

Optimal Joint Segmentation and Tracking of *Escherichia Coli* in the Mother Machine

Florian Jug¹, Tobias Pietzsch¹, Dagmar Kainmüller¹, Jan Funke²,
Matthias Kaiser³, Erik van Nimwegen³, Carsten Rother⁴, and Gene Myers¹

¹ Max Planck Institute of Molecular Cell Biology and Genetics, Germany

² Institute of Neuroinformatics, University Zurich / ETH Zurich, Switzerland

³ Biozentrum, University of Basel, and Swiss Institute of Bioinformatics, Switzerland

⁴ Computer Vision Lab Dresden, Technical University Dresden, Germany

Abstract. We introduce a graphical model for the joint segmentation and tracking of *E. coli* cells from time lapse videos. In our setup cells are grown in narrow columns (growth channels) in a so-called “Mother Machine” [1]. In these growth channels, cells are vertically aligned, grow and divide over time, and eventually leave the channel at the top. The model is built on a large set of cell segmentation hypotheses for each video frame that we extract from data using a novel parametric max-flow variation. Possible tracking assignments between segments across time, including cell identity mapping, cell division, and cell exit events are enumerated. Each such assignment is represented as a binary decision variable with unary costs based on image and object features of the involved segments. We find a cost-minimal and consistent solution by solving an integer linear program. We introduce a new and important type of constraint that ensures that cells exit the Mother Machine in the correct order. Our method finds a globally optimal tracking solution with an accuracy of > 95% (1.22 times the inter-observer error) and is on average 2 – 11 times faster than the microscope produces the raw data.

1 Introduction

The *Mother Machine* [1] is a microfluidic device designed to study live bacteria. It allows the observation of growth and division of the progeny of single “mother” cells over many generations using time lapse microscopy. Figure 1 illustrates the Mother Machine and the respective image data. In such data, individual bacteria need to be *tracked* over time. Tracking consist of two equally important tasks: (i) cells need to be segmented in each frame, and (ii) all segments of the same cell need to be linked between frames. Tracking large numbers of cells under different environmental conditions will allow biologists to better understand the stochastic dynamics of gene expression within living cells. Respective high throughput studies of cells in the Mother Machine would be greatly facilitated if the tracking task would be automated.

Many existing automated tracking systems perform the tasks of segmentation and linkage one after another to reduce overall model complexity and runtime [2].

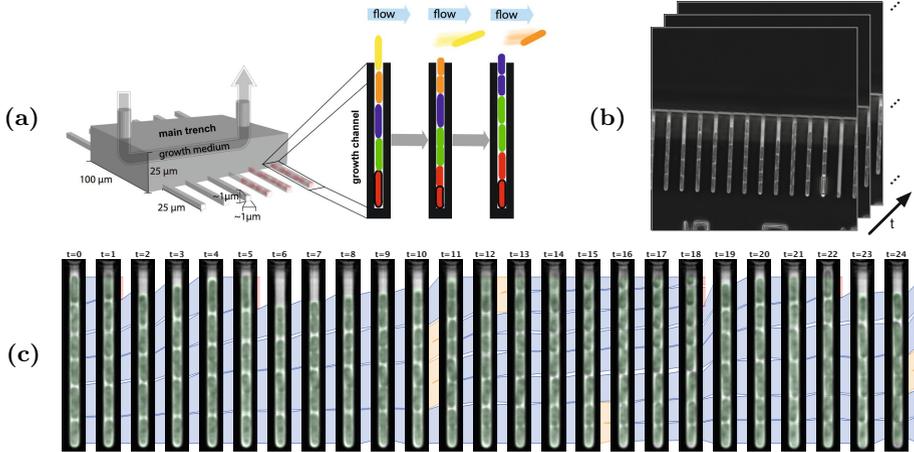


Fig. 1. (a) Illustration of the “Mother Machine”, a microfluidic device built to understand dynamic processes in *E. coli*. Individual ‘growth channels’ (narrow tubes, just wide enough for hosting a row of bacteria) are imaged every minute. (b) Raw images. (c) One growth channel in the first 25 frames of a time-lapse movie. A tracking is shown between frames, with mapping assignments in blue, division assignments in yellow, and exit assignments in red.

Model complexity is typically reduced even further by performing linkage in a locally optimal, *greedy* fashion [2], i.e. frame by frame, never considering the whole time series at once.

However, *globally optimal joint segmentation and linkage* can be achieved by so-called *Assignment Models* [3,4,5,6,7]. Assignment models pose the linkage problem as a global energy minimization task, where the energy is that of a graphical model (*factor graph*). Binary variables represent possible links (called *assignments*), with respective unary potentials capturing their plausibility. Higher order factors encode *continuity constraints*, that describe which link sequences form structurally sound tracks. Assignment models can elegantly handle an excess of non overlapping *segment hypotheses*¹. The only extra ingredient are additional unary factors assigning costs to all segment hypotheses. Energy minimization in such a model yields globally optimal, joint segmentation and tracking. The respective optimization task can be solved with existing discrete optimization methods [5,6,7].

A good assignment model should allow as many different segmentation hypotheses as possible to avoid missing segments (i.e. good *recall*). To this end, Kausler et al. [5] and Schiegg et al. [7] allow for an *over-segmentation* per time-frame. To be yet more robust against missed segments (false-negatives), Schiegg et al. propose a

¹ Superfluous segments do not have to be linked between frames but can be filtered by the tracking engine.

method capable of dealing with occasional under-segmentations. Funke et al. [6] introduced a model capable of dealing with a large pool of partially *conflicting* (overlapping) segment hypotheses per frame. Their model filters a conflict-free subset by introducing adequate higher order factors, called *tree constraints*. The work we presented here follows this “hypotheses-rich” approach of Funke et al. [6].

But in order to be as specific as possible for a given task (i.e. good *precision*), assignment models should be designed to restrict the space of possible solutions as much as possible. So far, relatively *generic* prior knowledge on cell movement and proliferation has been encoded into assignment models: Cells can be kept from moving too far between time frames; They can be allowed to divide but not merge; They can be kept from dividing into more than two, and kept from appearing from nowhere. However, none of the previously published assignment models captures a particular kind of prior knowledge that is important for cells in the Mother Machine, namely the *total order* of cells within growth channels which has to be maintained at any time.

The main technical contribution of this paper is a novel type of higher order factors which are concerned with the order of cells within growth channels. We call these factors *exit constraints*. We show that exit constraints considerably improve tracking accuracy in the Mother Machine (see Section 4.2). Another contribution is a new approach for generating nested segmentation hypotheses which outperforms previous approaches. The idea is to combine the benefits of parametric max flow [8] and random forest classifiers [9]. The random forest is used to improve the separation of recently divided cells which are otherwise hard to tell apart (see Section 3).

Our proposed assignment model can solve the problem of tracking cells in the Mother Machine with an error rate of 4.8%, which is only 1.22 times the inter-observer error (see Section 5). Hence our system renders high throughput imaging and tracking of bacteria in the Mother Machine possible.

2 Microscopic Setup and Data Preprocessing

The Mother Machine consists of a main trench and dead end growth channels that host the bacterial cells (see Figure 1). The width of the growth channels is chosen such that each of them fits only a single bacterial cell, thereby forcing the growing cells into a linear array. A constant flow in the main trench leads to continuous diffusion of nutrients and removes cells that emerge from the growth channels. Experiments are imaged by an inverted microscope equipped with an incubator. Images are taken every minute using a 100x objective.

Raw data from the microscope undergoes a few simple preprocessing steps. Two of those are of particular importance. First, movie frames are rotated into an upright orientation, because growth channels are usually tilted by up to $\pm 45^\circ$, see Figure 1(b). To determine the tilt angle, we smooth each image row, collect local maxima, and fit straight lines through each growth channel.

In a second step we correct for uneven background, caused by uneven lighting and different material thicknesses of the Mother Machine itself. For each growth

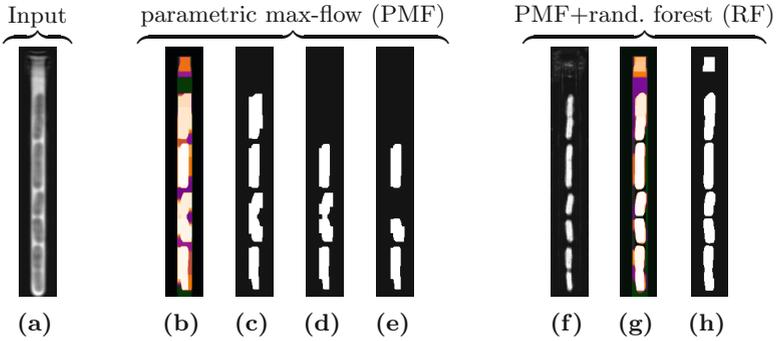


Fig. 2. Parametric max-flow based generation of segmentation hypotheses with and without using a random forest classifier (RF) to modulate unary and binary potentials. (a) Image to be segmented. (b-e) Results obtained using parametric max-flow. (f-h) Results when potentials are modified by a trained RF. (b,g) all graph-cut segmentations given by parametric max-flow. The color of a pixel is determined by the number of times this pixel is classified as foreground. (c,d,e) three graph cut solutions (of 5176). (f) probability map given by RF, trained to over-emphasize gaps between cells. (h) single graph-cut containing the correct segmentation and a false positive at the very top.

line we evaluate the background intensity at each height by averaging the intensities of automatically selected local image patches from within the “empty” areas to either side. This intensity is subtracted from each growth-channel pixel at the given height.

Images for each individual growth-channel are then cropped from the preprocessed image; an example is shown in Figure 2(a).

3 Segmentation Methods

Automated tracking approaches face the challenge that each segmentation error directly translates to at least one tracking error. Assignment models tackle this problem by not committing to a segment for as long as possible. Instead, an excess of potentially conflicting (overlapping) segment hypotheses is created and the model filters the best consistent subset [6]. Below we introduce 3 segmentation methods we use for this purpose.

3.1 Thresholding and Component Trees (CT)

The first segmentation methods we use is an intensity thresholding technique similar to [10,11]. Any threshold yields a binary image from which connected foreground components can be extracted. When the threshold level is gradually raised foreground components grow until they eventually merge. This allows

for grouping all components for all thresholds in a tree data structure, called a *component tree*. Nodes in the component tree, i.e. individual segmented regions rather than a global segmentation, correspond to segment hypotheses.

3.2 Parametric Max-Flow (PMF)

Parametric max-flow [8] is a graph-cut formulation with an additional, additive parameter λ . This parameter linearly scales the unary costs, leading to different segmentation results. The corresponding energy can be formalized as

$$E^\lambda(\mathbf{x}) = \sum_{u \in V} (a_u + \lambda)x_u + \sum_{(u,v) \in E} f_{uv}(x_u, x_v), \quad (1)$$

where \mathbf{x} is a vector of binary variables $x_u \in \{0, 1\}$, f_{uv} 's are hand tuned and submodular, $\lambda \in I \subseteq \mathbb{R}$, and $G = (V, E)$ is an undirected graph, in our case the 4-connected grid graph on the pixels of each frame. Values $x_u = 0$ and $x_u = 1$ represent pixel labels “foreground” (cell) and “background”, respectively. Unary costs a_u for a pixel u depend on measured intensity distributions for foreground and background pixels. Pairwise costs $f_{uv}(x_u, x_v)$ are inversely proportional to the intensity gradient between pixels u and v . More details can e.g. be found in [12]. The work by Kolmogorov et al. [8] offers an efficient way to compute all solutions for $E^\lambda(\mathbf{x})$ for all $\lambda \in \mathbb{R}$, which is a finite and nested set, typically counting between 10 and 10000 solutions.

Like components for increasing threshold values, also the components obtained by increasing λ are monotonically growing. Hence, we can again store all segment hypotheses in a tree. The benefit of PMF over thresholding alone is the additional smoothing that comes with the graph-cut formulation.

3.3 Parametric Max-Flow and Random Forest (PMFRF)

Since missing segments immediately lead to bad tracking performance we combine parametric max-flow and a trained random forest classifier (RFC). This predictor for cell vs. background pixels $P(x_u)$ is trained using the Fiji plugin “Trainable Weka-Segmentation” [13] and manually tuned to pick up even very small clefts between, for example, freshly divided cells. This is done to avoid undersegmentation in cases where the cleft between adjacent cells is not clearly visible (false positives can always be filtered by the model later on, but false negatives translate directly to tracking errors). For the data presented here we trained the RFC on only 3 raw images that were taken from a different raw dataset.

The probability map P for the ‘cell’-class is used to modify the costs a_u and $f_{uv}(x_u, x_v)$ of Equation (1) as follows (see Figure 2 for an illustration):

$$a_u^{\text{trained}} = a_u \cdot P(x_u), \text{ and} \quad (2)$$

$$f_{uv}^{\text{trained}}(x_u, x_v) = f_{uv}(x_u, x_v) \cdot (1 - |P(x_u) - P(x_v)|). \quad (3)$$

4 A Graphical Model for Segmentation and Tracking

We choose the language of factor graphs to describe a model for joint segmentation and tracking in Mother Machine datasets. Here, segmentation consists of selecting a consistent subset of the segment hypotheses $H^{(t)}$ for each time-point. See Figure 3 for an illustration. We use a *factor graph* $\mathcal{FG} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$ with \mathcal{V} being a set of binary variables or *variable nodes*, \mathcal{F} being a set of factors or *factor nodes*, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{F}$ [14].

Variable Nodes. The variable nodes $\mathcal{V} = \mathcal{H} \cup \mathcal{A}$ comprise *segmentation variables* $\mathcal{H} = \bigcup_{t=1}^T H^{(t)}$ and *assignment variables* $\mathcal{A} = \bigcup_{t=1}^{T-1} A^{(t)}$.

Each binary segmentation variable $h^{(t)} \in H^{(t)}$ indicates whether a particular segment hypothesis at time-point t is chosen as part of the solution. Assignment variables $a^{(t)} \in A^{(t)}$ link segment hypotheses at time-point t to segment hypotheses at time-point $t + 1$. We distinguish three types of assignment variables.

Mapping assignments: A mapping assignment $a_{i \rightarrow j}^{(t)}$ connects two segment hypotheses $h_i^{(t)}$ and $h_j^{(t+1)}$. It indicates that these segments correspond to the same segmented cell that is tracked between time-points t and $t + 1$.

Division assignments: A division assignment $a_{i \rightarrow jk}^{(t)}$ connects segment hypothesis $h_i^{(t)}$ to $h_j^{(t+1)}$ and $h_k^{(t+1)}$. It indicates that these segments correspond to a cell division event, where one segmented cell at time-point t divides into two daughter cells at $t + 1$.

Exit assignments: An exit assignment $a_{\perp i}^{(t)}$ is only connected to one segment hypothesis $h_i^{(t)}$. It indicates that this segment corresponds to a segmented cell at time-point t that is spilled out on top of a growth line at time-point $t + 1$.

Factor Nodes. Factor nodes connect to one or more variable nodes, assigning a potential to each joint configuration of these variables. The factor nodes \mathcal{F} comprise unary factors and higher order factors. Unary factors $f(v)$ are connected to each binary variable $v \in \mathcal{V}$, capturing the plausibility that v is active given the data. Formally we define

$$-\ln f(v) = \begin{cases} 0 & \text{if } v = 0 \\ c_v & \text{if } v = 1, \end{cases} \quad (4)$$

c_v is the cost for including the respective segmentation or assignment variable in the solution. These costs are derived from (image) features as described in the next subsection. Structural constraints are expressed as n -ary factors for which $-\ln f(\text{var}(E)) = 0$ if E holds, and ∞ otherwise, where E are (in-)equalities on the set of variables $\text{var}(E)$ connected to the factor. The constraints formalized by these (in-)equalities prohibit solutions involving conflicting segmentation hypotheses or assignments that are inconsistent with the selected segmentation.² Constraints are described in Section 4.2.

² Such inconsistent solutions correspond to events with infinite costs or 0 probability.

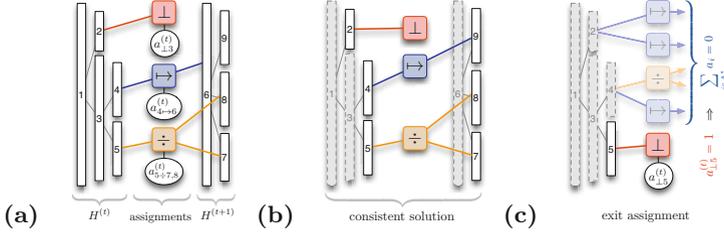


Fig. 3. Overview of the proposed model. Possibly contradictory segmentation hypotheses for two time frames are shown in gray. Between frames, binary assignments variables of three types are enumerated: continuations (blue), divisions (orange), and exits (red). Selecting an assignment variable jointly selects the involved segmentation hypotheses and assigns them to each other.

4.1 Costs

All costs c_v corresponding to activating a variables v are defined according to the following considerations.

We define negative costs for segmentation variables in order to provide an incentive to activate segment hypotheses. Otherwise the trivial solution of ‘seeing’ only empty growth lines, corresponding to a total cost of 0, would be optimal.

We derive segmentation costs from the image intensities along the pixel row at the center of the growth line with the following intuition in mind. A strong gradient on the upper and lower border of a hypothesis increases the likelihood of it being a correct segment and therefore lowers the cost. A strong gradient in the interior of a hypothesis decreases the likelihood (increases the cost) because it suggests that it might contain several cells. Finally, we scale the cost by the size of the segment hypothesis. The rationale for this is that we want to favor hypotheses that explain a larger part of the image in cases where equal support is given by the previously mentioned gradient based measures.

The costs for assignment variables are derived from the positions and sizes of segment hypotheses connected by this assignment. As time progresses from one frame in a given time-lapse movie to the next, we expect an average change in the size and position of a cell.

For mapping assignments we compare the segment sizes and centroids at time points t and $t + 1$. The cost for a mapping assignment is given by a suitably defined function that reflects how unlikely certain deviations from the expected size change and the expected centroid shift really are. This is actually a very natural way of utilizing the knowledge of biological experts.

Costs for division assignments are defined similarly. Here, a segment at time-point t is linked to two (adjacent) segments at $t + 1$. In addition we know that a dividing cell usually distributes its volume equally to its daughters. We compute size and centroid from the union of the two segment hypotheses at $t + 1$ and compute the cost as described for mapping assignments, plus some additional cost for unequally sized segments at $t + 1$.

Last but not least we have to define costs for exit assignments. With the rationale in mind that an early exit assignment already leads to not segmenting this cell in future time-points (thereby not ‘earning’ the corresponding negative cost) we assign 0 cost to all exit assignments.

4.2 Constraints

Tree Constraints. It is important to note that sequential thresholding as well as parametric max-flow respectively yield a monotonic sequence of solutions, inducing a partial order on the segment hypotheses to form a tree $(H^{(t)}, \supset)$.

We say that segment hypotheses $h_i^{(t)} \supset h_j^{(t)}$ are conflicting because they offer mutually exclusive interpretations of (parts of) the image data. Of all segment hypotheses on a branch $h_1^{(t)} \supset \dots \supset h_n^{(t)}$, only one can be simultaneously valid because we seek an assignment of each image pixel to exactly one segment (or background). Tree constraints enforce that conflicting segment variables cannot be simultaneously active. This is formalized in the set of inequalities

$$\forall t \in \{1, \dots, T\}, \forall \pi \in \mathcal{P}(H^{(t)}) : \sum_{h^{(t)} \in \pi} h^{(t)} \leq 1 \quad (5)$$

where $\mathcal{P}(H^{(t)})$ is the set of all paths π from the root node in $(H^{(t)}, \supset)$ to any of its leaf nodes.

Continuity Constraints. Continuity constraints enforce consistency between segmentation and assignment variables. If a segment hypothesis is selected, exactly one of the assignments entering it from the previous time-point, and exactly one of the assignments leaving it towards the next time-point must be selected as well. If a segment hypothesis is not selected, neither must any of these assignments be selected. This is formalized as the following sets of constraints. For the entering assignments we have

$$\forall t \in \{2, \dots, T\}, \forall h^{(t)} \in H^{(t)} : \sum_{a^{(t-1)} \in \Gamma_L(h^{(t)})} a^{(t-1)} = h^{(t)} \quad (6)$$

where the *left neighborhood* $\Gamma_L(h)$ is the set of all assignments entering h from the previous time-point. That is, $\Gamma_L(h_i^{(t)})$ contains assignments $a_{j \rightarrow i}^{(t-1)}$, $a_{j \div ik}^{(t-1)}$, and $a_{j \div ki}^{(t-1)}$ (for all j, k). Similarly, for the assignments leaving to the next time-point we have

$$\forall t \in \{1, \dots, T-1\}, \forall h^{(t)} \in H^{(t)} : \sum_{a^{(t)} \in \Gamma_R(h^{(t)})} a^{(t)} = h^{(t)} \quad (7)$$

where the *right neighborhood* $\Gamma_R(h)$ is the set of all assignments leaving h . That is, $\Gamma_R(h_i^{(t)})$ contains assignments $a_{\perp i}^{(t)}$, $a_{i \rightarrow j}^{(t)}$, and $a_{i \div jk}^{(t)}$ (for all j, k).

Exit Constraints. One of the main contributions of this article is the introduction of this specific type of constraint. It is obvious that cells can only exit the growth line at the very top. A cell in the middle of a growth line can impossibly be spilled out without all other cells above it being spilled out as well. Let us denote by $A_{\uparrow}(h^{(t)}) \subset A^{(t)}$ the set of mapping and division (but not exit) assignments that are leaving hypotheses located strictly above $h^{(t)}$. If the exit assignment is chosen for segment h , then none of the assignments in $A_{\uparrow}(h)$ can be active. (See Figure 3(c) for an illustration.) However, if the exit assignment for h is not chosen, any number of these assignments might be active. We express this as the set of inequalities

$$\forall t \in \{1, \dots, T-1\}, \forall h_i^{(t)} \in H^{(t)} : |H^{(t)}| \cdot a_{\perp i}^{(t)} + \sum_{a \in A_{\uparrow}(h_i^{(t)})} a \leq |H^{(t)}|. \quad (8)$$

Note that, in combination with the continuity constraints (7), this forces all active segments above an exiting hypothesis to exit as well, thereby maintaining the linear order of cells in the mother machine also in our tracking results.

To quantify the importance of exit constraints we removed all exit constraints from our model and tracked all available datasets. We then compared the results to ground truth as explained in Section 5. Error rates increased to 225% (on average to 123%), clearly hinting at the importance of these constraints.

4.3 Eliminating Segmentation Variables

Considering the costs and constraints defined above it can be seen that segmentation variables are redundant in the formulation of the factor graph. The continuity equality (7) provides a definition for each segmentation variable in terms of a sum over a set of assignment variables. Plugging these definitions into (5), and replacing (6) and (7) by

$$\forall t \in \{2, \dots, T-1\}, \forall h^{(t)} \in H^{(t)} : \sum_{a^{(t-1)} \in \Gamma_L(h^{(t)})} a^{(t-1)} - \sum_{a^{(t)} \in \Gamma_R(h^{(t)})} a^{(t)} = 0 \quad (9)$$

we can eliminate segmentation variables from the constraints.³

Similarly, the costs c_h can be dropped, and added to the cost of each exiting assignment c_a , where the constraints guarantee that at most one of these is active.⁴

³ One might fear that by replacing h by a sum over assignment variables might loose the restriction that h is binary. Note, however, that this is now effectively ensured by the tree constraints (5) (with $h^{(t)}$ replaced).

⁴ The costs of segmentation hypotheses $h^{(T)}$, which have no exiting assignments, are added to each entering assignment instead.

4.4 Finding The Globally Optimal Solution

A globally optimal segmentation and tracking is provided by a MAP (maximum *a posteriori* probability) or, equivalently, minimum energy solution of the factor graph. This amounts to finding a conflict-free variable assignment (not violating any constraint) with minimal summed cost.

Similarly to [5,6,7] we formulate the problem as an integer linear program (ILP) [15]: The cost of a conflict-free solution yields the linear objective we wish to minimize⁵. The feasible space is restricted to conflict-free solutions by the linear constraints discussed in Section 4.2 (and additional constraints $0 \leq v \leq 1$ to ensure that all variables $v \in \mathbb{Z}$ are binary). This approach guarantees to find a globally optimal solution, the worst-case complexity is though exponential. In all our experiments we observe runtimes (for ILP solving alone) in the range is a couple of seconds only. See also Figure 5.

We use the off-the-shelf ILP solver Gurobi™ to find the optimal solution.

5 Results

We tested our model on 2 movies containing a total of 21 datasets (growth channels). In order to measure the error of our fully automated tracking pipeline we have manually created ground truth (GT) for all given datasets.

We count (i) *segmentation mismatch*, and (ii) *tracking errors*. For both we greedily match all segments in a given solution with the corresponding segments in the GT. Segmentation mismatch is measured by adding offsets between uppermost pixels and lowermost pixels in each matched segment pair.

The tracking error counts over- and undersegmentations, computed by comparing the number of active segments at any given time-point in solution and the GT, and assignment-type mismatches. For those we count type-mismatches for all right-assignments (assignments towards next time-point) associated to pairs of matched segments. Note that this is a fairly pessimistic measure where errors that would intuitively be counted as one mistake are counted multiple times⁶.

Figure 4 shows the results of the ground truth comparison. The first three columns in each box-plot show how the fully automated solutions compare to GT. Each column corresponds to one of the segmentation methods introduced in Section 3. The last column shows an inter-observer reliability measure.

The inter-observer reliability tells us about how much homogeneity, or consensus, there is to expect when different users create “ground truth” for the same data. We gave the automatically generated PMFRF solution and a interactive tool to 2 users, asking them to fix all errors. We then compared their results to GT in the same way we described above. See Figure 5 for a detailed comparison of runtimes for the fully automated pipeline.

⁵ It is easily seen that the summed cost is a linear function by writing it as the inner product of the vectors of all binary variables and costs, $\langle (v_1, \dots, v_n), (c_{v_1}, \dots, c_{v_n}) \rangle$.

⁶ An early exit assignment is once counted as assignment-type mismatch and in all future time-points still containing this cell as undersegmentations.

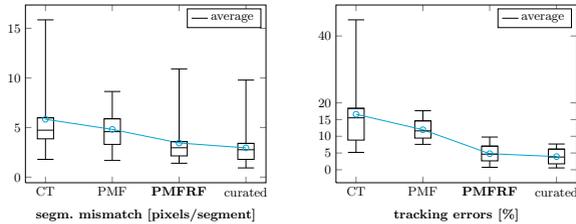


Fig. 4. Error measures for all 21 datasets. (Abbr.: CT→‘component tree’; PMF→‘parametrix max-flow’; PMFRF→PMF+trained random forest.) Left panel shows how well the chosen segments match to ground truth. We compare the pixel distance between the uppermost and lowermost segmented pixels between each segment and its corresponding ground truth segment. The right panel shows the fraction of assignments that do not match to ground truth.

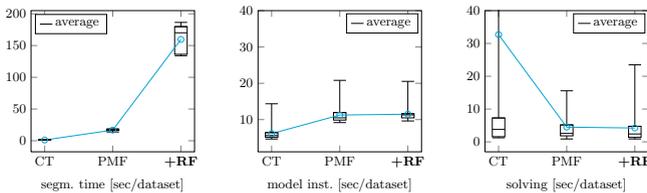


Fig. 5. Runtime for segmentation, model instantiation, and model solving. Shown times are in ‘wall-time’ seconds per dataset. We used a quadcore MacBook Pro Retina (Fall 2012). An excessive filter bank is main reason for slow RFs.

6 Summary and Discussion

We showed how cell tracking in the Mother Machine can be addressed using an adequately formulated assignment model. In order to achieve low error rates we needed to extend existing models [5,6,7] by additional constraints concerned with the linear order of cells in the Mother Machine and a specialized method to create nested segment hypotheses using a parametric max-flow formulation and trained random forests classifiers. Automated tracking and segmentation quality reaches a level that lies within a factor of 1.1 compared to the inter-observer variability we measured. Our system will be freely available open source software, enabling groups around the world to analyze cell cultured in the Mother Machine.

With this paper we contribute to a recent trend of formulating tracking problems as global optimization problems in the spirit of graphical models. We predict that the capabilities of assignment models is by far not reached yet.

Future extensions will focus on several important aspects such as (i) further increasing the set of segment hypotheses, thereby generalizing the concept of conflict trees to more general conflict graphs, (ii) development of more generic and task specific higher order factors that will capture ever more expert domain knowledge and therefore lead to better automated results, (iii) parametrization and parameter training of used cost functions, for example by means of

structured learning, and (*iv*) alternative solving strategies, either by means of divide-and-conquer like dual decomposition schemes or, means of approximate inference methods, or suitable combinations.

The last mentioned point will become increasingly important with growing problem instances and the need for interactive proofreading and data curation interfaces.

Acknowledgments. This work was supported by the German Federal Ministry of Research and Education (BMBF) under the funding code 031A099.

References

1. Wang, P., Robert, L., Pelletier, J., Dang, W., Taddei, F., Wright, A., Jun, S.: Robust growth of *E. coli*. *Current Biology* 20(12), 1099–1103 (2010)
2. Jug, F., Pietzsch, T., Preibisch, S., Tomancak, P.: Bioimage informatics in the context of *Drosophila* research. *Methods* (2014)
3. Padfield, D., Rittscher, J., Roysam, B.: Coupled Minimum-Cost Flow Cell Tracking. In: Prince, J.L., Pham, D.L., Myers, K.J. (eds.) *IPMI 2009*. LNCS, vol. 5636, pp. 374–385. Springer, Heidelberg (2009)
4. Padfield, D., Rittscher, J., Roysam, B.: Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical Image Analysis* 15(4), 650–668 (2011)
5. Kausler, B.X., et al.: A Discrete Chain Graph Model for 3d+t Cell Tracking with High Misdetection Robustness. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012, Part III*. LNCS, vol. 7574, pp. 144–157. Springer, Heidelberg (2012)
6. Funke, J., Anders, B., Hamprecht, F., Cardona, A., Cook, M.: Efficient automatic 3D-reconstruction of branching neurons from EM data. In: *CVPR. IEEE* (2012)
7. Schiegg, M., Hanslovsky, P., Kausler, B., Hufnagel, L.: Conservation Tracking. In: *ICCV* (2013)
8. Kolmogorov, V., Boykov, Y., Rother, C.: Applications of parametric maxflow in computer vision. In: *ICCV*, pp. 1–8. *IEEE* (2007)
9. Breiman, L.: Random Forests. *Machine Learning* 45(1), 5–32 (2001)
10. Jones, R.: Component trees for image filtering and segmentation. In: *IEEE Workshop on Nonlinear Signal and Image Analysis* (1997)
11. Nistér, D., Stewénius, H.: Linear Time Maximally Stable Extremal Regions. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 183–196. Springer, Heidelberg (2008)
12. Blake, A., Kohli, P., Rother, C.: *Markov Random Fields for Vision and Image Processing*. MIT Press (2011)
13. Arganda-Carreras, I., Cardona, A., Kaynig, V., Schindelin, J.: Trainable weka segmentation (May 2011), http://fiji.sc/Trainable_Weka_Segmentation
14. Frey, B., Kschischang, F., Loeliger, H., Wiberg, N.: Factor graphs and algorithms. In: *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 35, pp. 666–680 (1997)
15. Schrijver, A.: *Theory of Linear and Integer Programming*. J. Wiley & Sons (1998)