Biophysical Journal
Computational Tool

Biophysical Society

★Track: Inferred counting and tracking of replicating DNA loci

Robin Köhler,[1] Ismath Sadhir,[1] and Seán M. Murray[1,*]
[1]Max Planck Institute for Terrestrial Microbiology and LOEWE Centre for Synthetic Microbiology (SYNMIKRO), Marburg, Germany

ABSTRACT Fluorescent microscopy is the primary method to study DNA organization within cells. However, the variability and low signal/noise commonly associated with live-cell time-lapse imaging challenges quantitative measurements. In particular, obtaining quantitative or mechanistic insight often depends on the accurate tracking of fluorescent particles. Here, we present ★Track, an inference method that determines the most likely temporal tracking of replicating intracellular particles such DNA loci while accounting for missing, merged, and spurious detections. It allows the accurate prediction of particle copy numbers as well as the timing of replication events. We demonstrate ★Track's abilities and gain new insight into plasmid copy number control and the volume dependence of bacterial chromosome replication initiation. By enabling the accurate tracking of DNA loci, ★Track can help to uncover the mechanistic principles of chromosome organization and dynamics across a range of systems.

SIGNIFICANCE Microscopy is one of the main tools for studying the intracellular organization of cells. In particular, fluorescent proteins allow us to study the dynamics of many cellular processes. However, this requires the accurate tracking of fluorescent foci. Here, we present ★Track, a tool tailored to the tracking of replicating persistent subcellular particles such as DNA loci. ★Track provides accurate predictions of particle copy number and replication timing even in the presence of substantial noise. The knowledge of these quantities are critical for uncovering the mechanisms behind many cell-cycle-dependent processes, such the control of chromosome and plasmid replication initiation.

INTRODUCTION

Fluorescence live-cell microscopy is a powerful tool for the study of subcellular organization and dynamics. However, quantitative analysis requires accurate tracking of the detected fluorescent foci, which may visualize organelles, molecular complexes, or even single molecules. The conceptually simplest approach to tracking is to sequentially join particle localization on successive frames based on minimizing some distance metric, which could potentially account for the nature of the motion, e.g., diffusive or directed. For each particle on a given frame, the distances to all particles on the subsequent frame are calculated and the pair giving the smallest distance is selected. However, at high densities particles come into proximity of one another, introducing ambiguities in the linking process. One can attempt to reconstruct all trajectories simultaneously (multiparticle tracking), potentially using information from multiple frames (multiframe tracking) by minimizing a cost function over all possible links between particles (to obtain the global solution for that cost function). Unfortunately, since this combinatorial problem scales factorially in the number of particles and exponentially in the number of frames, finding that the global solution is computationally unfeasible for even tens of particles (there are $(10!)^9 \cong 10^{60}$ different ways to connect just 10 particles over 10 frames). As a result, many tracking algorithms have been developed that introduce constraints and approximations to solve the particle linking problem locally and provide a best-guess prediction of the global solution (1–4). Some of these have been released as publicly available software (5–10).

Further challenges come from permanent or transient appearance and disappearance of particles due to fluorescence photobleaching, foci moving out of the focal plane, or simple detection failure. Some tracking algorithms therefore allow trajectories to terminate or begin mid-sequence under certain conditions as well as close gaps in a trajectory due to missing localizations. Merge and splitting events may also be allowed. Given these different approaches and the different implementations used, it is not surprising that no single algorithm excels at all particle-tracking applications (4).

Submitted December 1, 2022, and accepted for publication March 22, 2023.

*Correspondence: sean.murray@synmikro.mpi-marburg.mpg.de

Editor: Dylan Myers Owen.

https://doi.org/10.1016/j.bpj.2023.03.033

The algorithms referenced above focus on the relatively high-density regime in which there are many particles with overlapping trajectories. However, for systems with fewer but still overlapping trajectories, more global and less-approximative methods become more feasible. One such application is the study of the dynamics and replication of chromosomal loci or low-copy extrachromosomal DNA such as plasmids within living cells. Under normal circumstances, these "particles" only increase in number during the cell cycle, typically by a factor of two and are persistent in that, while the fluorescent foci visualizing them may appear or disappear, the particles themselves do not. This additional information places a significant constraint that can be used to optimize their tracking. At the same time, the typically low copy numbers involved mean that the problem lies toward the other extreme of the efficiency-accuracy tradeoff, i.e., some of the computationally necessary simplifications required at higher numbers can be avoided. To our knowledge, no tracking software is tailored to this scenario and we found that existing tools did not perform with sufficiently high accuracy. Note that, despite the low copy numbers, generating the optimal tracking may not be trivial if there are many frames (e.g., there are $(2!)^{99} \cong 10^{30}$ ways to connect 2 particles over 100 frames). Furthermore, false positive (i.e., spurious) and false negative detections (due to the foci moving out of the focal plane, temporary merging, or detection failure) confound the tracking and make it challenging to determine the time point at which foci are duplicated. This is especially true for bacteria due their small size and the smaller pool of fluorescent protein. However, since DNA loci are persistent, both false positive and false negative detections can be more confidently identified.

Here, we present ★Track (pronounced "star-track") (as well as referring to the A* algorithm, we are inspired by the quote of Captain Kirk from Star Trek: "*There's no such thing as the unknown, only things temporarily hidden*"), a tool for the accurate tracking of replicating chromosomal and extrachromosomal DNA, and other low-copy-number persistent particles, from time-lapse images of live cells. The algorithm is transparent and user friendly and is completely specified, in the default case, by only four input parameters (there are no hidden internal parameters). MATLAB and Python implementations are provided so that it can be easily integrated into existing spot detection pipelines. We use ★Track to study the timing of plasmid replication within single cells, something that has not been achieved previously (11), and gain insight into the mechanisms of plasmid copy number control. We also investigate chromosome replication initiation in *Escherichia coli*.

MATERIALS AND METHODS

Strains and growth condition

F plasmid experiments use strain DLT3125 (12), a derivative of the *E. coli* K-12 strain DLT1215 (13) containing the mini-F plasmid derivative pJYB234. This plasmid carries a functional ParB-mVenus fusion. Overnight cultures were grown at 37°C in LB medium supplemented with 10 μg/mL thymine +10 μg/mL chloramphenicol. The strain IS130 was constructed by transduction of matP-YPet from RH3 (14) and oriC-$parS_{P1}$ from strain RM3 (15,16) into *E. coli* K-12 MG1655 strain (lab collection), which was then transformed with pFHCP1-mTurquoise2 plasmid. The plasmid pFHCP1-mTurquoise2 was derived from plasmid pFHC2973 by deletion of ygfp-$parB_{pMT1}$ and replacement of CFP with mTurquoise2 (15,17). Overnight cultures were grown in M9 minimal medium supplemented with 0.2% glucose, 2 mM $MgSO_4$, 0.1 mM $CaCl_2$, and 0.5 mg/mL BSA.

Microfluidics

A mother machine (18) is a microfluidic device consisting of long narrow growth channels, the width of a single bacterial cell, connected to a main channel through which nutrient medium flows. Cells can be trapped in the growth channels and monitored over many generations. Our device is broadly similar to the original. The growth channels are 35 μm long and are organized into blocks of 30. Channel widths vary between 0.8 and 1 μm. However, we follow Baltekin et al. (19) and also include: 1) a small opening at the end of each growth channel, 2) a waste channel connected to that opening to allow a continuous flow of nutrients through the growth channels, and 3) an inverted growth channel that is used to remove the background from fluorescence and phase contrast. Before imaging, the chip is bound to a glass slide using a plasma generator and baked for 30 min at 80°C.

Microscopy

We used a Nikon Ti microscope with a 100×/1.45 oil objective and a Hamamatsu Photonics camera for all imaging. For imaging cells of strain DLT3125 and IS130 we used a mother machine. Overnight cultures of DLT3125 were inoculated into fresh medium (M9 + 0.5% glycerol + 0.2% casamino acids + 0.04 mg/mL thymine + 0.2 mg/mL leucine + 10 μg/mL chloramphenicol) for 4 h at 30°C before imaging. For strain IS130, 50 μM IPTG (for induction of mTurquoise2-$ParB_{P1}$) was added to the medium, defined in the strains and growth condition, 2 h before and during the experiment. Cells were loaded into the chip through the main channel and the chip was placed into a preheated microscope at 30°C. The cells were constantly supplied with fresh medium by pumping 2 μL per min through the microfluidic chip. Cells were grown for at least 2 h inside the microscope before imaging. DLT3125 was imaged at 1 min intervals and IS130 was imaged at 5 min intervals for approximately 72 h. Both phase contrast and fluorescent signal were captured.

Image processing

Our image processing pipeline, Mothersegger (https://gitlab.gwdg.de/murray-group/MotherSegger), has been described previously (20). In brief, it consists of four parts: 1) preprocessing, 2) segmentation, 3) cell tracking, and 4) foci detection. Parts 1, 3, and 4 use custom MATLAB scripts, while part 2 is based on SuperSegger (21), a MATLAB-based package for segmenting and tracking bacteria within microcolonies, that we modified to better handle high-throughput data. In part 1 each frame of an acquired image stack is aligned (the offset between frames in x and y is removed). Afterward the image stack is rotated so the growth channels are vertical. A mask of the mother machine layout is fitted to the phase contrast, using cross correlation, to identify where the growth channels are located. Each growth channel is extracted from the image stack, and the flipped inverted channel is subtracted to remove the background from both the fluorescence signal and phase contrast. In part 3 the cells are tracked. Since cells cannot change their order inside the growth channel, they can be tracked by matching similar cell length between frames (starting from the bottom of the channel). Cell cycles that do not have exactly one parent and two daughters are excluded from

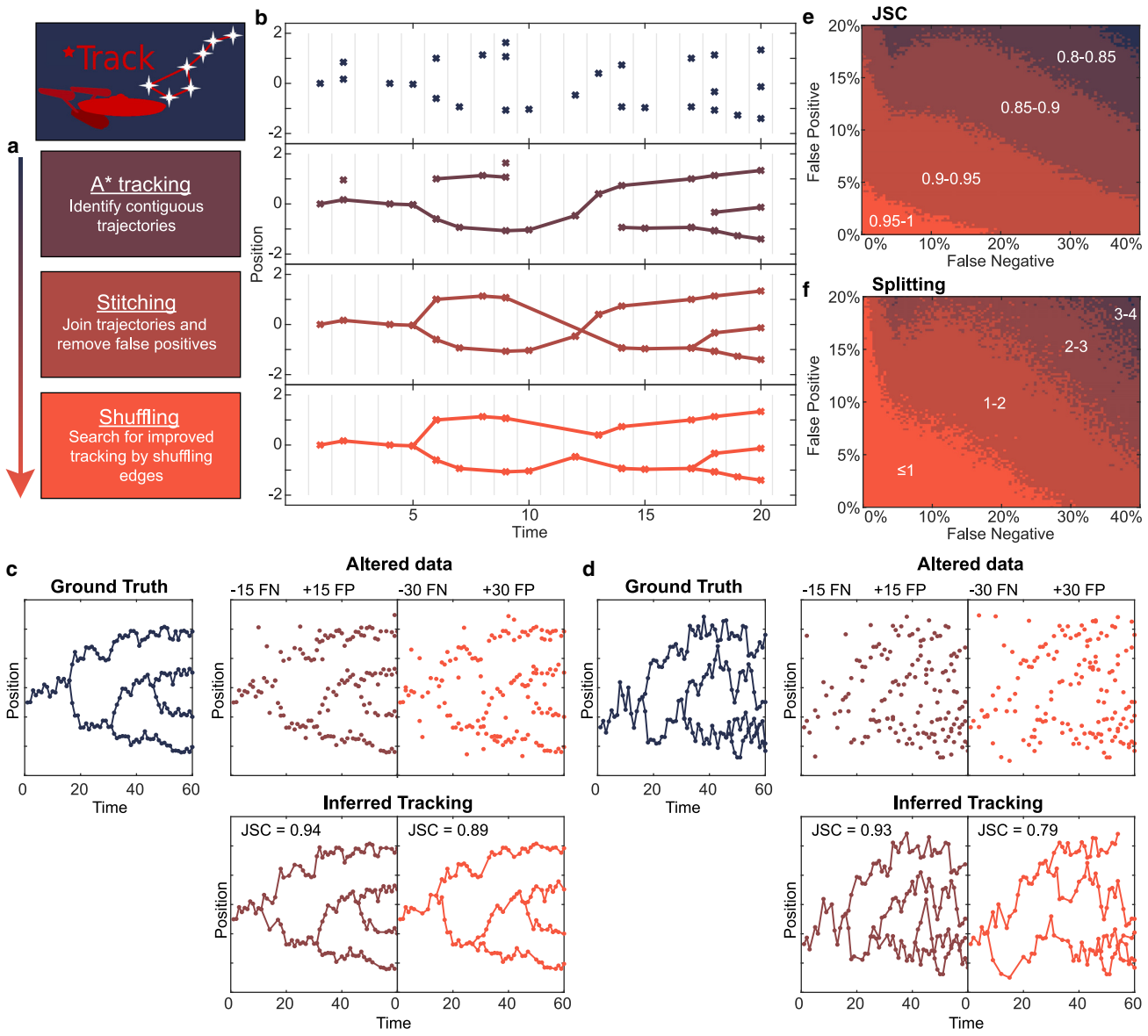

FIGURE 1 Overview and accuracy of ★Track. (*a*) All main steps of the algorithm with a short description. The arrow indicates the order. (*b*) Example time lapse of localizations and how the data look after each step in (*a*). (*c*) A simulated trajectory (150 foci) is used as ground truth to measure the accuracy of ★Track. The ground truth is altered by adding a number of spurious foci (false positive [FP]) and removing a number of real foci (false negative [FN]) (panels labeled Altered data). ★Track was then used to infer a tracking of the altered data (panels labeled Inferred). Accuracy is quantified using the Jaccard similarity coefficient (JSC).The JSC is equal to the number of real foci included in the tracking divided by the sum of the number of spurious foci in the tracking and the total number of real foci in the altered data. The JSC is between 0 and 1, where 1 is a perfect score (the tracking contains all the real and no spurious foci) and 0 is the worst score (no real foci included in the tracking). (*d*) Same as (*c*) but for more mobile particles. (*e*) JSC for a wide range of false positive and false negative rates. At each point, 100 trajectories were generated as in (*c*). (*f*) Same as in (*e*) but for splitting accuracy. The number indicates the mean absolute deviation of the splitting event in frames from the ground truth.

analysis along with their immediate relatives (with the exception of those who are pushed out of the growth channel). Part 4 detects fluorescent foci within cells. It is based on the SpotFinderZ tool from Microbetracker (22).

Computer-generated trajectories and manipulation

The trajectories used in Fig. 1, *c–f* were generated with our previously published model of plasmid positioning (20). This is a stochastic model of plasmid positioning by the interaction of plasmid-bound ParB with nucleoid-associated ParA. These trajectories were used as ground truth. As stated in Fig. 1 we manipulated the ground truth by adding (false positives) and removing (false negatives) a percentage of foci. If the ground truth contains 100 foci and 10% false positives are added and 5% false negatives are removed the resulting modified data set contains 105 foci (100 + 10 fp – 5 fn).

A* algorithm

The first step of ★Track is based on the A* graph-traversal algorithm (23), which we first summarize. A graph consists of nodes connected by edges

that have a weight corresponding to the cost of traversing that edge. The cost of a path on the graph is the cumulative cost of all its edges. Given a start and target node of a graph, the A* algorithm finds the shortest path between them. A path is considered the shortest if there exists no other path connecting these two nodes with a lower cost. The algorithm begins by initializing a candidate list of partial paths consisting of only the first edge (the edges of the starting node). The key feature of the algorithm is the generation of a lower bound for the cost of completing a partial path. The determination of this bound is problem specific. The algorithm then selects the first edge that gives the lowest lower bound for the total cost (the cost of the first edge plus the lower bound for completing the path starting from that edge). In the second iteration, all possible choices of second edge (with the chosen first edge) are added to the candidate list and again the best second edge is selected based on the cost estimate. This process repeats until either the target node is reached or the estimated cost of the current best partial path exceeds the estimated cost of completing a partial path further up the graph, e.g., a path with a different first edge. In this case, the process then continues from this different first edge. In this way the algorithm eventually finds the path with global minimum cost.

Cost of linking two foci

Use of the A* algorithm requires the specification of a cost for linking focus i on one frame to focus j on a later frame. For this, we assume that foci move diffusively between frames and consider the probability that the 2D distance traveled would be at least as great (the cumulative distribution function) as that observed

$$P(d,t,D) = e^{-d^2/(4Dt)}$$

where d is the 2D spatial distance between the foci, t is the number of frames between foci, and D is the diffusion coefficient. The movement cost is then $c_{i,j} = -log(P(d,t,D))$. An alternative model can be used if required. If the two foci are not from consecutive frames then we add a cost $c_{neg} = -log(p_{neg})$ for each of the intermediate $t-1 \leq m$ frames, where p_{neg} is the probability of a false negative and m is a user-provided upper bound for the number of frames on which the focus was missing. Below, we define a cost threshold for acceptable links that is used in the first part of the algorithm. Note that if $p_{neg} = 0$ tracking cannot contain false negatives as the tracking would otherwise fail due to having infinite cost.

Layered graph

The second requirement of the A* algorithm is that the search is between start and target nodes. On the other hand, the initial tracking consists of time-directed connections between foci that need not form a connected path. We therefore developed a mapping of the temporal tracking problem to a radial layered graph structure on which the A* algorithm could be applied.

We illustrate this with an example in Figure S1. Figure S1 *a* shows an example data set consisting of $n_{foci} = 6$ foci localizations across $t_{end} = 5$ frames. We first determine the set of acceptable links between these foci. Acceptable links are those with a cost below the threshold $c_{th} = min(-m\,log(p_{neg}), c_{max})$ based on the user provided maximum number of frame skips, m and the false negative probability. This threshold ensures no link is allowed between foci separated by more than m intermediate frames, as well as very spatially distant links across fewer frames. It corresponds to a spatial-temporal search radius for linking foci. We also include an upper bound on the threshold $c_{max} = -log(0.0001)$ to account for when $p_{neg} = 0$ (no false negatives allowed) in which case the threshold would otherwise be infinite. Note that this threshold is only used in this step of the algorithm and not in the subsequent steps (stitching/shuffling).

Given these acceptable links, we can then create the layered graph. We first label foci sequentially starting from one of the potentially multiple foci on the first frame (Fig. S1 *a*). The corresponding graph contains one layer for each focus plus an additional outermost layer to determine when the tracking is complete (Fig. S1 *b*). Every path from the first to the last layer corresponds to one possible tracking of the foci in Fig. S1 *a*. The relationship to a tracking is as follows. An edge from *any node* in *layer i* into a numbered *node j* in layer $i+1$ corresponds to an acceptable link between focus i and focus j (Fig. S1, *c* and *d*). An outgoing edge into a node labeled "x" corresponds to focus i having no outgoing link. Note that the layer into which an edge goes does not matter for the interpretation, nor does the number of the node it came from. A path from layer 1 to the outermost layer crosses each layer once, therefore each layer (except the outermost) has exactly one outgoing edge. This ensures that each focus has at most 1 outgoing link. Furthermore, each path contains each numbered node at most once. If a path traverses a numbered node i, node i will not appear again in the higher layers. This ensures that each focus has only one incoming link. The construction of the graph is illustrated in Video S1.

Each edge in the graph has a weight (referred to but not shown in Fig. S1 *b*) that corresponds to the cost of linking the foci associated with that edge, as described above. An edge going from layer i into a numbered node j has a cost $c_{i,j}$. An edge from layer i going into a node labeled "x" has a cost equal to $c_{i,x} = min(c_{th}, -log({p_{neg}}^{t_{end}-t_i}))$ where $t_{end} - t_i$ is the number of frames between that of focus i, t_i, and the last frame t_{end}. The cost of a path is the cumulative cost of its edges and the optimal tracking is the path from the root node in layer 1 to the outermost layer with the lowest cost.

To use the A* algorithm we require a heuristic lower-bound estimate of the cost of reaching the outermost layer from each node in the graph. A path from layer i has to traverse all subsequent layers to reach the outermost layer. A lower bound for the cost is then the sum of all the lowest costing outgoing links $c_{est}(i) = \sum_{j=i}^{n_{foci}} min(\{c_{j,\cdot}\})$, where the minimum is over the cost of all acceptable outgoing links from layer (focus) j (including to the "x" node).

Since the number of nodes in each layer increases exponentially, it is infeasible, and also unnecessary, to pregenerate the entire graph when tracking more than a few foci over a small number of frames. Therefore, the graph is generated dynamically as the A* algorithm traverses it. An example can be seen in Video S2. Note that the optimal tracking is found without generating the entire graph. To further reduce the complexity, the candidate list is filtered once a new layer is reached. All candidate paths ending $1 + max(\{j - i : t_j - t_i \leq m\})$ layers below the newly reached layer are removed from the candidate list. Therefore, the greater the number m of allowed frame skips (consecutive false negatives), the more of the graph is retained. In addition, if the candidate list exceeds 10^6 entries, the 10% of partial paths with the highest cost estimates are removed. While formally making the algorithm approximative, in practice this threshold has little effect on the accuracy of the results (Fig. S2 *a*) and can be reduced to as low as 10^5 to reduce the memory usage of the program and reduce runtime (Fig. S2 *b*).

Stitching

The goal of this procedure is to integrate the trajectory segments returned by the A* algorithm such that the number of foci monotonically increases. This is done in three steps. First, loose beginnings (foci which have no incoming link) are linked to earlier foci (Fig. S3 *b*), either to a real focus according to the lowest cost of the link or to an imaginary focus before the first frame. The latter is for implementation reasons only.

In the next step, loose ends (foci which have no outgoing link) are integrated into the tracking (Fig. S3 *d*). This occurs in three different ways: 1) a loose end can be linked to an imaginary focus after the last frame (Fig. S3 *c* ii and iii), implying some number of missed foci (false negatives) when the loose end is not on the last frame; 2) all the foci of the segment containing the loose end up to any branching point are considered as false positives; 3) a loose end can be interwoven into the tracking by replacing an existing link, which can but does not have to be a link created by integrating a loose beginning (Fig. S3 *c* iv and v). This may create a different loose end, which is treated as in 2). Which method occurs is determined by the cost of the entire tracking.

The cost of the tracking is calculated differently than in the A* step as it incorporates the concept of false positives and applies to the entire tracking and not just disconnected segments. Furthermore, the threshold cost (c_{th}) of the A* step is no longer used. Each false positive focus has a cost $c_{pos} = -log(p_{pos})$, where p_{pos} is the false positive probability. Each real focus then has a cost arising from its probability to not be a false positive $c_{real} = -log(1 - p_{pos})$. The cost of a tracking is then $c_t = n_{fp}c_{pos} + n_{real}c_{real} + n_{fn}c_{neg} + \sum_{links\ i \to j, for\ real\ foci\ i,j} c_{i,j}$, where n_{fp} is the number of false positives, n_{real} is the number of detected foci, and n_{fn} is the number of false negative foci. Note that if $p_{pos} = 0$ false positives are not allowed, the $n_{fp}c_{pos}$ term is not included in the total cost.

Returning to the integration of loose ends, the change in the cost of the tracking is calculated for every possible way to integrate a loose end and the method with the lowest cost is chosen. The procedure is iterative starting at the last loose end. It continues until there are no loose ends left or the remaining loose ends consist of false positives.

The third and final step is simply the removal of the foci identified as false positives from the tracking (Fig. S3 *e*).

The result of this procedure is a tracking in which every focus is path connected to the first and last frames and the number of foci never decreases. All the steps above were developed with this goal in mind. Connecting loose beginnings first and then handling loose ends was found to be the best approach for stitching together trajectories produced by the A* algorithm, compared with connecting loose ends first or doing both simultaneously. This is because of disconnected fragments, which possess both a loose end and loose beginning. Multiples of these fragments might be a part of a long branch but remain unlinked due to the cost threshold during the A* step. This is problematic because these fragments might be labeled as false positives and removed, instead of being incorporated as a single long branch. To circumvent this issue, loose beginnings are connected first, which links all disconnected fragments to the tracking and provides the necessary context for making the most appropriate decisions how and where to incorporate loose ends.

Shuffling

Shuffling resolves remaining inaccuracies in the tracking after the stitching procedure. Inaccuracies sometimes appear in the output of the A* algorithm due to false negative foci. For example, the crossing links in the third panel of Fig. 1 *b* is due an inaccurate trajectory segment from the A* step, which in turn is due to multiple consecutive missed foci causing the upper segment in the top panel to terminate. The most straightforward way to resolve these types of inaccuracies is by shuffling links as follows. Two links connecting four foci, focus i linked with focus j and focus n linked with focus m, have a combined movement cost of $c_{i,j} + c_{n,m}$. The links between the foci can be shuffled, subject to $max(i,m) < min(j,n)$, such that focus i is linked with focus m and focus n is linked with focus j. If the new cost of $c_{i,m} + c_{n,j}$ is lower than $c_{i,j} + c_{n,m}$ then the new configuration is accepted. Foci are shuffled until there are no pairs of links left that can be shuffled to reduce the cost of the tracking (Fig. 1 *b* bottom). A limitation of this procedure is that it can only find an improved tracking that is obtainable by shuffling two links. Improved trackings obtainable through more complicated rearrangements cannot be identified.

RESULTS AND DISCUSSION

★Track operates on a provided list of foci localizations (t,x,y), consisting of the frame number and location of the fluorescent foci within a cell or region of interest (corrected for movement and growth as required). The algorithm requires a cost function that specifies, for any given focus, how likely it is that another focus on the next or some later frame is the next detection of the same underlying particle (a higher likelihood results in a lower cost). The latter case implies that the same focus was not detected on some intermediate frames and ★Track requires the user to specify the maximum number of consecutive frames for which this is allowed to occur ("not detected" includes foci that have merged to within the diffraction limit). This naturally defines a minimum spatiotemporal distance (cost threshold) between foci that greatly increases the computational efficiency of the first step of the algorithm. In subsequent steps connections greater than the threshold are allowed provided they improve the overall tracking.

The default cost function is relatively simple and uses only the positions of foci. However, it could easily be modified to incorporate the spot intensity, goodness of fit, or specific models of movement. We have found for the applications studied that spot intensity and shape can vary considerably between frames due to stochastic variation and movement out of the focal plane and we therefore base the cost function only on the foci positions. The default function assumes that foci move diffusively between frames (this is always true on a short enough timescale) and requires only the diffusion coefficient D. The user must also specify p_{fn}, the probability of a spot not being detected on a given frame (false negative), and p_{fp}, the probability of a detected spot being spurious (false positive). These three parameters can be estimated by analyzing cells with a sufficiently low number of foci and no splitting events (Fig. S4). Precise values are not required as the algorithm is robust with respect to these parameters (Fig. S2, *c* and *d*).

The pipeline consists of three steps (Fig. 1 *a*). First, the tracking problem is formulated as a pathfinding problem on a layered graph (Fig. S1). The A* algorithm (23) then uses the cost function, with a threshold cost derived from the max number of frame skips, to find the optimal path on this graph and thereby generate candidate trajectory segments (Figs. 1 *b*, *second panel* and S1). The resulting trajectories are then stitched or interwoven together in such a way that the total cost of the tracking is minimized. Afterward, each focus is either marked as a false positive and removed or is part of a trajectory that can be traced from the first frame to the last (Figs. 1 *b*, *third panel* and S3). Finally, a lower cost tracking is searched for by shuffling links in the tracking (Fig. 1 *b*, *bottom*). This helps to mitigate inaccuracies in the first step due to false negative/positive foci. The contributions made by each step of the algorithm are given in Fig. S5. Note that ★Track does not predict the location of missing foci, but simply their existence. A detailed description of the A* algorithm and the stitching and shuffling steps can be found in the materials and methods.

To assess the accuracy of ★Track, we generated ground truth data using a stochastic model of plasmid positioning (20) and mimicked the presence of false positive and false negative foci by randomly adding and removing data points,

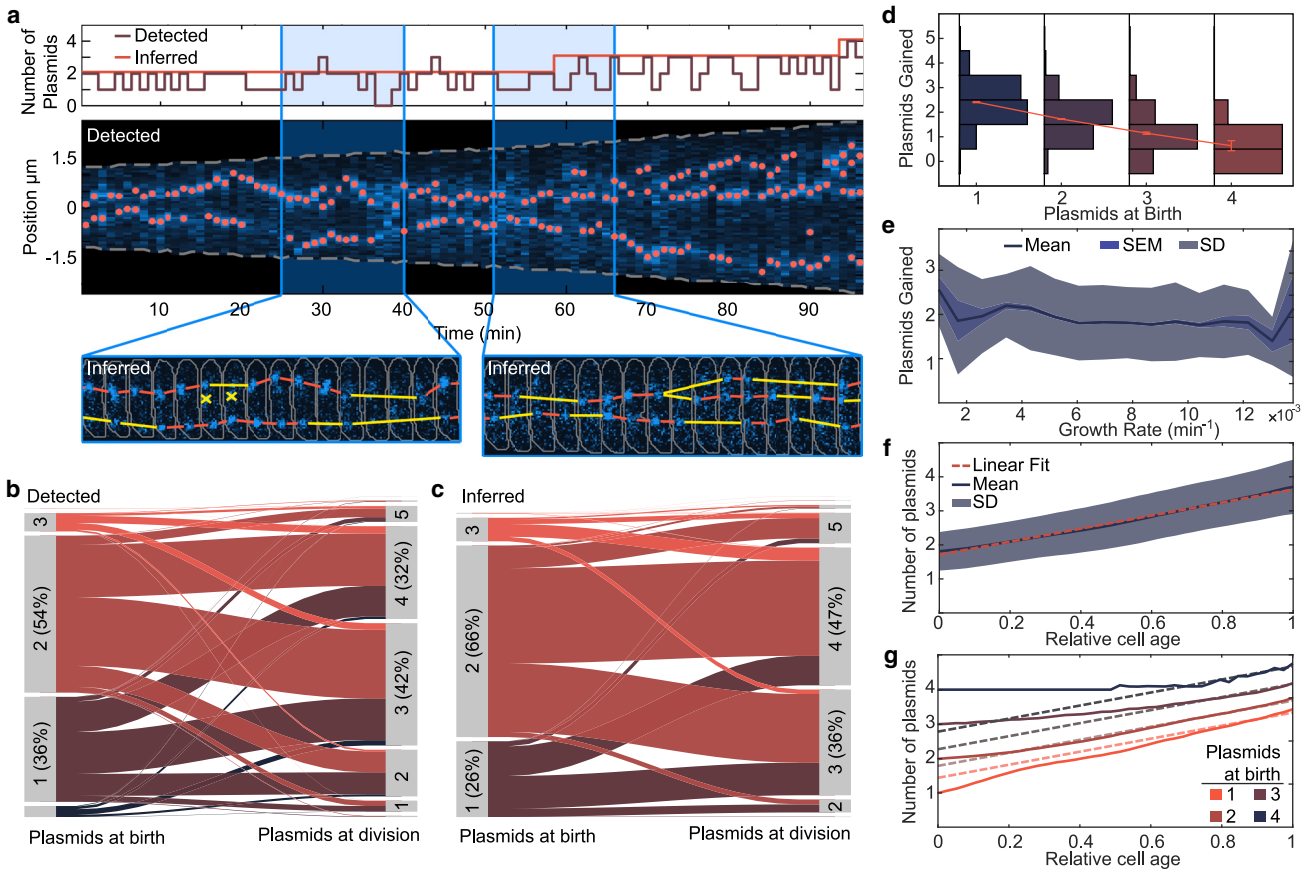

FIGURE 2 Tracking F plasmid. (*a*) Example cell cycle of the *E. coli* strain DLT3125, hosting an F plasmid encoding an mVenus-ParB fusion, which binds and labels the plasmid. Top: detected and inferred number of plasmids during the cell cycle. Middle: kymograph of the fluorescent signal of the example cell with detected plasmid locations (*red dots*). Gray dashed lines indicated cell boundaries. Bottom: detailed view of the indicated regions. Red and yellow lines indicate the inferred tracking. Yellow lines indicate frames in which missing foci were inferred (false negatives). Red cross indicates foci identified as false positives. Gray lines indicate cell contours. (*b*) Alluvial plot showing how the number of plasmids changes from birth to division in the raw data. Frame rate is 1 min^{-1}. (*c*) Same as (*b*) but with the inferred data. (*d*) Side-ways histograms of plasmids gained during a cell cycle plotted against the number of plasmids at birth. The red line depicts the mean number of plasmids gained for each group mean ± SE. (*e*) Plasmids gained during a cell cycle plotted against the growth rate mean ± SE and SD. (*f*) Mean number of plasmids ± SD plotted against relative cell age (0 is birth and 1 is division). The red dashed line is a linear fit. (*g*) The mean number of plasmids grouped by the number of plasmids at birth plotted against relative cell age. The dashed lines are linear fits with the same slope as in (*f*) with the intercept chosen by fitting to the last portion of each group.

respectively. We found that, even after removing a substantial fraction of localizations and adding many false positives, ★Track could accurately reproduce the ground truth tracking including correctly identifying the time points of plasmid replication (Fig. 1, *c* and *d*). Performing a sweep over a range of false positive and false negative rates, we found that ★Track performs with consistently high precision (Fig. 1, *e* and *f*) and outperformed the state of the art and widely employed method u-track (5,6,24) (Fig. S6).

To test ★Track on real data, we applied it to the study of plasmid copy number control. This has been a topic of active research for several decades (25–29) but progress has more recently slowed due, in part, to the inability to accurately measure plasmid copy numbers in individual cells (11). Having recently performed a high-throughput study of the partitioning mechanism of the low-copy-number F plasmid using a microfluidic "mother machine" device (20), we decided to revisit our data in the context of copy number control. F plasmid is a tractable system in this regard since duplicated plasmids segregate beyond the diffraction limit within about a minute of replication (30,31). We expected that ★Track should therefore be able to accurately determine the temporal changes in copy number during the cell cycle.

In Fig. 2 *a* we present an example time lapse showing how the number of detected foci (of plasmid-labeling ParB-mVenus) changes substantially from frame to frame (see Fig. S7 for more examples). As discussed above, this can occur due to foci moving out of the focal plane, merging together, or simply due to stochastic fluctuations in the number of plasmid-bound ParB-mVenus. However, ★Track inferred a consistent tracking that produces a stepwise increasing copy number and predicts the number and frame of the replication events (Fig. 2 *a*). We then applied the

algorithm to the entire data set of 4096 cell cycles and examined how the copy number changes between birth and division. The raw data displayed several irregularities such as cells having no detected plasmids at birth and cell cycles in which the number of plasmids decreased (Fig. 2 *b*). All of these inconsistencies were removed by ★Track (Fig. 2 *c*), which also inferred a mean copy number at birth ∼12% greater than that of the mean number of detected foci. We found a decreasing linear relationship between the number of plasmids gained during the cell cycle and the number at birth (Fig. 2 *d*), indicating a "sizer"-like mechanism for copy control (32). Interestingly, the mean number of plasmids gained did not depend on the growth rate of the individual cell (Fig. 2 *e*), indicating that plasmid production is coupled to the growth rate of the host so as to produce the same number of plasmids irrespective of the cycle duration.

★Track allows us to probe the timing of plasmid replication within the cell cycle. We found that the mean number exhibits a surprising linear relationship with time (Fig. 2 *f*). This is consistent with the net replication rate being constant in time, i.e., plasmids are produced at the same rate irrespective of how many are present. However, the explicit nature of the sizer regulation could be seen by binning cell cycles according to the number of plasmids at birth (Fig. 2 *g*). Cells born with fewer/more plasmids than average have a higher/lower net plasmid replication rate initially before returning to the population mean production rate toward the end of the cell cycle. This results in outliers converging rapidly to the mean. While negative feedback has long been known to underlie plasmid copy number control (25–29), we believe that this is the first time it has been characterized at the level of the cell cycle. Furthermore, our result that the effect of the regulation is to push the system back to a constant net replication rate can now be used to test models of copy number control and motivate further study.

Next we used ★Track to analyze the replication of the origin region of the *E. coli* chromosome. We used a strain in which the origin of replication (*ori*) is visualized through the P1 *parS*/ParB labeling system using an mTurquoise2-ParB fusion and imaged several thousand cell cycles using the same mother machine device as we used for studying F plasmid. As for that case, foci are not always visible or detected and spurious foci can confound interpretation (Fig. 3 *a*, see Fig. S7 *b* for further examples). This makes it challenging to determine the time of duplicated *ori* separation with certainty. Recent studies have used a replisome reporter to identify initiation events (33,34), which occur approximately 15 min before separation of *ori* foci (35); however, similar detection issues occur. These irregularities could be seen by comparing the number of *ori* on the first and last frames of the cell cycle (Fig. 3 *b*). Significant cell populations have no detected foci at birth or only one focus at division. However, ★Track corrected almost all these inconsistencies (Fig. 3 *c*).

Surprisingly, given the slow growth conditions used (median doubling time of 140 min), we found that 14% of cells were born with two *ori* (this is a lower bound since a single focus could consist of two unsegregated *ori*). Plotting the number of *ori* against cell age revealed two periods of the cell cycle during which *ori* separation, and hence, replication initiation occurs (Fig. 3, *d* and *e*). The majority population, cells born with a single *ori* focus, replicated their origin in the first third of the cell cycle. However, in about 15% of these cells additional replication events occurred toward the end of the cell cycle producing predivisional cells with three or four *ori* foci (Fig. 3 *e* and *f*). Cells born with two foci only initiated replication in this later period and never at the beginning of the cell cycle, although the majority of such cells do not initiate replication at all and so produce daughter cells with one *ori* focus each.

While analyzing replication initiation in terms of relative cell age is useful, the prevailing understanding is that chromosome replication initiates at an invariant cell volume per *ori* (33,34,36). Consistent with this, we found that *ori* focus duplication in cells with a single *ori* occurred at a cell volume of about 0.8 μm^3, whereas in cells with two *ori* it occurred at twice this volume (Fig. S9). As such, there is really only one replication period defined by the volume per *ori*, but this period overlaps the division event and therefore we are able to observe two different values for the volume at initiation in the same population. This supports previous evidence for the invariant volume per *ori* hypothesis, which has been largely based on comparing populations under different growth conditions.

★Track is a powerful new tool for the tracking of plasmid and chromosomal DNA loci. It provides predictions for the timing of replication (initial foci splitting) events that should provide mechanistic insight across a range of systems, and not only for bacteria. ★Track can also be applied to any persistent particles that increase in copy number during the cell cycle and could therefore be used to study protein complexes that display this behavior (37–40). The algorithm is deterministic (there are no stochastic or deep learning components) and therefore produces reproducible results with the user having to specify only four estimatable parameters. ★Track is available at https://gitlab.gwdg.de/murray-group/StarTrack.

Limitations of the study

★Track is designed for the tracking of small numbers of persistent particles. However, the first part of the algorithm (the A* step) does not require persistence and can be used in isolation to identify trajectory segments of nonpersistence particles. This can be achieved through the setting of a flag in the code. However, with higher particle numbers the A* step will become increasingly computationally expensive (take too long to run). This will depend on the provided parameters

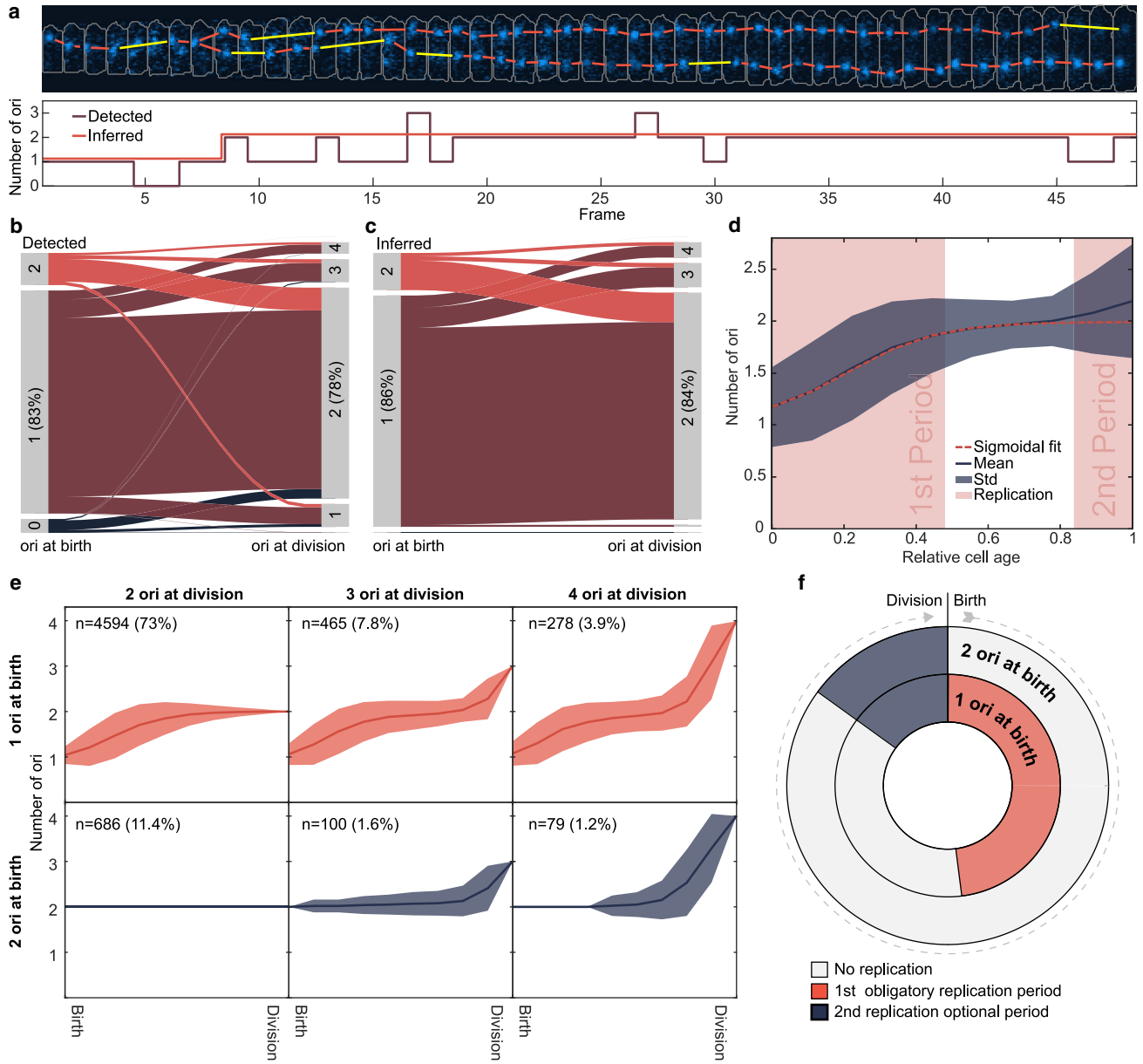

FIGURE 3 Replication of *ori* occurs during two periods in the cell cycle. (*a*) An example cell cycle of the *E. coli* strain IS130. It encodes mTourquise2-ParB, which labels the origin of replication (*ori*). Top: fluorescent signal with tracked *ori*. Bottom: number of *ori*, both detected and inferred. Frame interval is 5 min. (*b*) Alluvial plot showing how the number of *ori* changes from birth to division of the detected foci. (*c*) Same as (*b*) but after ★Track was used. (*d*) Mean number of *ori* ± SD plotted against relative cell age (0 is birth and 1 is division). The two areas shaded in light red indicate periods during the cell cycle in which *ori* is replicated. The red dashed line is a sigmoidal fit from 0 to 0.85 (excluding the second period). See also Fig. S8. (*e*) Data from (*d*) grouped by number of *ori* at birth (rows) and number of *ori* at division (columns). (*f*) Cartoon of replication periods. The plots in (*b*)–(*e*) were created from 6265 cell cycles.

through the threshold cost for acceptable links c_{th} (see materials and methods). Nevertheless, the method should also be useful for applications with intermediate particle numbers in which accuracy is paramount.

SUPPORTING MATERIAL

Supporting material can be found online at https://doi.org/10.1016/j.bpj.2023.03.033.

AUTHOR CONTRIBUTIONS

R.K. and S.M.M. designed the research. R.K. developed, tested, and applied the method. I.S. performed origin tracking experiments. R.K. and S.M.M. wrote the paper.

ACKNOWLEDGMENTS

This work was funded through grant MU 4469/2 from the Deutsche Forschungsgemeinschaft (DFG). We thank Frédéric Boccard for strain RM3

and plasmid pFHC2973, Jaan Mannik for strain RH3, and Jean-Yves Bouet for strain DLT1215 and plasmid pJYB234.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Manzo, C., and M. F. Garcia-Parajo. 2015. A review of progress in single particle tracking: from methods to biophysical insights. *Rep. Prog. Phys.* 78:124601.
2. Shen, H., L. J. Tauzin, …, C. F. Landes. 2017. Single particle tracking: from theory to biophysical applications. *Chem. Rev.* 117:7331–7376.
3. Meijering, E., O. Dzyubachyk, and I. Smal. 2012. Chapter nine - methods for cell and particle tracking. *In* Methods in Enzymology, *504*. P. M. conn, ed. Academic Press, pp. 183–200.
4. Chenouard, N., I. Smal, …, E. Meijering. 2014. Objective comparison of particle tracking methods. *Nat. Methods.* 11:281–289.
5. Tinevez, J.-Y., N. Perry, …, K. W. Eliceiri. 2017. TrackMate: an open and extensible platform for single-particle tracking. *Methods.* 115:80–90.
6. Jaqaman, K., D. Loerke, …, G. Danuser. 2008. Robust single-particle tracking in live-cell time-lapse sequences. *Nat. Methods.* 5:695–702.
7. Sage, D., F. R. Neumann, …, M. Unser. 2005. Automatic tracking of individual fluorescence particles: application to the study of chromosome dynamics. *IEEE Trans. Image Process.* 14:1372–1383.
8. Chenouard, N., I. Bloch, and J. C. Olivo-Marin. 2013. Multiple hypothesis tracking for cluttered biological image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.* 35:2736–3750.
9. Oviedo-Bocanegra, L. M., R. Hinrichs, …, P. L. Graumann. 2021. Single molecule/particle tracking analysis program SMTracker 2.0 reveals different dynamics of proteins within the RNA degradosome complex in *Bacillus subtilis*. *Nucleic Acids Res.* 49:e112.
10. Sbalzarini, I. F., and P. Koumoutsakos. 2005. Feature point tracking and trajectory analysis for video imaging in cell biology. *J. Struct. Biol.* 151:182–195.
11. Tal, S., and J. Paulsson. 2012. Evaluating quantitative methods for measuring plasmid copy numbers in single cells. *Plasmid.* 67:167–173.
12. Sanchez, A., D. I. Cattoni, …, J. Y. Bouet. 2015. Stochastic self-assembly of ParB proteins builds the bacterial DNA segregation apparatus. *Cell Syst.* 1:163–173.
13. Bouet, J. Y., J. Rech, …, D. Lane. 2005. Probing plasmid partition with centromere-based incompatibility. *Mol. Microbiol.* 55:511–525.
14. Männik, J., D. E. Castillo, …, J. Männik. 2016. The role of MatP, ZapA and ZapB in chromosomal organization and dynamics in *Escherichia coli*. *Nucleic Acids Res.* 44:1216–1226.
15. Mercier, R., M. A. Petit, …, O. Espéli. 2008. The MatP/matS site-specific system organizes the terminus region of the E. coli chromosome into a macrodomain. *Cell.* 135:475–485.
16. Li, Y., B. Youngren, …, S. Austin. 2003. Segregation of the *Escherichia coli* chromosome terminus. *Mol. Microbiol.* 50:825–834.
17. Nielsen, H. J., J. R. Ottesen, …, F. G. Hansen. 2006. The *Escherichia coli* chromosome is organized with the left and right chromosome arms in separate cell halves: *E. coli* chromosome segregation. *Mol. Microbiol.* 62:331–338.
18. Wang, P., L. Robert, …, S. Jun. 2010. Robust growth of Escherichia coli. *Curr. Biol.* 20:1099–1103.
19. Baltekin, Ö., A. Boucharin, …, J. Elf. 2017. Antibiotic susceptibility testing in less than 30 min using direct single-cell imaging. *Proc. Natl. Acad. Sci. USA.* 114:9170–9175.
20. Köhler, R., E. Kaganovitch, and S. M. Murray. 2022. High-throughput imaging and quantitative analysis uncovers the nature of plasmid positioning by ParABS. *Elife.* 11:e78743.
21. Stylianidou, S., C. Brennan, …, P. A. Wiggins. 2016. SuperSegger: robust image segmentation, analysis and lineage tracking of bacterial cells: robust segmentation and analysis of bacteria. *Mol. Microbiol.* 102:690–700.
22. Sliusarenko, O., J. Heinritz, …, C. Jacobs-Wagner. 2011. High-throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics. *Mol. Microbiol.* 80:612–627.
23. Hart, P. E., N. J. Nilsson, and B. Raphael. 1968. A formal basis for the heuristic determination. *IEEE Trans. Syst. Sci. Cybern.* 8
24. Ershov, D., M. S. Phan, …, J. Y. Tinevez. 2022. TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nat. Methods.* 19:829–832.
25. Pinto, U. M., K. M. Pappas, and S. C. Winans. 2012. The ABCs of plasmid replication and segregation. *Nat. Rev. Microbiol.* 10:755–765.
26. Chattoraj, D. K. 2000. Control of plasmid DNA replication by iterons: no longer paradoxical. *Mol. Microbiol.* 37:467–476.
27. Del Solar, G., and M. Espinosa. 2000. Plasmid copy number control: an ever-growing story. *Mol. Microbiol.* 37:492–500.
28. Paulsson, J., and M. Ehrenberg. 2001. Noise in a minimal regulatory network: plasmid copy number control. *Q. Rev. Biophys.* 34:1–59.
29. Konieczny, I., K. Bury, …, K. Wegrzyn. 2014. Iteron plasmids. *Microbiol. Spectr.* 2
30. Onogi, T., T. Miki, and S. Hiraga. 2002. Behavior of sister copies of mini-F plasmid after synchronized plasmid replication in *Escherichia coli* cells. *J. Bacteriol.* 184:3142–3145.
31. Walter, J. C., J. Rech, …, J. Y. Bouet. 2020. Physical modeling of a sliding clamp mechanism for the spreading of ParB at short genomic distance from bacterial centromere sites. *iScience.* 23:101861.
32. Jun, S., F. Si, …, M. Scott. 2018. Fundamental principles in bacterial physiology—history, recent progress, and the future with focus on cell size control: a review. *Rep. Prog. Phys.* 81:056601.
33. Wallden, M., D. Fange, …, J. Elf. 2016. The synchronization of replication and division cycles in individual E. coli cells. *Cell.* 166:729–739.
34. Si, F., G. Le Treut, …, S. Jun. 2019. Mechanistic origin of cell-size control and homeostasis in bacteria. *Curr. Biol.* 29:1760–1770.e7.
35. Reyes-Lamothe, R., C. Possoz, …, D. J. Sherratt. 2008. Independent positioning and action of Escherichia coli replisomes in live cells. *Cell.* 133:90–102.
36. Si, F., D. Li, …, S. Jun. 2017. Invariance of initiation mass and predictability of cell size in Escherichia coli. *Curr. Biol.* 27:1278–1287.
37. Murray, S. M., and V. Sourjik. 2017. Self-organization and positioning of bacterial protein clusters. *Nat. Phys.* 13:1006–1013.
38. Hofmann, A., J. Mäkelä, …, S. M. Murray. 2019. Self-organised segregation of bacterial chromosomal origins. *Elife.* 8:e46564.
39. Richards, D. M., A. M. Hempel, …, M. Howard. 2012. Mechanistic basis of branch-site selection in filamentous bacteria. *PLoS Comput. Biol.* 8:e1002423.
40. Roberts, M. A. J., G. H. Wadhams, …, J. P. Armitage. 2012. ParA-like protein uses nonspecific chromosomal DNA binding to partition protein complexes. *Proc. Natl. Acad. Sci. USA.* 109:6698–6703.