

Cell Lineage Tracing in Lens-free Microscopy Videos

Markus Rempfler^{1,2}, Sanjeev Kumar², Valentin Stierle³, Philipp Paulitschke³,
Bjoern Andres⁴, and Bjoern H. Menze^{1,2}

¹ Institute for Advanced Study, Technical University of Munich, Germany

² Department of Informatics, Technical University of Munich, Germany

³ Faculty of Physics, Ludwig-Maximilians University of Munich, Germany

⁴ Max Planck Institute for Informatics, Saarbrücken, Germany

Abstract. In vitro experiments with cell cultures are essential for studying growth and migration behaviour and thus, for gaining a better understanding of cancer progression and its treatment. While recent progress in lens-free microscopy (LFM) has rendered it an inexpensive tool for continuous monitoring of these experiments, there is only little work on analysing such time-lapse sequences.

We propose (1) a cell detector for LFM images based on residual learning, and (2) a probabilistic model based on *moral lineage tracing* that explicitly handles multiple detections and temporal successor hypotheses by clustering and tracking simultaneously. (3) We benchmark our method on several hours of LFM time-lapse sequences in terms of detection and tracking scores. Finally, (4) we demonstrate its effectiveness for quantifying cell population dynamics.

1 Introduction

Cell growth and migration play key roles in cancer progression: abnormal cell growth can lead to formation of tumors and cancer cells can spread to other parts of the body, a process known as metastasis. In vitro experiments are essential to understand these mechanisms and for developing anti-cancer drugs. In these experiments, the cells are typically observed with conventional light microscopes. Thanks to recent advances in CMOS sensor technology, lens-free microscopy (LFM) [13,4] has become a promising alternative. In LFM a part of the incident wavefront originating from the light source is scattered by the sample, in this case the cell. The scattered light then interferes with the unscattered part of the wavefront and the resulting interference pattern is recorded with a CMOS sensor. The components required for LFM are extremely small and cheap. Thus, LFM provides the means for a wide range of applications where a conventional light microscope would be either too big or simply too expensive, such as the continuous monitoring of growing cell cultures inside standard incubators [9].

To quantify the clinically relevant information on cell growth and migration from the large amount of images that are acquired in such continuous monitoring, reliable automatic image analysis methods are crucial. Counting the number of

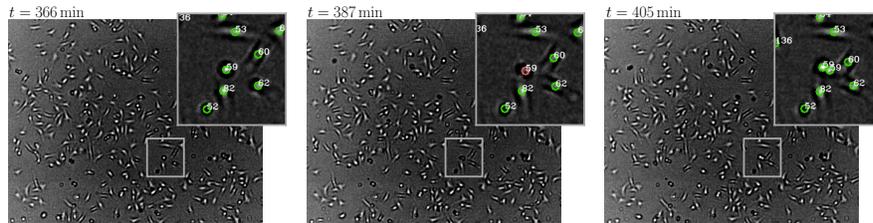


Fig. 1. The cell lineage tracing problem with LFM data. We aim to detect all cells and establish their relation over time, i.e. determine the lineage forest. While the LFM technology allows for frequent image acquisition (3 min / frame in this case), challenges arise due to overlapping interference patterns of close objects, fluctuating shape and size of the cells appearance, and particles that generate similar patterns as the cells. The detail views show cell locations as a circle and identify their lineage tree.

cells in a time series of images gives access to the dynamics of cell growth. Locating and tracing individual cells provides information about cell motility, and over the course of a sequence, reconstructing the lineage trees gives insights into cell cycle timings and allows more selective analysis of cell sub-cultures.

There are several works on these tasks in traditional light microscopy, e.g. focussing on cell segmentation [15], detection and counting [8,10,16] or tracking [1,6,7,12,14], but very few deal with LFM data. One of the few exceptions is [3] which employs a regression framework for estimating the total cell count per image. We aim at the more complex goal of not only counting but also localizing cells and reconstructing their spatio-temporal lineage forest (c.f. Fig. 1). Methods for the latter task range from Kalman filtering [1] to keep track of moving cells, or iteratively composing tracklets by using the Viterbi algorithm [11], and have been compared in [12]. More recently, Jug et al. [7] have proposed a mathematically rigorous framework for lineage reconstruction, the so-called moral lineage tracing problem (MLTP). The MLTP differs fundamentally from all mathematical abstractions of cell tracking whose feasible solutions are either disjoint paths or disjoint trees of detections. Unlike these approaches that select only one detection for each cell in every image, feasible solutions of the MLTP select and cluster an arbitrary set of such detections for each cell. This renders the lineage trees defined by feasible solutions of the MLTP robust to the addition of redundant detections, a property we will exploit in this work.

In this paper, we contribute a framework for analysis of LFM sequences. First, we design and benchmark robust cell detectors for LFM time-lapse sequences derived from most recent work on convolutional neural networks and residual learning. Second, we discuss the MLTP in the context of LFM data. In particular, we define a probability measure for which the MLTP is a maximum a posteriori (MAP) estimator. This allows us to define the costs in the objective function of the MLTP w.r.t. probabilities that we estimate from image data. We validate it experimentally on two annotated sequences. Finally, we demonstrate

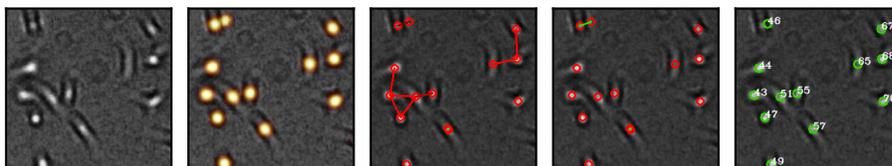


Fig. 2. Illustration of our workflow. From left to right: 1) Raw microscopy image, 2) image overlaid with cell probability map generated by the detector, 3) nodes of the hypothesis graph with spatial edges constructed from cell probabilities, 4) optimized lineage where spatial edges that were cut are removed, and 5) each cluster is represented as one cell with its lineage tree identifier. Temporal edges are not depicted for simplicity.

the capability of our approach to quantify biologically relevant parameters from sequences of two in vitro experiments with skin cancer cells.

2 Methods

We consider the lineage tracing task as a MAP inference over a hypothesis graph containing a multitude of potential lineage forests. We discuss the probability measure and its MAP estimator, the MLTP in Sec. 2.1. In order to construct the hypothesis graph from a sequence of LFM images, we devise a cell detector in Sec. 2.2, which estimates a cell probability map for each given image. The workflow is illustrated in Fig. 2.

2.1 Lineage Tracing

Hypothesis graph. We construct a spatio-temporal hypothesis graph $G = (V, E)$ as follows: For every image I_t in the sequence, we apply a cell detector and define one node $v \in V_t$ for every local maximum in $P(c_s = 1|I_t)$, the estimated probability map for finding a cell at a particular location s in image I_t . Additionally, we define *hypothesized* successors to each node that has one or more favourable parents in the previous frame but no immediate successor. This helps avoiding gaps in the final tracklets. The nodes $v \in V$ represent cells, yet do not need to be unique, i.e. one cell may give rise to several nodes. We then construct edges in space $E_t^{\text{sp}} = \{uv \in V_t \times V_t : d(u, v) \leq d_{\text{max}}\}$, i.e. between any two nodes that lie within a distance of d_{max} , and in the same fashion, we construct temporal edges $E_t^{\text{tmp}} = \{uv \in V_t \times V_{t+1} : d(u, v) \leq d_{\text{max}}\}$ between nodes in adjacent frames.

Probabilistic model. We introduce a family of probability measures, each defining a conditional probability of any lineage forest, given an image sequence. We describe the learning of this probability from a training set of annotated image sequences as well as the inference of a maximally probable lineage forest, given a previously unseen image sequence. The resulting MAP estimation problem will assume the form of an MLTP with probabilistically justified costs.

First, we encode subgraphs in terms of cut edges with binary indicator variables $\mathbf{x} \in \{0, 1\}^E$. If edge uv is cut, i.e. $x_{uv} = 1$, it means that nodes u and v do *not* belong together. In order to ensure that the solution describes a lineage forest, we rely on the formulation of the MLTP [7], which describes the set of inequalities that are required to do so. In short, these constraints ensure: 1) *spatial* and *temporal consistency*, i.e. if nodes u and v as well as v and w belong together, then u and w must also belong together. 2) Distinct tracklets cannot merge at a later point in time. These are the so called *morality* constraints. 3) *Bifurcation* constraints allow cells to split in no more than two distinct successors. We will denote the set of \mathbf{x} that describe valid lineage forests with X_G . For a more extensive discussion of these constraints, we refer to [7,14]. We next model the measure of probability:

$$P(\mathbf{x}|X_G, \Theta) \propto P(X_G|\mathbf{x}) \prod_{uv \in E} P(x_{uv}|\Theta) \prod_{v \in V} P(x_v^+|\Theta) \prod_{v \in V} P(x_v^-|\Theta), \quad (1)$$

$$\text{where } P(X_G|\mathbf{x}) \propto \begin{cases} 1 & \text{if } \mathbf{x} \in X_G, \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

It is comprised of four parts. First, we have $P(X_G|\mathbf{x})$ representing the uniform prior over X_G . Second, the cut probability $P(x_{uv}|\Theta)$ describing the probability of u and v being part of the same cell (either in space or along time), and third and fourth, the birth and termination probabilities $P(x_v^+|\Theta)$ and $P(x_v^-|\Theta)$ for each node $v \in V$. The variables $x_v^+, x_v^- \in \{0, 1\}$ are indicating whether the respective event, birth or termination, occurs at node v . Θ denotes the joint set of parameters. We use these parts to incorporate the following assumptions: Two detections u and v that are close are more likely to originate from the same cell, hence we choose $P(x_{uv} = 1|\Theta) = \min(\frac{d(u,v)}{\theta^{\text{sp}}}, 1)$. Similarly, two successive detections u at t and v at $t+1$ are more likely to be related the closer they are, is captured by $P(x_{uv} = 1|\Theta) = \min(\frac{d(u,v)}{\theta^{\text{tmp}}}, 1)$. Finally, we assume that birth and termination events occur at a low rate, which is incorporated by $P(x_v^+ = 1|\Theta) = \theta^+$ and $P(x_v^- = 1|\Theta) = \theta^-$. We fit these parameters Θ on training data in a maximum likelihood fashion: For θ^- and θ^+ this boils down to calculating the relative frequency of the respective events on the annotated lineage. For the spatial and temporal parameters θ^{sp} and θ^{tmp} , we first complement the lineage forest with edges within d_{max} as \mathcal{E} . We then maximize the log-likelihood $\log \mathcal{L}(\theta) = \sum_{uv \in \mathcal{E}} \log P(x_{uv}|\theta)$ by an extensive search over the interval $\theta \in [\theta_{\text{min}}, \theta_{\text{max}}]$, where we found [1, 80] to be appropriate.

The MAP estimate $\mathbf{x}^* = \arg \max_{\mathbf{x} \in X} P(\mathbf{x}|\Theta, X_G)$ can be written as solution to the MLTP:

$$\min \left\{ \sum_{uv \in E} c_{uv} x_{uv} + \sum_{v \in V} c_v^+ x_v^+ + \sum_{v \in V} c_v^- x_v^- \mid \mathbf{x} \in X_G \cap X_V \right\}, \quad (3)$$

where the coefficients become $c_{uv} = -\log \frac{P(x_{uv}=1|\Theta)}{1-P(x_{uv}=1|\Theta)}$ for edges, and vice versa for c_v of the node events. X_V is the set of \mathbf{x} that satisfy the auxiliary constraints

which tie birth and termination indicator variables x_v^- and x_v^+ to the respective edge variables. We optimize (3) with the KLB algorithm described in [14].

2.2 Cell Detection with Residual Networks

Cells in LFM images are usually only marked at their center of mass and not segmented since their interference pattern, i.e. their appearance in the image, does not accurately describe their true shape and would therefore be ambiguous in many cases. Thus, we are interested in a detector that outputs the set of cell centers in image I_t . Strong performance of the detector is crucial for the lineage reconstruction as its errors can affect the final lineage trees over many frames. To achieve this, we build on the recent work on residual networks [5]. However, instead of directly regressing bounding boxes or center coordinates in a sliding window fashion, we train our network, denoted with $f(I_t)$, on a surrogate task: We approximate $f(I_t) \approx P(c_s = 1|I_t)$, the probability map of finding a cell at a particular location s in I_t . This detector is fully convolutional and its output $f(I_t)$ has the same size as I_t . We found this to facilitate the training as it enlargens the spatial support of the sparse cell center annotations and gracefully handles the strongly varying cell density. Similar findings were made with techniques that learn a distance transform to detect cells, e.g. in [8]. We describe next how we arrive at a suitable architecture for this task and how to construct $P(c_s = 1|I_t)$ from point-wise cell annotations.

Network Architecture. We start from the architecture of *ResNet-50* [5]. We first truncate the network at layer 24 to obtain a fully convolutional detector. We found that truncating in the middle of the original ResNet-50, i.e. at layer 24, results in best resolution of the output response maps and allows to distinguish close cells. We then add one convolutional layer of $1 \times 1 \times 256$ and one up-convolutional layer (also known as deconvolutional layer) of $8 \times 8 \times 1$ with a stride of 8. The former combines all feature channels, while the latter compensates for previous pooling operations and ensures that the predicted cell probability map has the same resolution as the input image I_t . Finally, a sigmoid activation function is used in the last layer to ensure that $f(I_t)$ is within the interval $[0, 1]$ at any point.

Loss Function & Training. We sample training images of size 224×224 from all frames of the training corpus. For each training image I_k , we construct a corresponding cell probability map $P(c_s = 1|I_k)$ by placing a Gaussian kernel G^σ with $\sigma = 8$ at each annotated center. This implicitly represents the assumption that all cells have about the same extent, which is reasonable for our microscopy data. Each image is normalized to zero mean and unit variance. During training, we minimize the cross entropy loss between the predicted map $f(I_t)$ and $P(c_s|I_t)$ in order to let our network approximate the constructed cell probability map. We fine tune the network (pre-trained weights from ResNet-50 [5]) with a learning rate of 10^{-3} for 100 epochs with batch size of 8. In each epoch, we sample 4000 training images. Since the annotated dataset for training is typically small and shows strong correlation between cells in consecutive frames, we used dropout of 0.5 after the last convolutional layer to avoid overfitting.

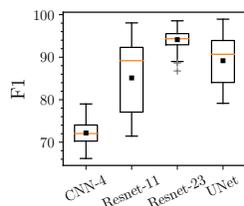


Fig. 3. Performance of different detectors over all test frames. Boxplots depict median as orange line, mean as black square and outliers as grey +. The F1 scores are shown in %. We find that ResNet-23 is the most robust detector in our experiment with an average F1 of 94.1 %. It is followed by the UNet with 89.2%, ResNet-11 with 85.1% and finally, CNN-4 with only 72.2%.

3 Experiments & Results

Datasets. We use a sequence depicting A549 cells, annotated over 250 frames in a region of interest (ROI) of 1295×971 px, for all training purposes. For testing, we annotated two distinct sequences monitoring 3T3 cells of 350 and 300 frames in a ROI of 639×511 px (*3T3-I*) and 1051×801 px (*3T3-II*), respectively. Images were acquired at an interval of 3 min with $1.4 \mu\text{m} \times 1.4 \mu\text{m}$ per pixel.

Benchmarking detectors. We compare four different network configurations, including the described ResNet-23, ResNet-11, a variant of it which was truncated at layer 11, the UNet [15] and CNN-4. In UNet, we obtained better results when replacing the stacks in the expansive path with single up-convolution layers which are merged with the corresponding feature maps from contracting path. CNN-4 is a plain vanilla CNN with three 5×5 convolutional layers followed by max pooling and finally, one up-convolutional layer of $8 \times 8 \times 1$ to compensate for the down-sampling operations. We use the same training procedure (Sec. 2.2) for all detectors, but adjust the learning rate for UNet and CNN-4 to 10^{-2} .

We match annotated cells to detections within each frame with the hungarian algorithm and consider only matches closer than 10 px (\approx a cell center region) as a true positive (TP). Unmatched annotations are counted as false negative (FN), unmatched detections as false positive (FP). The results are presented in Fig. 3, where we find the ResNet-23 to be the most robust detector.

Lineage tracing. To compare the quality of different lineages, we match again annotations and detections within each frame to calculate the number of TP, FP and FN as described before. We then determine the number of false links, i.e. how often two matched nodes do not have the same parent. From these, we calculate multiple object detection accuracy (MODA) and multiple object tracking accuracy (MOTA) [2]. Moreover, we derive the number of edit operations needed to get from the predicted lineage to the ground truth lineage, and calculate the tracking accuracy (TRA) score proposed in [12]. We use unit weight for each type of edit (add or delete node or edge). This is justified by the fact that we have point annotations for cells instead of segmentations, making both addition and deletion equally expensive to correct.

For the MLTP, we compare the effect of varying θ^{tmp} , θ^{sp} together with hypothesis graphs generated from the different detectors in Fig. 4. The optimal parameter choice for ResNet-23 is at 10, i.e. a relatively small merge radius of favourable merges, while the other detectors considerably benefit from wider ranges. In Table 1, we compare different lineage tracing approaches. Our

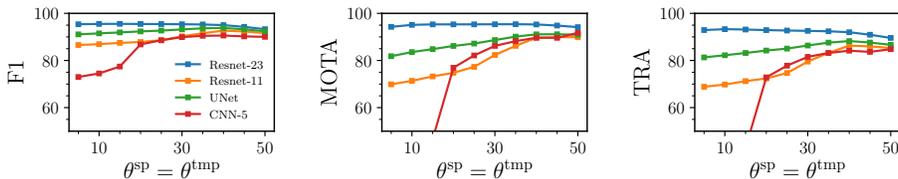


Fig. 4. Sensitivity analysis of the lineage tracing model with different detectors. We increase both edge cut parameters θ^{tmp} and θ^{sp} together. While the optimal choice in combination with ResNet-23 is relatively small, i.e. at 10, the other detectors, which suffer from many spurious detections, benefit from a wider range. Most notably, the performance with CNN-4 improves up to a competitive TRA of 84.8%.

Table 1. Quantitative evaluation of traced lineages. Precision, recall, F1 and MODA are averaged over all frames of a sequence, while MOTA and TRA are global scores for the entire lineage forest. All measures are in %. Disjoint trees (DTP) uses our ResNet-23 detections and is equivalent to MLTP with disabled spatial edges and no hypothesized successors. LAPT is linear assignment problem tracking [6] and our baseline.

Instance	Method	Precision	Recall	F1	MODA	MOTA	TRA
3T3-I	LAPT	86.39	88.99	87.63	85.88	83.87	80.46
	DTP	93.67	92.84	93.22	93.67	90.22	87.11
	MLTP	97.09	93.19	95.07	97.18	95.67	92.58
3T3-II	LAPT	85.12	87.35	86.19	84.68	82.65	79.13
	DTP	94.02	95.89	94.93	93.85	91.49	89.87
	MLTP	96.46	96.12	96.28	96.45	95.43	93.76

baseline is linear assignment problem tracking (LAPT) [6]. The disjoint trees method (DTP), uses our ResNet-23 detections but solves the disjoint trees problem instead, i.e. it considers only one detection per cell. We find that MLTP outperforms both in terms of detection and tracking metrics.

Assessing cell population dynamics. We apply our method on data from two experiments with skin cancer cells. In each, one population is exposed to an inhibitor substance while the control is not. Figure 5 depicts the resulting statistics. We observe the expected difference in growth rate, yet a more constrained motility of the control cells, which is caused by the limited space.

4 Conclusions

We have presented a framework for automatic analysis of LFM time-lapse sequences. It transfers two recently proposed methods, residual learning [5] and moral lineage tracing [7,14], to the task at hand. We have shown experimentally that it is able to determine cell lineage forests of high quality and thereby quantify several measures of interest for the analysis of in vitro experiments, including cell population dynamics and motility.

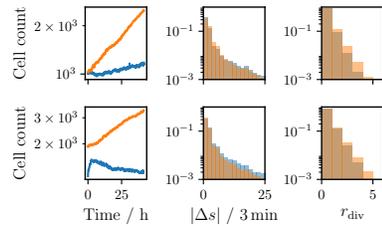


Fig. 5. Cell dynamics measured on two experiments with skin cancer cell lines. One population (blue) is exposed to an inhibitor substance, while the other (orange) is not. From left to right: Cell count over time, histograms on cell motility ($\mu\text{m}/3 \text{ min}$) and divisions r_{div}/h . Cells that divide often are more abundant in the control group.

Acknowledgements. With the support of the Technische Universität München – Institute for Advanced Study, funded by the German Excellence Initiative (and the European Union Seventh Framework Programme under grant agreement n 291763).

References

1. Arbellet, A., et al.: Analysis of high-throughput microscopy videos: Catching up with cell dynamics. In: MICCAI. pp. 218–225. (2015)
2. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP JIVP(1), 1–10 (2008)
3. Flaccavento, G., et al.: Learning to count cells: applications to lens-free imaging of large fields. Microscopic Image Analysis with Applications in Biology (2011)
4. Greenbaum, A., et al.: Imaging without lenses: achievements and remaining challenges of wide-field on-chip microscopy. Nature methods 9(9), 889–895 (2012)
5. He, K., et al.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
6. Jaqaman, K., et al.: Robust single-particle tracking in live-cell time-lapse sequences. Nature methods 5(8), 695–702 (2008)
7. Jug, F., et al.: Moral lineage tracing. In: CVPR pp. 5926–5935 (2016)
8. Kainz, P., et al.: You should use regression to detect cells. In: MICCAI. pp. 276–283 (2015)
9. Kesavan, S.V., et al.: High-throughput monitoring of major cell functions by means of lensfree video microscopy. Scientific reports 4, 5942 (2014)
10. Khan, A., et al.: Deep convolutional neural networks for human embryonic cell counting. In: ECCV. pp. 339–348. (2016)
11. Magnusson, K.E.G., Jaldén, J.: A batch algorithm using iterative application of the viterbi algorithm to track cells and construct cell lineages. In: ISBI. (2012)
12. Maška, M., et al.: A benchmark for comparison of cell tracking algorithms. Bioinformatics 30(11), 1609–1617 (2014)
13. Mudanyali, O., et al.: Compact, light-weight and cost-effective microscope based on lensless incoherent holography for telemedicine applications. Lab on a Chip 10(11), 1417–1428 (2010)
14. Rempfler, M., et al.: Efficient algorithms for moral lineage tracing. arXiv:1702.04111 (2017)
15. Ronneberger, O., et al.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. (2015)
16. Xie, W., et al.: Microscopy cell counting with fully convolutional regression networks. In: MICCAI 1st WDLMIA (2015)