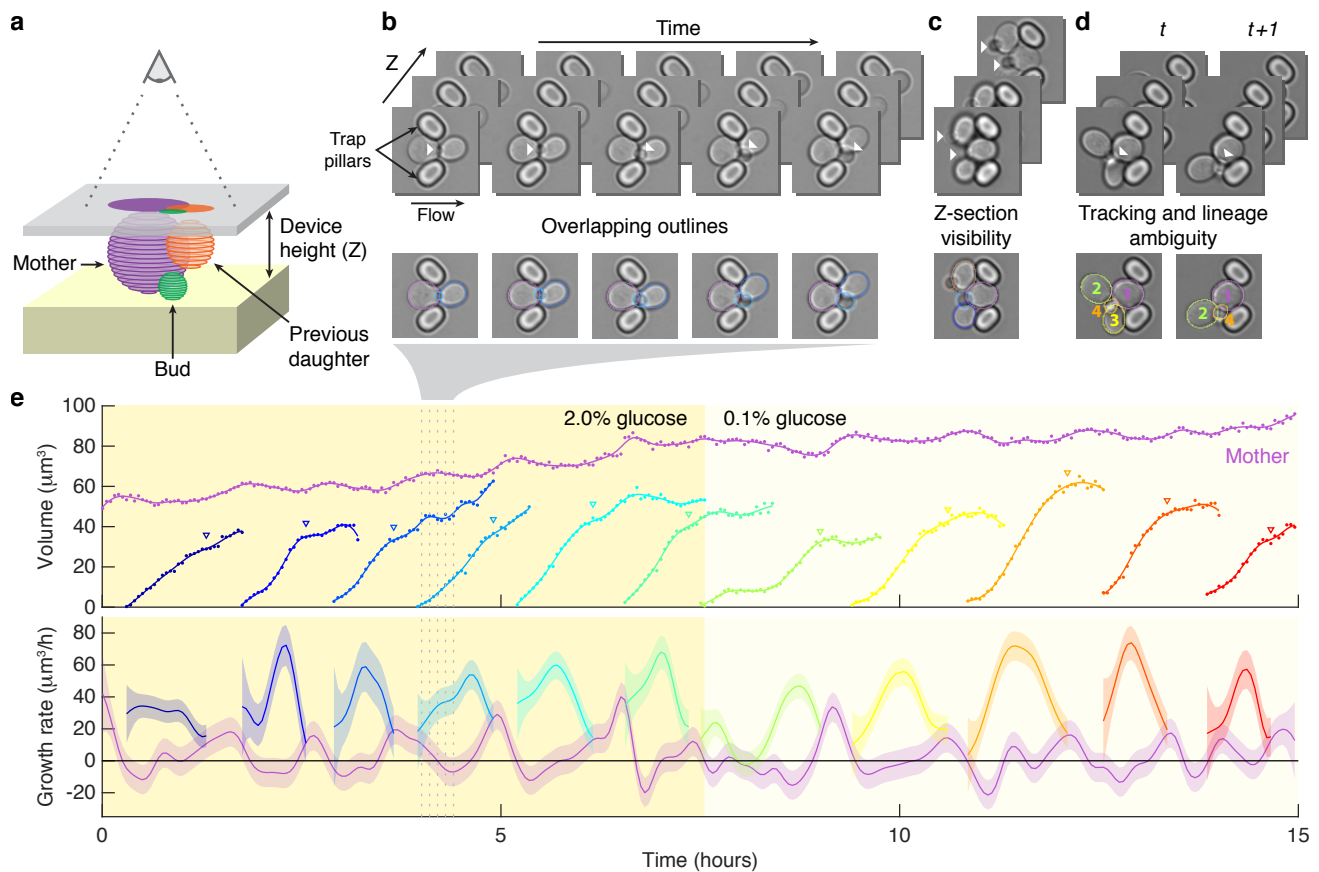


Figure 1: Reliably identifying individual cells makes automating label-free segmentation of budding cells challenging. **a** A schematic of a budding cell constrained in a microfluidic device showing how a mother cell can produce a bud beneath the previous daughter. The microscope, denoted by the eye, sees a projection of these cells. **b** A time series of bright-field images of budding yeast trapped in an ALCATRAS microfluidic device [38], in which a growing bud (white arrowheads) overlaps with both its sister and mother. On the duplicated images below, we show outlines produced by BABY. **c** Bright-field images of growing buds (white arrowheads) taken at different focal planes demonstrate how the appearance of small buds may change. **d** Cells can move substantially from image to image. Here medium flowing through the microfluidic device causes a cell to wash out between time points and the remaining cells to pivot. We show the correct lineage assignment with white arrowheads and the correct tracking by the numbers within the BABY outlines. **e** A time series of a mother (purple) and its buds/daughters for a switch from 2% to 0.1% glucose is shown using volumes and growth rates estimated by BABY. Bud growth rates are truncated to the predicted time of cytokinesis (triangles). Shaded areas are twice the standard deviation of the fitted Gaussian process.

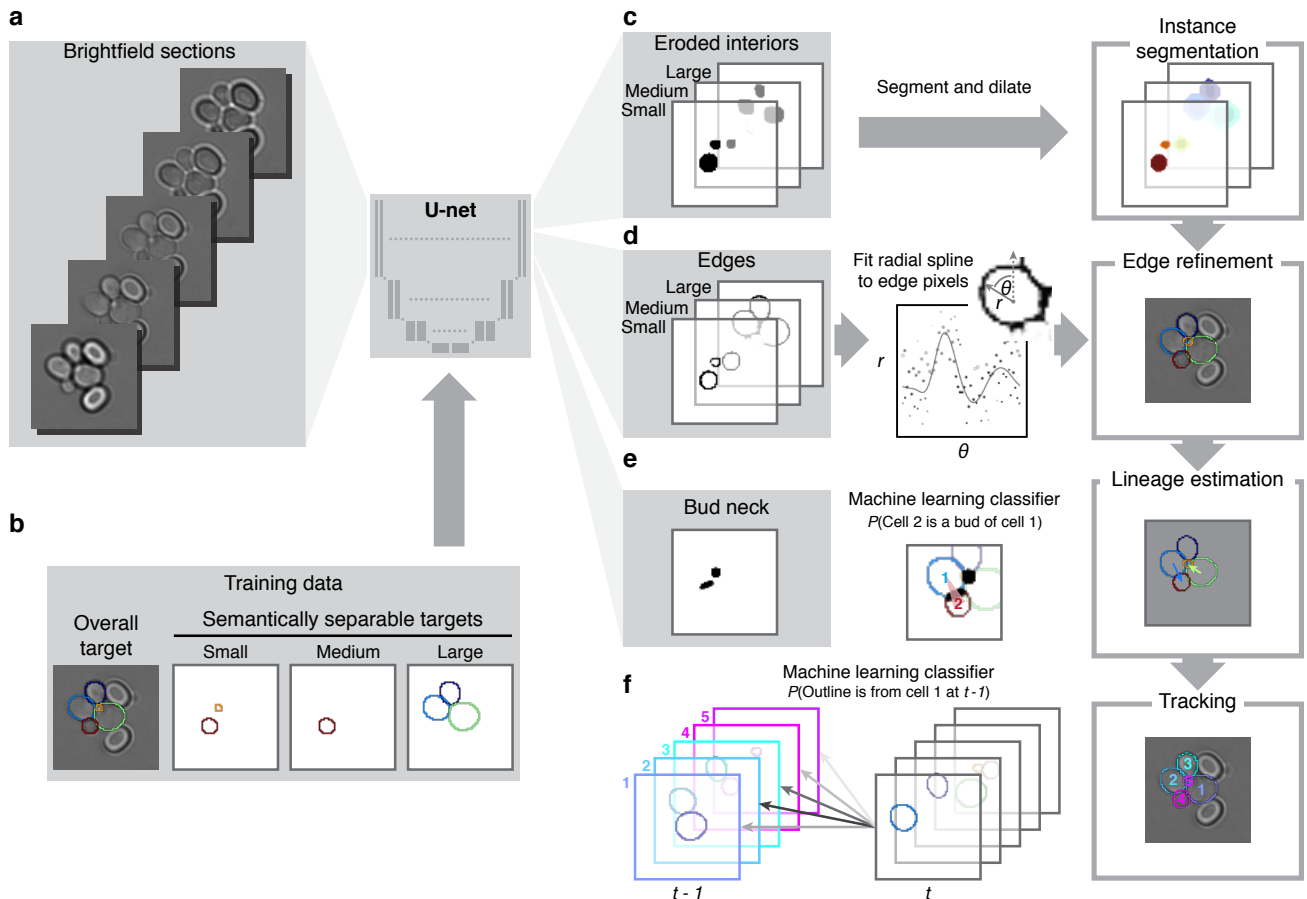


Figure 2: **BABY** uses multiple bright-field Z-sections, a multi-target convolutional neural network followed by instance segmentation, and two machine learning classifiers to identify cells and their buds reliably from image to image. **a** Multiple bright-field Z-sections are input into a multi-target U-net CNN. **b** The curated training data comprises multiple outlines that are categorised by size to reduce overlap within each category. **c** The CNN is trained to predict a morphological erosion of the target masks, which act as seeds for segmenting instances. **d** Edge targets from the CNN are used to refine each cell's outline, parameterised as a radial spline. **e** A bud-neck target from the CNN and metrics characterising the cells' morphologies are used to estimate the probability that a pair of cells is a mother and bud with a machine-learning classifier. **f** Another classifier uses the same morphological metrics to estimate the probability that an outline in the previous time point matches the current one.

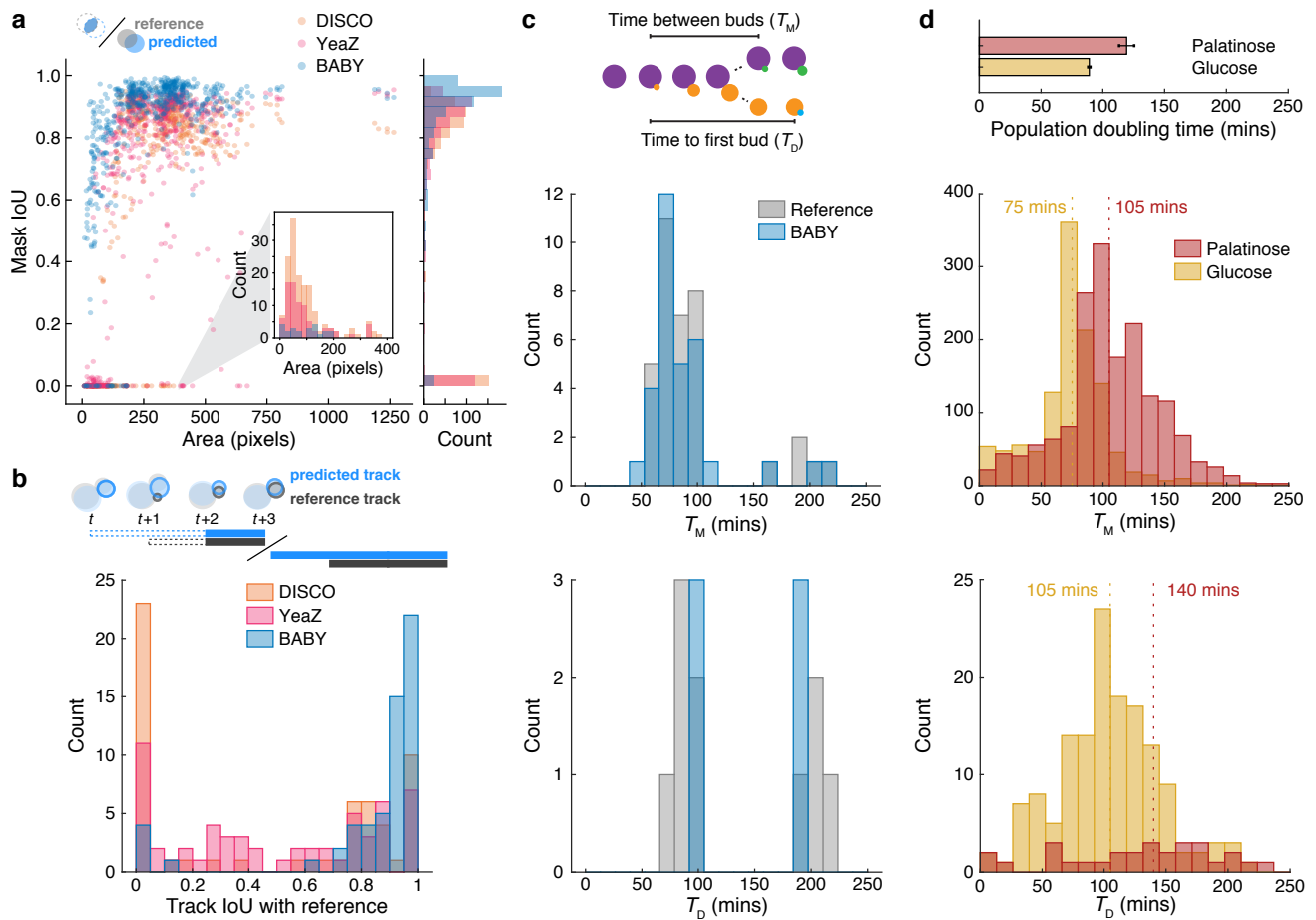


Figure 3: **BABY outperforms existing algorithms for segmenting and tracking as well as accurately estimating doubling times.** **a** Comparing the intersection-over-union (IoU) score between manually curated masks of single-cell outlines and masks predicted by the BABY, YeaZ and DISCO algorithms shows that BABY performs best, particularly for buds and small cells. **b** BABY tracks best when comparing the distribution of track IoUs between manually curated single-cell tracks and predicted tracks. **c** BABY accurately predicts both the times between budding events, T_M , and the times from birth to a cell's first budding event, T_D . **d** The predicted distributions of T_M and T_D shift to longer times, as expected, when steady-state growth in 2% glucose is compared to 2% palatinose, a poorer carbon source. We estimate too the population doubling times [39] using the median times (upper bar plot with mean and 95% confidence interval from 10 bootstraps of the doubling time distributions).

105 previous algorithm DISCO [30], which segments instances, and a recent algorithm YeaZ [34], which uses a U-net. For
106 smaller cell sizes, there is substantial improvement. By using size categories, BABY overcomes the difficulties from
107 the imbalance in size of mothers and buds and identifies buds overlapping with mother cells that both DISCO and
108 YeaZ therefore miss.

109 Using machine learning to track lineages robustly

110 To determine growth rates accurately, estimates of both the volumes of the mother and the bud are necessary because
111 the majority of growth occurs in the bud [13,39]. Not only must we track cells from one time point to the next, but
112 also correctly identify, track, and assign buds to their mothers (Appendix 4 Fig. 4).

113 Budding raises multiple issues. With their small initial size, buds frequently first appear surrounded by cells and
114 displace their neighbours as they grow (Fig. 1b), potentially making assigning a bud to its mother ambiguous (Fig. 1d).
115 Additional challenges arise when the connected mother and bud react to the flow of medium. Buds may pivot around
116 the mother, and such movements are especially prevalent, and can even involve multiple cells (Fig. 1d), in microfluidic
117 devices that allow long-term imaging by washing away daughter cells. If tracked incorrectly, a pivoting bud is likely
118 to be treated as newly detected – both overestimating budding rates and underestimating lag times.

119 To help assign buds to mothers, we included an additional ‘bud-neck’ training target for the CNN (Fig. 2e; Appendix
120 4) because cytokinesis is sometimes visible as a darkening of the bud neck in bright-field images. We then estimate
121 the probability that a pair of cell outlines is a mother and bud using a machine-learning classifier trained on both
122 morphological features derived from the segmented outlines and the CNN’s predictions of bud-necks (Appendix 4 Fig.
123 3). Although the CNN’s performance in this task is lower than it is for identifying cells (Appendix 3 Fig. 3), the
124 mother-bud classifier achieves a precision of 80% on test data.

125 We construct the most probable lineage tree by first identifying cells that are likely to be buds and then assigning
126 their mothers. For each bud, we pick its mother cell as the cell that has the highest accumulated probability of being
127 in a pair with the bud, where the probability is accumulated over all previous images containing the bud and the
128 potential mother cell (Appendix 4).

129 We use another classifier to track cells between time points. This classifier is trained on the morphological properties
130 of the cell outlines with only weak constraints on distance – the cells should only be near a microfluidic trap – allowing
131 pivots (Fig. 1d) to be correctly tracked. Other tracking algorithms, in contrast, typically rely on distance, either
132 explicitly by using the Hungarian algorithm to assign pairs of cells between images [25,28,34] or implicitly by seeding
133 the detection of tracked cells either forwards [30] or backwards [31] in time. Using the classifier we find the probability
134 of a match for all pairs of outlines between two time points (Fig. 2f; Appendix 4) and identify unmatched outlines
135 too – those that have no pairings above a threshold probability. Such outlines are treated either as new buds or as
136 disappearing or appearing cells because cells may be washed into or out of the field of view by the medium. To be
137 resilient to transient errors in segmentation and in acquisition, like a loss of focus, we aggregate tracking predictions
138 over the last three time points and defer to a classifier with stronger constraints on distance if assignment is ambiguous
139 (Appendix 4).

140 Comparing with the DISCO [30] and YeaZ [34] algorithms, BABY finds more than three times as many complete
141 or near-complete tracks (Fig. 3b). DISCO uses previously detected cells as priors for tracked cells in subsequent time
142 points, and YeaZ uses the Hungarian algorithm to assign pairs that minimise differences in space and morphology.
143 Each algorithm was assessed against manually curated data by calculating the intersection-over-union score (IoU)
144 between cell masks in a reference track with those in a predicted track. We report the track IoU – the time points
145 where the masks match relative to the total duration of both tracks. Where multiple predicted tracks could match
146 a reference track, we use the match with the highest track IoU. Any predicted tracks left unassigned receive a track
147 IoU of zero. BABY excels because it detects buds earlier, which both increases the track IoU and prevents buds from
148 being tracked to daughters rather than mothers.

149 We are unaware of other algorithms that assign lineages from bright-field images of budding yeast and so compared
150 estimates of doubling times derived by BABY with those from manual curation (Fig. 3c). The mother’s time between
151 budding events, T_M , is expected to be shorter than the time a newly born daughter takes to produce its first bud,
152 T_D [39], and so we consider both types of doubling separately (Fig. 3c). There are fewer results for T_D because
153 observing a daughter cell that buds before being lost in the flow is rare – the microfluidic device is designed to retain
154 mothers not daughters [38]. Although the estimates of T_D are consistent, BABY does not give estimates for all curated
155 daughters, usually because some buds forming on daughter cells appear for only a few time points before both daughter
156 and bud are washed away.

157 As an additional test, we ran BABY for images of cells growing in palatinose and in glucose (Fig. 3d). Estimates
158 of T_M are on average shorter than T_D , and, at least for glucose where we have more division events from the faster
159 growth, we observe that the distribution of T_D is broader than that for T_M .

160 A useful control is to compare the population’s doubling time with that in batch growth. We estimated the
161 doubling time from the medians of T_M and T_D with the formula of Hartwell and Unger [39]. The results for glucose

(Fig. 3d) agree with our measurements of batch growth (Figure 3—figure supplement 2). BABY, however, estimates a shorter doubling time in palatinose, a sugar similar to maltose. In batch growth, the concentration of the carbon source is not constant like it is in a microfluidic device, and cells both prepare for and switch from fermentation to respiration [40], slowing growth. This switch complicates comparing with constant conditions and is likely to happen for higher concentrations of palatinose than glucose because cells strongly prefer to ferment glucose.

Estimating growth rates

From the time series of segmented cells, we estimate instantaneous single-cell growth rates as time derivatives of volumes. We use a conical method to find each cell’s volume from its outline in an image [27] (Fig. 1e) and then a Gaussian process to both smooth the time series of volumes and to estimate its time derivative [41].

We independently estimate the growth rates of mothers and their buds. Both are informative. Like others [13,42], we observe periodic changes in growth rate across the cell cycle (Fig. 1e).

BABY predicts cytokinesis from bright-field images

To better characterise how growth rate varies during the cell cycle, we ran experiments for cells with the gene MYO1 tagged with Green Fluorescent Protein. As part of the actin-myosin contractile ring, Myo1 localises to the bud neck in late G1 and delocalises at cytokinesis [43] (Fig. 4a), marking the birth of a daughter. We could therefore identify the G1 and S/G2/M phases of the mother cells and the first G1 of independent daughters (Appendix 5 Fig. 1). By aligning the growth rates at cytokinesis, we observed two phases of growth (Fig. 4b). The bud dominates growth during S/G2/M, with its growth rate peaking approximately midway through (Fig. 4b) [13]. We further observe that the bud’s average growth rate positively correlates with the volume of the new daughter it becomes (Figure 4—figure supplement 1). The mother’s growth rate, in contrast, peaks during G1.

We determined that we could predict cytokinesis from growth rate because at cytokinesis the mother’s and bud’s growth rates often reach a similar magnitude (Appendix 5). The predictions coincide with the drop in Myo1’s fluorescence as expected (Fig. 4c) and hold in different media, with the Myo1-based and predicted points of cytokinesis having a Pearson correlation coefficient above 0.94 (Appendix 5 Fig. 2).

Nutrient modulation of birth size occurs after the peak in growth rate

This tight coordination between growth rate and cytokinesis suggested that the peak in growth rate preceding cytokinesis marks a regulatory transition. Comparing growth rates over S/G2/M for buds in a rich versus a poor carbon source, we found that the maximal growth rate occurs at similar times for both carbon sources despite substantial differences in the durations of the S/G2/M phases (Fig. 4d).

Daughters born in rich media are larger than those born in poor media, and some of this regulation is known to occur during S/G2/M [5,6,24]. Understanding the mechanism, however, is confounded by the longer S/G2/M phases in poorer media (Fig. 4d) [24], which counterintuitively gives daughters that should be smaller longer to grow.

Given that the time between maximal growth and cytokinesis appears approximately constant in different carbon sources (Fig. 4d), we hypothesised that the growth rate falls because the bud has reached a critical size and that size is regulated during the subsequent period of falling growth to reach the desired daughter size. The faster growth in richer carbon sources would generate larger daughters. Consistently, the buds’ growth rates peak when the buds reach a similar size in both carbon sources, whereas their sizes at birth are distinct (Fig. 4e). Size regulation in S/G2/M is therefore likely to be implemented as cells approach cytokinesis.

Changes in ribosome biogenesis precede changes in growth

An important advantage of the BABY algorithm is that single-cell growth rates can be estimated without fluorescence markers, freeing fluorescence channels for other reporters. Here we focus on Sfp1, a transcription factor that helps coordinate ribosome synthesis with the availability of nutrients [6].

Sfp1 promotes the synthesis of ribosomes by activating the ribosomal protein (RP) and ribosome biogenesis (RiBi) genes [6,44] and enters the nucleus in response to two conserved nutrient-sensing kinases – TORC1 and PKA [6,45,46] (Fig. 5a). In steady-state conditions, levels of ribosomes positively correlate with growth rate [47], and we therefore assessed whether Sfp1’s activity predicts changes in instantaneous single-cell growth rates.

Shifting cells from glucose to the poorer carbon source palatinose and back again, we observed that Sfp1 responds quickly to both the upshift and downshift and that growth rate responds as quickly to downshifts but more slowly to upshifts (Fig. 5b). As a target of TORC1 and PKA, Sfp1 acts as a fast read-out of the cell sensing a change in nutrients [20]. In contrast, synthesising more ribosomes is likely to be slower and explains the lag in growth rate after the upshift. The fast drop in growth rate in downshifts is more consistent, however, with ribosomes being deactivated, rather than their numbers being regulated. Measuring the half-times of these responses (Fig. 5b boxplots), there is a mean delay of 30 ± 2 minutes (95% confidence; $n = 245$) from Sfp1 localising in the nucleus to the rise in growth rate in the upshift. This delay is only 8 ± 1 minutes (95% confidence; $n = 336$) in the downshift, and downshift half-times are less variable than those in upshifts, consistent with fast post-translational regulation. Although changes in Sfp1

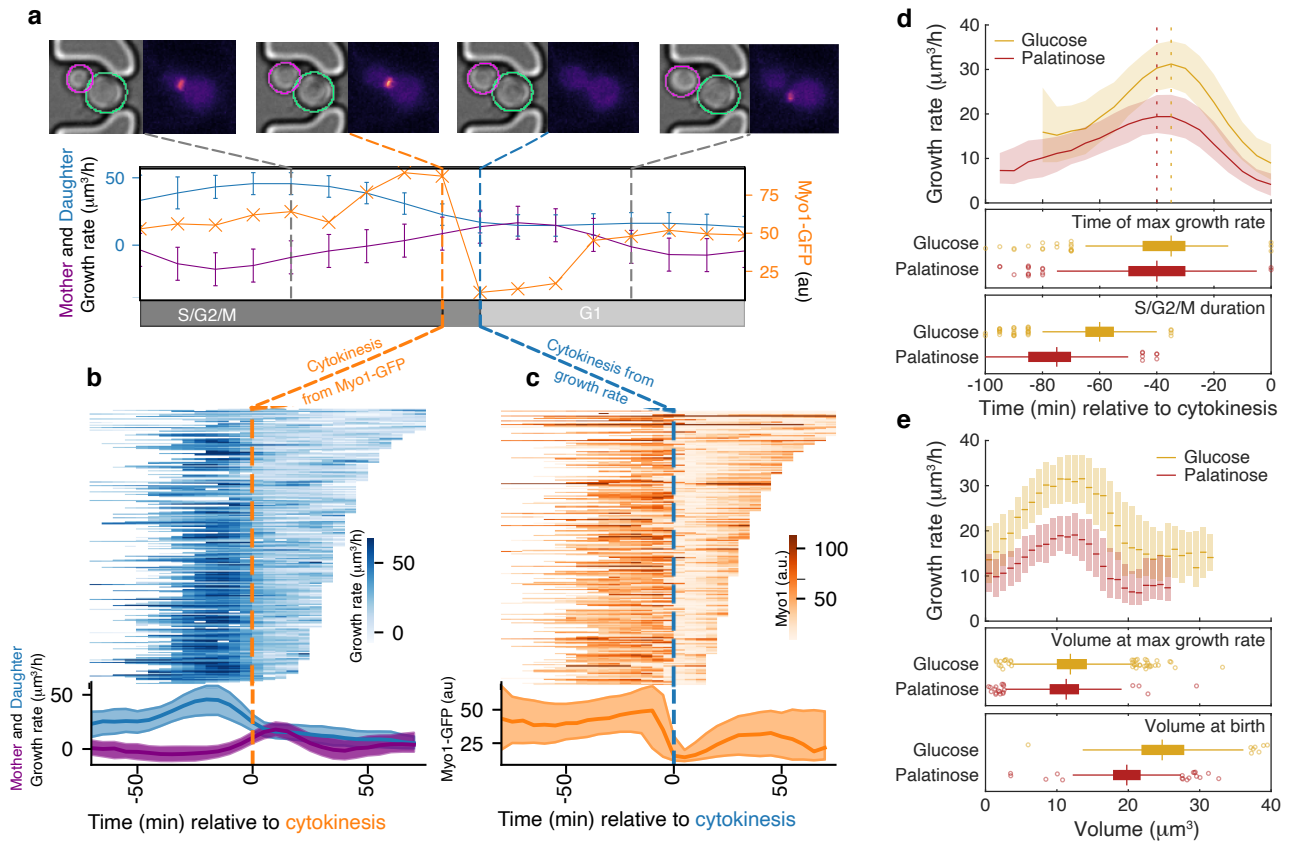


Figure 4: A peak in growth rate of the bud is predictive of cytokinesis and implicated in size control. **a** Using MYO1-GFP as a marker, we split individual cell cycles of mother cells into G1 and S/G2/M phases. We show bright-field and fluorescence images for a representative cell during one cell cycle. Myo1's fluorescence at the bud neck abruptly disappears at cytokinesis. The cell's growth rate (purple) and that of its bud (blue) are found from their volumes and shown with a measure of Myo1's localisation (orange). We estimate the point of cytokinesis from the drop in Myo1 fluorescence (orange dashed line) and also from the mother's and bud's growth rates (blue dashed line). **b** The time series with shading indicating growth rates for the buds/daughters of 268 cells growing initially in 2% and switched to 0.1% glucose are shown as horizontal lines. Cells are aligned by the time of the drop in Myo1's fluorescence. The corresponding median growth rates of mother and bud/daughter are plotted below, with the interquartile range shaded. The longer cell cycles likely occur during the glucose switch. **c** We show the fluorescence signal for the same cells, but align by the point of cytokinesis predicted from the mother's and daughter's growth rates. For almost all cells, Myo1's fluorescence drops near the predicted time of cytokinesis. **d** Although buds grow faster in richer media, the time of the maximal growth rate relative to cytokinesis is approximately constant, unlike the durations of the mothers' S/G2/M phases. Cells are grown in either 2% glucose or 2% palatinose. We show median growth rates with the interquartile range shaded. **e** Binning median growth rates according to volume, with the interquartile range shaded, shows that the buds' volumes when their growth rate is maximal are similar in both carbon sources, although those at birth are not.

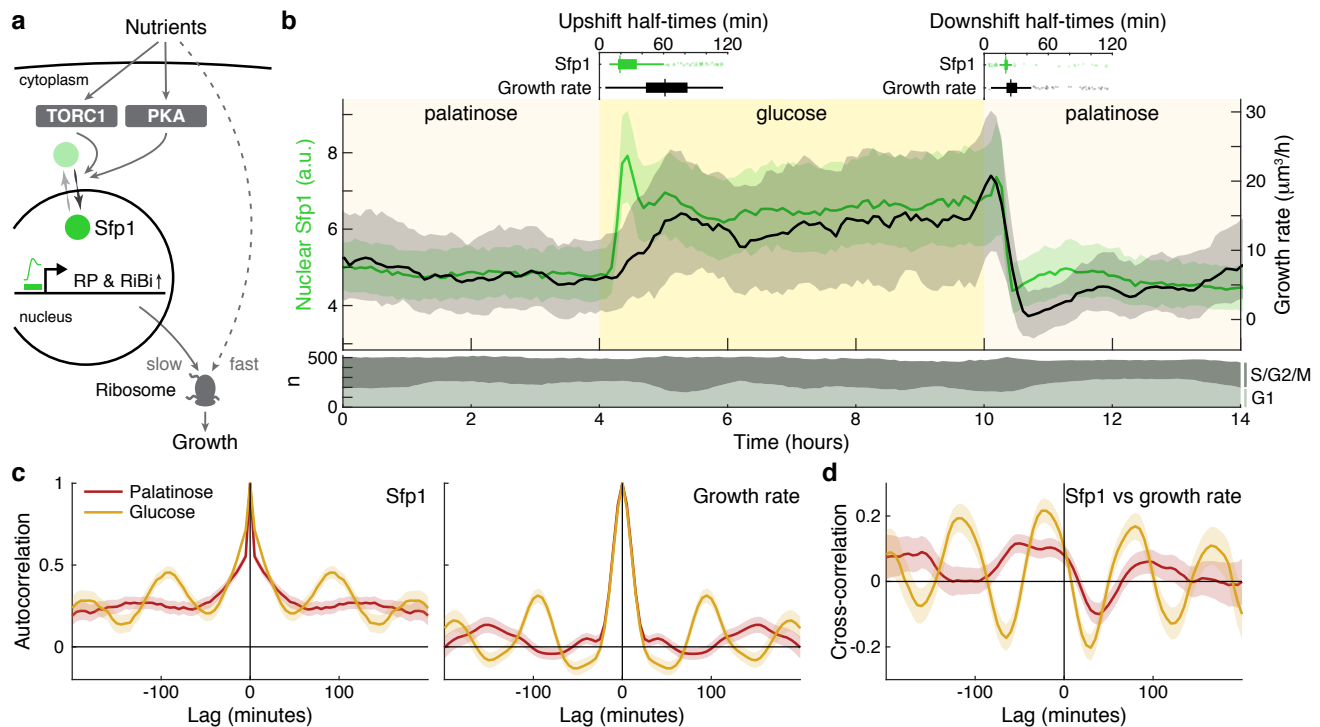


Figure 5: The dynamics of the ribosomal regulator Sfp1 anticipate changes in single-cell growth rates. **a** The transcription factor Sfp1 is activated by the kinases TORC1 and PKA when extracellular nutrients increase and promotes synthesis of ribosomes and so higher growth rates. **b** Growth rate follows changes in Sfp1's nuclear localisation if nutrients decrease but lags if nutrients increase. We show the median time series of nuclear Sfp1-GFP in mothers (green) and the summed bud and mother growth rates (black) for cells switched from 2% palatinose to 2% glucose and back. The interquartile range is shaded. Data was filtered to those cell cycles that could be unambiguously split into G1 and S/G2/M phases by a nuclear marker, and we display the number in each phase in the area plot. Above the switches, we show box plots for the distributions of half-times: the time of crossing midway between the minimal and maximal values. **c** The mean single-cell autocorrelation of nuclear Sfp1 and the summed mother and bud growth rates are periodic because both vary during the cell cycle. We calculate the autocorrelations for constant medium using data four hours before each switch (Appendix 6). The 95% confidence interval is shaded. **d** The mean cross-correlation between nuclear Sfp1 and the summed mother and bud growth rate shows that fluctuations in Sfp1 precede those in growth.

217 consistently precede those in growth rate, the higher variability in half-times for the growth rate is not explained by
218 Sfp1's half-time (Pearson correlation 0.03, $p = 0.6$).

219 By enabling both single-cell fluorescence and growth rates to be measured, BABY permits correlation analyses [21]
220 (Appendix 6). Both Sfp1's activity and the growth rate vary during the cell cycle. The autocorrelation functions for
221 nuclear Sfp1 and for the growth rate are periodic with periods consistent with cell-division times (Fig. 5c): around
222 90 minutes in glucose and 140 minutes in palatinose for Sfp1 and 95 minutes and 150 minutes for the growth rate. If
223 Sfp1 acts upstream of growth rate, then its fluctuations in nuclear localisation should precede fluctuations in growth
224 rate. Cross-correlating nuclear Sfp1 with growth rate shows that fluctuations in Sfp1 do lead those in growth rate, by
225 an average of 25 minutes in glucose and by 50 minutes in palatinose (Fig. 5d). Nevertheless, the weak strength of this
226 correlation suggests substantial control besides Sfp1.

227 During the downshift, we note that the growth rate transiently drops to zero (Fig. 5b), irrespective of a cell's stage
228 in the cell cycle (Figure 5—figure supplement 1), and there is a coincident rise in the fraction of cells in G1 (Fig. 5b
229 bottom), suggesting that cells arrest in that phase.

230 Using growth rate for real-time control

231 With BABY, growth rate can be used as a control variable in real time because BABY's speed and accuracy enables
232 cells to be segmented and their growth rates estimated during an experiment (Fig. 6a). As an example, we switched
233 the medium to a poorer carbon source and used BABY to determine how long cells should be kept in this medium if we
234 want approximately 50% the cells to have resumed dividing before switching back to the richer medium (Appendix 7).
235 After five hours in glucose, we switched the carbon source to ethanol (or galactose – Figure 6—figure supplement 1).
236 There is a lag in growth as cells adapt. Using BABY, we automatically determined the fraction of cells that have
237 escaped the lag at each time point – cells that have at least one daughter whose growth rate exceeds a threshold (Fig.
238 6b). This statistic is read by the software running the microscope, and the switch back to glucose is triggered when
239 50% of the cells have escaped (Fig. 6c). All cells resume dividing in glucose and initially grow synchronously because
240 of the rapid change of media. This synchrony is most obvious in those cells that did not divide in ethanol (Fig. 6c).

241 This proof-of-principle shows that BABY can be used in more complex feedback control, where a desired response
242 is achieved by comparing behaviour with a computational model to predict the necessary inputs, such as changes
243 in media [48–53]. Unlike previous approaches though, which typically measure fluorescence, BABY not only allows
244 single-cell fluorescence but also growth rates to be control variables, and growth rate correlates strongly with fitness [1].

245 Discussion

246 Here we presented BABY, a freely available algorithm to routinely extract growth rates from label-free images obtained
247 by time-lapse microscopy. BABY tackles previously unaddressed challenges in analysing images of budding cells and
248 provides fully automated tracking of lineages of budding yeast – from identifying cells to estimating births. We
249 introduce both a segmentation algorithm that allows instances to overlap and general machine-learning methods to
250 track and assign lineages. These approaches individually and together provide substantial gains in performance. As
251 a result, BABY reliably estimates time series of cellular volumes and consequently instantaneous single-cell growth
252 rates. There is no prohibitive penalty on speed, and BABY performs real-time processing.

253 BABY improves segmentation largely because it allows cells to overlap. Samples for microscopy are often prepared
254 to encourage cells to grow in a monolayer [11], but growth can be more complex because cells inevitably have different
255 sizes. We observe substantial and frequent overlaps between buds and neighbouring cells in ALCATRAS microfluidic
256 devices [38]. Inspecting images obtained by others, we believe overlap is a common, if undeclared, problem, occurring
257 during growth in CellASIC devices [31,34], against an agar substrate [15,28], in a microfluidic dissection platform [26],
258 and by surface attachment [54]. BABY alleviates having to constrain cells to completely exclude their neighbours,
259 at least when determining growth rates. Moreover we anticipate that BABY will segment mixtures of cells with
260 a range of sizes even if they do not bud, such as persister cells that transition at different rates from smaller to
261 larger morphologies [55]. For substantial growth beyond a monolayer, however, a different approach with full three-
262 dimensional imaging and segmentation is likely required [32].

263 Our algorithm uses a convolutional neural network that is smaller than the typical networks, with performance
264 relying instead on the choice of output targets. U-nets usually involve five contracting layers [32,34,35], whereas we
265 found four layers were optimal, presumably because our problem is of lower than usual complexity. Unlike others [30,
266 56], we need not explicitly ignore features in the image because the network quickly learns to disregard both the traps
267 in ALCATRAS devices and any debris.

268 The neural network represents cell edges at their native resolution, unlike alternative algorithms that also allow
269 instances to overlap: the Mask-RCNN scales regions of interest to a standard low-resolution image [36] and StarDist
270 represents an object's edge pixels as star-convex shapes with a constant and necessarily limited number of rays at
271 fixed angles [37]. Though BABY's final masks for segmentation are also star-convex, we optimise both the number
272 of rays and their angles to fit the high-resolution edge and use smoothing splines for the cell outlines. Instances are

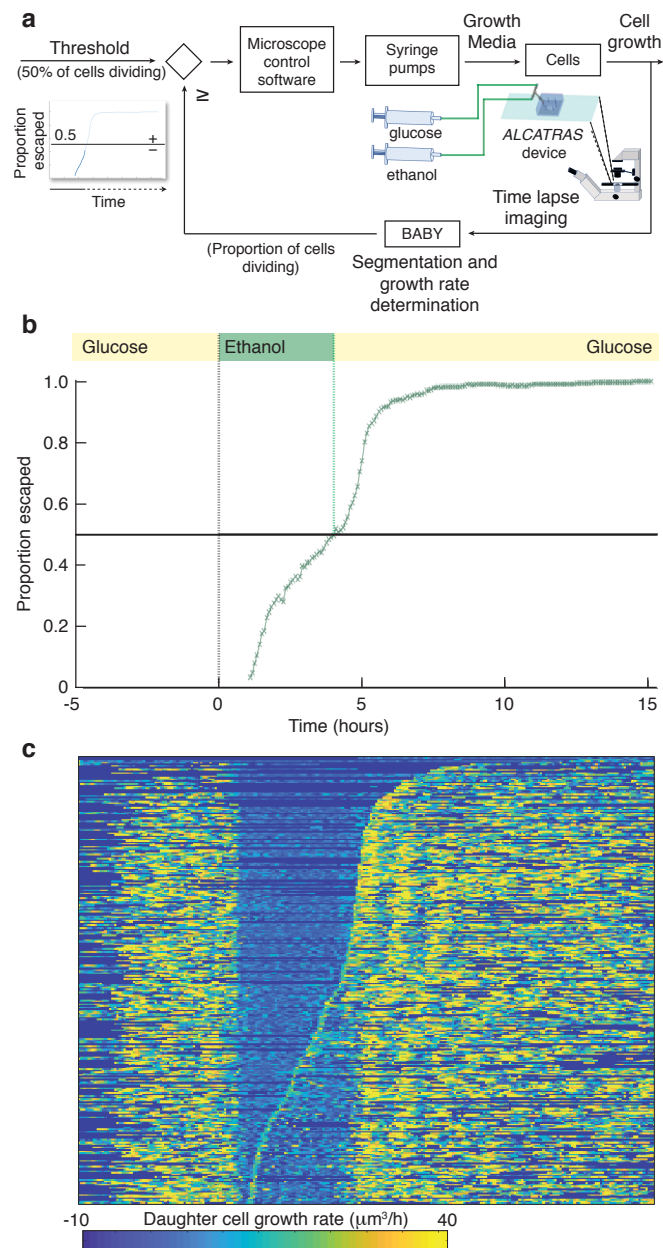


Figure 6: **BABY allows growth rate to be used as a variable for real-time control.** **a** Integrating BABY in real time with an experiment allows the cells' growth rate to control changes in media. Cells growing in the ALCATRAS microfluidic device are fed from one of two syringe pumps containing differing media. Following five hours in 0.5% glucose, growth is arrested by a switch to 2% ethanol, a poorer carbon source. The images collected are analysed by BABY and growth rates are determined. When the majority of cells have resumed dividing, detected by the growth rate of at least one of their daughters exceeding $15 \mu\text{m}^3/\text{hour}$, the software triggers a change in pumping and returns glucose to the device. **b** The fraction of cells that have escaped the lag and resumed dividing increases with the amount of time in ethanol. All cells divide shortly after glucose returns. **c** The growth rates of the buds for each mother cell drop substantially in ethanol and resume in glucose. Each row shows data from a single mother cell with the daughters' growth rate indicated by the heat map. Rows are sorted by the time each cell resumes dividing in ethanol.

therefore represented in an expanded morphological space, which aligns the rays with how a cell is oriented. With extra training data, we expect BABY to segment atypical morphologies, such as those of ageing cells, but for more deviant shapes specialised algorithms may be required [34].

We use a flexible, generalizable approach to tracking cells and lineages. To identify the same cell from one image to the next, tracking algorithms often use both Euclidean distance and morphological similarity, with the weighting between the two tuned to the type of image [34, 57]. We generalise this approach by applying machine learning to assign pairs. We thus both formalise how weights are determined and find a probability for each assignment. With these probabilities, we are able to distinguish rigorously between a cell that is correctly tracked and a new cell replacing another and so avoid having to arbitrarily set thresholds. Because buds appear as independent objects that need pairing with mother cells, we apply the same approach to tracking lineages and obtain high accuracies by using features derived from the neural network’s prediction of bud necks.

Once we have generated time series of cellular volumes, we make only weak assumptions about the nature of growth to estimate instantaneous growth rates. Single-cell growth rates in yeast have been variously modelled – for example, as bilinear [13, 24, 25, 42] and as exponential [7, 15, 23, 58] – but that choice has implications for size control [8]. Instead we use Gaussian processes to estimate growth rates [41] and so make assumptions only on the class of functions that describe growth rather than their precise functional form. As well as consistently estimating errors and interpolating to new time points, we show that this non-parametric analysis yields detailed instantaneous growth rates both across the cell cycle and in response to shifts in nutrients.

A well-known challenge in automating segmentation is determining when two cells become independent [57], which we answer by exploiting characteristics of yeast’s growth. Like others [13, 42], we observe that growth rate varies during the cell cycle and in particular peaks during budded growth. This peak is sufficiently regular that we are able to use it to reliably predict the time of cytokinesis.

Our results highlight the importance of single-cell growth rates as phenomenological readouts. We hypothesise that the peak in the bud’s growth rate marks a regulatory transition, triggered when the bud’s size reaches a nutrient-independent threshold. Although the bud’s size is known to be regulated during M phase [24, 25], our data suggest a sequential mechanism that matches size to growth rate, with a nutrient-independent sizer followed by a nutrient-dependent timer. This peak in bud growth may be monitored by Gin4-related kinases [59]. We also report that activation of Sfp1 – a downstream target of TOR kinase [60] – anticipates growth rate, both within a cell-cycle and after a shift in nutrients, consistent with growth being powered by translation [6].

Much of cell biology is focused on understanding how cells respond to change [10], and watching individual cells in real time as their environment alters gives great insight [11]. Together time-lapse microscopy, microfluidic technology, and fluorescent proteins allow us to control extracellular environments, impose dynamic changes, and phenotype cellular responses over time. With BABY, we add the ability – with no extra imaging costs – to measure what is often our best estimate of fitness, single-cell growth rates. The strategies used by cells in their decision making are of high interest [61, 62]. With BABY, or comparable software, we are not only able to rank the strategies that each cell adopts by their fitness, but also to investigate the strategies used to regulate fitness itself through how cells control their growth, size, and divisions.

Methods

Strains and media

Strains included in the curated training images were all derivatives of BY4741 [63]. Both BY4741 Myo1-GFP Whi5-mCherry and BY4741 Sfp1-GFP Nhp6A-mCherry were derived from the respective parent in the *Saccharomyces cerevisiae* GFP collection [64] by PCR-based genomic integration of mCherry-Kan^R from pBS34 (EUROSCARF) to tag either Whi5 or the chromatin-associated Nhp6A protein. All tags were validated by sequencing. Media used for propagation and growth was standard synthetic complete (SC) medium supplemented either with 2% glucose, 2% palatinose or 0.5% glucose depending on the starting condition in the microfluidics devices. Cells were grown at 30°C.

Microscopy and microfluidics

Device preparation and imaging

Overnight cultures were inoculated with low cell numbers such that they would reach mid-log phase in 13-16 hours. Cells were diluted in fresh medium to OD₆₀₀ of 0.1 and incubated an additional 3–4 hours before loading into microfluidic devices at OD₆₀₀ of 0.3–0.4. To expose multiple strains to the same environmental conditions and to optimise data acquisition, we use multi-chamber versions of ALCATRAS [38, 65, 66], which allow for either three or five different strains to be observed in separate chambers while exposed to the same extracellular medium. The ALCATRAS chambers were pre-filled with growth medium with added 0.05% bovine serum albumin (BSA) to facilitate cell loading and reduce clumping. All microfluidics media were passed through 0.2 μm filters before use.

327 We captured images on a Nikon Ti-E microscope using a 60 \times , 1.4 NA oil immersion objective (Nikon), OptoLED
328 light source (Cairn Research) and sCMOS (Prime95B) or EMCCD (Evolve) cameras (both Photometrics) controlled
329 through custom MATLAB software using Micro-manager [67]. The microscope and syringe pumps containing media
330 were maintained at 30 $^{\circ}$ C in a custom-made incubation chamber (Okolabs).

331 **Changing the extracellular environment**

332 For experiments in which the cells experience a change of media, two syringes (BD Emerald, 10ml) mounted in syringe
333 pumps (Aladdin NE-1002X, National Instruments) were connected via PTFE tubing (Smiths Medical) to a sterile
334 metal T-junction. The T-junction's output was delivered via PTFE tubing to the microfluidic device. Initially the
335 syringe with the first medium infused at 4 μ L/min while the second pump was off. To remove back pressure and
336 achieve a rapid switch, medium was infused at 150 μ L/min for 40s from the second pump while the first withdrew at
337 the same rate. The second pump was then set to infuse at 4 μ L/min and the first switched off. This sequence was
338 reversed to achieve a second switch in some experiments. All changes of the flow rates and directions of the pumps
339 were controlled through Matlab software, via RS232 serial ports.

340 **Data and code availability**

341 Data is available at <https://doi.org/10.7488/ds/3427> and code from
342 <https://git.ecdf.ed.ac.uk/swain-lab/baby>.

343 **Birth Annotator for Budding Yeast (BABY) algorithm**

344 The BABY algorithm takes either a stack of bright-field images or a single Z-section as input and coordinates multiple
345 machine-learning models to output individual cell masks annotated for both tracking and lineage relationships.

346 Central to segmenting and annotating lineages is a multi-target convolutional neural network (CNN). Each target
347 of the CNN is semantic – binary labels are assigned to each pixel. These targets are defined to ease both segmenting
348 overlapping instances and assigning lineages in post-processing steps. The reverse task of identifying cell instances from
349 the semantic outputs of the CNN comprises two steps: image processing to identify candidate masks and optimising
350 a radial spline to refine mask edges. Details are given in Appendix 3.

351 To track cells and lineages, we use machine-learning classifiers both to link cell outlines from one time point to the
352 next and to identify mother-daughter relationships. The classifier converts a feature vector, representing quantitatively
353 how two cell masks are related, into probabilities for two possible classes. For cell tracking, the probability of interest
354 is the probability that the two cells at different time points are in fact the same cell. For assigning lineages, the
355 probability of interest is the probability that the two cells have a mother-daughter relationship. A target of the CNN
356 dedicated to assigning lineages is aggregated over time to determine this probability. Details are given in Appendix 4.

357 Scripts automate the training process, including optimising hyperparameters – like the size categories and CNN
358 architecture – and post-processing parameters. We split the training data into training, validation, and test sets [68],
359 with these categories consistently carried through the training pipeline. The training set is used for training the CNN
360 and the validation set for optimising hyperparameters and post-processing parameters. The test set is excluded during
361 this optimisation and is used only to assess performance and generalisability after training.

362 The algorithm is implemented in Python and depends on Tensorflow [69] for the deep-learning models, Scikit-
363 learn [70] for machine learning, and Scikit-image [71] for image processing. The code can be run either directly from
364 Python or as an HTTP server that processes submitted requests and enables access from other languages, such as
365 Matlab.

366 **Measuring growth**

367 **Calculating cell volumes**

368 To estimate cell volumes, we model a 3D cell from our 2D outline. Although the neural network uses information in
369 the z -direction, the resulting segmentation is 2D not 3D, partially because 3D measurements made by epifluorescence
370 microscopy are anisotropic. In our experiments, the distance between two z -positions is usually approximately three
371 times larger than the pixel size in the x and y directions.

372 We use a conical method [27], which is robust to various common cell shapes, to estimate cell volume from an
373 outline. The method makes two assumptions: that the outline obtained cuts through the widest part of the cell and
374 that the cell is ellipsoidal. We build a cone with a base shape that is the filled segmentation outline of the cell by
375 iteratively eroding the mask of the cell and stacking these masks in the z dimension. We find the volume of the cone
376 by summing the voxels in the corresponding 3D mask. Finally, we multiply this sum by four to obtain the volume of
377 the cell, because a cone whose base is the equatorial plane of an ellipsoid will have a volume that is a quarter of the
378 corresponding ellipsoid's volume [27].

379 **Estimating single-cell growth rates**

380 Depending on the need for computational speed, we use one of two methods for estimating instantaneous growth rates.

381 For long-term, and stored, analysis, we estimate growth rates by fitting a Gaussian process with a Matern covariance
382 function to the volume time series for each cell [41]. We set the bounds on the hyperparameters to prevent overfitting.
383 Maximising the likelihood of the hyperparameters, we are able to obtain the mean and first and second time derivatives
384 of the volume, as well as estimates of their errors. The volume's first derivative is the single-cell growth rate.

385 During real-time processing where the growth rate may be used to control the microscope, fitting a Gaussian
386 process is too slow. Instead we estimate growth rates from the smoothed first derivative obtained by Savitzky-Golay
387 filtering of each cell's volume time series. Though faster, this method is less reliable than a Gaussian process and does
388 not estimate errors. For time series of mothers, we use a third-order polynomial with a smoothing window of seven
389 time points; for time series of daughters, we use a fourth-order polynomial also with a smoothing window of seven
390 time points.

391 We estimate growth rates separately for mothers and their buds because both are informative. We find that the
392 summed results are qualitatively similar to previous estimates of growth rate, which fit the time series of the combined
393 volume of the mother and its bud [13, 42].

394 Acknowledgements

395 We thank Yu Huo for providing the data shown in Supplementary Figure 3 and gratefully acknowledge support
396 from the Leverhulme Trust (JMJP & PSS – grant number RPG-2018-04), the BBSRC (IF, IBNC, & PSS – grant
397 number BB/R001359/1), and the European Union's Horizon 2020 research and innovation programme under the Marie
398 Skłodowska Curie grant agreement no. 764591 — SynCrop (AFM & DYA).

399 References

- 400 [1] Orr, H. A. Fitness and its role in evolutionary genetics. *Nat Rev Genet* **10**, 531–539 (2009).
- 401 [2] Broach, J. R. Nutritional control of growth and development in yeast. *Genetics* **192**, 73–105 (2012).
- 402 [3] Levy, S. & Barkai, N. Coordination of gene expression with growth rate: a feedback or a feed-forward strategy?
403 *FEBS Lett* **583**, 3974–3978 (2009).
- 404 [4] Scott, M., Klumpp, S., Mateescu, E. M. & Hwa, T. Emergence of robust growth laws from optimal regulation of
405 ribosome synthesis. *Mol Syst Biol* **10**, 747 (2014).
- 406 [5] Johnston, G. C., Pringle, J. R. & Hartwell, L. H. Coordination of growth with cell division in the yeast *Saccha-*
407 *romyces cerevisiae*. *Exp Cell Res* **105**, 79–98 (1977).
- 408 [6] Jorgensen, P. *et al.* A dynamic transcriptional network communicates growth potential to ribosome synthesis and
409 critical cell size. *Genes Dev* **18**, 2491–505 (2004).
- 410 [7] Di Talia, S., Skotheim, J. M., Bean, J. M., Siggia, E. D. & Cross, F. R. The effects of molecular noise and size
411 control on variability in the budding yeast cell cycle. *Nature* **448**, 947–51 (2007).
- 412 [8] Turner, J. J., Ewald, J. C. & Skotheim, J. M. Cell size control in yeast. *Curr Biol* **22**, R350–9 (2012).
- 413 [9] Ginzberg, M. B., Kafri, R. & Kirschner, M. Cell biology. On being the right (cell) size. *Science* **348**, 1245075
414 (2015).
- 415 [10] Murugan, A. *et al.* Roadmap on biology in time varying environments. *Phys Biol* **18**, 041502 (2021).
- 416 [11] Locke, J. C. & Elowitz, M. B. Using movies to analyse gene circuit dynamics in single cells. *Nat Rev Microbiol*
417 **7**, 383–392 (2009).
- 418 [12] Bennett, M. R. & Hasty, J. Microfluidic devices for measuring gene network dynamics in single cells. *Nat Rev*
419 *Genet* **10**, 628–638 (2009).
- 420 [13] Ferrezuelo, F. *et al.* The critical size is set at a single-cell level by growth rate to attain homeostasis and adaptation.
421 *Nat Commun* **3**, 1012 (2012).
- 422 [14] Schmoller, K. M., Turner, J. J., Kõivomägi, M. & Skotheim, J. M. Dilution of the cell cycle inhibitor Whi5
423 controls budding-yeast cell size. *Nature* **526**, 268–72 (2015).
- 424 [15] Soifer, I., Robert, L. & Amir, A. Single-cell analysis of growth in budding yeast and bacteria reveals a common
425 size regulation strategy. *Curr Biol* **26**, 356–61 (2016).

- 426 [16] Coates, J. *et al.* Antibiotic-induced population fluctuations and stochastic clearance of bacteria. *Elife* **7**, e32976
427 (2018).
- 428 [17] El Meouche, I. & Dunlop, M. J. Heterogeneity in efflux pump expression predisposes antibiotic-resistant cells to
429 mutation. *Science* **362**, 686–690 (2018).
- 430 [18] Levy, S. F., Ziv, N. & Siegal, M. L. Bet hedging in yeast by heterogeneous, age-correlated expression of a stress
431 protectant. *PLoS Biol* **10**, e1001325 (2012).
- 432 [19] Granados, A. A. *et al.* Distributing tasks via multiple input pathways increases cellular survival in stress. *Elife*
433 **6**, e21415 (2017).
- 434 [20] Granados, A. A. *et al.* Distributed and dynamic intracellular organization of extracellular information. *Proc Natl*
435 *Acad Sci U S A* **115**, 6088–6093 (2018).
- 436 [21] Kiviet, D. J. *et al.* Stochasticity of metabolism and growth at the single-cell level. *Nature* **514**, 376–9 (2014).
- 437 [22] Chen, K. L., Crane, M. M. & Kaeberlein, M. Microfluidic technologies for yeast replicative lifespan studies. *Mech*
438 *Ageing Dev* **161**, 262–269 (2017).
- 439 [23] Chandler-Brown, D., Schmoller, K. M., Winetraub, Y. & Skotheim, J. M. The adder phenomenon emerges from
440 independent control of pre- and post-start phases of the budding yeast cell cycle. *Curr Biol* **27**, 2774–2783.e3
441 (2017).
- 442 [24] Leitao, R. M. & Kellogg, D. R. The duration of mitosis and daughter cell size are modulated by nutrients in
443 budding yeast. *J Cell Biol* **216**, 3463–3470 (2017).
- 444 [25] Garmendia-Torres, C., Tassy, O., Matifas, A., Molina, N. & Charvin, G. Multiple inputs ensure yeast cell size
445 homeostasis during cell cycle progression. *Elife* **7** (2018).
- 446 [26] Litsios, A. *et al.* Differential scaling between G1 protein production and cell size dynamics promotes commitment
447 to the cell division cycle in budding yeast. *Nat Cell Biol* **21**, 1382–1392 (2019).
- 448 [27] Gordon, A. *et al.* Single-cell quantification of molecules and rates using open-source microscope-based cytometry.
449 *Nat Methods* **4**, 175–81 (2007).
- 450 [28] Falconnet, D. *et al.* High-throughput tracking of single yeast cells in a microfluidic imaging matrix. *Lab Chip* **11**,
451 466–473 (2011).
- 452 [29] Versari, C. *et al.* Long-term tracking of budding yeast cells in brightfield microscopy: Cellstar and the evaluation
453 platform. *J R Soc Interface* **14** (2017).
- 454 [30] Bakker, E., Swain, P. S. & Crane, M. M. Morphologically constrained and data informed cell segmentation of
455 budding yeast. *Bioinformatics* **34**, 88–96 (2018).
- 456 [31] Wood, N. E. & Doncic, A. A fully-automated, robust, and versatile algorithm for long-term budding yeast
457 segmentation and tracking. *PLoS One* **14**, e0206395 (2019).
- 458 [32] Falk, T. *et al.* U-net: deep learning for cell counting, detection, and morphometry. *Nat Methods* **16**, 67–70 (2019).
- 459 [33] Lu, A. X., Zarin, T., Hsu, I. S. & Moses, A. M. Yeastspotter: accurate and parameter-free web segmentation for
460 microscopy images of yeast cells. *Bioinformatics* **35**, 4525–4527 (2019).
- 461 [34] Dietler, N. *et al.* A convolutional neural network segments yeast microscopy images with high accuracy. *Nat*
462 *Commun* **11**, 5723 (2020).
- 463 [35] Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation.
464 In *International conference on medical image computing and computer-assisted intervention*, 234–241 (Springer,
465 2015).
- 466 [36] He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference*
467 *on computer vision*, 2961–2969 (2017).
- 468 [37] Schmidt, U., Weigert, M., Broaddus, C. & Myers, G. Cell detection with star-convex polygons. In *Medical Image*
469 *Computing and Computer Assisted Intervention - MICCAI 2018 - 21st International Conference, Granada, Spain,*
470 *September 16-20, 2018, Proceedings, Part II*, 265–273 (2018).

- 471 [38] Crane, M. M., Clark, I. B. N., Bakker, E., Smith, S. & Swain, P. S. A microfluidic system for studying ageing
472 and dynamic single-cell responses in budding yeast. *PLoS One* **9**, e100042 (2014).
- 473 [39] Hartwell, L. H. & Unger, M. W. Unequal division in *Saccharomyces cerevisiae* and its implications for the control
474 of cell division. *J Cell Biol* **75**, 422–35 (1977).
- 475 [40] Cerulus, B. *et al.* Transition between fermentation and respiration determines history-dependent behavior in
476 fluctuating carbon sources. *Elife* **7** (2018).
- 477 [41] Swain, P. S. *et al.* Inferring time derivatives including cell growth rates using Gaussian processes. *Nat Commun*
478 **7**, 13766 (2016).
- 479 [42] Cookson, N. A., Cookson, S. W., Tsimring, L. S. & Hasty, J. Cell cycle-dependent variations in protein concen-
480 tration. *Nucleic Acids Res* **38**, 2676–81 (2010).
- 481 [43] Bi, E. *et al.* Involvement of an actomyosin contractile ring in *Saccharomyces cerevisiae* cytokinesis. *J Cell Biol*
482 **142**, 1301–12 (1998).
- 483 [44] Albert, B. *et al.* Sfp1 regulates transcriptional networks driving cell growth and division through multiple
484 promoter-binding modes. *Genes Dev* **33**, 288–293 (2019).
- 485 [45] Lempiäinen, H. *et al.* Sfp1 interaction with TORC1 and Mrs6 reveals feedback regulation on TOR signaling. *Mol*
486 *Cell* **33**, 704–16 (2009).
- 487 [46] Singh, J. & Tyers, M. A Rab escort protein integrates the secretion system with TOR signaling and ribosome
488 biogenesis. *Genes Dev* **23**, 1944–58 (2009).
- 489 [47] Metz-Raz, E. *et al.* Principles of cellular resource allocation revealed by condition-dependent proteome profiling.
490 *Elife* **6**, e28034 (2017).
- 491 [48] Harrigan, P., Madhani, H. D. & El-Samad, H. Real-time genetic compensation defines the dynamic demands of
492 feedback control. *Cell* **175**, 877–886.e10 (2018).
- 493 [49] Miliadis-Argeitis, A. *et al.* In silico feedback for in vivo regulation of a gene expression circuit. *Nat Biotechnol* **29**,
494 1114–6 (2011).
- 495 [50] Toettcher, J. E., Gong, D., Lim, W. A. & Weiner, O. D. Light-based feedback for controlling intracellular signaling
496 dynamics. *Nat Methods* **8**, 837–9 (2011).
- 497 [51] Uhlenhof, J. *et al.* Long-term model predictive control of gene expression at the population and single-cell levels.
498 *Proc Natl Acad Sci U S A* **109**, 14271–6 (2012).
- 499 [52] Lugagne, J.-B. *et al.* Balancing a genetic toggle switch by real-time feedback control and periodic forcing. *Nat*
500 *Commun* **8**, 1671 (2017).
- 501 [53] Menolascina, F. *et al.* In-vivo real-time control of protein expression from endogenous and synthetic gene networks.
502 *PLoS Comput Biol* **10**, e1003625 (2014).
- 503 [54] Hansen, A. S., Hao, N. & O’Shea, E. K. High-throughput microfluidics to control and measure signaling dynamics
504 in single yeast cells. *Nat Protoc* **10**, 1181–97 (2015).
- 505 [55] Bakshi, S. *et al.* Tracking bacterial lineages in complex and dynamic environments with applications for growth
506 control and persistence. *Nat Microbiol* **6**, 783–791 (2021).
- 507 [56] Lugagne, J.-B. *et al.* Identification of individual cells from z-stacks of bright-field microscopy images. *Sci Rep* **8**,
508 11455 (2018).
- 509 [57] Ulman, V. *et al.* An objective comparison of cell-tracking algorithms. *Nat Methods* **14**, 1141–1152 (2017).
- 510 [58] Godin, M. *et al.* Using buoyant mass to measure the growth of single cells. *Nat Methods* **7**, 387–90 (2010).
- 511 [59] Jasani, A., Huynh, T. & Kellogg, D. R. Growth-dependent activation of protein kinases suggests a mechanism
512 for measuring cell growth. *Genetics* **215**, 729–746 (2020).
- 513 [60] Lempiäinen, H. & Shore, D. Growth control and ribosome biogenesis. *Curr Opin Cell Biol* **21**, 855–63 (2009).
- 514 [61] Perkins, T. J. & Swain, P. S. Strategies for cellular decision-making. *Mol Syst Biol* **5**, 326 (2009).

- 515 [62] Balázsi, G., van Oudenaarden, A. & Collins, J. J. Cellular decision making and biological noise: from microbes
516 to mammals. *Cell* **144**, 910–925 (2011).
- 517 [63] Brachmann, C. B. *et al.* Designer deletion strains derived from *saccharomyces cerevisiae* s288c: a useful set of
518 strains and plasmids for pcr-mediated gene disruption and other applications. *Yeast* **14**, 115–32 (1998).
- 519 [64] Huh, W.-K. *et al.* Global analysis of protein localization in budding yeast. *Nature* **425**, 686–91 (2003).
- 520 [65] Granados, A. A. *et al.* Distributing tasks via multiple input pathways increases cellular survival in stress. *eLife*
521 **6** (2017).
- 522 [66] Crane, M. M. *et al.* DNA damage checkpoint activation impairs chromatin homeostasis and promotes mitotic
523 catastrophe during aging. *eLife* **8** (2019).
- 524 [67] Edelstein, A. D. *et al.* Advanced methods of microscope control using μ Manager software. *J Biol Methods* **1**
525 (2014).
- 526 [68] Goodfellow, I., Bengio, Y. & Courville, A. *Deep learning* (MIT press, 2016).
- 527 [69] Abadi, M. *et al.* TensorFlow: Large-scale machine learning on heterogeneous systems (2015). URL
528 <https://www.tensorflow.org/>. Software available from tensorflow.org.
- 529 [70] Pedregosa, F. *et al.* Scikit-learn: Machine learning in python. *J Mach Learn Res* **12**, 2825–2830 (2011).
- 530 [71] van der Walt, S. *et al.* scikit-image: image processing in Python. *PeerJ* **2**, e453 (2014). URL
531 <https://doi.org/10.7717/peerj.453>.
- 532 [72] Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation
533 (2015). URL <https://arxiv.org/abs/1505.04597>. 1505.04597.
- 534 [73] Pelt, D. M. & Sethian, J. A. A mixed-scale dense convolutional neural network for image analysis. *Proc Natl*
535 *Acad Sci U S A* **115**, 254–259 (2018).
- 536 [74] Bishop, C. M. *Pattern recognition and machine learning* (Springer, 2006).
- 537 [75] Chen, T. & Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd*
538 *international conference on knowledge discovery and data mining*, 785–794 (2016).
- 539 [76] Costanzo, M. *et al.* CDK activity antagonizes Whi5, an inhibitor of G1/S transcription in yeast. *Cell* **117**,
540 899–913 (2004).
- 541 [77] Dunlop, M. J., Cox, R. S., 3rd, Levine, J. H., Murray, R. M. & Elowitz, M. B. Regulatory activity revealed by
542 dynamic correlations in gene expression noise. *Nat. Genet.* **40**, 1493–8 (2008).
- 543 [78] Lichten, C. A., White, R., Clark, I. B. N. & Swain, P. S. Unmixing of fluorescence spectra to resolve quantitative
544 time-series measurements of gene expression in plate readers. *BMC Biotechnol.* **14**, 11 (2014).

Appendix 1

Quantifying localization

During each experiment, we acquired bright-field and fluorescence images at five z -sections spaced 0.6 micron apart. The maximum projection of these images (the maximum pixel values across all z -sections) was used for quantification.

For each cell, we determined its fluorescence image by subtracting the median fluorescence of the cell's pixels and setting all non-cell pixels to zero. This median fluorescence will be determined by a cytoplasmic pixel, and we assume that it results from autofluorescence only, which requires sufficiently low numbers of fluorescent markers.

To quantify fluorescent markers in the nucleus, we noted that fluorescence in the nucleus appears in an image as a two-dimensional Gaussian distribution because of point spreading in epifluorescence microscopes. We therefore identified the most probable location of the nucleus for each cell by convolving a Gaussian filter with the fluorescence image. The maximal value in the resulting filtered image marks the location that most closely matches this filter.

Using data from nuclei segmented via Nhp6A-mCherry reporters [20], we observed that the area of the nucleus A_{nuc} scales as a fraction of cell area A_{cell} with a scaling factor $f_{\text{nuc}} \simeq 0.085$. We used this result to estimate a standard deviation σ for the Gaussian filter. If the nucleus is approximately circular then its radius can be estimated as

$$r_{\text{nuc}} = \sqrt{\frac{f_{\text{nuc}} A_{\text{cell}}}{\pi}}. \quad (1)$$

Assuming that 95% of the fluorescence in the nucleus is contained within the segmented area of nucleus, then we choose the σ of the Gaussian filter so that 95% of its probability is obtained by integrating over a circle of radius r_{nuc} . Writing $\alpha = 0.95$, we have

$$\begin{aligned} \alpha &= \int_{A_{\text{nuc}}} dx dy \frac{e^{-\frac{x^2+y^2}{2\sigma^2}}}{2\pi\sigma^2} \\ &= 2\pi \int_0^{r_{\text{nuc}}} dr r \frac{e^{-\frac{r^2}{2\sigma^2}}}{2\pi\sigma^2} \\ &= 1 - e^{-\frac{r_{\text{nuc}}^2}{2\sigma^2}} \end{aligned} \quad (2)$$

switching to polar coordinates. Using that the cumulative distribution function of the χ^2 distribution with two degrees of freedom is $F_{\chi^2}(x) = 1 - e^{-x^2/2}$, we can rearrange Eq. 2 and combine with Eq. 1 to give

$$\sigma = \sqrt{\frac{f_{\text{nuc}} A_{\text{cell}}}{\pi F_{\chi^2}^{-1}(\alpha)}}. \quad (3)$$

We next assume an ideal fluorescence image of the nucleus can be described by the same Gaussian filter but rescaled by some amplitude a_f . If we apply the Gaussian convolution G to the pixel in this ideal image with maximal fluorescence, we obtain

$$a_f \|G^2\| \quad (4)$$

where $\|G^2\|$ is the sum of the squared values of the Gaussian filter. This quantity should in principle be equal to $\alpha \max(C)$, where C is the Gaussian filtered fluorescence image of the actual cell. Therefore

$$a_f = \frac{\alpha \max(C)}{\|G^2\|}. \quad (5)$$

Finally, a_f is our prediction of the total nuclear fluorescence, but the concentration is more biologically relevant and, if denoted N , is

$$\begin{aligned} N &= \frac{a_f}{f_{\text{nuc}} A_{\text{cell}}} \\ &= \frac{\alpha \max(C)}{f_{\text{nuc}} A_{\text{cell}} \|G^2\|} \end{aligned} \quad (6)$$

which is the measure we use.

For quantifying the localization of Myo1-GFP to the bud neck, we note that N is a sensitive proxy for localization and assume that it applies equally well in this case.

574 **Appendix 2**

575 **Training data and a graphical user interface for curating**

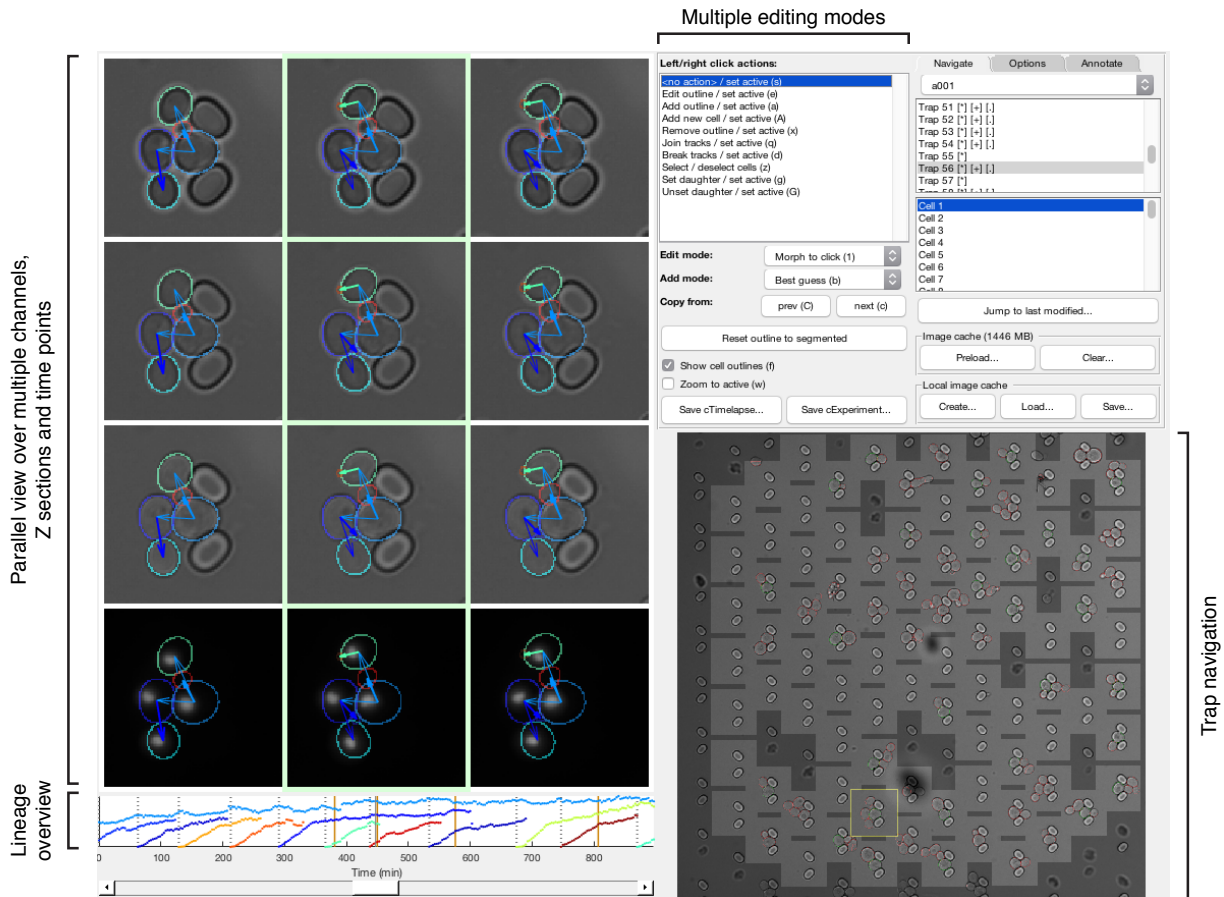


Figure 1: **Main features of the graphical user interface used for annotation.** A custom graphical user interface (GUI) was developed in Matlab for efficient annotation of overlapping cell instances, tracks and lineages over long time courses. The screen-shot shows the GUI in its horizontal layout with 3 bright-field sections and a fluorescence channel selected for parallel view. Annotated outlines and arrows indicating lineage relationships have each been toggled on for display. Up to 9 time points can be displayed in parallel; the slider at the bottom allows fast scrolling through the entire time-lapse. A time-course summary panel is displayed above the slider and has been set to show the outline areas for a mother and all its daughters. An overview image of the entire position allows navigation between traps. Multiple editing modes can be selected for manipulating annotations in the parallel view region, including modes for draggable outline editing, track merging/splitting and lineage reassignments.

576 Training data for the machine learning models consisted of bright-field time-lapse images of yeast cells trapped
577 in ALCATRAS devices [38] and manually curated annotations: a bit-mask outline for each cell and its associated
578 tracking label and lineage assignment, if any. All images were taken with five Z sections and 60× lens, including
579 examples taken using cameras with different pixel sizes (0.182 μm and 0.263 μm).

580 To ease annotating overlapping instances, cell tracks and lineage relationships, we developed a Graphical User
581 Interface (GUI) in Matlab that allows parallel viewing of multiple Z sections and time points (Appendix 2 Fig. 1).
582 This parallel view helps curate buds obscured by a lack of focus and those that might be missed without simultaneously
583 observing multiple time points. Manipulations made to outlines and tracks are mirrored to all views in real time. The
584 interface is highly customisable, with multiple layouts available and the ability to select which sections and channels
585 are displayed. To edit outlines for smaller cells, the level of zoom can be adjusted. Further, starting outlines can be
586 copied across time points and interpolated forwards or backwards in time (interpolated outlines are annotated as such
587 until they are manually adjusted).

588 Annotations are saved in a custom format for computational efficiency, but various export options are available.
589 For training we exported annotations in PNG format with one image per time point. Because outlines can potentially

590 overlap, they are tiled, with one cell instance per tile. The metadata of the PNG file is used to store track and lineage
591 annotations.

592 The GUI also includes features to efficiently detect and correct rare errors. A track display panel provides visual
593 aids to summarise tracks across the entire time course. In particular, the ‘Display mother and daughter areas’ mode
594 uses this panel to plot the area of the currently selected cell and all of its daughters over the time course. Using this
595 mode, many segmentation and tracking errors are highlighted as unexpected jumps in area or changes in track label
596 (denoted in colour). We use a slider to navigate to these errors where they can be either corrected in place or saved
597 for future curation.

598 Although the GUI can be used on whole images, it includes features to navigate and annotate images that can be
599 partitioned into regions, such as the traps of our ALCATRAS devices. Then the trap navigation image shows trap
600 locations and can be used to move between traps.

Appendix 3

The BABY algorithm: identifying cells and buds

Mapping cell instances to a semantic representation

For epifluorescence microscopy, samples are typically prepared to constrain cells in a monolayer. If the cells have similar sizes and match the height of this constraint, they will be physically prevented from overlapping. If cells are of different sizes, however, then a small cell can potentially fit in gaps and overlap with others. This phenomenon is especially prevalent for cells that divide asymmetrically, where a small daughter grows out of a larger mother.

Few segmentation algorithms identify instances of overlapping cells. Most, including recent methods for budding yeast [31, 34, 56], make the assumption that cells can be labelled semantically – each pixel of the image is identified with at most one cell. Similarly, most tools for annotating also label semantically, and consequently curated training data does not allow for overlaps [34], even when the segmentation algorithm could [33]. Our laboratory’s previous segmentation algorithm included limited overlap between neighbouring cells [30], but not the substantial overlap seen between the smaller buds and their neighbours.

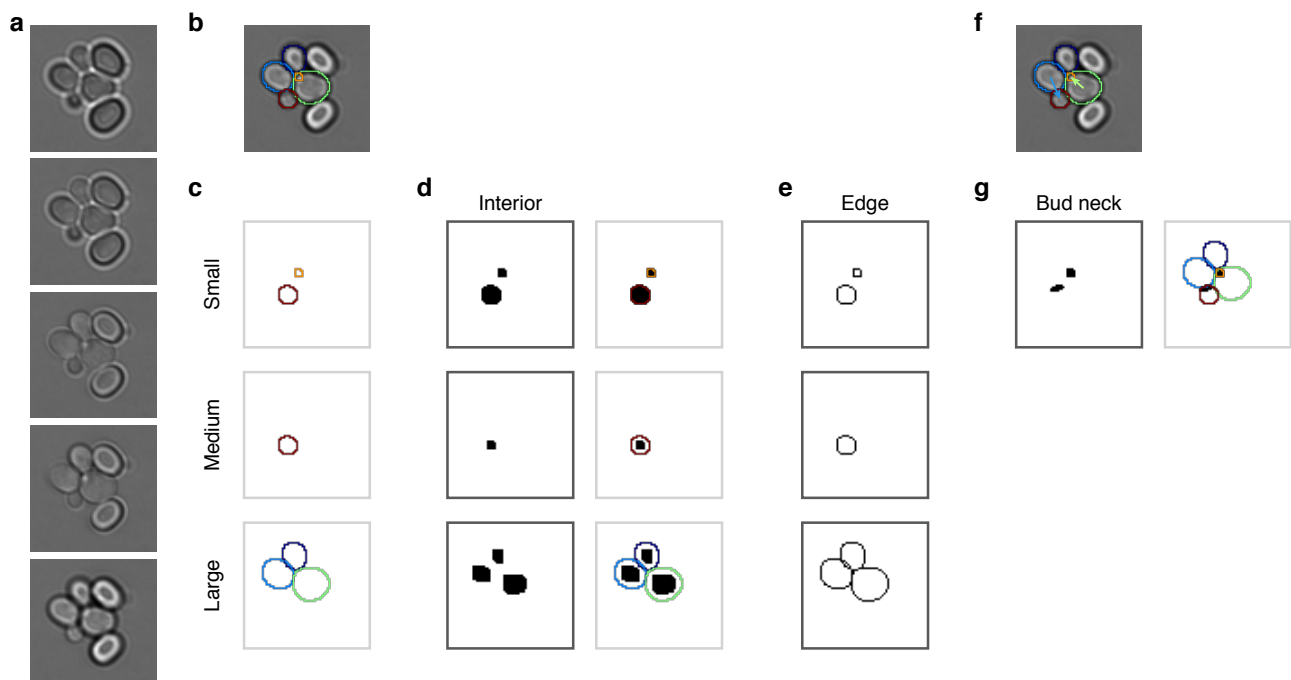


Figure 1: **Mapping cell instances to semantic targets of a CNN.** **a** Bright-field Z-sections of cells trapped in an ALCATRAS device. **b** Curated cell outlines overlaid on one bright-field section. **c** Outlines are separated into categories by size. Each category is chosen to retain some overlap with neighbouring categories. Here the red outline in the medium category is duplicated in the small category. **d** Interior targets are cell masks after different rounds of morphological erosions appropriate for each size category: no erosion for small cells, four iterations for medium, and five for large. On the right, outlines are overlaid on the target masks for reference. **e** Edge targets are the outlines for each size category. **f** The curated cell outlines of **b**, but with arrows to show the lineages assigned during curation. **g** Using these curated lineages, the ‘bud neck’ target is defined as the overlap of the bud mask with a morphological dilation of the mother mask (right).

Separating cells by size to disjoin overlapping cells We rely on two consequences of the height constraint to segment overlapping instances. First, cells of different sizes show different patterns of overlap; second, cells rarely have coincident centres. Very occasionally, we do observe small buds stacked directly, one on top of the other, but neglecting these rare cases does not degrade performance. We therefore use morphological erosions to obtain semantic images by shrinking cell masks within a size category and morphological dilations to approximate the original cell outlines from each resulting connected region.

To separate overlapping cells, we define three size categories and treat instances in each category differently. Appendix 3 Fig. 1 illustrates our approach, where we segment a bud (orange outline) that overlaps a mother cell (green outline). The bud is only visible in the third and fourth Z sections of the bright-field images (Appendix 3

623 Fig. 1a). If used for training, the manually curated outlines in this example (Appendix 3 Fig. 1b) would be split into
 624 different size categories (Appendix 3 Fig. 1c). The bud is assigned to the small category. When the outlines in this
 625 category are filled and the image converted into a binary one (Appendix 3 Fig. 1d), the cell masks are distinct from
 626 each other. For the large category, however, the masks are not separable when immediately converted, but become
 627 so when the filled outlines are first morphologically eroded (Appendix 3 Fig. 1d). The largest size category tolerates
 628 more erosions than smaller ones, for which the mask may disappear or lose its shape.

629 **Determining the size categories** Using the training data – curated masks for each cell present at each trap at
 630 each time point, we identify the size categories that best separate overlapping cells. To begin, we calculate the overlap
 631 fraction – the intersection over union – between all pairs of cell masks. Its distribution reveals that the most substantial
 632 overlaps occur between cells of different sizes (Appendix 3 Fig. 2a – upper triangle).

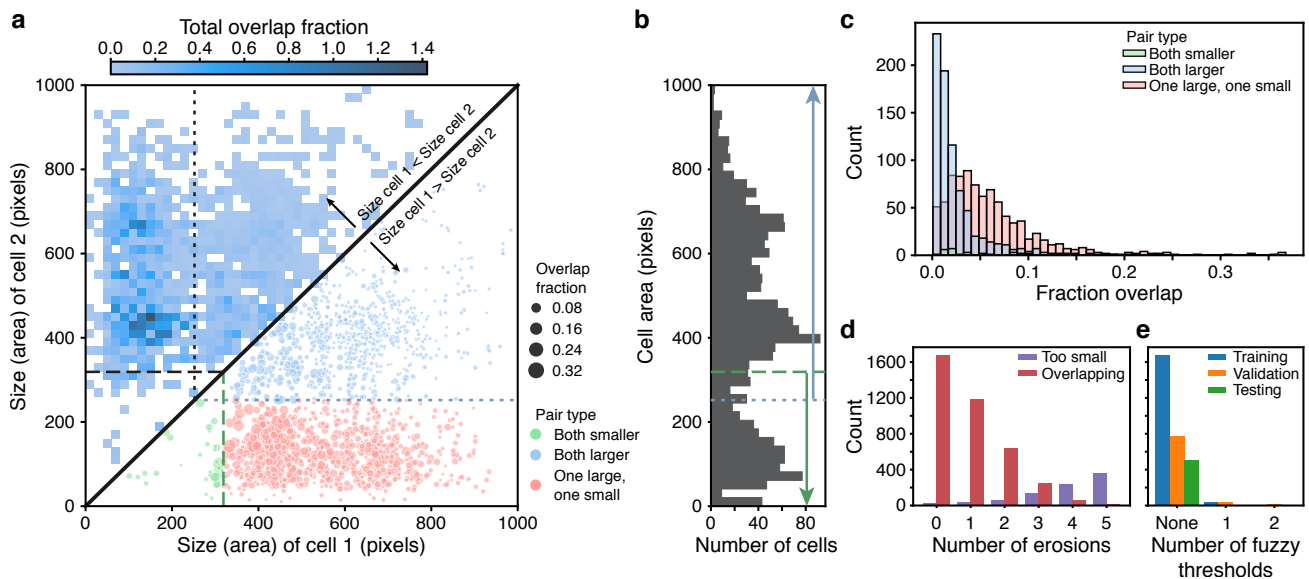


Figure 2: **Overlaps between cells are reduced through categorising cells by size.** **a** Upper triangle: plotting the overlap fraction for each pair of cells – the intersection over union of their bit masks, shows that the majority of overlaps occur for cells of different sizes. Almost all overlaps have the size of cell 2 greater than the size of cell 1 and lie off the diagonal. Lower triangle: With a single fuzzy size threshold, cells in the small category have sizes less than the upper threshold (dashed line), and cells in the larger category have sizes greater than the lower threshold (dotted line). Within each category, the overlap fractions are mostly small; between the two categories, overlaps are mostly large (excluded cells in red). By converting the bit masks into two binary images, one for each size category, rather than a single binary image, we therefore eliminate, or exclude, most of the substantial overlaps. **b** The distribution of all mask areas in the same training data for comparison. Size thresholds are indicated as in **a**. **c** The distributions of overlap fractions for mask pairs grouped using the fuzzy size threshold described in **a**. We omit pairs that do not overlap for clarity. **d** Applying morphological erosions of the cell masks reduces the number of overlapping cell pairs, but generates smaller masks. We judge masks with areas below 10 pixels squared to be too small. **e** The numbers of overlapping cell pairs remaining from the training, validation and test sets either before (denoted None) or after splitting into size categories and applying an optimised number of erosions.

633 We therefore choose the size categories so that most overlaps occur between pairs of cells in different categories and
 634 little overlap occurs between pairs of cells within a category. For example, rather than converting the cell masks
 635 directly into a single binary image for training, if first we divide cells into two size categories and convert the masks
 636 within each category to a separate binary image, giving two images rather than one, then in these two images we have
 637 eliminated all overlaps occurring between cells in the smaller category with cells in the larger category (Appendix 3
 638 Fig. 1 & 2a – lower triangle).

639 To divide the cell masks into two categories, we define a fuzzy size threshold using a threshold T and padding
 640 value P . The set of smaller masks is all masks whose area is less than $T + P$; the set of larger masks is all masks
 641 whose area is greater than $T - P$. Consequently, it is possible to have the same mask in both sets (Appendix 3 Fig.
 642 1c). This redundancy ensures the CNN can produce confident predictions even for cells close to the size threshold –
 643 we eliminate any resulting duplicate predictions in post-processing. A pair of masks are prevented from overlapping,
 644 by being converted into distinct binary images, if their sizes are separated by the threshold (after padding): the two

645 masks must be in different size categories, and so the smaller cell must have a size $< T - P$ and the larger cell must
646 have a size $> T + P$. To scale with pixel size, we set P to be 3% of the area of the largest mask in the training set.

647 To determine an optimal fuzzy threshold, we test 29 values evenly spaced between the minimal and maximal mask
648 sizes and choose the threshold that minimises the summed overlap fraction for all mask pairs not excluded by the
649 threshold. Even with just one fuzzy threshold (Appendix 3 Fig. 2a), most of the pairs with substantial overlap –
650 typically buds with neighbouring cells – are excluded (Appendix 3 Fig. 2c).

651 After applying the threshold, overlaps between cells within a size category remain, and we reduce such overlaps
652 using morphological erosions (Appendix 3 Fig. 1). We use the training data to optimise the number of erosions applied
653 to each size category. As the number of iterations increases, there is a trade-off between the number of overlapping
654 mask pairs and the number of masks whose eroded areas become too small to be confidently predicted by the CNN
655 (Appendix 3 Fig. 2d). Without erosion, many of the large cells show overlaps; with too much erosion, the smallest
656 masks distort their shapes or disappear completely. We therefore optimise the number of iterations separately for each
657 size category, picking the highest number of iterations that can be applied without any of the training masks in that
658 size category falling below an absolute minimal size, defined as 10 pixels squared.

659 Combining categorising by size with eroding reduces the number of pairs of overlapping masks almost to zero
660 (Appendix 3 Fig. 2e). We arrive at three size categories by first introducing an additional fuzzy threshold for each
661 of the two initial size categories. These thresholds are similarly determined by testing 29 fuzzy threshold values and
662 calculating the overlap fraction for all mask pairs not excluded by either the original or the new threshold. We only
663 keep one of the new thresholds – the one minimising the overlap fraction, giving three size categories in total. This
664 extra category results in a further, although proportionally smaller, decrease in the number of overlapping masks.

665 After erosion, mask interiors within each size category are easily identified, but with less resolved edges. To help
666 alleviate this loss, we generate edge targets from the training data (Appendix 3 Fig. 1e) – the outlines of all cells
667 within each size category.

668 **Three types of training targets** The curated data is further annotated with lineage assignments (Appendix 3
669 Fig. 1f), which we use to generate ‘bud neck’ targets (Appendix 3 Fig. 1g).

670 In total, then, the training targets for the CNN are the mask interiors and the edges for three size categories and
671 the bud necks.

672 Predicting semantic targets with a convolutional neural network

673 We trained fully convolutional neural networks [68] to map a stack of bright-field sections to multiple binary target
674 images. Example inputs and outputs are shown in Appendix 3 Fig. 1, but we also trained networks with only
675 one or three bright-field sections. The intensities of the bright-field sections were normalised to the interval $[-1, 1]$
676 by subtracting the median and by scaling according to the range of intensities expected between the 2nd and 98th
677 percentiles.

678 Each output layer of the CNN approximates the probability that a given pixel belongs to the target class, being a
679 convolution with kernel of size 1×1 and sigmoidal activation. All other convolutions had kernels of size 3×3 with
680 ReLU activation and used padding to ensure consistent dimensions for input and output layers. Different core network
681 architectures were tested and trained.

682 **Augmenting the training data** To prevent over-fitting and improve generalisation, we augmented the training
683 data [68]. Each time a training example was presented to the CNN, a randomly selected series of image manipulations
684 was applied to the input and target. The same training example therefore typically appears differently for each epoch.

685 One augmentation was always applied and the others applied with a certain probability. If the bright-field input
686 had more Z sections than expected by the network, we selected a random subset, excluding any subsets with selected
687 sections that were separated by two or more missing sections. The remaining augmentations were applied with a
688 probability p and comprised vertical and horizontal translation (each with $p = 0.25$), image rotation ($p = 0.2$), re-
689 scaling ($p = 0.08$), vertical and horizontal flips (each with $p = 0.08$), addition of white noise ($p = 0.08$), and a step
690 shift of the Z sections ($p = 0.08$). The probability of not augmenting was thus $p = 0.30$. To show a different region of
691 each image-mask pair at each epoch, translation, rotation and re-scaling were all applied to images and masks before
692 cropping to a consistent size (96×96 pixels for a pixel size of $0.182 \mu\text{m}$). Using reflection to handle the boundary,
693 translations were over a random distance and rotations over a random angle. The factor used for re-scaling was
694 randomly adjusted by up to 5%. Augmentation by addition of white noise involved adding random Gaussian noise
695 with a standard deviation picked from an exponential distribution with rate 0.003 to each pixel of the (normalised)
696 bright-field images.

697 To reduce aliasing errors when manipulating binary masks during augmentation, all image transformations were
698 independently applied to each filled mask before converting the transformed masks into one binary image. Further,
699 before a transformation, each binary filled outline was smoothed with a 2D Gaussian filter, and after the transformation,
700 the transformed binary outline was found by the Canny algorithm. To determine the standard deviation of this
701 Gaussian filter, σ , we tested a range of values on the training outlines. For each filled outline and σ , we applied

702 the filter followed by edge detection and filling. We then calculated the intersection over union of the resulting filled
 703 outline with the original filled outline. We observed that as a function of edge length, defined as the number of pixels
 704 in the outline, the σ producing the highest intersection over union increased exponentially. We consequently used an
 705 exponential fit of this data to estimate an appropriate σ for each outline.

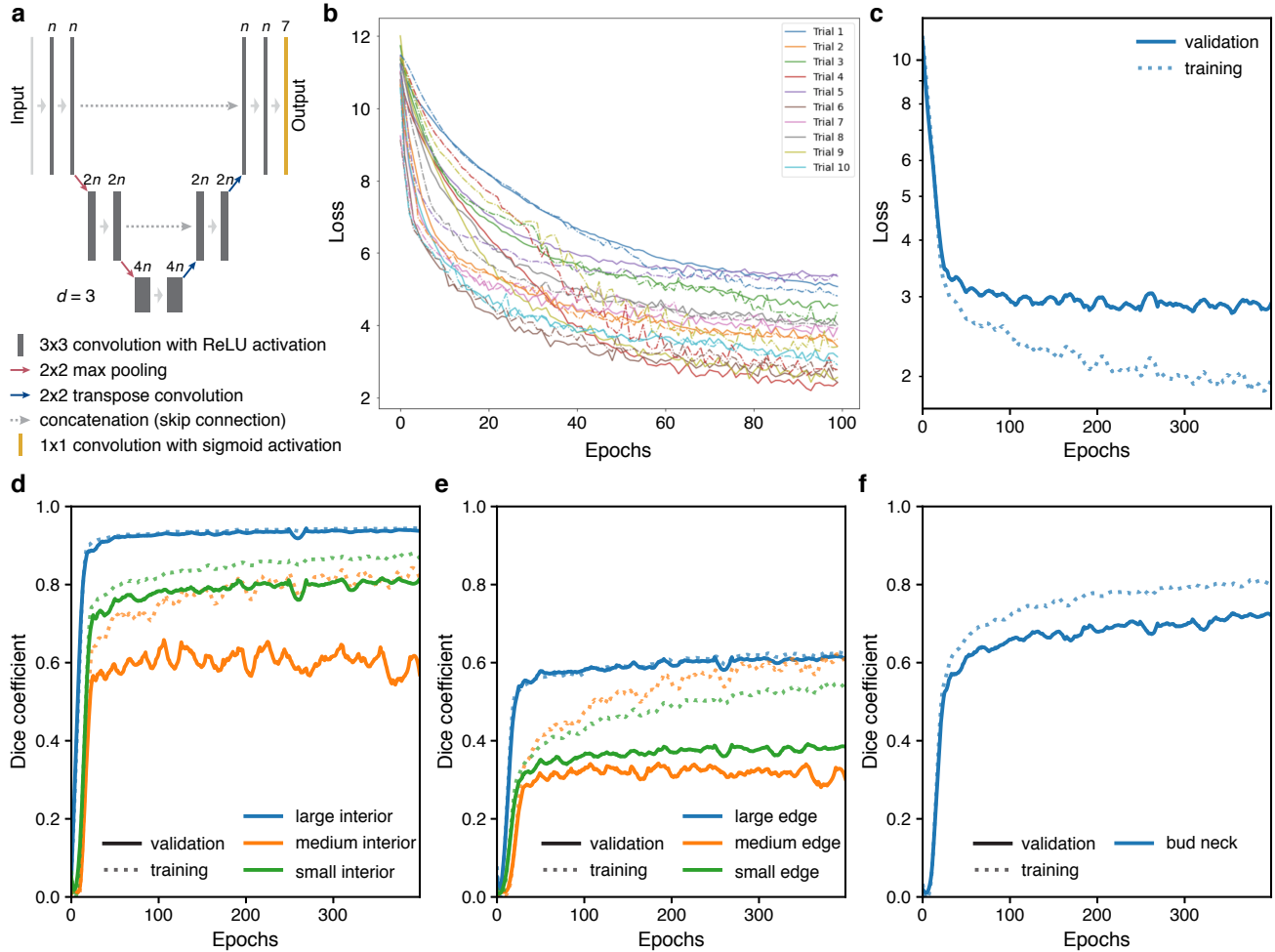


Figure 3: **Training performance of the multi-target U-Net.** **a** Schematic U-Net architecture with depth $d = 3$. Labels above convolution operations indicate number of output filters as a multiple of n . Layer heights indicate reduction in image width/height with network depth. **b** Training (solid) and validation (dot-dash) loss for 10 training trials using the U-Net architecture each with randomly chosen hyperparameters. **c** Loss for the fully trained U-Net with hyperparameters chosen from the trial network with the lowest final validation loss (a U-Net with depth $d = 4$, filter factor $n = 8$ and batch normalisation). **d-f** Performance of (d) interior, (e) edge and (f) bud neck targets by the U-Net of **c** decomposed into the three different size categories when possible. The Dice coefficient reports similarity between prediction probabilities and target masks with a value of 1 indicating identity. For two sets X and Y , the Dice coefficient is $\frac{2|X \cap Y|}{|X| + |Y|}$.

706 **Training** Networks were trained using Keras with TensorFlow 1.14. We used Adam optimisation with default
 707 parameters except for a learning rate of 0.001 and regularised by keeping only the network weights from the epoch
 708 with the lowest validation loss (similar in principle to the early stopping method) [68]. We train for 400 epochs, or
 709 complete iterations over the training data set.

710 The loss function is the sum of the binary cross-entropy and one minus the Dice coefficient across all targets:

$$L = - \sum_i \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right] + 1 - \frac{2 \sum_i y_i \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i} \quad (7)$$

711 where y is the tensor of true values, \hat{y} is the CNN's sigmoid tensor output of the CNN, and i is a vectorised index.

712 Each CNN is trained to a specific pixel size, and we ensured that training images and masks with different pixel
713 sizes were re-scaled appropriately

714 **CNN architectures** We trialled two core architectures for the CNN – U-Net [72] (Appendix 3 Fig. 3a) and Mixed-
715 Scale-Dense (MSD) [73] – and optimised hyperparameters to find the smallest loss on the validation data.

716 The U-Net performed best (see ‘Optimising hyperparameters’ below). The U-Net has two parts: an encoder that
717 reduces the input into multiple small features and a decoder that upscales these features into an output [72]. Each step
718 of the encoder comprises a convolutional block, which creates a new, larger set of features from its input. To force the
719 network to keep only small, relevant features, a down-sampling step is applied after three convolutional blocks. This
720 maximum pooling layer reduces the size of the features by half by replacing each two-by-two block of pixels by their
721 maximal value. The decoder also comprises convolutional blocks, but with up-sampling instead of down-sampling.
722 The up-sampling step is the inverse of down-sampling: each pixel is turned into a two-by-two block by repeating its
723 value. Finally, most characteristic of the U-Net is its skip layers. These layers preserve information on the global
724 organisation of the pixels by passing larger-scale information from the encoder to the decoder after each up-sampling
725 step. They act by concatenating the same-size layer of the encoder into the decoder layers, which are then used as
726 inputs for the next step of the decoder. The decoder is therefore able to create an output from both the local features
727 that it upsampled and from the global features that it obtains from the skip layers.

728 For the U-Net, we optimised for depth, for a scaling factor for the number of filters output by each convolution,
729 whether or not to include batch normalisation, and for the proportion of neurons to drop out on each batch. For the
730 MSD, we optimised for depth, defined as the total number of convolutions, for the number of dilation rates to loop
731 over, with each loop increasing dilation by a factor of two, for an overall dilation-rate scaling factor, and whether or
732 not to include batch normalisation.

733 **Optimising hyperparameters** We used *KerasTuner* with TensorFlow 2.4 to optimise hyperparameters (Appendix
734 3 Fig. 3b), choosing random search with default settings, training for a maximum of 100 epochs, and having 10 training
735 and validation steps per epoch. The U-Net and MSD networks with the lowest final validation loss were then re-trained
736 as described (Appendix 3 Fig. 3c), and the network with the lowest validation loss chosen.

737 For our data, the best performing model was a U-Net with depth four, and so three contractions, with a scaling
738 factor of eight for the number of filters output by each convolution, giving 8, 16, 32 and 64 filters for each of the two
739 chained convolution layers of the encoding and decoding blocks, with batch normalization, and with no drop-out. Its
740 performance is shown in Appendix 3 Fig. 3d–f.

741 Identifying cells

742 To identify cell instances from the semantic predictions of the CNN, we developed a post-processing pipeline (Appendix
743 3 Fig. 4a). Post-processing can be split into two parts: proposing unique cell outlines and refining edges.

744 The pipeline includes multiple parameters that we optimise on validation data by a partial grid search. We favour
745 precision (the fraction of predicted positives that are true) over recall (the fraction of ground truth positives that are
746 predicted) by maximising the F_β score with $\beta = 0.5$. Recall that for true positives TP, false negatives FN, and false
747 positives FP,

$$748 F_\beta = \frac{(1 + \beta^2)TP}{(1 + \beta^2)TP + \beta^2FN + FP} \quad (8)$$

749 We measure how well two masks match using the intersection over union (IoU) and consider a match to occur when
750 the $\text{IoU} > 0.5$. Nevertheless, multiple predictions may match a single target mask because predicted masks can
751 overlap too. We therefore count true positives as target masks for which there is at least one predicted mask with
752 $\text{IoU} > 0.5$. Any predicted masks in excess of the true positive count become false positives, thus avoiding double
753 counting. Unmatched target masks are false negatives.

754 **Proposing cell outlines** The post-processing pipeline starts by identifying candidate outlines independently for
755 each size category. The CNN’s outputs are images $p_{xy}^{(S,C)} \in [0, 1]$ approximating the probability that a pixel at position
756 (x, y) belongs to either the small, medium, or large size categories, denoted S , and to either the interior (Appendix 3
757 Fig. 4b) or edge (Appendix 3 Fig. 4e & f) target classes, denoted C .

758 In principle, we could find instances for each size category by thresholding the interior probability $p_{xy}^{(S,\text{interior})}$
759 and identifying connected regions as outlines. To further enhance separability, however, we also reweight the interior
760 probabilities using the edge probabilities. Specifically, we identify connected regions from semantic bitmasks $b_{xy}^{(S,\text{interior})}$
761 by those pixels that satisfy

$$762 p_{xy}^{(S,\text{interior})} \left[1 - \text{Dilate}^{N_{\text{dilate}}} \left(p_{xy}^{(S,\text{edge})} \right) \right] > T_{\text{interior}} \quad (9)$$

763 where Dilate^N specifies N iterations of a grayscale morphological dilation and T_{interior} is a threshold. We optimise
764 the thresholds $T_{\text{interior}} \in [0.3, 0.95]$, number of dilations $N_{\text{dilate}} \in \{0, 1, 2\}$, and the order of connectivity (1- or 2-
765 connectivity) for each size category.

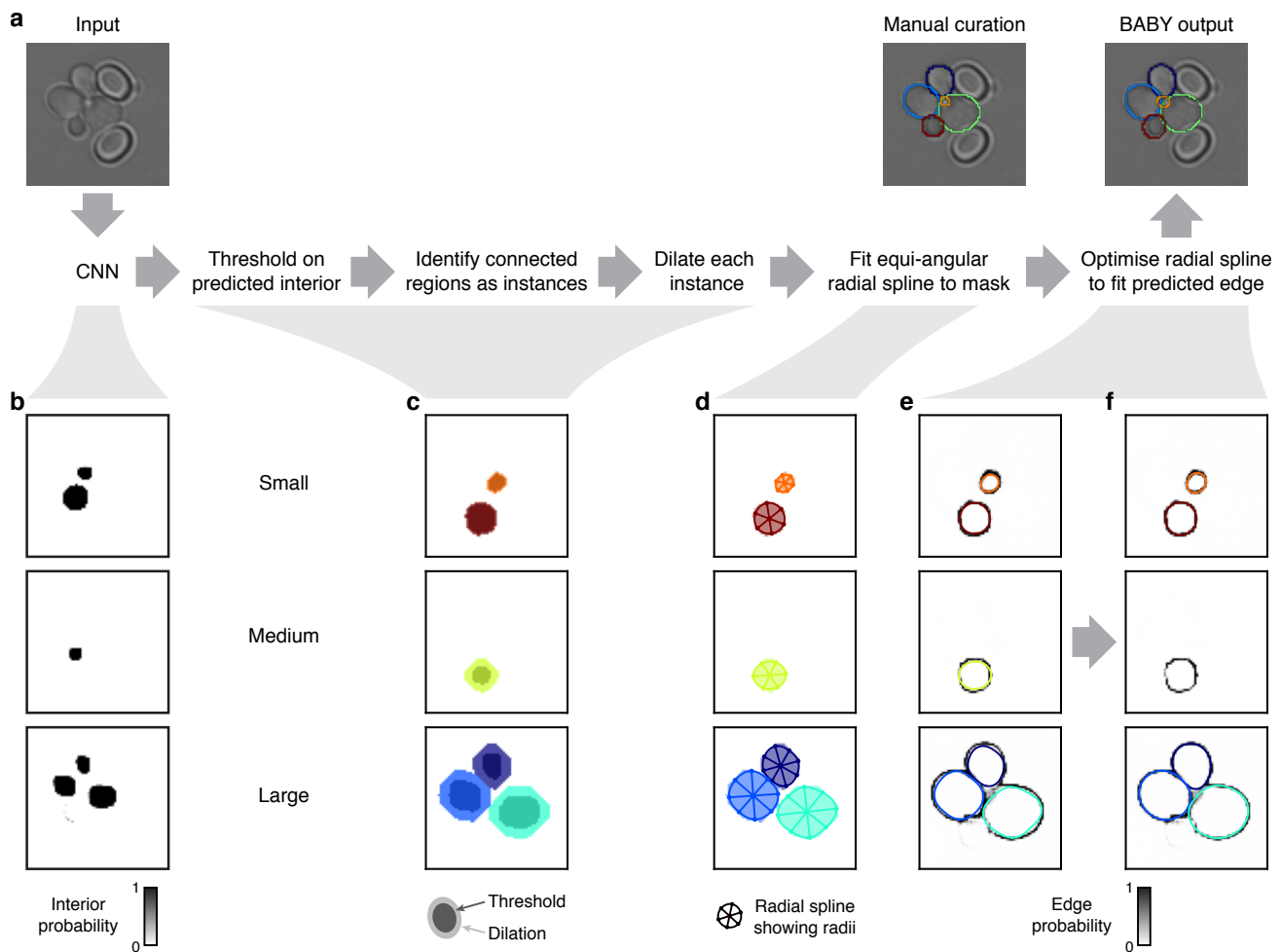


Figure 4: **Segmenting overlapping cell instances from the CNN output.** **a** Flow chart summarising the post-processing for identifying individual instances from the CNN’s multi-target output. Here and below, we show results using the five Z sections of Appendix 3 Fig. 1 as input for the CNN. One of which is repeated here. **b** Probability maps output by the CNN for the interiors of small, medium and large cells. **c** Bitmasks obtained by thresholding on the CNN’s output. Darker shading shows bitmasks before each instance is dilated to compensate for the erosion applied when generating the training targets. Colour indicates distinctly identified instances. **d** The initial, equiangular radial splines proposed for each instance overlaid on the dilated bitmasks from **c**. The rays defining placement of the knots are shown as spokes. **e** The same initial proposed radial splines overlaid on the edge target probability maps output by the CNN. **f** The radial splines after optimisation to match edge probabilities. The outline in the medium size category is detected as a duplicate and is not optimised.

764 The connected regions in $b_{xy}^{(S,interior)}$ define masks that are initial estimates of the cells’ interiors (darker shading in
 765 Appendix 3 Fig. 4c). The actual cell interiors used to train the CNN are generated by iterative binary morphological
 766 erosions of the full mask, where the number of iterations $N_{erosion}$ is pre-determined for each size category. First, we
 767 remove small holes and small foreground features by applying up to two binary morphological closings followed by up
 768 to two binary morphological openings. Second, we estimate full masks $b_{xy}^{(S,full)}$ from each putative mask by applying
 769 $N_{erosion}$ binary dilations (light shading in 4c), undoing the level of erosion on which the CNN was trained. Both the
 770 numbers of closing, $N_{closing}$, and opening, $N_{opening}$, operations are optimised.

771 Masks whose area falls outside the limits for a size category are discarded. For each category, however, we soften
 772 these limits, on top of the fuzzy thresholds, by optimising an expansion factor $F_{exp} \in [0, 0.4]$, which extends the limits
 773 by a fractional amount of that category’s size range. We also optimise a single hard lower threshold $T_{min} \in [0, 20]$ on
 774 mask area.

775 **Using splines to describe mask edges** To prepare for refining edges and to further smooth and constrain outlines,
 776 we use a radial spline to match the edge of each of the remaining masks (Appendix 3 Fig. 4d). As in DISCO [30],
 777 we define radial splines as periodic cubic B splines using polar coordinates whose origin is at the mask’s centroid.

Parameter	$S = \text{small}$	$S = \text{medium}$	$S = \text{large}$
T_{interior}	0.35	0.5	0.95
N_{dilate}	0	0	0
Connectivity	2	1	1
N_{closing}	0	0	0
N_{opening}	0	0	2
F_{exp}	0.32	0.06	0.28
T_{edge}	0.0012	0.0028	0.0
T_{min}		19	
$T_{\text{containment}}$		0.85	

Table 1: **Optimised post-processing parameters for standard model.** The standard model takes five brightfield Z sections with a pixel size of $0.182 \mu\text{m}$ as input. Excepting T_{min} and $T_{\text{containment}}$, parameters are optimised separately for each size category.

We generalise this representation to have a variable number $n^{(S,i)}$ of knots per mask specified by $n^{(S,i)}$ -dimensional vectors of radii $\mathbf{r}^{(S,i)}$ and angles $\boldsymbol{\theta}^{(S,i)}$:

$$\boldsymbol{\theta} \mapsto s \left(\boldsymbol{\theta}, \mathbf{r}^{(S,i)}, \boldsymbol{\theta}^{(S,i)} \right) \quad (10)$$

A mask's outline is then fully specified by those pixels that intersect with this spline.

To initially place the knots, we search along rays originating at the centroid of each mask $b_{xy}^{(S,\text{full})}$ and find where these rays intersect with the mask edge. We determine the edge by applying a minimum filter with 2-connectivity to the mask and set to true all pixels in the filtered image that are different from the original one. We then smooth the resulting edge image using a Gaussian filter with $\sigma = 0.5$. For a given polar angle θ , we find the radius of the corresponding knot by averaging the edge pixels that intersect with the ray, weighted by their values. We use the major axis of the ellipse with same normalized second central moment (`regionprops` function from Scikit-image [71]) as the mask to determine both the number of rays, and so knots, and their orientations. The length $\ell^{(S,i)}$ of the major axis gives the number of rays: four for $0 < \ell^{(S,i)} < 5$; six for $5 \leq \ell^{(S,i)} < 20$; and eight for $\ell^{(S,i)} \geq 20$. For this initial placement, we choose equiangular $\boldsymbol{\theta}^{(S,i)}$, with the first knot on the ellipse's major axis.

Discarding poor or duplicated outlines The quality of the outline masks $\hat{o}_{xy}^{(S,i)}$ derived from these initial radial splines are then assessed against the edge probabilities generated by the CNN (Appendix 3 Fig. 4e) and masks of poor quality discarded. We calculate the edge score for a given outline as

$$\eta^{(S,\text{edge},i)} = \frac{1}{N_x N_y} \sum_{xy} p_{xy}^{(S,\text{edge})} \text{Dilate}^2 \left(\hat{o}_{xy}^{(S,i)} \right). \quad (11)$$

We discard those outlines for which the edge score is less than a threshold, where the thresholds $T_{\text{edge}} \in [0, 1]$ are optimised for each size category based on the range of edge scores observed.

With a smoothed and filtered set of outlines, we proceed by detecting and eliminating any outlines duplicated between size categories. We start by filling the outlines to form a set of full masks $\hat{m}_{xy}^{(S,i)}$. We then compare these masks between neighbouring size categories S_j and S_k . We consider the pair of masks i_1 and i_2 to be duplicated if one of the masks is almost wholly contained within the other:

$$\frac{\sum_{xy} \hat{m}_{xy}^{(S_j,i_1)} \cap \hat{m}_{xy}^{(S_k,i_2)}}{\min \left(\sum_{xy} \hat{m}_{xy}^{(S_j,i_1)}, \sum_{xy} \hat{m}_{xy}^{(S_k,i_2)} \right)} > T_{\text{containment}} \quad (12)$$

for some threshold $T_{\text{containment}} \in [0, 1]$, optimised on validation data. For pairs that exceed this threshold, we keep only the mask with the highest edge score, Eq. 11.

For each size category, the first part of the post-processing pipeline finishes with the set of outlines that pass these size, edge probability, and containment thresholds. Appendix 3 Table 1 gives values for the optimised post-processing parameters.

Refining edges The outlines $\hat{o}_{xy}^{(S,i)}$, defined by the radial splines, do not directly make use of the CNN's edge targets for their shape and deviate from $p_{xy}^{(S,\text{edge})}$, particularly for those in the large size category (Appendix 3 Fig. 4e).

We therefore optimise the radial splines to better match the predicted edge. This optimisation is challenging because $p_{xy}^{(S,\text{edge})}$ provide only a semantic representation of the edge – the association of a given pixel (x, y) with a particular instance i is unknown. Our approach is to use the outlines to generate priors on whether predicted edge

809 pixels associate with a given instance. We then apply standard techniques to optimise the fit of the radii and angles
 810 of the knots for each outline's spline to its instance's likely edge pixels.

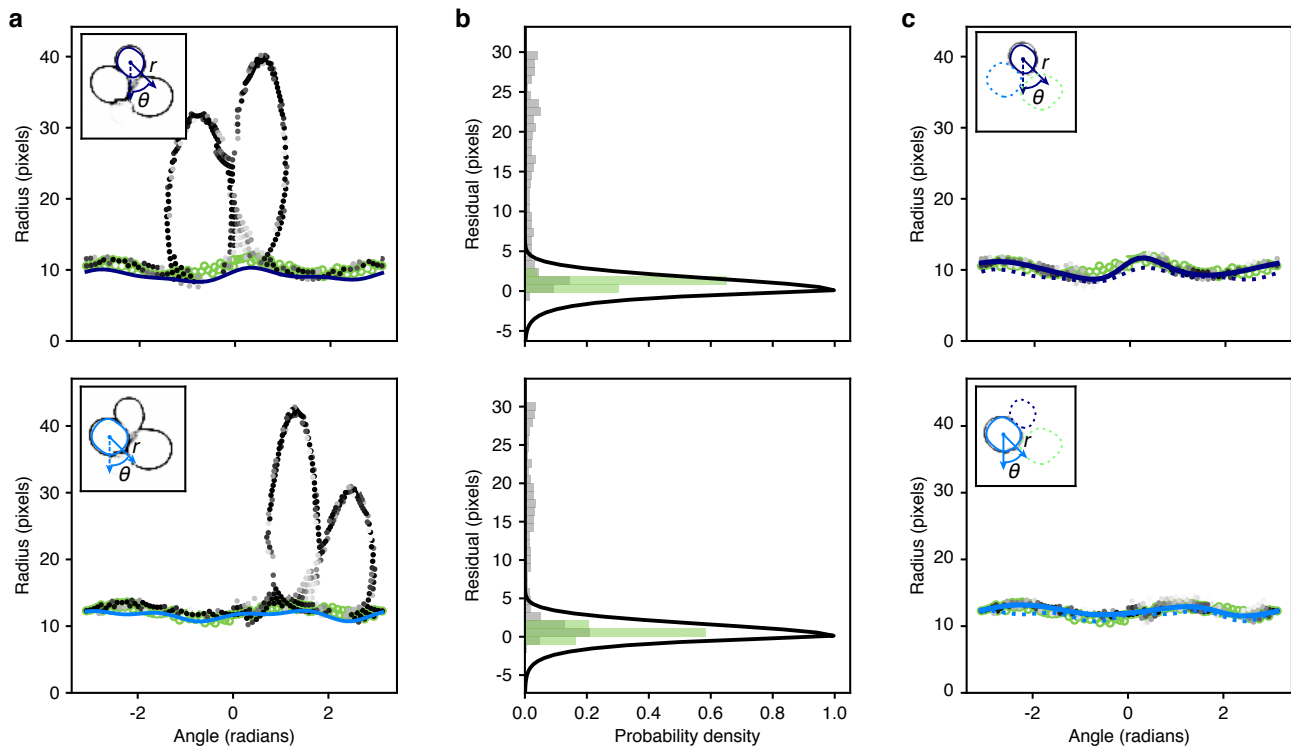


Figure 5: **Optimisation of the radial spline to fit the predicted edge.** **a** Predicted edge pixels by the CNN in polar coordinates centred on two different instances (top and bottom panels). Darker shading indicates a higher probability of being an edge. Open green circles are the manually curated ground truth. Solid lines are the initial radial splines estimated from the CNN's predictions of interiors. Insets show the predicted edge in cartesian coordinates with the instance providing the origin marked by its initial outline and the indicated polar coordinates. **b** Binned residuals of the predicted edge pixels with the initial radial spline for the examples shown in **a**. Only edge pixels with probability greater than 0.2 are considered. Binned residuals for the ground truth are in green. Black lines show the function used to re-weight pixel probabilities for each instance. **c** As for **a**, but after the edge pixels have been re-weighted for each instance. Solid lines indicate the optimised radial spline. The inset shows with a solid line the outline favoured by the instance-association probability and the dashed lines those outlines disfavoured.

811 To associate pixels with instances, we first calculate the radial distance of each pixel from the initial radial spline
 812 function s proposed for an instance, 10. To increase speed, we consider only pixels where $p_{xy}^{(S, \text{edge})} > 0.2$. Expressing
 813 the edge pixels in polar coordinates as (ρ, ϕ) with the origin at the instance's centroid, this distance is

$$R_{xy}^{(S,i)} = \rho - s(\phi, \mathbf{r}^{(S,i)}, \boldsymbol{\theta}^{(S,i)}) \quad (13)$$

814 which we will refer to as a pixel's residual. We give two examples of edge pixels (Appendix 3 Fig. 5a) and of the
 815 corresponding residuals (Appendix 3 Fig. 5b), which highlight the need to associate pixels with a given instance before
 816 attempting to optimise the spline.

817 We use the residuals, Eq. 13, to assign prior weights to pixels:

$$W(R) = \begin{cases} e^{-R^2/\sigma_G} & \text{if } R \geq 0 \\ e^{R/\sigma_E} & \text{if } R < 0 \end{cases} \quad (14)$$

818] where $\sigma_G = 5$ and $\sigma_E = 1$. The function W is a Gaussian function of the residual for pixels exterior to the proposed
 819 outline and an exponential function for pixels interior (Appendix 3 Fig. 5b). This asymmetry is intended to increase
 820 tolerance for interior edge pixels, which may belong to neighbouring instances that overlap with the cell of interest.
 821 In such cases, instance association should be improved, particularly where the edges of each of the cells intersect.

822 With these prior weights, we find the probability that each edge pixel is associated with a particular instance and
823 not with the others, via:

$$p_{xy}^{(S,\text{edge},i)} = p_{xy}^{(S,\text{edge})} \times W(R_{xy}^{(S,i)}) \times \left(1 - \frac{1}{n-1} \sum_{j \neq i} W(R_{xy}^{(S,j)}) \right) \quad (15)$$

824 where n is the number of detected instances in this and adjacent size categories, with j running over all these instances.
825 We filter the result, keeping only those edge pixels with $p^{(S,\text{edge},i)} > 0.1$. Examples are shown in Appendix 3 Fig. 5c.

826 We optimise the knot radii $r^{(S,i)}$ and angles $\theta^{(S,i)}$ for each radial spline by minimising the squared radial residual
827 between the spline and the edge pixels, Eq. 13. Residuals are weighted by $p_{xy}^{(S,\text{edge},i)}$, Eq. 15, and initial values are
828 taken from each $\hat{o}_{xy}^{(S,i)}$. Radii are constrained to a 30% change from their initial values; angles are constrained to
829 a change of $\pm 25\%$ of the initial angular separation between knots: $\theta_{i+1} - \theta_i$. The resulting optimised radial spline
830 provides the outlines output by the BABY algorithm.

831 Appendix 4

832 The BABY algorithm: tracking cells and identifying lineages

833 To track cells and lineages, we have two tasks: first, link cell outlines from one time point to the next (Appendix 4
834 Fig. 4a), and second, identify mother-daughter relationships (Appendix 4 Fig. 4b).

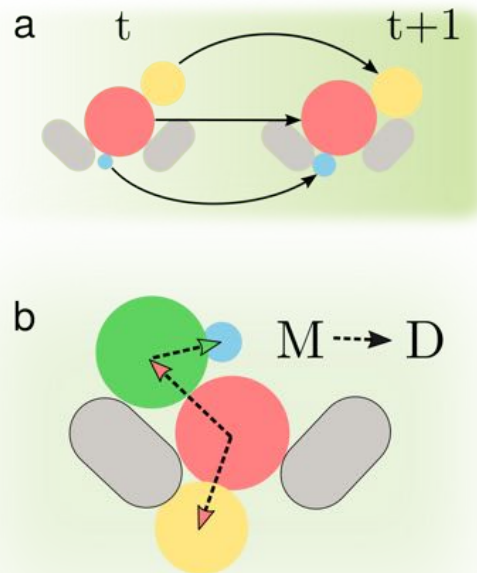


Figure 1: **Determining accurate lineages requires solving two independent tasks.** **a.** We must identify cells across time points regardless of how they grow and move within the images. **b.** We have to find the mother-daughter relationship between cells at every time point.

835 Tracking cells from image to image

836 In our setup, daughter cells may be washed out of the microfluidic device and so disappear from one time point to the
837 next. These absences undermine other approaches to tracking, such as the Hungarian algorithm [29].

838 To track cells, we use the changes in their masks over time to indicate identity. From each mask, we extract an
839 array of attributes, such as the mask’s area, major axis length, etc., and to compare a mask at one time point to a
840 mask at another time point, we subtract the two corresponding arrays of features. This array of differences is the
841 array of features for the classification algorithm.

842 Our training data comprises a series of manually labelled time-lapse images from four experiments. For two
843 consecutive time points, we calculated the difference in feature arrays between all pairs of cells and grouped these
844 difference arrays into two classes: one for identical cells – cells with the same label – and one for all other cells.

845 **Using multiple time points in the past** To generate additional training data, we use multiple time points
846 backwards in time. For example, for time t , we generate not only feature vectors by comparing with cells at $t - 1$,
847 but also with cells at $t - 2$ and $t - 3$. We found this additional data increased generalisability, allowing accuracy to
848 be maintained across a wider range of imaging intervals and growth rates. For the purpose of training, we treat the
849 additional data as consecutive time points – the algorithm does not know whether the features come from one or more
850 than one time point in the past.

851 As part of testing that all features contribute to the learning, we divided the features into two overlapping sets.
852 One set had no features that explicitly depend on distance, comprising area, lengths of the minor and major axes,
853 convex area, and area of the bounding box; the second set did include distance-dependent features, comprising area,
854 lengths of the minor and major axes, and convex area again, but additionally including the mask’s centroid, and the
855 distance obtained from the x - and y -axis locations.

856 We compared three standard algorithms for classification [74]: the Support Vector Classifier (SVC), Random
857 Forest, and Gradient Boosting, specifically Xtreme Gradient Boosting [75]. We used `scikit-learn` [70] to optimise

858 over a grid of learning hyperparameters.

859 For the SVC, we considered a regularisation parameter C of 0.1, 10, or 100; a Γ kernel coefficient of 1, 10^{-2} , or
 860 10^{-4} ; no shrinking heuristic to speed up training; and either a radial basis function or sigmoid kernel.

861 For the Random Forest, we explored a range between 10 and 80 estimators and a depth between 2 and 10 levels.

862 For Gradient Boosting, we used a maximal depth of either 2, 4, or 8 levels; a minimal child weight of 1, 5, or 10;
 863 gamma, the minimal reduction in loss to partition a leaf node, of 0.5, 1, 1.5, 2, or 5; and a sub-sampling ratio of 0.6,
 864 0.8, or 1.

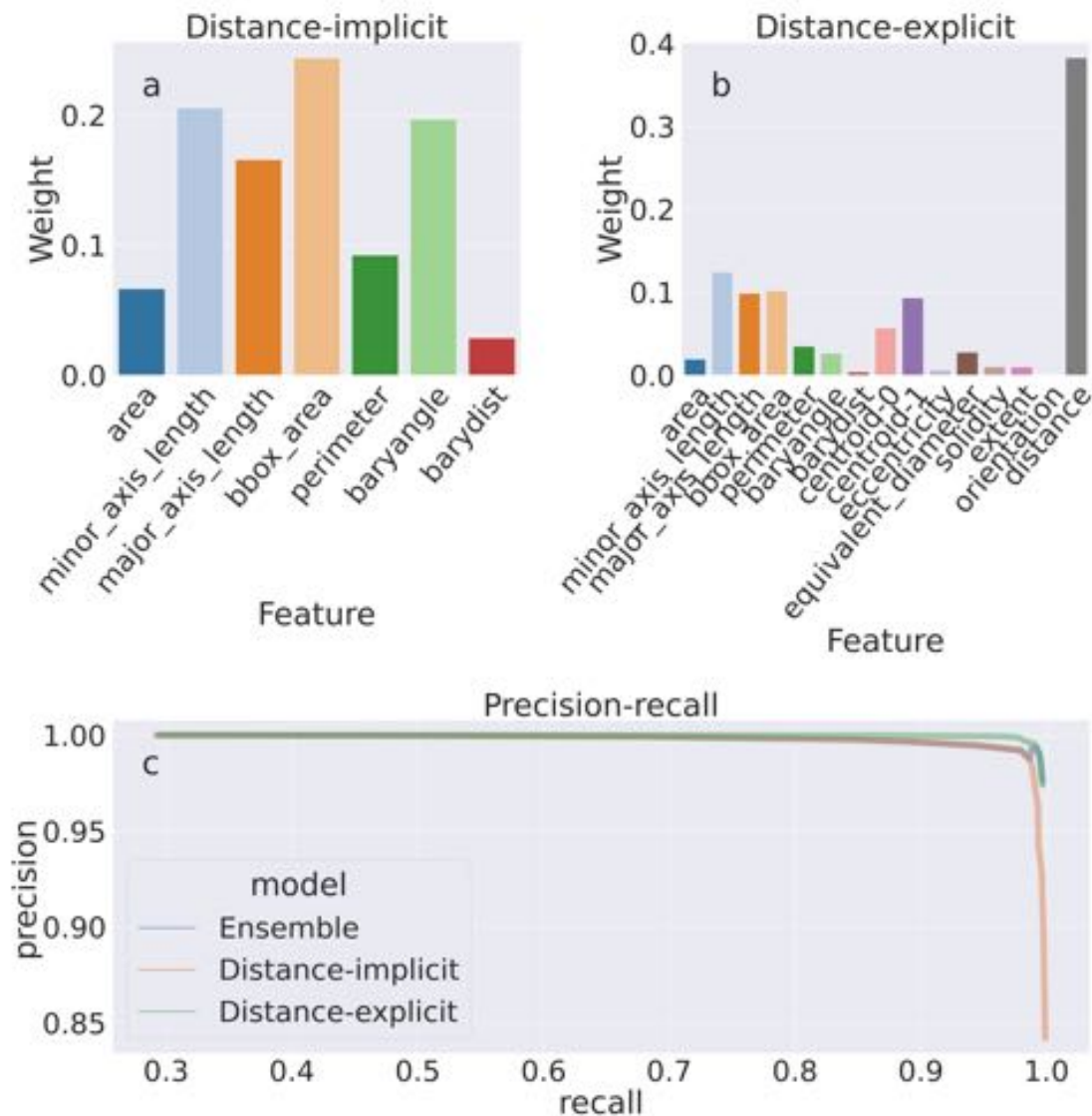


Figure 2: **The importance of the features used by the Random Forest classifier for tracking cells from time point to time point.** Depending on the features with which we train the classifier, the weights, or feature importances, are more evenly spread. **a** If we train the classifier using features that explicitly include distance-dependence, distance drives the decisions, and the remaining features are only used for marginal cases. **b** If we train the classifier using distance-implicit features, however, the weights are more uniform. **c** The precision-recall curve shows high accuracy for both sets of features.

865 Within the training data, the number of time points for each experiment is different and the feature arrays will
 866 favour linking cells within an experiment. To prevent biases toward long experiments, we define the accuracy as
 867 the fraction of true positives – cells correctly linked between images, and compare the precision and recall of this
 868 time-averaged accuracy.

869 After training, we evaluated which features are important for classifying, using the Random Forest. The distribution
870 of the features' importance depends on whether distance is included (Appendix 4 Fig. 2): excluding distance distributes
871 the weight of decision more evenly.

872 **An ensemble model** The precision-recall curve indicates that using the distance-explicit features is best, although
873 both sets of features have high accuracy (Appendix 4 Fig. 2c). Although performing better on our test data, we
874 expect that using the distance-explicit features may perform worse if the cells pivot or are displaced in some other
875 way. Therefore, we use the non-explicit features as our main model, but also use the distance-explicit features to
876 resolve any ambiguous predictions. The ensemble model performs similarly to the distance-implicit classifier, but for
877 more stringent thresholds becomes behaves like the distance-explicit model.

878 Making predictions

879 To predict with the classifier, we use data from the current time point and the two most recent previous time points.
880 We generate feature arrays between t and independently $t - 1$ and $t - 2$ and feed both arrays to the classifier. If the
881 probability returned is greater than 0.9, we accept the result; if the probability lies between 0.1 and 0.9, we use instead
882 the probability returned by the backup classifier, which uses the distance-explicit features.

883 Using multiple time points to track cells has two advantages: first, it reduces noise when there is an artefact, either
884 in image acquisition, such as a loss of focus, or in segmentation; second, it ensures that cells are more consistently
885 identified if their position or shape transiently changes. Including data further back in time is neither computationally
886 efficient nor more accurate – greater than three time points is over 15 minutes in our experiments, about a sixth of a
887 cell cycle.

888 We apply the linear sum assignment algorithm, via SciPy, on the probability matrix of predictions to assign
889 labels (Appendix 4 Algorithm 1), which guarantees at most one outline assigned to each cell by choosing the set of
890 probabilities whose total sum is highest. To match a cell with its previous self, we pick the cell in the recent past that
891 generates the highest probability when paired with the cell of interest, providing this probability is greater than 0.5.
892 A cell is labelled as new if the probabilities returned from pairing with all cells in the recent past is below 0.5.

Algorithm 1: Cell labelling

```
Data: probMat, threshold, oldLabels, maxLabel  
Result: New cell labels (newLabels)  
let newLabels be zeros(ncols(probMat));  
for old, new  $\in$  linearSumAssignment( $-$ ProbMat) do  
    | if probMat[old, new] > threshold then  
    |   | newLabels[new]  $\leftarrow$  oldLabels[old];  
    | end  
893 end  
for label  $\in$  newLabels do  
    | if label! = 0 then  
    |   | label = maxLabel+ 1;  
    |   | maxLabel = label;  
    | end  
end  
return newLabels
```

894 Assigning lineages

895 We wish to identify which cells are buds of mothers and which mothers have buds. This problem is analogous to
896 tracking, but, rather than identifying pairs of cells that are the same cell at different time points, we must identify
897 pairs of cells that are a mother-bud pair at one time point. We therefore seek to determine the probability that a pair
898 of cells is a mother-bud pair (Appendix 4 Fig. 3). Unlike for tracking, however, we anticipated that the cell outlines
899 alone would be at best a weak indicator of this probability.

900 **Defining mother-bud features** We observed that cytokinesis is sometimes visible in bright-field images as a
901 darkening of the bud neck and designed features to exploit this characteristic of mother-bud pairs.

902 Multiple of these features rely on the CNN's prediction of bud necks. For generalisability and to avoid ambiguity,
903 we chose to define the corresponding training target using manually annotated outlines and lineage relationships,
904 rather than relying on a fluorescent bud-neck marker. Specifically, we define a binary semantic 'bud-neck' training
905 target that is true only at pixels where a mother mask, dilated twice by morphological dilation, intersects with its
906 assigned bud (Appendix 3 Fig. 1). Assigning a time of cytokinesis by eye is challenging, and so we included two
907 constraints to identify a bud. First, the bud must be current – as soon as another bud is found associated to the
908 mother, the previous bud is excluded. Second, buds are excluded if their area is larger than $10 \mu\text{m}^2$ (300 pixels for

909 our standard training target with a pixel size of $0.182 \mu\text{m}$ and corresponding to a sphere of $\sim 24 \mu\text{m}^3$).

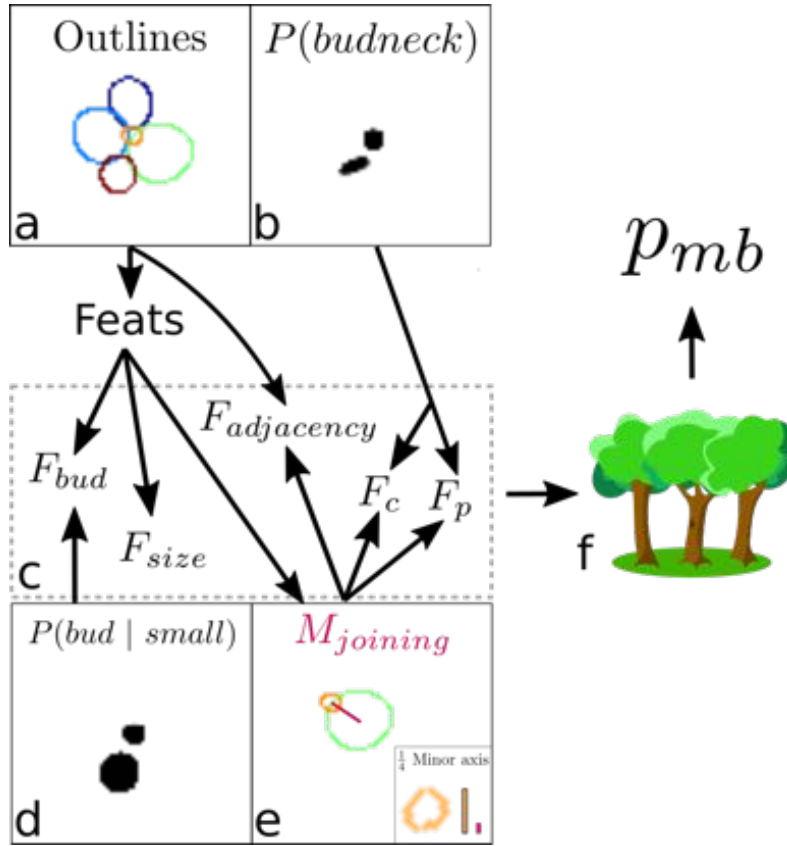


Figure 3: **Overview of the algorithm for assigning lineages.** **a** Cell outlines. Colours are only illustrative. **b** The CNN's predicted probabilities per pixel of being a bud neck are shown for small cells. White is zero, and different colour intensities are probabilities. **c** Composite features used by the classifier to solve the task. **d** The probability of small cells being a bud. **e** An intermediate element of assigning lineages is defining $M_{joining}$ – the red rectangular box. **f** Feeding the features into a trained random forest model returns the probability of a pair of cells being a mother and bud.

910 We used five image features to characterise a mother-bud relationship. For an ordered pairing of all cells in an
 911 image, we consider a putative mother-bud pair and define a mask, $M_{joining}$, as the joining rectangle between the
 912 centres of the mother and bud with a width equal to one quarter the length of the bud's minor axis. Given $M_{joining}$,
 913 the features are:

- 914 (i) F_{size} , which is the ratio of the mother's to bud's area. Mothers generally have a greater size than their bud so
 915 that $F_{size} > 1$.
- 916 (ii) $F_{adjacency}$, which is the fraction of $M_{joining}$ intersecting with the union of the mother's and bud's masks. Mothers
 917 should be proximal to their buds so that $F_{adjacency}$ is close to one – only a small fraction of $M_{joining}$ should lie
 918 outside of the mother and bud outlines.
- 919 (iii) F_{bud} , which is the mean over the union of the CNN's output for a small, interior targets and all pixels contained
 920 in the bud. F_{bud} approximates the probability that a cell is a bud and should be close to one for mother-bud
 921 pairs.
- 922 (iv) F_p , which is the mean over the union of the CNN's output for bud-necks, only including those pixels whose
 923 probability is greater than 0.2, with the pixels contained in $M_{joining}$. F_p approximates the probability that the
 924 mother and bud are joined by a bud neck.
- 925 (v) F_c , which is the number of the CNN's bud-neck target pixels that have a probability greater than 0.2 and are in
 926 $M_{joining}$ normalised by the square root of the bud's area, or effectively the bud's perimeter. We interpret F_c as
 927 a confidence score on F_p because a single spurious pixel with high bud-neck probability could produce high F_p .

928 **Training a classifier** With these features, we train a random forest classifier to predict the probability that a pair
 929 of cells is a mother and bud. We train on all pairs of cells in the validation data to ensure that the classifier does not
 930 rely on the CNN's high performance on the training data. We optimised the hyperparameters, including the number
 931 of estimators and tree depth, using a grid search with five-fold cross-validation. We optimise for precision because
 932 true mother-bud pairs are in the minority and because our strategy for assigning lineages aggregates over multiple
 933 time points (as detailed below).

934 For our standard CNN trained to a pixel size of $0.182 \mu\text{m}^2$, the random forest classifier had a precision of 0.80 and
 935 recall of 0.47 on the test data. This data has 211 true mother-bud pairs out of 1654 total pairs. The classifier assigned
 936 feature importances of 0.42 for F_{size} , 0.10 for $F_{\text{adjacency}}$, 0.26 for F_{bud} , 0.08 for F_{p} , and 0.13 for F_{c} .

937 **Assigning each cell a unique mother** To establish lineage relationships, we need to assign at most one tracked
 938 mother cell to each tracked cell object. We use the classifier to assign a mother-bud probability $p_{mb}^{(t)}$ for each time
 939 point at which a pair of tracked objects are found together. We then estimate the probability that a tracked object \hat{c}
 940 has ever been a mother using

$$p_m^{(t)}(\hat{c}) = \max \left[p_m^{(t-1)}(\hat{c}), \sum_{c_b \in C_t} p_{mb}^{(t)}(\hat{c}, c_b) \right], \quad (16)$$

941 as well as the probability that it has ever been a bud with

$$p_b^{(t)}(\hat{c}) = \max \left[p_b^{(t-1)}(\hat{c}), \max_{c_m \in C_t} \left(p_{mb}^{(t)}(c_m, \hat{c}) \right) \right]. \quad (17)$$

942 Finally, for a putative mother-bud pair we calculate a cumulative score that is reduced if the candidate bud has
 943 previously shown a high probability of being a mother:

$$S_{mb}^{(t)}(\hat{c}_m, \hat{c}_b) = S_{mb}^{(t-1)}(\hat{c}_m, \hat{c}_b) + p_{mb}^{(t)}(\hat{c}_m, \hat{c}_b) \left(1 - p_m^{(t)}(\hat{c}_b) \right). \quad (18)$$

944 At each time point, we then propose lineages by assigning each putative cell object \hat{c}_b with a bud probability
 945 $p_b^{(t)}(\hat{c}_b) > 0.5$ and a mother $\hat{c}_m = \text{argmax}_{\hat{c}} \left(S_{mb}^{(t)}(\hat{c}, \hat{c}_b) \right)$. We treat the mother-bud assignments proposed at the
 946 final time point as definitive, because they have integrated information over the entire time series, and avoid spurious
 947 assignments by requiring all buds to be present for at least three time points.

948 Post-processing

949 Though rare, we do have to mitigate occasional detection, tracking and assignment errors. For example, debris can
 950 occasionally be mistakenly identified as a cell and tracked.

951 We discard tracks that have both small volumes and show limited growth over the experiment. Specifically, we
 952 discard a given cell track i with duration T_i , minimal volume $V_i^{(\text{min})}$ at time $T_i^{(\text{min})}$, and maximal volume $V_i^{(\text{max})}$ at
 953 time $T_i^{(\text{max})}$, if both $V_i^{(\text{max})} < 7 \mu\text{m}^3$ and the estimated average growth rate $G_i < 10 \mu\text{m}^3/\text{hour}$, where

$$G_i = \text{sign} \left(T_i^{(\text{max})} - T_i^{(\text{min})} \right) \times \left(V_i^{(\text{max})} - V_i^{(\text{min})} \right) / T_i. \quad (19)$$

954 We also ignore any tracks that have no identified lineage relationships: they are neither assigned as a mother nor as
 955 a bud/daughter at any time point.

956 Our tracking algorithm correctly identifies many instances where a mother and bud pivot with the flow of the
 957 medium, but exceptions do arise. For a given mother, we therefore join contiguous daughter tracks – pairs of daughter
 958 tracks where one ends with the other starting on the next time point – if the extrapolated volume of the old track
 959 falls within a threshold difference of the volume of the new track. Specifically, for the pair of contiguous tracks i and
 960 j , with track i ending at time point t and track j beginning at time point $t + 1$, we calculate

$$V_{ij}^{(\text{diff})} = \min \left(\left| V_i^{(t)} + G_i \Delta T - V_j^{(t+1)} \right|, \left| V_j^{(t+1)} - G_j \Delta T - V_i^{(t)} \right| \right), \quad (20)$$

961 where $V_i^{(t)}$ is the volume of track i at time point t and ΔT is the time step between time points t and $t + 1$. We join
 962 these tracks if $V_{ij}^{(\text{diff})} < 7 \mu\text{m}^3$.

963 Finally, we discard any daughter tracks with fewer than five time points.

964 **Appendix 5**

965 **Estimating birth times from fluorescent markers**

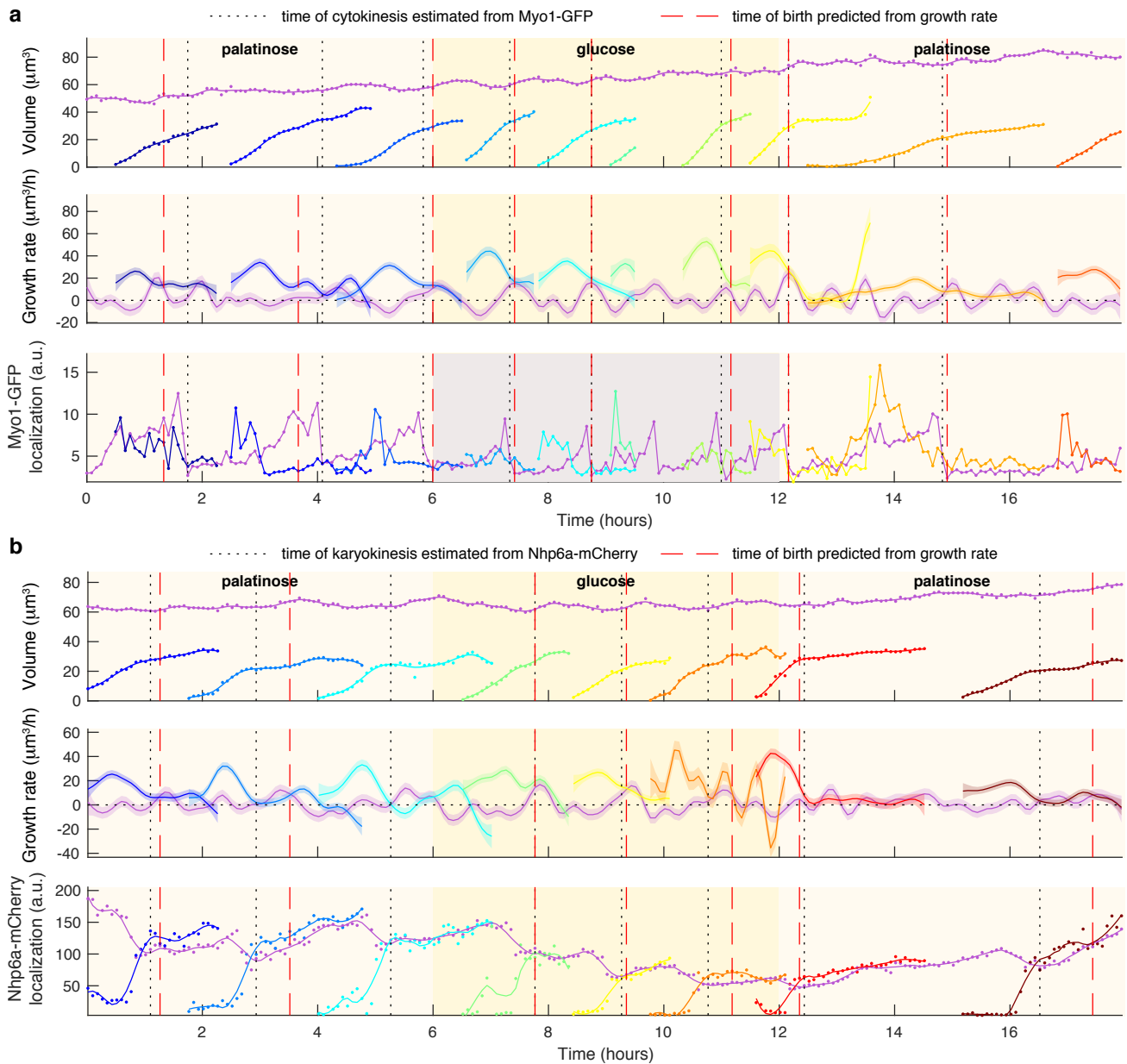


Figure 1: **Markers for karyokinesis and cytokinesis reveal coincidence with a crossing point in mother and daughter growth rates.** **a** Time series for a mother (purple) and its buds/daughters for a switch from 2% palatinose to 2% glucose and back. Volumes and growth rates are as estimated by BABY. Localization of Myo1-GFP to the bud neck is used to identify times of cytokinesis (vertical black dotted lines) as described in the Supplementary Text. For comparison, birth times predicted by our growth rate heuristic are also shown (vertical red dashed lines). **b** As for (a), but using localization of Nhp6A-mCherry to the nucleus to identify times of karyokinesis (vertical black dotted lines). Both the raw (points) and smoothed (lines; Savitzky-Golay filter with third degree polynomial and smoothing window of 15 time points) localization of Nhp6A-mCherry is shown.

966 We used either Myo1-GFP or Nhp6A-mCherry to estimate the time at which a bud becomes an independent daughter.
 967 Myo1 – a type II myosin – localises to the bud neck and shows a drop in intensity upon cytokinesis (Appendix 5
 968 Fig. 1a); Nhp6A, a histone-associated protein localised to the nucleus, shows a drop in intensity upon karyokinesis
 969 (Appendix 5 Fig. 1b). Although karyokinesis and cytokinesis are distinct events, their timing is similar [24], and

970 we assume both mark the birth of an independent daughter. For Fig. 4a & b, Appendix 5 Fig. 1a and 2, we used
971 Myo1-GFP; for Fig. 4d & e, Fig. 5, Appendix 5 Fig. 1b, Figure 4—figure supplement 1, Figure 5—figure supplement 1,
972 and Figure 6—figure supplement 1, we used Nhp6a-mCherry.

973 Cytokinesis is accompanied by a drop in Myo1-GFP intensity at the bud neck in the mother cell (Fig. 4a), and we
974 use this phenomenon to determine when cytokinesis occurs, assuming that it is fast compared to the time interval of
975 our imaging. We use the time series of fluorescence localisation within the mother cell over the period where the mother
976 has a bud and estimate its time derivative using finite differences (NumPy’s `diff` function). To obtain candidate time
977 points for cytokinesis, we find the dips in this derivative with a minimum prominence, via SciPy’s `findpeaks` function.
978 The actual time of cytokinesis is taken to be the candidate point with the minimal derivative, corresponding to the
979 strongest down-shift in fluorescence.

980 For Nhp6a-mCherry, the timing of karyokinesis is less obvious, although when the nucleus partitions between the
981 mother and its bud there is both a fall in fluorescence localisation in the mother and a rise in the bud (Appendix 5 Fig.
982 1b). As karyokinesis completes, both localisation signals level to a similar, often slowly increasing value. We therefore
983 identify the completion of karyokinesis as the time point when fluorescence localisation shifts from a fast increase in
984 the bud and decrease in the mother to slow increase in both. We thus search for a minimum in the time derivative of
985 the localisation signal for the mother that occurs near a maximum in the time derivative of the daughter.

986 Specifically, we consider a mother and bud/daughter each with a time series of fluorescence localisation. We apply
987 a Savitzky-Golay filter with a third order polynomial and a smoothing window W to obtain the smoothed localisations,
988 denoted m_t and b_t , and their time derivatives. We find candidate time points of karyokinesis by identifying peaks in
989 the series $(\frac{3}{2}\partial_t b_t - \frac{1}{2}\partial_t m_t)$ that have a minimal peak prominence P , using Matlab’s `findpeaks` function. For each
990 peak, we define candidate intervals of karyokinesis as $\{|t - t_i| \leq \text{round}(w_i/2)\}$ where t_i is the time that the peak
991 occurs and w_i is its width. This set of candidates is filtered by thresholding on peak height, the maximal value of the
992 bud’s smoothed localisation, the range of the bud’s localisation, the minimal value of the mother’s localisation minus
993 the maximal value of the bud’s, and the minimal value of the time derivative of the mother’s localisation, all calculated
994 over the candidate time interval. The smoothing window W , minimal peak prominence P , and each threshold are
995 manually curated for every experiment. The first occurring peak that passes all filters defines the time interval of
996 karyokinesis, and karyokinesis’s time of completion is taken as this interval’s final time point.

997 Predicting cytokinesis from growth rate

998 All together, we are able to determine key events of the cell cycle. First, we define a cell cycle for each mother as the
999 duration between two birth points, obtained from the lineage assignment. These points approximately correspond to
1000 shortly after the START checkpoint [76]. Second, assuming that the births are accurately predicted, we identify a
1001 single point of cytokinesis within the corresponding cell cycle.

1002 We observe three phases of growth during a cell cycle (Fig. 4a-b). First, the bud dominates growth during S/G2/M,
1003 with its growth rate peaking midway through that period while simultaneously the mother’s growth rate falls. Second,
1004 the bud’s growth rate decreases as cytokinesis is approached. Near cytokinesis, the mother’s and bud’s growth rate
1005 have similar magnitudes, becoming identical at multiple timepoints. Finally, the mother’s growth rate increases after
1006 cytokinesis, peaking during G1.

1007 Using these observations, we developed an algorithm to identify the point of cytokinesis. Our growth rates are
1008 defined by both a mean and a standard deviation because they are estimated using Gaussian processes, and we use
1009 both quantities to compare growth rates of a mother and its bud. Our algorithm determines the point of cytokinesis as
1010 the first time point where the mother and bud’s growth rates are similar, within a threshold g , and the bud’s growth
1011 rate is sufficiently large, above a threshold g_d . This additional constraint eliminates false negatives when the growth
1012 rate of both mother and daughter undergo a sudden drop, which may occur when the extracellular media is changed.

1013 In practice, we set $g_d = g$, and the threshold g is a parameter whose optimal value varies from cell cycle to cell
1014 cycle, especially with changing extracellular media. A too small g gives poor accuracy for fast-growing cells where the
1015 difference between the mother’s and bud’s growth rates is larger; a too large g gives poor accuracy because cytokinesis
1016 is predicted too early. We therefore choose g from a range g_0, \dots, g_n chosen manually to increase the number of cell
1017 cycles for which cytokinesis is found, without affecting the error (Appendix 5 Algorithm 2). If the algorithm fails to
1018 find a value of g that gives a point of cytokinesis, the cell cycle is marked as unknown and ignored in further analysis.

1019 We evaluated the algorithm by the correlation between the real time of cytokinesis and the predicted time (Appendix
1020 5 Fig. 2a & c). The prediction is both accurate and robust to different media, with correlation coefficients between
1021 0.939 and 0.998 for two different experiments and three different media. The algorithm is also usually accurate at
1022 transitions between two media, when START occurs before and ends after the transition. This result suggests that
1023 the relationship between the mother’s and bud’s growth rates is informative despite transient changes in growth rate
1024 (Fig. 4b).

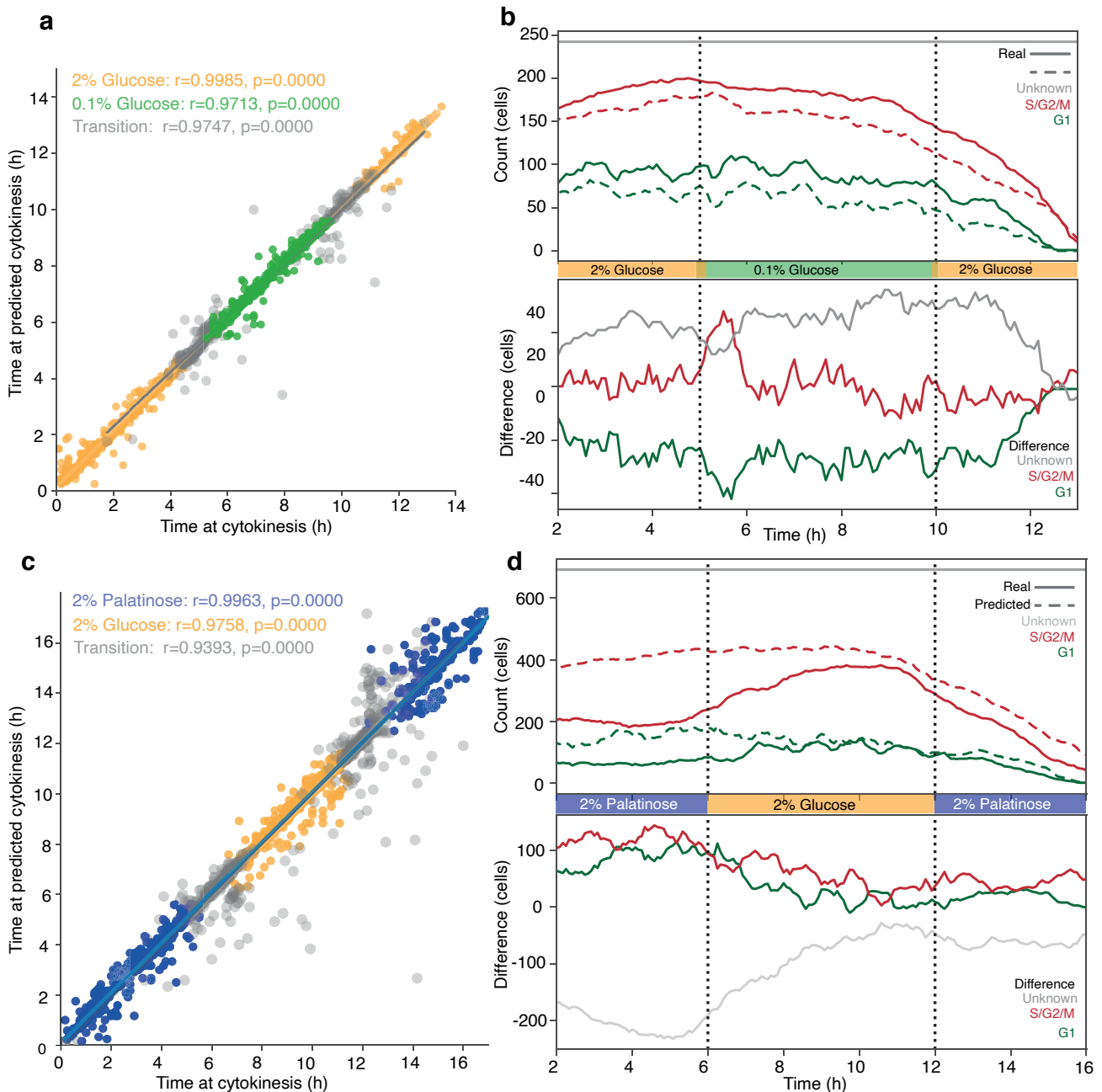


Figure 2: Performance of the prediction of cytokinesis from growth rate. The cells in **a** and **b** were imaged for five hours in SC media with 2% glucose, five hours in SC media with 0.1% glucose, and five hours in SC media with 2% glucose. The cells in **c** and **d** were imaged for six hours in SC media with 2% palatinose, six hours in SC media with 2% glucose, and six hours in SC media with 2% palatinose. **a**, **c** Comparison of the real vs predicted (absolute) time of cytokinesis for each cell cycle. The points are separated by color based on which condition the cell cycle occurs in, with cell cycles that span a switch marked as Transition (grey). **b**, **d** Cumulative counts of mother cells in each cell cycle stage throughout the experiment, with vertical lines representing the switch (top), with real (filled) and predicted (dashed) cytokinesis prediction used to delineate between G1 and S/G2/M. The gray line shows the total number of cells considered in the experiment, defining how many were in an unknown cell cycle state at any given time. The bottom panel shows the absolute difference between real and predicted number of cells in each cell cycle phase, as well as in unknown states, across the experiment.

1025 Further, we verified that any prediction errors do not change the distribution of cell cycle phases over time. We used
 1026 the real and predicted points of cytokinesis to split cell cycles into G1 and S/G2/M and counted how many mother
 1027 cells are in each phase at any given time. We then compared these cumulative counts (Appendix 5 Fig. 2b & d). For a

1028 switch from high to low glucose, although the number of unknowns is larger for the prediction – the cumulative count
1029 of G1 and S/G2/M is smaller, the prediction still captures the dynamics of the population (Appendix 5 Fig. 2b). For
1030 a switch from palatinose into glucose, the accuracy is less (Appendix 5 Fig. 2d). The oscillations in the number of
1031 cells in G1 that occur in glucose are weaker, but we do predict cells accumulating in G1 once palatinose returns.

1032 We note that these comparisons necessarily use only cells that are assigned at least one valid cell cycle with the
1033 Myo1 method. In the palatinose-glucose-palatinose experiment in particular, there are multiple cells that are therefore
1034 ignored despite being assigned at least one valid cell cycle with the growth-rate method. It is possible that discarding
1035 these cells affects the overall result.

Algorithm 2: Cytokinesis from growth rate

Choose a cell cycle, defined between two birth points $t \in [b_i, b_{i+1}]$

Get mother growth rate in $[b_i, b_{i+1}]$: (μ_m, σ_m)

Get growth rate of daughter d_i born at time b_i : (μ_d, σ_d)

Get mother-daughter distance in $[b_i, b_{i+1}]$: $d_{m,d} = \sigma_m^2 + \sigma_d^2 + (\mu_m - \mu_d)^2$

1036 **for** $g \in [g_0, \dots, g_n]$ **do**

for $t \in [b_i, b_{i+1}]$ **do**

if $d_{m,d}(t) < g$ **and** $\mu_d(t) + \sigma_d^2(t) > g$ **then**

 | Set cytokinesis time to t ;

end

end

end

Appendix 6

Correlating nuclear Sfp1 with growth rate

The cross correlation of time series can reveal regulatory relationships [77]. Here we apply cross correlations to investigate if fluctuations in Sfp1's localisation anticipate fluctuations in growth rate. Analysis by the method of Kiviet *et al.* [21] assumes steady-state cells. We nonetheless make use of data with a switch from palatinose to glucose and back (Fig. 4b), but limit to time points from either the four hours preceding the switch to glucose – approximately steady growth in palatinose – or the four hours preceding the switch back to palatinose – approximately steady growth in glucose.

Correlations may occur on scales longer than the duration of a cell cycle, so we limit our analysis to mother cells that are present over the full four hours of steady growth. We use the summed mother and bud growth rates whenever a bud is present because most of the mother's growth is in the bud. We identify when a daughter separates from its mother using Nhp6A-mCherry (Appendix 5). Almost all daughters are washed away before they become mothers, making the lineage trees in our data unbranched and simplifying the analysis.

For each mother i , we have a time series $\ell_1^{(i)}, \dots, \ell_N^{(i)}$ of the degree of localisation of Sfp1-GFP and a time series of instantaneous growth rates $g_1^{(i)}, \dots, g_N^{(i)}$. For our sampling interval of $\Delta t = 5$ min, N is 48. We denote the total number of mother cells by M and calculate the deviation from the population mean for each time series:

$$\delta\ell_t^{(i)} = \ell_t^{(i)} - \frac{1}{M} \sum_j \ell_t^{(j)} \quad \text{and} \quad \delta g_t^{(i)} = g_t^{(i)} - \frac{1}{M} \sum_j g_t^{(j)}. \quad (21)$$

The cross-covariance of Sfp1 localisation and growth rate at a time lag of $r\Delta t$ is then [21]:

$$C_{lg}^{(i)}(r\Delta t) = \begin{cases} \frac{1}{N-r} \sum_{t=1}^{N-r} \delta\ell_t^{(i)} \cdot \delta g_{t+r}^{(i)} & \text{if } r \geq 0 \\ C_{gl}^{(i)}(-r\Delta t) & \text{otherwise.} \end{cases} \quad (22)$$

We find the cross-correlation through normalising by the standard deviations:

$$R_{lg}^{(i)}(r\Delta t) = \frac{C_{lg}^{(i)}(r\Delta t)}{\sqrt{C_{\ell\ell}^{(i)}(0)C_{gg}^{(i)}(0)}}. \quad (23)$$

We determine the auto-correlation for Sfp1 localisation, $R_{\ell\ell}^{(i)}(r\Delta t)$, and for growth rate, $R_{gg}^{(i)}(r\Delta t)$, similarly. In Fig. 5c–d of the main text, we show the mean and 95% confidence interval over all mother cells (all i).

1057 Appendix 7

1058 Real-time feedback control

1059 In these experiments, we wished to trigger a change in media based on the cells' growth rate. As an example, we
1060 switched medium from a richer to a poorer carbon source and used BABY to determine how long cells should be kept
1061 in this medium if we want approximately 50% to have resumed dividing before switching back to the richer medium.

1062 Code to implement the feedback control was run on two computers, one controlling the microscope (Appendix 7
1063 Algorithm 3) and the other both segmenting images (via calls to Python functions) and determining growth rates
1064 (Appendix 7 Algorithm 4). The code was written in Matlab and is available on request.

1065 We defined the fraction of escaped cells as the proportion of included mothers that have had a bud/daughter
1066 exceed a threshold in growth rate of $15\mu\text{l}/\text{hour}$ at any time point after the onset of the lag in growth caused by the
1067 poorer carbon source. We define this lag period to begin at the time point when the median daughter growth rate
1068 first drops below $5\mu\text{l}/\text{hour}$. To be included a mother cell must satisfy two requirements: be present in our data for at
1069 least 95% of the time points from the 20 time points before the first switch to the current time point; and be assigned
1070 a bud/daughter for at least 10% of the time it was observed.

1071 To increase processing speed, we use Savitzky-Golay filtering to estimate growth rates. The resulting first derivative
1072 is not well constrained at the end-points, making instantaneous growth rates vary widely at the most recently acquired
1073 time point. We therefore used growth rates up to and including the time point three steps before the most recent
1074 when determining the fraction of escaped cells.

1075 We used the strain BY4741 Sfp1-GFP Nhp6A-mCherry in both experiments.

Algorithm 3: Feedback control – pseudocode for microscope acquisition software

Set glucose pump to infuse at $4\mu\text{l}/\text{min}$;

Set ethanol pump off;

for 270 *timepoints* **do**

 image acquired time = current time + 5 min Acquire images at 6 stage positions

 Save images in networked directory

while *current time < image acquired time* **do**

if *time since start \geq 5 hours and first switch has not happened* **then**

 Run switch protocol (fast infuse/withdraw to remove back pressure);

 Set glucose pump off;

 Set ethanol pump to infuse at $4\mu\text{l}/\text{min}$;

end

 read *onlinedata.txt*

if *fraction of escaped mothers is recorded and second switch has not happened* **then**

if *fraction of escaped mothers \geq 0.5* **then**

 Run switch protocol (fast infuse/withdraw to remove back pressure);

 Set glucose pump to infuse at $4\mu\text{l}/\text{min}$;

 Set ethanol pump off;

end

end

end

end

Algorithm 4: Feedback control – pseudocode for segmentation software

```
for 270 timepoints do
  while segmentation of all positions is not complete do
    for 6 positions do
      Check networked data directory
      if all images for current position are recorded then
        | Run BABY segmentation on current position
      end
    end
  end
end
for 6 positions do
  Calculate growth rates by Savitzky-Golay filtering
  Append result to array for all positions
end
Write median growth rate to onlinedata.txt
if first switch has happened then
  Calculate lag start time as first time point where median growth rate < 5 $\mu$ l/hour
  if lag start has happened then
    | Calculate fraction of escaped mothers and write to onlinedata.txt
  end
end
end
end
```

1077

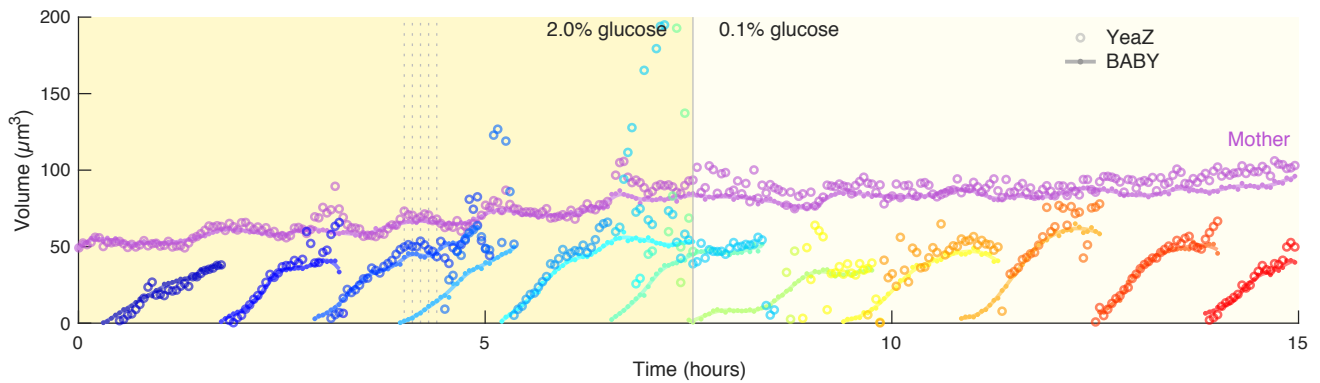


Figure 1—figure supplement 1: Segmentation methods without overlaps can underestimate bud area or fail to detect buds. The YeaZ segmentation algorithm [34], which segments cells from the semantic output of a CNN without allowing for overlaps, was used to segment and track the same images used to generate the time series of Fig. 1e. We show volume time series – estimated from the area of each segmented mask assuming a spherical shape – for all tracks that overlap (mask IoU > 0.5) with those of the segmented output of BABY (light filled circles). As YeaZ only accepts a single Z section as input, we performed a systematic search over both Z section and segmentation parameters and show the closest match to the data obtained with BABY. Vertical dashed lines correspond to the images shown in Fig. 1b.

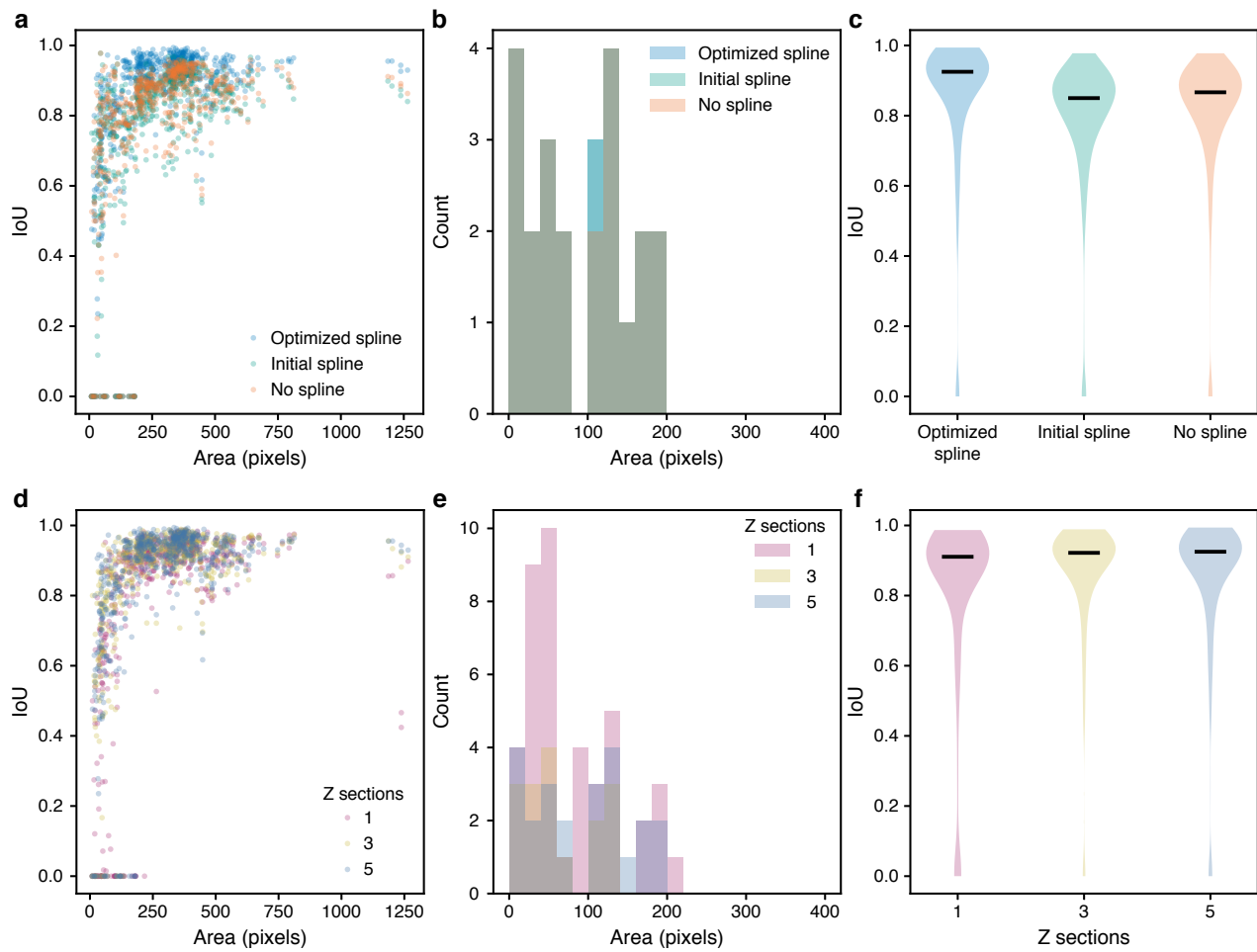


Figure 3—figure supplement 1: Segmentation accuracy is improved by edge optimisation and use of multiple Z sections as input. **a–c** The intersection over union (IoU) with curated masks was evaluated on test data for the predictions of selected stages of the BABY algorithm: masks predicted directly from CNN output with appropriate morphological dilation (no spline), preliminary fits of these masks with an equal-angle radial spline (initial spline), or optimised fits of the radial spline to the edge output of the CNN (optimised spline). **d–f** The IoU with curated masks was evaluated on test data for the predictions of the full BABY algorithm with CNNs trained to use either 1, 3 or 5 bright-field Z sections as input. **a, d** Distribution of IoU with curated outline area. **b, e** Distribution of curated outline area for predicted outlines with IoU of zero. **c, f** Violin plots summarising IoU distributions. Black horizontal line is the median.

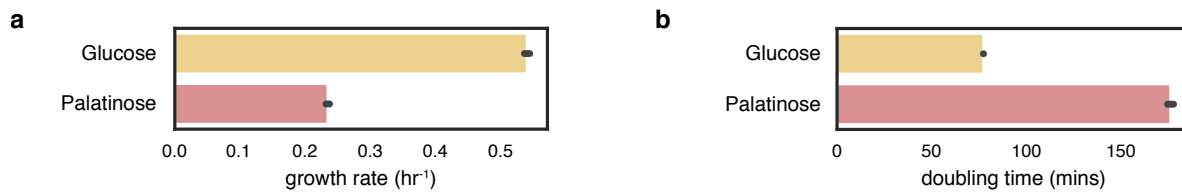


Figure 3—figure supplement 2: Population growth rates measured by optical density. **a** The maximal growth rate for cultures of BY4741 grown in SC with either 2% glucose or 2% palatinose. We plot the greatest local maxima in growth rate as a function of time for 47 hours of growth in a plate reader (Infinity M200, Tecan Group Ltd., Switzerland). Cells were cultured for two days in SC with 2% sodium pyruvate at 30°C, diluted 6× with fresh SC-pyruvate and grown a further 6 hours before being spun down, washed in SC with no carbon source, spun down again, resuspended in SC, added to a 96-well plate with appropriate carbon sources in each well and placed in the plate reader [78]. The time-varying growth rate was found from the optical density using a Matern Gaussian process [41]. Error bars are 95% confidence limits estimated by bootstrapping two biological replicates, each with two technical replicates. **b** Doubling times as calculated from the growth rates in **a**.

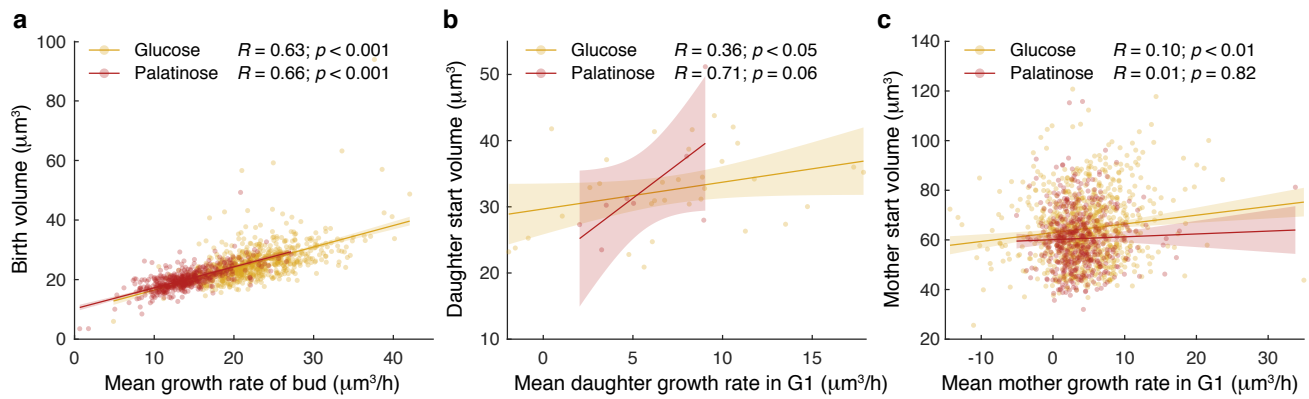


Figure 4—figure supplement 1: Growth rates estimated with BABY show expected correlations with volume. **a** The mean growth rate of a bud from first detection to birth (completion of karyokinesis determined from an Nhp6A-mCherry marker as described in the Supplementary Text) correlates with its volume at birth for buds growing in either 2% glucose or 2% palatinose. **b** The mean growth rate of a daughter across its first G1 phase (from birth to detection of its first bud) correlates with its volume at START (detection of its first bud). In our devices, most daughters are washed away before the appearance of their first bud. **c** In contrast, the mean growth rate of a mother during G1 (from birth of a daughter to appearance of the next bud) shows little to no correlation with its volume at START (appearance of its next bud).

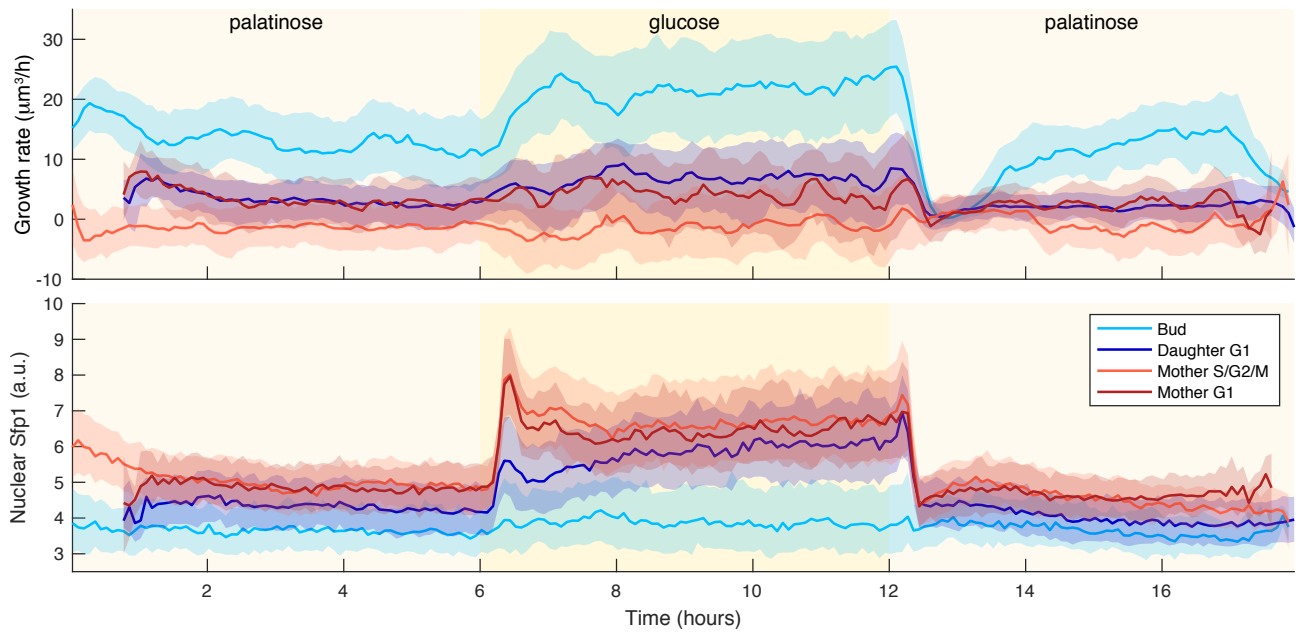


Figure 5—figure supplement 1: Irrespective of cell cycle phase, growth rates transiently drop for a shift to a poorer carbon source. Time series of median growth rate (upper) and median Sfp1-GFP localization (lower) across different cell cycle phases for a switch from 2% palatinose to 2% glucose and back. A Nhp6A-mCherry reporter was used to identify completion of karyokinesis (Supplementary Text) so that the time series for each cell could be partitioned into budding – detection of bud up to karyokinesis, daughter G1 – from karyokinesis up to the detection of its first bud, mother S/G2/M – from detection of a bud to karyokinesis, and mother G1 – from karyokinesis to the appearance of its next bud. The interquartile range is shaded.

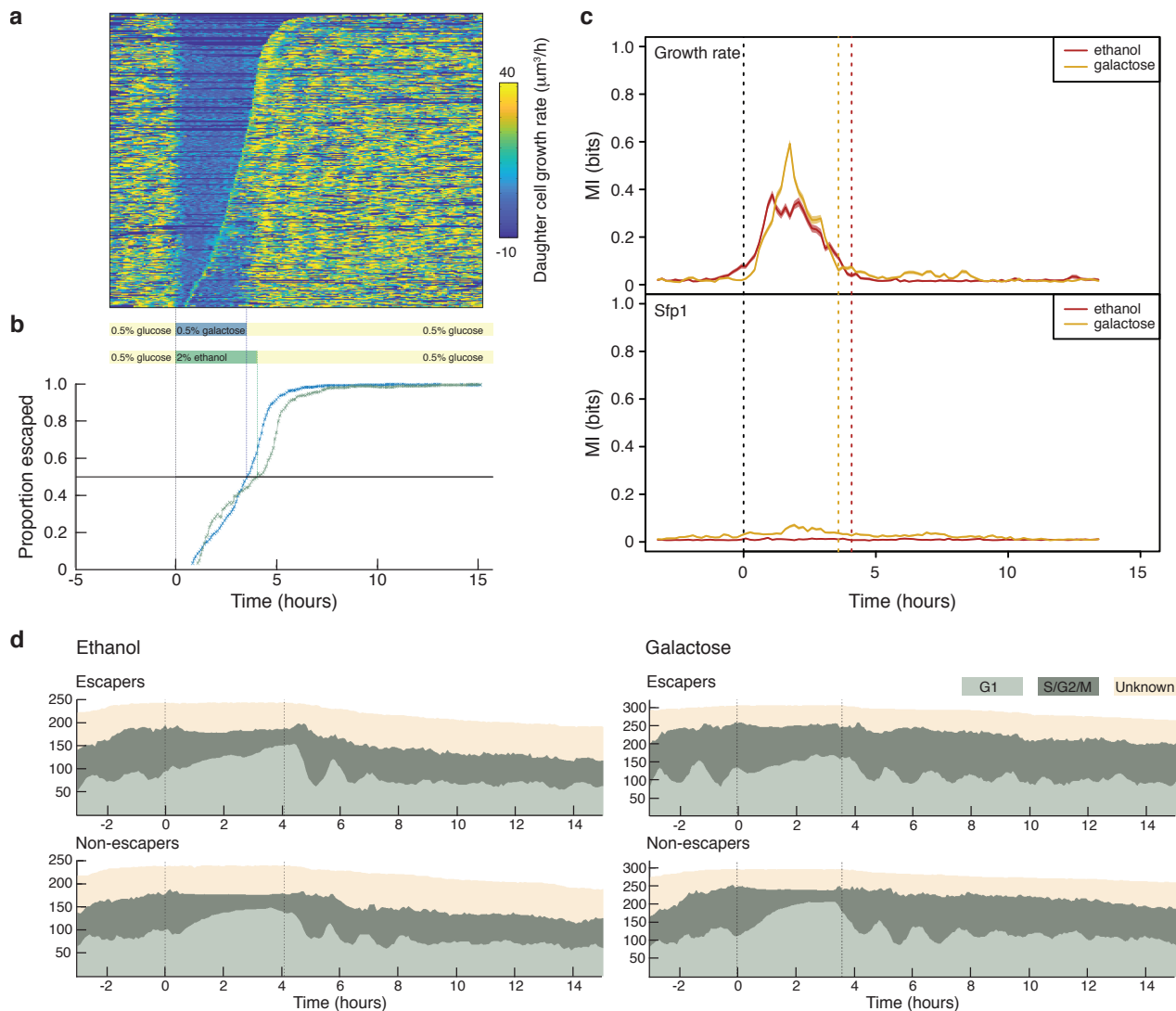


Figure 6—figure supplement 1: Live growth rate control of media flow. **a** Kymograph showing the daughter cell growth rates calculated during image acquisition for an experiment in which the carbon source in of the second media is 0.5% galactose. The experimental method is otherwise identical to the experiment with ethanol shown in Fig. 6 of the main text. Rows representing mother cells are sorted by the time each cell resumes division, with cells resuming later shown toward the top. Times of the media changes are indicated below. **b** Time course of the proportion of cells that have resumed division ("escaped") following the switch to 2% ethanol (green) or 0.5% galactose. The times of the two media switches are indicated for each experiment. **c** Estimated lower bound on the mutual information between escaping/non-escaping cells and either daughter growth rate (top) or nuclear localization of Sfp1 (bottom) in experiments with ethanol or galactose as the second carbon source. Mutual information was estimated by the decoding method [20] for a sliding time-window of 100 minutes, with the mean and 95% confidence interval (shaded area) of 100 bootstraps of the estimation algorithm shown. Decoding was by a gradient boosting model trained on concatenated raw, Fourier-transformed, and rank-ordered versions of each time window. **d** Time courses of estimations of cell cycle stage for escapers and non-escapers for the experiments using ethanol (left) and galactose (right). There is an accumulation of cells in G1 during the period in the second carbon source and synchronisation in cell cycles is evident from the periodic increases and decreases of the proportions of cells in G1 following the reintroduction of glucose at the second media switch.