

# Tracing Cell Lineages in Videos of Lens-free Microscopy

Markus Rempfler<sup>a,b,\*</sup>, Valentin Stierle<sup>c</sup>, Konstantin Ditzel<sup>c</sup>, Sanjeev Kumar<sup>a</sup>,  
Philipp Paulitschke<sup>c</sup>, Bjoern Andres<sup>d</sup>, Bjoern H. Menze<sup>a,b</sup>

<sup>a</sup>*Department of Informatics, Technical University of Munich, Germany*

<sup>b</sup>*Institute for Advanced Study, Technical University of Munich, Germany*

<sup>c</sup>*Faculty of Physics, Ludwig-Maximilians University of Munich, Germany*

<sup>d</sup>*Max Planck Institute for Informatics, Saarbrücken, Germany*

---

## Abstract

In vitro experiments with cultured cells are essential for studying their growth and migration pattern and thus, for gaining a better understanding of cancer progression and its treatment. Recent progress in lens-free microscopy (LFM) has rendered it an inexpensive tool for label-free, continuous live cell imaging, yet there is only little work on analysing such time-lapse image sequences.

We propose (1) a cell detector for LFM images based on fully convolutional networks and 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 in terms of detection and tracking scores on a dataset of three annotated sequences of several hours of LFM, where we demonstrate our method to produce high quality lineages. (4) We evaluate its performance on a somewhat more challenging problem: estimating cell lineages from the LFM sequence as would be possible from a corresponding fluorescence microscopy sequence. We present experiments on 16 LFM sequences for which we acquired fluorescence microscopy in parallel and generated annotations from them. Finally, (5) we showcase our methods effectiveness for quantifying cell dynamics in an experiment with skin cancer cells.

*Keywords:* Cell detection, Cell lineage tracing, Lens-free microscopy, Fully

---

\*Corresponding author

## 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 order to understand these mechanisms, in vitro experiments are essential. Such experiments allow, for example, to compare the behaviour of modified and wildtype cell lines or measure the influence of certain chemicals on the cells culture, and can therefore be designed to gain insights into biological processes. More fields of biology that rely on such in-vitro experiments with cultured cells include the study of embryonic development [1, 2] or tissue formation [3]. Typically, the cells are observed with conventional light microscopes, but thanks to recent advances in CMOS sensor technology, lens-free microscopy (LFM) [4, 5] has become a promising alternative to the traditional light microscope for live cell imaging. 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. Since the components required for building an LFM are both small and inexpensive, LFM provides the means for a wide range of applications where a conventional light microscope would be either too big or too expensive. Continuous monitoring of growing cell cultures inside standard incubators can be considered such an application scenario [6]. Another advantage of LFM for this purpose is that it is a label-free method and thus, does not suffer from effects like photobleaching (limiting temporal resolution and length of the sequence) or difficulties with markers that alter the cell behaviour.

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. By detecting and counting individual cells in each frame of the microscopy video, we gain infor-

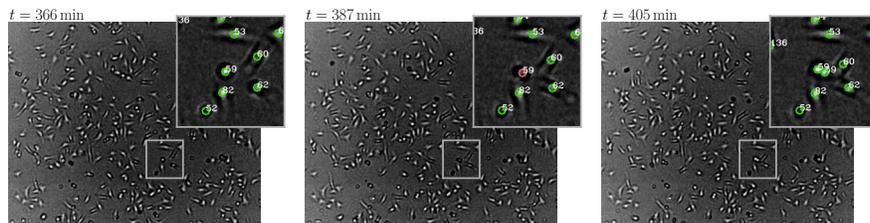
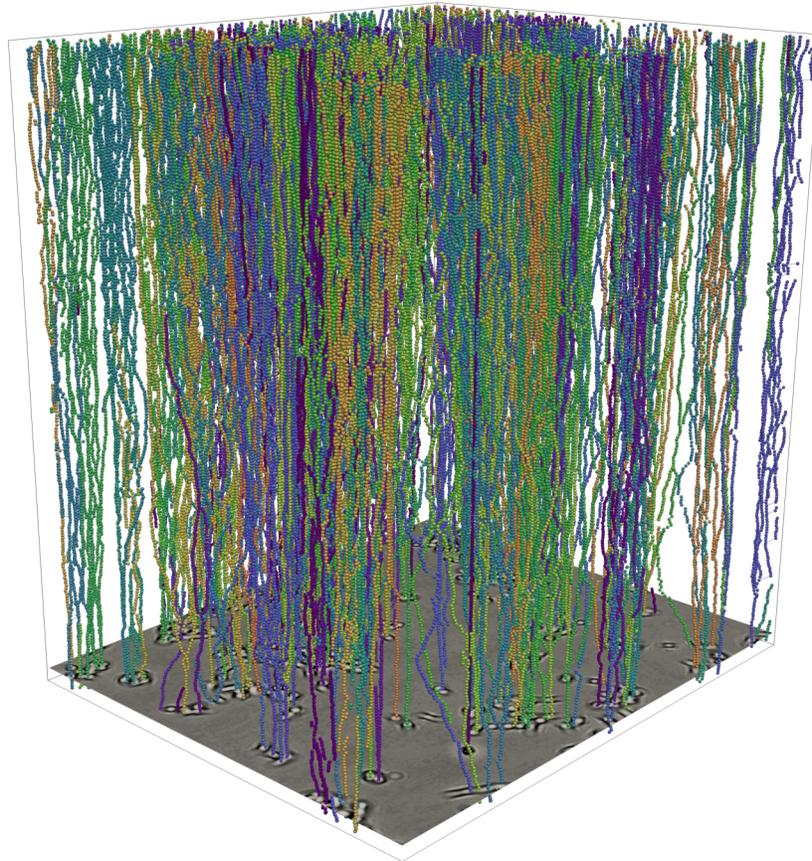


Figure 1: The cell lineage tracing problem on LFM videos. We aim to detect all cells and establish their relation over time, i.e. determine the lineage forest. A visualisation of a cell lineage forest is given in the top of the figure. Individual trees are depicted with one color. While the LFM technology 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 bottom row shows three images from an LFM sequence with detailed views indicating cell locations with a circle and identifying their lineage tree.

mation on the dynamics of the cell growth. Locating and tracing individual  
30 cells along time provides information about cell motility. Reconstructing the  
*lineage trees* over the course of a sequence, gives insights into cell cycle timings  
and allows more selective analysis of cell sub-cultures.

These tasks have received considerable attention in traditional light mi-  
croscopy [7, 8, 9, 10, 11, 12, 13, 14], focussing on certain aspects such as cell seg-  
35 mentation [15], detection and counting [16, 17, 18] or tracking [19, 7, 20, 12, 21].  
However, very few deal with LFM data. One of the few exceptions is [22] which  
employs a regression framework for estimating the total cell count per image.  
In this work, we not only count but also localize cells and trace their spatio-  
temporal behaviour and descendance, i.e. we estimate the underlying cell lineage  
40 trees (cf. Fig. 1). Methods for this task on traditional microscopy data include,  
for example, Kalman filtering [19] to keep track of moving cells, or iteratively  
composing tracklets by using the Viterbi algorithm [23], and have been com-  
pared in [12, 10, 14]. Typically, they formulate their solution as either disjoint  
paths or disjoint trees of detections or segmentations of cells. More recently,  
45 Jug et al. [20] 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,  
50 feasible solutions of the MLTP select and cluster an arbitrary set of such detec-  
tions 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.

*Contributions of this work.* We contribute a framework for analysis of LFM  
55 sequences. (1) We design and benchmark robust cell detectors for LFM time-  
lapse sequences derived from recent work on fully convolutional neural net-  
works [24, 15] and residual learning [25]. (2) 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

60 the costs in the objective function of the MLTP with respect to probabilities  
that we estimate from image data. (3) We validate it experimentally on three  
manually annotated sequences. This work extends our previous conference pa-  
per [26] with an enhanced discussion and technical description of the method,  
and a more comprehensive experimental evaluation. In particular, (4) we con-  
65 tribute experiments on an additional dataset consisting of 16 sequences of three  
different cell lines and varying cell culture densities. Other than our previ-  
ous dataset, we have acquired fluorescence microscopy videos in parallel to the  
LFM and generated ground truth annotations directly from the fluorescence mi-  
croscopy, where the cell nuclei were labelled with a fluorescent dye (cf. Fig. 6 for  
70 an illustration of both microscopy types). The purpose of this dual microscopy  
setup is a validation experiment. It not only allows quantitative experiments  
when LFM images would be difficult to annotate, e.g. in the presence of dense  
cell clusters or when there are simply too many. Further, it enables us to evalu-  
ate our method on a more challenging problem: We estimate cell lineages solely  
75 from LFM (which has no need for a fluorescent labeling) and compare them  
to those reconstructed from a corresponding fluorescence microscopy sequence.  
Finally, (5) we demonstrate the capability of our approach to quantify biologi-  
cally relevant parameters from LFM sequences of two in vitro experiments with  
skin cancer cells.

## 80 **2. Background**

The task of cell lineage tracing is commonly considered a two-stage problem.  
First, individual cells are detected and/or segmented in every image. Second,  
each individual cell is linked over time to either itself or its ancestor cell, in  
case of a cell division. Depending on the microscopy modality and setup, the  
85 task can be complicated by various issues: low spatial resolution makes accurate  
detection and distinction of individual cells difficult. Low temporal resolution  
allows large displacements within one time step and cells entering or leaving the  
field of view render tracking along time difficult. Furthermore, mistakes in the

detection stage influence the resulting lineage forest, often leading to spurious  
90 divisions or missing branches. In the context of LFM data, challenges arise in  
particular due to overlapping interference patterns of close objects, fluctuating  
appearance in terms of shape and size of the cells appearance, and particles that  
generate similar patterns as the cells. Especially objects that agglomerate and  
remain close over an extended period of time can be difficult to distinguish in  
95 the LFM image, even for a human observer.

Cell lineage tracing is related to multiple-object tracking [27, 28, 29, 30,  
31, 32], where the tracked objects do not split and the result is a collection of  
trajectories, or the reconstruction of tree- and network-like structures [33, 34, 35,  
36, 37], where the object of interest is delineated in the spatial domain. Similar  
100 to several methods for these tasks, many works formulate cell lineage tracing as  
an optimization problem [38, 39, 23, 40, 41, 42, 43], dealing with some of the  
mentioned difficulties by using spatio-temporal information across the sequence.  
They formulate the optimization, for example, as a dynamic program [23], an  
assignment problem [7], a network flow [40] or, more general, by using integer  
105 programming [43, 41]. Another class of methods, sometimes categorized as  
*tracking by model evaluation* [12], define a recursive tracking model that uses  
outcome of frame  $t_n$  as initial condition for the subsequent frame  $t_{n+1}$  [11, 8, 9,  
10, 19]. Their recursive nature facilitates incorporating shape or behavioural  
priors. Both types of approaches deploy a variety of segmentation models,  
110 ranging from watershed [23], Gaussian mixture models [19] to local classifiers [41]  
including deep learning techniques [15]. We refer the interested reader to [12, 10,  
14] for more comprehensive comparisons of cell tracking methods on traditional  
light microscopy data.

Jug et al. [20] proposed a rigorous mathematical abstraction for cell lineage  
115 tracing called moral lineage tracing problem (MLTP). While previous formu-  
lations define their feasible solutions as disjoint paths or trees of individual  
detections, the feasible solutions to the MLTP are disjoint trees of *clusters* of  
an arbitrary set of detections. This enables the MLTP to be used to jointly  
segment and track cells by starting from an oversegmentation of the cells, as

120 showcased in [20], or to account for redundant detections that originate from  
a single cell as in the case of LFM videos in this work. The resulting opti-  
mization problem is considered a hybrid of the *minimum cost multicut problem*  
(*MCMCP*), an NP-hard [44, 45] but well-studied optimization problem in the  
context of image decomposition [46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57],  
125 and the *minimum cost disjoint arborescence* problem, which has been used for  
cell lineage tracing in different forms in [38, 39, 40, 42, 41]. Efficient algorithms  
to solve the MLTP have been proposed in [21] and include a branch-and-cut  
algorithm as well as primal feasible local search heuristics.

Other than most microscopy video analysis methods on traditional light mi-  
130 croscopy videos, we do not aim to segment the cells in our LFM videos but  
only localize their center of mass. This is due to the fact that the cells appear-  
ance in LFM images, i.e. their interference pattern, does not describe their true  
shape and would therefore be ambiguous to define in most cases. Learning-based  
methods for similar detection problems, such as centerline detection of tubular  
135 structures, have been proposed using Hough voting [58] or regressing a distance  
transformation [59]. Kainz et al. [16] apply such a distance regression approach  
for localizing cells of a certain type in histopathology images and demonstrate  
its advantages over a classification-based detection. We follow these findings  
and pose the detection learning task as a regression of an auxiliary probability  
140 map. This probability map is constructed such that spatial maxima indicate the  
presence of cell centers. Thus, it can be considered a special case of a learned  
distance transform. To address the somewhat elusive appearance of cells in  
LFM images, we rely on fully convolutional neural networks for detecting the  
cells. These methods have been successfully applied in several complex object  
145 detection and multi-target tracking tasks [15, 25, 32, 31] and are well-suited for  
processing video sequences.

### 3. Methods

We consider the cell lineage tracing task as a MAP inference over a hypothesis graph containing a multitude of potential lineage forests. In this section, we discuss the probability measure and derive its MAP estimator, the MLTP in Sec. 3.1. We describe the hypothesis graphs semantics and its construction from a sequence of LFM images. For the latter, we devise a fully convolutional cell detector in Sec. 3.2, which estimates a cell probability map for each given image. The resulting workflow is illustrated in Fig. 2.

#### 3.1. Lineage Tracing

##### 3.1.1. Hypothesis Graph

We construct a spatio-temporal hypothesis graph  $G = (V, E)$  as follows: Assume that we have a cell probability map  $P(c_s = 1|I_t)$  for every image  $I_t$  in the sequence, which assigns a probability of finding a cell center at each location  $s$  in image  $I_t$ . How this cell probability map is generated follows in Sec. 3.2. We create one node  $v \in V_t$  for every local maximum in this cell probability map  $P(c_s = 1|I_t)$ . The nodes  $v \in V$  represent cells, yet do not need to be unique, i.e. one cell may give rise to several nodes. In fact, lateral constructive interference of the captured pattern of cells in LFM often causes multiple maxima from one cell and thus, spurious nodes. To avoid gaps in otherwise consistent tracklets, we define *hypothesized* successors to each node that has one or more favourable, potential parents over the  $k$  previous frames but no immediate, favourable successor. Here, we define *favourable* as the nodes being at most  $d_{\text{gap}}$  away from each other. By default, we use  $k = 3$  and  $d_{\text{gap}} = 10$  and fill gaps of up to 3 time steps.

Having constructed all the nodes, we continue with the edges. We first 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  $d_{\text{max}}$ , and then 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. Temporal edges simply represent the possibility that

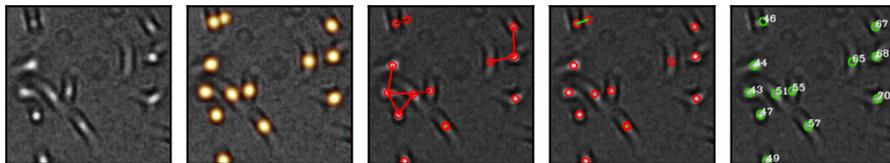


Figure 2: Illustration of our workflow. From left to right: 1) Raw microscopy image, 2) image overlaid with cell probability map generated by the detector, 3) nodes of the hypothesis graph with spatial edges constructed from cell probabilities, 4) optimized lineage where spatial edges that were cut are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

two nodes describe the same cell in two different frames, while spatial edges represent the possibility that two nodes originate from a single individual cell. Unlike the hypothesis graph in [20], where vertices represent superpixels, our hypothesis graph is not planar within each frame, implicitly preventing clusters  
 180 from expanding arbitrarily far in the spatial domain.

### 3.1.2. 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  
 185 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 of our hypothesis graph 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 [20]. It consists of three classes of inequality constraints to ensure: 1) *spatial* and *temporal consistency*, i.e. if nodes  $u \in V_t$  and  $v \in V_t$  belong together and  $v$  and  $w \in V_t \cup V_{t+1}$  belong together, then  $u$  and  $w$  must also belong together. 2) Distinct tracklets cannot merge at a later point in time, that is, no cell can have more than one parent. These are the so called *morality* constraints. 3) *Bifur-*

*cation* 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 mathematically rigorous definition and a more extensive discussion of these constraints, we refer to [20, 21]. We next model the measure of probability as:

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

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

It is comprised of four parts. The first part  $P(X_G|\mathbf{x})$  represents a uniform prior over all feasible solutions  $X_G$ . Second, the cut probability  $P(x_{uv}|\Theta)$  describing the probability of  $u$  and  $v$  being part of the same cell (either in space  
190 if  $u, v \in V_t$  or along time if  $u \in V_t$  and  $v \in V_{t+1}$ ). Third and fourth are 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  $v$ . The state of these birth and termination variables  
195  $x_v^+$  and  $x_v^-$  is therefore deterministically tied to the incident edge cut variables  $x_e$ . This relationship can be encoded with another set of auxiliary inequality constraints whose feasible set we will denote with  $X_V$ .  $\Theta$  denotes the joint set of parameters. We use these four parts to incorporate the following assumptions: Two detections  $u$  and  $v$  that are close are more likely to originate from the same  
200 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  $\Theta$  on training data in a  
205 maximum likelihood fashion: For  $\theta^-$  and  $\theta^+$  this boils down to calculating the relative frequency of the respective events on the annotated lineage. For the spatial and temporal parameters  $\theta^{\text{sp}}$  and  $\theta^{\text{tmp}}$ , we first complement the lineage forest with edges within  $d_{\text{max}}$  as  $\mathcal{E}$ . We then maximize the log-likelihood

210  $\log \mathcal{L}(\theta) = \sum_{uv \in \mathcal{E}} \log P(x_{uv}|\theta)$  by an extensive search over the interval  $\theta \in [\theta_{\min}, \theta_{\max}]$ , where we found  $[1, 80]$  to be an appropriate search range.

The MAP estimate  $\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathbf{X}} P(\mathbf{x}|\Theta, X_G)$  for (2) can be derived as solution to the MLTP:

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

with the coefficients  $c_{uv} = -\log \frac{P(x_{uv}=1|\Theta)}{1-P(x_{uv}=1|\Theta)}$  for edges, and vice versa for  $c_v$  of the node events.  $X_G$  is the set of feasible lineages and  $X_V$  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. The derivation can be found in [Appendix A](#). We optimize (3) with the KLB algorithm described in [21].

### 3.2. Cell Detection with Fully-convolutional Regression 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. Hence, 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 affect the quality of the final lineage trees over many frames. To achieve this, we build on the recent work on fully convolutional neural networks [24] and residual networks [25]. 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  $f(I_t) \approx P(c_s = 1|I_t)$  a dense probability map of finding a cell at a particular location  $s$  in the given image  $I_t$ . This is the cell probability map we introduced in [Sec. 3.1.1](#), from which we construct the vertices  $V$  of the hypothesis graph. We found that predicting a dense probability map facilitates training as it enlarges the spatial support of the sparse cell center annotations and gracefully handles the strongly varying cell culture density. Similar findings were made with techniques that learn a distance transform, e.g. to detect cells in [16]. Designing the detector as fully convolutional network renders it computationally

efficient for long sequences and wide field of views, and facilitates scaling them with extended training data. We describe next how we arrive at suitable architectures for this task and how we construct  $P(c_s = 1|I_t)$  from point-wise cell annotations.

### 3.2.1. Network Architectures

Our main detector network architecture is the *Resnet-23*, which we derive from the *Resnet-50* [25]. We first truncate it 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, is a good trade-off between spatial context and loss of resolution, resulting in output response maps that allow 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, encoding presence and location of cells, while the latter places the corresponding probability kernels, thereby compensating for previous pooling operations and ensuring that the predicted cell probability map has the same resolution as the input image  $I_t$ . A sigmoid output function in the last layer ensures that  $f(I_t)$  is within the interval  $[0, 1]$  at any point. We used dropout of 0.5 after the last convolutional layer to avoid overfitting. We also experiment with a smaller variant called *Resnet-11*, where we already truncate the original Resnet-50 at layer 11 before appending the above described final layers.

As an alternative to the residual networks, we adjust a UNet [15] architecture for our purpose. We use a UNet with 5 levels, i.e. 5 pooling operations in the contracting and 5 up-sampling operations in the expanding path. To this end, we replace the stacks in the expansive path with single up-convolution layers and concatenate them with the corresponding feature maps of the contracting path. Each of these up-convolution layers upsamples by a factor 2 and has kernels of  $5 \times 5 \times k$ , where  $k$  is the number of feature channels according to the level. The last upsampling-layer finally ends with a sigmoid activation to rescale activation into the interval of  $[0, 1]$ . With about  $40.5 \times 10^6$  parameters, this architecture

is considerably larger than the Resnet-23 that has about  $1.75 \times 10^6$ .

### 3.2.2. Loss Function & Training Setup

This subsection describes the general training procedure for the cell detectors. Particular training parameter choices such as learning rate or batch size  
270 are specified in each respective experiment in Sec. 4. We sample  $n$  new (sub-)images from the training corpus in each epoch. The size of these training images depends on the detectors architecture and are large enough to contain several cells. For example, all Resnet-derived detectors such as Resnet-11 and Resnet-  
275 23 take inputs of size  $224 \times 224$ . Each image is normalized to zero mean and unit variance. The sampling procedure chooses uniformly amongst the available datasets and, then, amongst its frames. While this means that frames of shorter sequences are more frequently sampled than those of longer sequences, it helps keeping the balance between sequences of different cell types and cell culture  
280 density. For each training image  $I_k$ , we construct a corresponding cell probability map  $P(c_s = 1|I_k)$  by placing a Gaussian kernel  $G^\sigma$  at each annotated center. The size of the Gaussian kernel, controlled by  $\sigma$ , has to be chosen accordingly such that it gives appropriate spatial support but does not overlap too strongly within cell clusters. The fact that we use a single  $\sigma$  implicitly represents the  
285 assumption that all cells take approximately the same spatial extent, which we find reasonable for our microscopy data. In our experiments, we will work with  $\sigma = 8$  and  $\sigma = 4$ . An illustration of a training image  $I_k$  and its corresponding probability map  $P(c_s = 1|I_k)$  is given in Fig. 3.

We train the detector network to approximate these probability maps  $P(c_s = 1|I_t)$   
290 by minimizing a pixel-wise mean absolute error (MAE) loss. We found that the MAE yields better results than the cross entropy that was used in [26], since MAE does not pull the output probabilities towards extreme probabilities as the cross entropy loss does. This helps in particular in the presence of sequences with dense cell cultures. Furthermore, we mask the loss to be insensitive to  
295 the border region (15 px from the images edge) in order to not penalize the detector for its behaviour under incomplete information, i.e. where some of the

convolutional layers operate on zero-padded regions. This boundary region is also accounted for when applying the detector to larger images than its native image size: patches to process are chosen with an overlap of at least twice the border region. Since the MAE is a surrogate for the actual metric of interest, the detection rate, it can be difficult to judge progress of training just based on this MAE pixel-wise loss. We handle this as follows: In the beginning of training, we randomly select a small number of (entire) frames from each training sequence as validation frames. Then, at the end of each epoch, we calculate the actual detection precision, recall and F1-score by taking the local maxima on the current probability map for these validation frames as cell detections and matching them to the annotations (i.e. as described in Sec. 4.1). We found it sufficient to only calculate a greedy matching between detections and annotations instead of an optimal one, saving valuable time. Keeping in mind that this estimate is slightly optimistic due to the potentially high correlation between the validation frames and other training frames that are within close temporal proximity, we found it a valuable approach for monitoring training progress in practice.

#### 4. Experiments & Results

In this section, we present both experiment setup and their results. We distinguish two main sets of experiments: The first set (Sec. 4.2) is done on three LFM sequences that were manually annotated based on LFM images only. This is the same dataset that was used in [26]. The second set of experiments (Sec. 4.3) is concerned with a more extensive dataset of 16 sequences for which we acquired fluorescence microscopy images in parallel to the LFM images. Examples for both types of microscopy images can be found in Fig. 6. The annotations for these sequences are generated from the fluorescence microscopy images, in which the cells nuclei are labeled with a fluorescent dye and can be tracked with high spatial accuracy. These fluorescence-based annotations serve two purposes: First, they enable us to validate our approach in the presence of

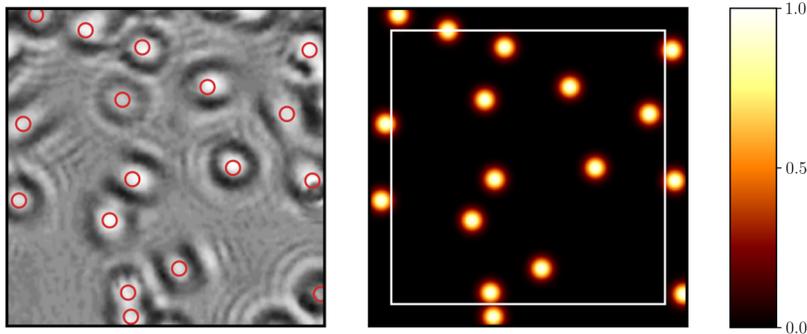


Figure 3: Illustration of the cell detection learning task. **Left:** A training image  $I$  with cell annotations indicated as red circles. **Right:** The corresponding, constructed probability map  $P(c_S = 1|I)$  which we train our fully convolutional network to approximate. Local maxima in the probability map  $P(c_S = 1|I)$  correspond to cell annotations. Both input and output are of identical size,  $224 \times 224$  in this case. The overlaid white rectangle indicates the region within which the loss is evaluated during training. By excluding the loss outside this region, i.e. the border, we don't penalize the detector for its performance under incomplete information.

settings that are otherwise difficult to annotate, e.g. in the presence of dense clusters or when there are simply too many cells in the field of view. Second, they allow us to measure to which degree we are able to reconstruct the same lineages as from traditional fluorescent microscopy, yet using LFM only (i.e. without any fluorescence labeling). In a third and final experiment (Sec. 4.4), we showcase our method for analysing cell dynamics on four sequences of skin cancer cells.

#### 4.1. Metrics

We match annotated cells (vertices in the ground truth graph) to detections (vertices in the estimated graph) within each frame such that each vertex is matched to at most one vertex in the other graph. We consider only matches closer than 10 px ( $\approx$  a cell center region) as a true positive (TP). All unmatched annotations are counted as false negative (FN), unmatched detections as false positive (FP). The matching can be computed efficiently with the Hungarian algorithm [60, 61]. Using these definitions, we calculate precision, recall and F1-score. These three metrics provide information on the detection accuracy

and can be calculated for each frame individually.

To compare the quality of different lineages, we use again the same vertex-matching between the annotated cell lineage forest and the predicted cell lineage forest to calculate the number of TP, FP and FN as described for the detection metrics. 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) [62] and multiple object tracking accuracy (MOTA) [27], two metrics commonly used in multi-object tracking tasks. Furthermore, we compute the number of edit operations needed to get from the predicted lineage to the ground truth lineage, and calculate the tracking accuracy (TRA) score as proposed in [12, 63] for cell tracking from it. This metric uses a normalized count of necessary edits to judge the quality of a lineage, that is  $\text{TRA} = 1 - \frac{|\Delta(G_{\text{Pred}}, G_{\text{GT}})|}{|\Delta(0, G_{\text{GT}})|}$ , where  $\Delta(G, H)$  is the set of edits required to get from  $G$  to  $H$ , and  $\Delta(0, G)$  are the operations needed to build  $G$  from scratch. Other than in the original definition of TRA, we set a unit weight for each type of edit (addition or deletion of a node or edge). This is justified by the fact that we work with point annotations for cells instead of segmentations, rendering both addition and deletion equally expensive to correct.

#### 4.2. Experiments on Original Datasets

The subsequent experiments are conducted on the manually annotated dataset used in [26]. It consists of three individual sequences: one sequence of A549 cells, annotated over 250 frames in a region of interest (ROI) of  $1295 \times 971$  px, used for all training purposes, as well as two sequences of 3T3 cells for testing. These two testing sequences are annotated over 350 and 300 frames in a ROI of  $639 \times 511$  px (*3T3-I*) and  $1051 \times 801$  px (*3T3-II*), respectively. The reason for this training/test split is to avoid an overestimation of the methods performance through a potential overfit. Frames were acquired at an interval of 3 min with a spacing of  $1.4 \mu\text{m} \times 1.4 \mu\text{m}$  per pixel.

370 *4.2.1. Benchmarking Detectors*

We compare the discussed network configurations, including the described Resnet-23, Resnet-11 and the UNet, plus a plain vanilla CNN with three  $5 \times 5$  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, denoted  
375 as CNN-4. We train the networks for 100 epochs, in each of which we sample 4000 training images in batches of 8 from the training sequence. We employ Adam [64] as optimization strategy. For the two Resnet-derived architectures, we use pre-trained weights from Resnet-50 [25], which we fine tune with a learning rate of learning rate of  $10^{-3}$ . For UNet and CNN-4, we adjust the learning  
380 rate to  $10^{-2}$ , but otherwise, keep the same training schedule.

The results of this experiment in terms of detection metrics on the two test sequences are presented in Fig. 4. We find that the Resnet-23 obtains the best average F1-score of 94.1%, followed by the UNet with 89.2%, Resnet-11 with 85.1% and finally, the CNN-4 with 72.2%. The major source of mistakes on  
385 these two test sequences occur in the form of spurious detections, deteriorating the precision of the detectors. The CNN-4 is particularly prone to such mistakes, achieving a considerably lower precision than its competitors.

*4.2.2. Lineage Tracing*

For the MLTP, we compare the effect of varying  $\theta^{\text{tmp}}$ ,  $\theta^{\text{sp}}$  together with  
390 hypothesis graphs generated from the different detectors in Fig. 5. The optimal parameter choice for Resnet-23 is at  $\theta = 10$  (corresponding to about  $14 \mu\text{m}$ ), i.e. a relatively small radius of favourable merges, while the other detectors considerably benefit from wider ranges. Most notably, the CNN-4 that suffers from a low precision due to many spurious detections benefits from a wider range  
395 of spatial merges. In Table 1, we compare different lineage tracing approaches. Here we use the described maximum-likelihood parameter fitting approach to determine the MLTP parameters  $\theta^{\text{tmp}}$ ,  $\theta^{\text{sp}}$  and the birth and termination rates from the training sequence. Our baseline is linear assignment problem tracking (LAPT) [7]. The disjoint trees method (DTP), uses our Resnet-23 detections

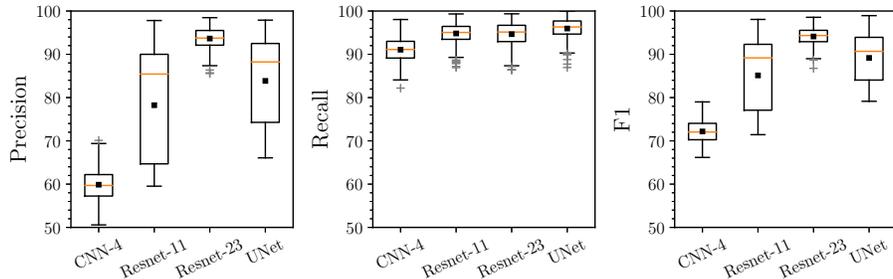


Figure 4: Performance of different detectors measured in terms of precision, recall and F1-score of their detections calculated on the original test datasets of [26]. Boxplots depict median as orange line, mean as black square and outliers as grey +. All scores are shown in %. We find that Resnet-23 is the most robust detector in our experiment with an average F1 of 94.1%. It is followed by the UNet with an F1 of 89.2%, Resnet-11 with 85.1% and finally, CNN-4 with 72.2%. We note that the main source of error on these test sequences lies in spurious detections that decrease the detectors precision.

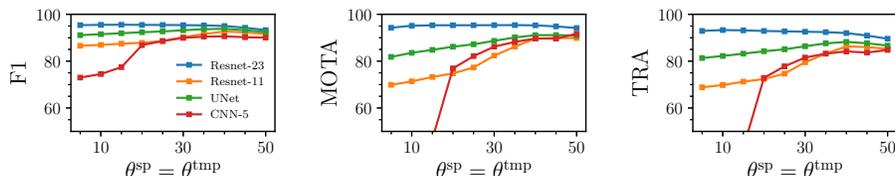


Figure 5: Sensitivity analysis of the lineage tracing model with different detectors. We increase both edge cut parameters  $\theta^{tmp}$  and  $\theta^{SP}$  together. While the optimal choice in combination with Resnet-23 is relatively small, i.e. at 10, the other detectors, which suffer from many spurious detections, benefit from a wider range. Most notably, the performance with CNN-4 improves up to a competitive TRA of 84.8%.

400 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.

### 4.3. Experiments on Fluorescence-annotated Sequences

405 The next experiments are conducted on a new dataset of sequences for which we acquired fluorescence microscopy images concurrently to the LFM images. Thanks to this, we are able to generate cell lineage annotations not only from visual appearance of the LFM, but based on the fluorescence microscopy, where

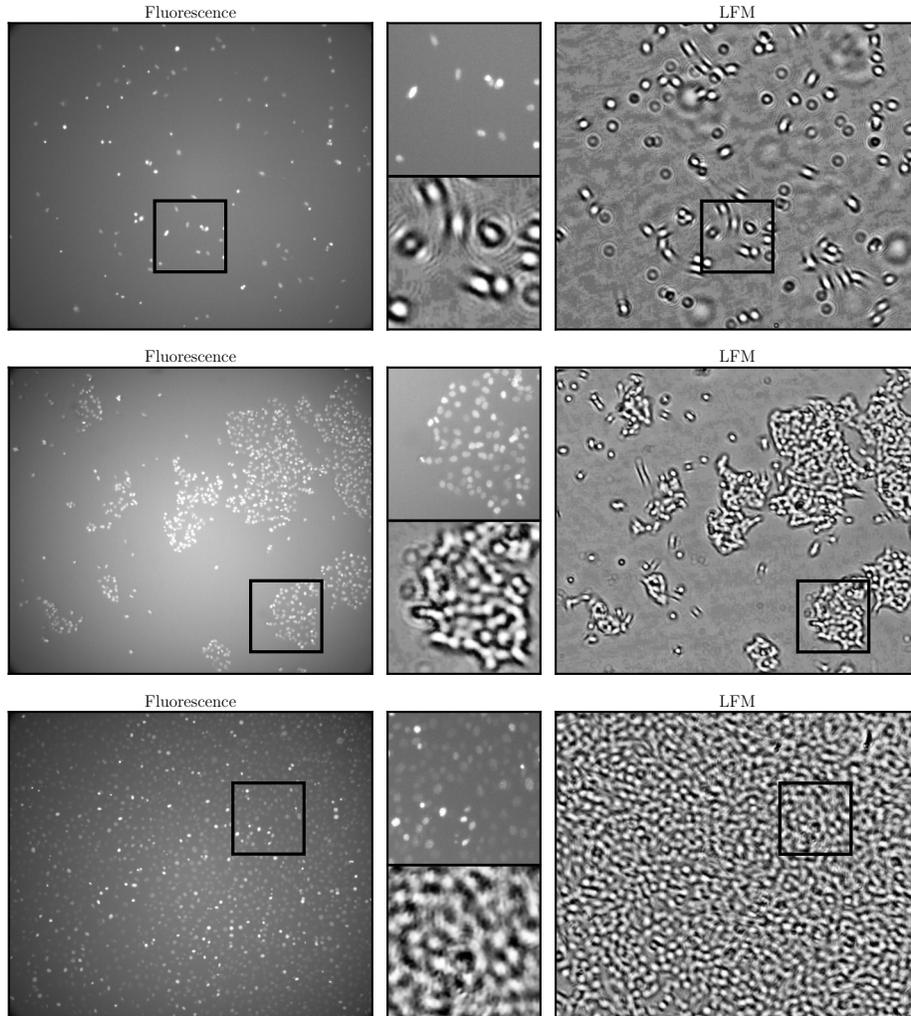


Figure 6: Examples for sequences of different cell culture density (top to bottom: low, medium, high) and different cell lines (again top to bottom: 3T3, A549, HuH7). The left hand side shows the fluorescence microscopy image, while the corresponding LFM image is shown on the right. The central column presents a detailed view of the indicated region for each comparison. The fluorescence microscopy image, in which the cell nuclei are labeled with a fluorescent dye, provides valuable information to generate ground truth annotations for otherwise ambiguous scenarios, such as dense cell clusters. The LFM, on the other hand, is a label-free technique and thus avoids all the limitations inherent to fluorescence labeling, such as temporally limited observation due to bleaching or the fact that the dye may alter the cells behaviour.

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 %. Our proposed method is MLTP. 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 [7] 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 |

cell nuclei were labeled with a fluorescent dye. This allows us to generate a large corpus of validation sequences with annotations even in dense cell clusters that would otherwise be ambiguous to annotate or with several hundreds per frame  
410 simply too many. From these fluorescence-based annotations, we quantify how far we are able to retrieve the same information from only LFM videos. This is interesting because LFM is a label-free technique and thus avoids all the limitations inherent to fluorescence labeling, such as limited temporal observation  
415 due to bleaching or the fact that the dye may alter the cells behaviour. We acknowledge that the fluorescent labeling may not succeed for a small fraction of cells, as a few of them do not incorporate the fluorescent dye sufficiently, and are thus not visible in the fluorescence image. However, we consider this fraction to be negligible for our study. The dataset includes 16 sequences in total of three  
420 different cell lines, A549, 3T3 and HuH7 cells, and a wide variety of settings, which we mainly categorize with respect to the sequences cell culture density (from low over medium to high density). Examples are illustrated in Fig. 6. Because of the concurrent fluorescence microscopy setup, the acquisition interval is larger than for those sequences in Sec. 4.2, ranging from 14.95 min to

425 28.75 min. The videos are between 82 and 279 frames. All of them have a ROI  
of  $847 \times 1007$  px and a pixel spacing of  $1.65 \mu\text{m} \times 1.65 \mu\text{m}$ . The annotations  
in the fluorescence microscopy videos are generated using TrackMate [65] with  
manually tweaked parameters for each sequence. Fluorescence and LFM im-  
430 ages are registered with an affine transform estimated from several manually  
selected key point pairs. We refer to this dataset as *fluorescence-annotated* in  
the remainder of the paper.

#### 4.3.1. Detection Performance and Retraining

We evaluate our best-performing models, the Resnet-23 and UNet architec-  
435 ture, on the fluorescence-annotated dataset. We apply the two detectors trained  
on the original dataset of Sec. 4.2.1, to the new dataset. Then, we retrain both  
models on the fluorescence-annotated sequences in two variations. In the first,  
we only train on sequences of low and medium cell density. In the second, we  
also use dense sequences. In both cases, we employ a 3-fold cross-validation and  
report only scores on the hold-out sequences of the according split. Again, we  
440 treat each individual sequence as one entity in order to avoid over-optimistic  
performance estimates. We stratify the folds to ensure that all training sets  
contain sequences of each cell line and density. The models are retrained for  
100 epochs of 4000 sampled images at a learning rate of  $10^{-4}$  with Adam [64]  
as optimization strategy. We set the batch size to 50 and 25 for Resnet-23 and  
445 UNet, respectively, and we reduce the size of the placed Gaussians to  $\sigma = 4$  in  
order to account for dense cell clusters that require finer grained kernels.

The comparison is presented in Fig. 7. We observe that the detectors trained  
on the original datasets perform best on sequences of low cell culture density,  
which are most similar to the original ones, and exhibit decreasing performance  
450 with increasing density. This is also seen in the performance conditioned on  
the cell line. A549 cells, which tend to agglomerate and build dense clusters,  
are more difficult to detect than the other two cell lines, which spread more  
evenly. Overall, the performance decreases compared to the benchmark on  
manual annotations of Fig. 4. To a certain degree, this is to be accounted

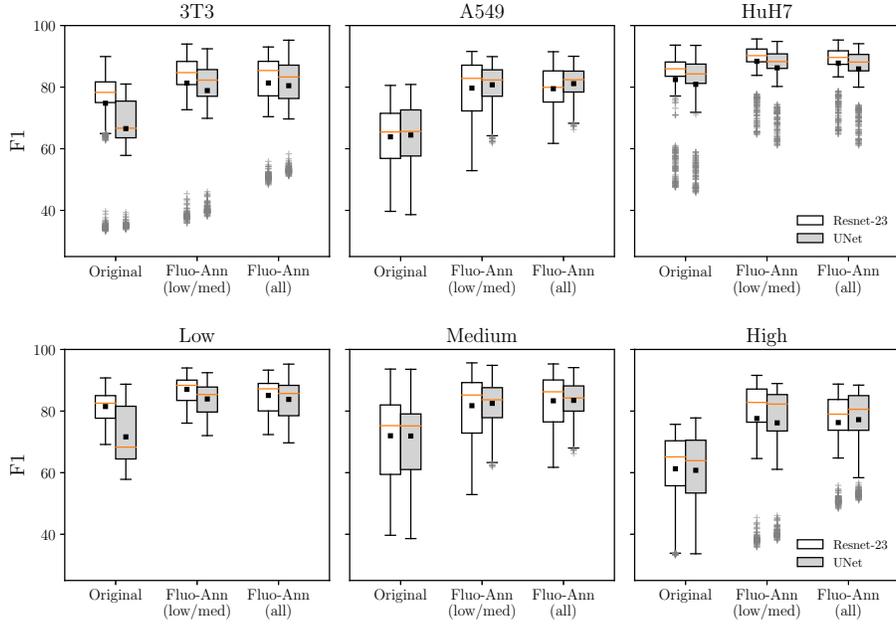


Figure 7: Performance comparison of Resnet-23 and UNet on the fluorescence-annotated dataset. *Original* refers to the detectors trained on the original dataset of Sec. 4.2.1, while *Fluo-Ann* are the detectors that were retrained on fluorescence-annotated sequences (cf. Sec. 4.3). We distinguish between retraining with all sequences (*all*) and retraining only on sequences that were considered having a low or medium cell density (*low/med*). F1-scores are shown in % and are calculated per-frame on the respective hold-out set of a 3-fold crossvalidation. The **top** row shows performance with respect to the observed cell lines, i.e. 3T3, A549 and HuH7 cells, while the **bottom** row categorizes them according to their cell culture density. Median values are indicated with an orange line, mean values with a black square. Outliers are depicted as grey +. Retraining improves performance in all categorizations, yet we find the fluorescence-based annotations considerably more challenging to reproduce than the original ones that were created from LFM only.

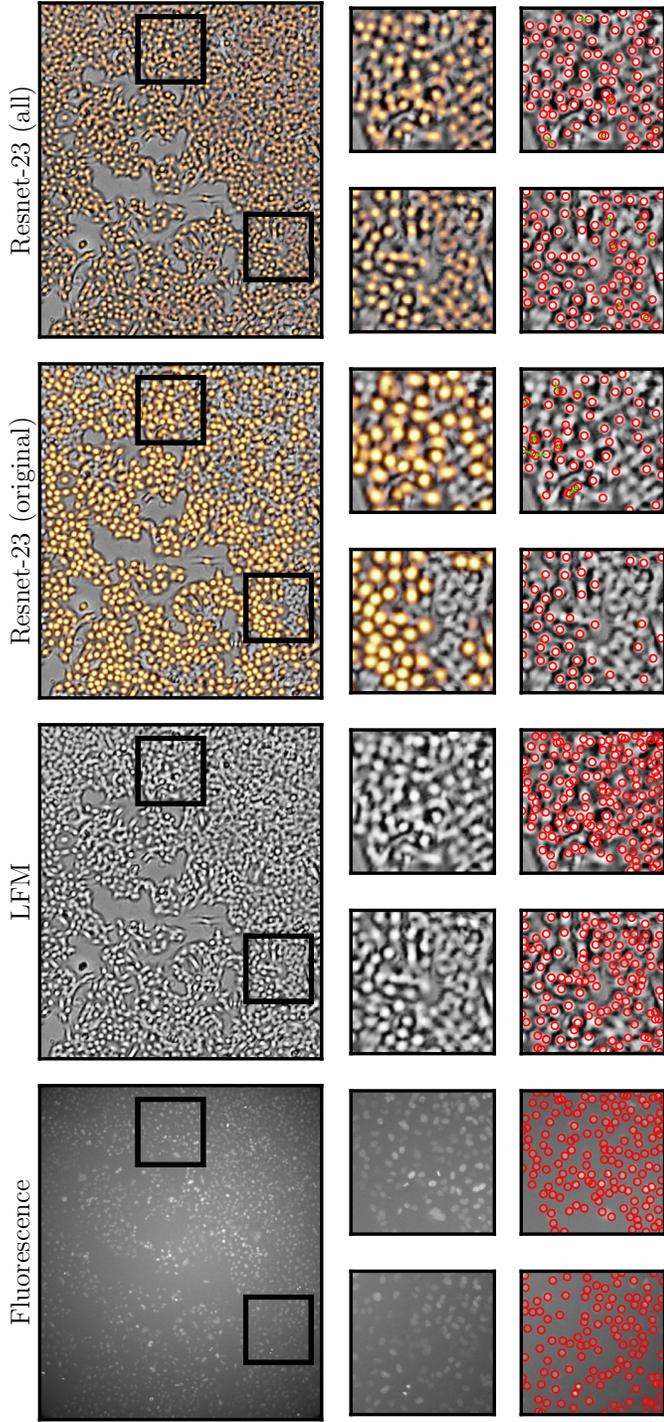


Figure 8: Qualitative comparison on a sequence with a *dense* cell culture. From left to right: Fluorescence microscopy image, raw LFM image, LFM image with heatmap as overlay from a Resnet-23 trained on the original dataset (*original*) and retrained on fluorescence-based annotations (*all*). Detail views corresponding to the drawn rectangles are presented in the two lower rows. Red circles in the bottom row represent cells, either from fluorescence-based annotations (Fluorescence and LFM columns) or generated from the detectors probability map. For the latter two, green edges connect detections that were clustered together in the solution of the MLTP. We notice that the original Resnet-23 struggles with the varying sharpness in dense regions. The retrained Resnet-23 handles this better and is able to distinguish close cells, partly due to the finer heatmap kernels. We further observe that determining the presence of certain cells only based on the LFM is extremely challenging, e.g. in the bottom right corner of the right detail. Additional visual comparisons on are found in [Appendix B](#), including low and medium density cases.

455 to the more diverse dataset, but the main difficulty comes from the fact that  
some of the cells that are well-visible in the fluorescence microscope are hard to  
spot in the corresponding LFM image, even for a human annotator (cf. Fig. 6,  
in particular the medium and high density example, or the visual comparison  
in Fig. 8 and Appendix B). The *original* detectors are thus simply trained to  
460 another standard.

Retraining the detectors on fluorescence-based annotations helps. On low  
density sequences, F1-scores improve from a mean of 81.4% to 87.0% (Resnet-  
23) and from 71.6% to 83.9% (UNet), indicating that the more challenging  
fluorescence-based annotations can be learned to a certain degree. Similar find-  
465 ings are made on medium density sequences, with the only difference being that  
both architectures benefit by about the same amount: Resnet-23 improves from  
71.9% to 83.3% and UNet from 71.9% to 83.4%. The improvements on high  
density sequences are the largest. Both original detectors obtain a mean F1-  
score of 61.2% (Resnet-23) and 60.7% (UNet), while the retrained ones obtain  
470 76.2% and 77.2%, respectively. An interesting observation is that including  
dense sequences into the training set does not considerably improve the av-  
erage when compared to retraining only with low/medium density sequences  
(c.f. Fig. 7, *all* vs *low/mid*). A partial reason could be that medium sequences  
contain sequences with clusters of similar appearance like completely dense se-  
475 quences, thereby providing information at training time that transfers to the  
high density setting. On the other hand, we observe that prediction on the  
worst frames ameliorate when training on all (cf. outliers in Fig. 7: High) while  
the median decreases. This could be considered an indicator that the mod-  
els have insufficient capacity to capture all the necessary appearances, or that  
480 some of the relevant information is simply not present in the LFM image and  
the models try to compensate for it by shifting their expectation. A qualitative  
example of the detection probability maps with and without retraining can be  
found in Fig. 8, and additional examples are found in Appendix B. In general,  
we notice that the considerably extended training dataset especially benefits the  
485 model with more capacity, that is the UNet, which we find to be on par with

the Resnet-23 after retraining.

#### 4.3.2. Lineage Tracing

We next evaluate the tracing performance of the MLTP combined with either of the two retrained detectors of the previous section on the fluorescence-annotated sequences. Retraining is again done in a 3-fold crossvalidation using  
490 annotated sequences. Retraining is again done in a 3-fold crossvalidation using sequences of all densities (i.e. the ones denoted as *Fluo-Ann. (all)* in Fig. 7). Otherwise, we keep the MLTP parameters fixed as in Sec. 4.2.2. All scores are calculated on the according hold-out sets of the detectors crossvalidation.

The results are presented in Fig. 9 and Table 2. Average MOTA over all  
495 sequences is 73.1% and 70.78% for Resnet-23 and UNet, respectively, while average TRA is 64.2% and 63.2%. In both cases, the MLTP obtains better F1, MOTA and TRA scores than the alternative model of solving a disjoint trees problem (denoted with DTP). Analyzing the results with respect to cell culture density in the sequence, we observe that TRA decreases with higher cell culture  
500 density from 71.3% to 59.9%, whereas MOTA increases slightly from 71.6% to 76.2%. The same trend is observed in combination with the UNet with a TRA decrease from 70.45% to 58.9% and a minor MOTA increase from 70.9% to 73.2%. This is due to the fact that performance on dense sequences suffers from a higher amount of false negatives, i.e. a lower recall. TRA penalizes false  
505 negatives twice (through both a vertex and an edge insertion), while MOTA only counts them once (as missed detection). MOTA puts more weight on the tracking accuracy of detected nodes, which becomes easier in dense sequences due to the constrained spatial motion, thus explaining the slight improvement. Both MOTA and TRA are lower than in the evaluation on manually annotated  
510 datasets of Sec. 4.2.2, mainly because there is no mechanism to compensate for entirely missed tracklets, which would be visible in the fluorescence sequence but are hard to retrieve from only the LFM sequence and thus, are the major source of error on the fluorescence-annotated dataset.

Table 2: Average performance metrics obtained on the fluorescence-annotated dataset. MLTP is our method based on moral lineage tracing, while DTP solves a disjoint trees problem. All metrics are shown in %. Averages are computed over all sequences with unit weight and reported in the last row. We find that MLTP achieves better scores with either detector.

| Detector  | Model | Precision | Recall | F1    | MODA  | MOTA  | TRA   |
|-----------|-------|-----------|--------|-------|-------|-------|-------|
| Resnet-23 | DTP   | 87.27     | 77.30  | 80.76 | 88.27 | 63.87 | 60.46 |
|           | MLTP  | 93.09     | 73.92  | 81.14 | 94.21 | 73.11 | 64.26 |
| UNet      | DTP   | 82.70     | 79.32  | 80.12 | 83.27 | 56.64 | 57.34 |
|           | MLTP  | 91.40     | 74.85  | 81.21 | 92.66 | 70.78 | 63.25 |

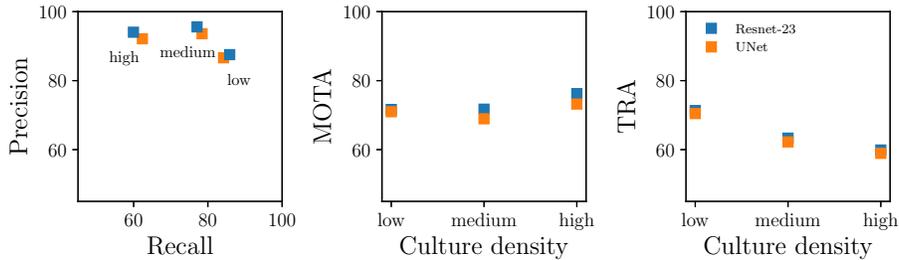


Figure 9: Comparison of lineage tracing metrics on the fluorescence-annotated dataset. We compare the MLTP together with Resnet-23 and Unet, both being retrained on fluorescence-annotated sequences of all densities. All scores are calculated on hold-out sets of a 3-fold crossvalidation. The left plot shows mean precision and recall, while the middle and right show mean tracking metrics MOTA and TRA. Low, medium and high refer to the cell culture density of the analyzed sequences. Average MOTA over all sequences is 73.1% and 70.78% for Resnet-23 and UNet, respectively, while average TRA is 64.2% and 63.2%. We observe that TRA decreases with increasing density, while MOTA shows a slight improvement. This is due to the fact that performance on dense sequences suffers from a higher amount of false negatives, i.e. a lower recall. TRA penalizes these false negatives twice (through both a vertex and an edge insertion), while MOTA only counts each of them once. MOTA puts more weight on the tracking accuracy of detected nodes, which becomes slightly easier in dense sequences due to the constrained spatial motion.

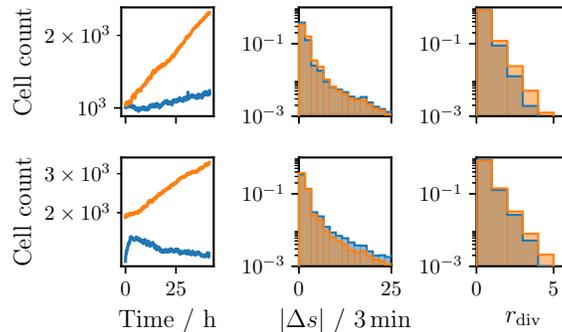


Figure 10: Cell dynamics measured on two experiments with skin cancer cell lines. One population (blue) is exposed to an inhibitor substance, while the other (orange) is not. From left to right: Cell count over time, histograms on cell motility ( $\mu\text{m}/3 \text{ min}$ ) and divisions  $r_{\text{div}}/\text{h}$ . While many cells are steady, we observe displacements of up to  $25 \mu\text{m}$  between two frames. Cells that divide often are more abundant in the control group.

#### 4.4. Assessing Cell Population Dynamics

515 As a proof of concept, 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 10 depicts the resulting statistics. We observe the expected difference in growth rate, yet a more constrained motility of the control cells, which we attribute to the limited space.

## 520 5. Conclusions

We have presented a framework for automatic analysis of LFM time-lapse sequences. It transfers recent work on fully convolutional neural networks [24, 15] and residual learning [25], and moral lineage tracing [20, 21], to the task at hand. We have shown experimentally that it is able to determine cell lineage  
 525 forests of high quality. Furthermore, we examined its performance on the more challenging problem of reconstructing cell lineages from LFM equivalent to the ones obtainable from a corresponding fluorescence microscopy sequence. To this end, we present experiments on an extended dataset for which we acquired fluorescence microscopy in parallel to LFM and generated ground truth lineages

530 from the former. While our framework is able to adapt to these annotations  
through retraining, reaching an average MOTA of up to 73.1%, it does not  
reach the same level of performance we observed on manually annotated datasets  
based on LFM images only. Future work will need to investigate up to which  
point this performance can be pushed and to which degree the information  
535 on certain cell locations can no longer be retrieved from LFM images. While  
our method is designed to be fully automated, certain challenging acquisition  
environments might require human interaction, e.g. for re-training the detectors.  
To this end, future research on integrating it into a interactive workflow would  
be valuable and could use, for example, recent active learning techniques as  
540 in [66].

### 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). We  
545 acknowledge the support of NVIDIA Corporation with the donation of the Titan  
X GPU used in this research.

### References

- [1] P. J. Keller, A. D. Schmidt, A. Santella, K. Khairy, Z. Bao, J. Wittbrodt, E. H. Stelzer, Fast, high-contrast imaging of animal development  
550 with scanned light sheet-based structured-illumination microscopy, *Nature methods* 7 (8) (2010) 637–642.
- [2] P. J. Keller, A. D. Schmidt, J. Wittbrodt, E. H. Stelzer, Reconstruction of zebrafish early embryonic development by scanned light sheet mi-  
croscopy, *Science* 322 (5904) (2008) 1065–1069. [arXiv:http://science.  
555 sciencemag.org/content/322/5904/1065.full.pdf](http://arxiv.org/http://science.sciencemag.org/content/322/5904/1065.full.pdf).

- [3] C. Guillot, T. Lecuit, Mechanics of epithelial tissue homeostasis and morphogenesis, *Science* 340 (6137) (2013) 1185–1189.
- [4] O. Mudanyali, D. Tseng, C. Oh, S. O. Isikman, I. Sencan, W. Bishara, C. Oztoprak, S. Seo, B. Khademhosseini, A. Ozcan, Compact, light-weight and cost-effective microscope based on lensless incoherent holography for telemedicine applications, *Lab on a Chip* 10 (11) (2010) 1417–1428.
- [5] A. Greenbaum, W. Luo, T.-W. Su, Z. Göröcs, L. Xue, S. O. Isikman, A. F. Coskun, O. Mudanyali, A. Ozcan, Imaging without lenses: achievements and remaining challenges of wide-field on-chip microscopy, *Nature methods* 9 (9) (2012) 889–895.
- [6] S. V. Kesavan, F. Momey, O. Cioni, B. David-Watine, N. Dubrulle, S. Shorte, E. Sulpice, D. Freida, B. Chalmond, J. Dinten, et al., High-throughput monitoring of major cell functions by means of lensfree video microscopy, *Scientific reports* 4 (2014) 5942.
- [7] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, G. Danuser, Robust single-particle tracking in live-cell time-lapse sequences, *Nature methods* 5 (8) (2008) 695–702.
- [8] F. Amat, W. Lemon, D. P. Mossing, K. McDole, Y. Wan, K. Branson, E. W. Myers, P. J. Keller, Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data, *Nature methods*.
- [9] F. Amat, E. W. Myers, P. J. Keller, Fast and robust optical flow for time-lapse microscopy using super-voxels, *Bioinformatics* 29 (3) (2013) 373–380.
- [10] N. Chenouard, I. Smal, F. De Chaumont, M. Maska, I. F. Sbalzarini, Y. Gon, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Cohen, W. J. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. Magnusson, J. Jalden, H. M. Blau, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J.-Y. Tinevez, S. L. Shorte, J. Willemse, K. Celler, G. P. Van Wezel, H.-W. Dan, Y.-S. Tsai, C. Ortiz De Solorzano,

- J.-C. Olivo-Marin, E. Meijering, Objective comparison of particle tracking  
585 methods, *Nature Methods* 11 (3) (2014) 281–289.
- [11] K. Li, E. D. Miller, M. Chen, T. Kanade, L. E. Weiss, P. G. Campbell, Cell  
population tracking and lineage construction with spatiotemporal context,  
*Medical image analysis* 12 (5) (2008) 546–566.
- [12] M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Ur-  
590 biola, T. España, S. Venkatesan, D. M. W. Balak, P. Karas, T. Bolcková,  
M. Streitová, C. Carthel, S. Coraluppi, N. Harder, K. Rohr, K. E. G.  
Magnusson, J. Jaldn, H. M. Blau, O. Dzyubachyk, P. Křížek, G. M. Ha-  
gen, D. Pastor-Escuredo, D. Jimenez-Carretero, M. J. Ledesma-Carbayo,  
A. Muñoz-Barrutia, E. Meijering, M. Kozubek, C. Ortiz-de Solorzano,  
595 A benchmark for comparison of cell tracking algorithms, *Bioinformatics*  
30 (11) (2014) 1609–1617.
- [13] E. Meijering, O. Dzyubachyk, I. Smal, Methods for cell and particle track-  
ing, *Methods in Enzymology* 504 (9) (2012) 183–200.
- [14] V. Ulman, M. Maška, K. E. Magnusson, O. Ronneberger, C. Haubold,  
600 N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, et al., An  
objective comparison of cell-tracking algorithms, *Nature methods* 14 (12)  
(2017) 1141.
- [15] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for  
biomedical image segmentation, in: *MICCAI*, Springer, 2015, pp. 234–241.
- 605 [16] P. Kainz, M. Urschler, S. Schuler, P. Wohlhart, V. Lepetit, You should  
use regression to detect cells, in: *MICCAI*, Springer, 2015, pp. 276–283.
- [17] A. Khan, S. Gould, M. Salzmann, Deep convolutional neural networks for  
human embryonic cell counting, in: *ECCV*, Springer, 2016, pp. 339–348.
- [18] W. Xie, J. A. Noble, A. Zisserman, Microscopy cell counting with fully con-  
610 volutional regression networks, in: *MICCAI 1st Workshop on Deep Learn-  
ing in Medical Image Analysis*, 2015.

- [19] A. Arbelle, N. Drayman, M. Bray, U. Alon, A. Carpenter, T. R. Raviv, Analysis of high-throughput microscopy videos: Catching up with cell dynamics, in: MICCAI, Springer, 2015, pp. 218–225.
- 615 [20] F. Jug, E. Levinkov, C. Blasse, E. W. Myers, B. Andres, Moral lineage tracing, in: CVPR, 2016.
- [21] M. Rempfler, J. Lange, F. Jug, C. Blasse, E. W. Myers, B. H. Menze, B. Andres, Efficient algorithms for moral lineage tracing, in: ICCV, IEEE, 2017.
- 620 [22] G. Flaccavento, V. Lempitsky, I. Pope, P. Barber, A. Zisserman, J. A. Noble, B. Vojnovic, Learning to count cells: applications to lens-free imaging of large fields, *Microscopic Image Analysis with Applications in Biology*.
- [23] K. E. G. Magnusson, J. Jaldén, A batch algorithm using iterative application of the viterbi algorithm to track cells and construct cell lineages, in: 625 ISBI, 2012, pp. 382–385.
- [24] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: CVPR, 2015, pp. 3431–3440.
- [25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.
- 630 [26] M. Rempfler, S. Kumar, V. Stierle, P. Paulitschke, B. Andres, B. H. Menze, Cell Lineage Tracing in Lens-Free Microscopy Videos, Springer International Publishing, Cham, 2017, pp. 3–11.
- [27] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the clear mot metrics, *EURASIP Journal on Image and Video Processing* 2008 (1) (2008) 1–10.
- 635 [28] J. Berclaz, F. Fleuret, E. Türetken, P. Fua, Multiple object tracking using k-shortest paths optimization, *IEEE transactions on pattern analysis and machine intelligence* 33 (9) (2011) 1806–1819.

- [29] S. Tang, B. Andres, M. Andriluka, B. Schiele, Subgraph decomposition for  
640 multi-target tracking, in: CVPR, 2015, pp. 5033–5041.
- [30] X. Wang, E. Türetken, F. Fleuret, P. Fua, Tracking interacting objects  
optimally using integer programming, in: ECCV, Springer, 2014, pp. 17–  
32.
- [31] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. An-  
645 dres, B. Schiele, ArtTrack: Articulated multi-person tracking in the wild,  
in: CVPR, 2017.
- [32] S. Tang, M. Andriluka, B. Andres, B. Schiele, Multiple people tracking by  
lifted multicut and person re-identification, in: CVPR, 2017.
- [33] J. Funke, B. Andres, F. A. Hamprecht, A. Cardona, M. Cook, Efficient au-  
650 tomatic 3d-reconstruction of branching neurons from em data, in: CVPR,  
IEEE, 2012, pp. 1004–1011.
- [34] M. Rempfler, M. Schneider, G. D. Ielacqua, X. Xiao, S. R. Stock, J. Klohs,  
G. Székely, B. Andres, B. H. Menze, Reconstructing cerebrovascular net-  
works under local physiological constraints by integer programming, *Med-*  
655 *ical Image Analysis* 25 (1) (2015) 86 – 94.
- [35] M. Rempfler, B. Andres, B. Menze, The minimum cost connected subgraph  
problem in medical image analysis, in: MICCAI, 2016, pp. 397–405.
- [36] E. Türetken, G. González, C. Blum, P. Fua, Automated reconstruction of  
dendritic and axonal trees by global optimization with geometric priors,  
660 *Neuroinformatics* 9 (2-3) (2011) 279–302.
- [37] E. Türetken, F. Benmansour, B. Andres, P. Glowacki, H. Pfister, P. Fua,  
Reconstructing curvilinear networks using path classifiers and integer pro-  
gramming, *IEEE Transactions on Pattern Analysis and Machine Intelli-*  
*gence* 38 (12) (2016) 2515–2530.

- 665 [38] F. Jug, T. Pietzsch, D. Kainmüller, J. Funke, M. Kaiser, E. van Nimwegen, C. Rother, G. Myers, Optimal joint segmentation and tracking of escherichia coli in the mother machine, in: Bayesian and graphical Models for Biomedical Imaging, Springer, 2014, pp. 25–36.
- [39] B. X. Kausler, M. Schiegg, B. Andres, M. Lindner, U. Koethe, H. Leitte,  
670 J. Wittbrodt, L. Hufnagel, F. A. Hamprecht, A discrete chain graph model for 3d+ t cell tracking with high misdetection robustness, in: ECCV, Springer, 2012, pp. 144–157.
- [40] D. Padfield, J. Rittscher, B. Roysam, Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis, Medical image analysis  
675 15 (4) (2011) 650–668.
- [41] M. Schiegg, P. Hanslovsky, C. Haubold, U. Koethe, L. Hufnagel, F. A. Hamprecht, Graphical model for joint segmentation and tracking of multiple dividing cells, Bioinformatics 31 (2015) 948–956.
- [42] M. Schiegg, P. Hanslovsky, B. X. Kausler, L. Hufnagel, F. A. Hamprecht,  
680 Conservation tracking, in: ICCV, 2013, pp. 2928–2935.
- [43] E. Türetken, X. Wang, C. J. Becker, C. Haubold, P. Fua, Network flow integer programming to track elliptical cells in time-lapse sequences, IEEE Transactions on Medical Imaging 36 (4) (2017) 942–951.
- [44] N. Bansal, A. Blum, S. Chawla, Correlation clustering, Machine Learning  
685 56 (1-3) (2004) 89–113.
- [45] E. D. Demaine, D. Emanuel, A. Fiat, N. Immerlica, Correlation clustering in general weighted graphs, Theoretical Computer Science 361 (2-3) (2006) 172–187.
- [46] B. Andres, J. H. Kappes, T. Beier, U. Köthe, F. A. Hamprecht, Probabilistic image segmentation with closedness constraints, in: ICCV, 2011.  
690

- [47] B. Andres, T. Kröger, K. L. Briggman, W. Denk, N. Korogod, G. Knott, U. Köthe, F. A. Hamprecht, Globally optimal closed-surface segmentation for connectomics, in: ECCV, 2012.
- [48] B. Andres, J. Yarkony, B. S. Manjunath, S. Kirchhoff, E. Türetken, C. C. Fowlkes, H. Pfister, Segmenting planar superpixel adjacency graphs w.r.t. non-planar superpixel affinity graphs, in: EMMCVPR, 2013. 695
- [49] S. Bagon, M. Galun, Large scale correlation clustering optimization, CoRR abs/1112.2903.
- [50] T. Beier, B. Andres, U. Köthe, F. A. Hamprecht, An efficient fusion move algorithm for the minimum cost lifted multicut problem, in: ECCV, 2016. 700
- [51] T. Beier, F. A. Hamprecht, J. H. Kappes, Fusion moves for correlation clustering, in: CVPR, 2015, pp. 3507–3516.
- [52] T. Beier, T. Kroeger, J. H. Kappes, U. Köthe, F. A. Hamprecht, Cut, glue, & cut: A fast, approximate solver for multicut partitioning, in: CVPR, 2014, pp. 73–80. 705
- [53] J. H. Kappes, M. Speth, G. Reinelt, C. Schnörr, Higher-order segmentation via multicuts, *Computer Vision and Image Understanding* 143 (2016) 104–119.
- [54] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, B. Andres, Efficient decomposition of image and mesh graphs by lifted multicuts, in: ICCV, 2015, pp. 1751–1759. 710
- [55] S. Kim, C. D. Yoo, S. Nowozin, P. Kohli, Image segmentation using higher-order correlation clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (9) (2014) 1761–1774.
- [56] J. Yarkony, A. Ihler, C. C. Fowlkes, Fast planar correlation clustering for image segmentation, in: ECCV, 2012, pp. 568–581. 715

- [57] J. E. Yarkony, C. Fowlkes, Planar ultrametrics for image segmentation, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), NIPS, Curran Associates, Inc., 2015, pp. 64–72.
- 720 [58] M. Schneider, S. Hirsch, B. Weber, G. Székely, B. H. Menze, Joint 3-d vessel segmentation and centerline extraction using oblique hough forests with steerable filters, *Medical Image Analysis* 19 (1) (2015) 220 – 249.
- [59] A. Sironi, E. Türetken, V. Lepetit, P. Fua, Multiscale centerline detection, *IEEE Transactions on Pattern Analysis & Machine Intelligence* 1 (2015) 1–14.
- 725 [60] H. W. Kuhn, The hungarian method for the assignment problem, *Naval research logistics quarterly* 2 (1-2) (1955) 83–97.
- [61] J. Munkres, Algorithms for the assignment and transportation problems, *Journal of the society for industrial and applied mathematics* 5 (1) (1957) 32–38.
- 730 [62] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, J. Zhang, Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2) (2009) 319–336.
- 735 [63] P. Matula, M. Maška, D. V. Sorokin, P. Matula, C. Ortiz-de Solorzano, M. Kozubek, Cell tracking accuracy measurement based on comparison of acyclic oriented graphs, *PLOS ONE* 10 (12) (2015) 1–19.
- [64] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: *ICLR*, 2015, pp. 1–13.
- 740 [65] J.-Y. Tinevez, N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, S. Y. Bednarek, S. L. Shorte, K. W. Eliceiri, Trackmate: An open and extensible platform for single-particle tracking, *Methods* 115 (2017) 80–90.

- <sup>745</sup> [66] A. Mosinska, J. Tarnawski, P. Fua, Active learning and proofreading for delineation of curvilinear structures, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017, pp. 165–173.

## Appendix A. Derivation of the MAP Estimate

Here, we derive the MLTP as the MAP estimate of our probability measure  $P(\mathbf{x}|X_G, \Theta)$  of (2) over lineages encoded by the hypothesis graph. We start by absorbing the uniform prior over feasible solutions  $X_G$  into the constraints and follow by taking the logarithm,

$$\begin{aligned} \arg \max_{\mathbf{x} \in X_V} P(\mathbf{x}|X_G, \Theta) &= \arg \max_{\mathbf{x} \in X_G \cap X_V} \prod_{uv \in E} P(x_{uv}|\Theta) \prod_{v \in V} P(x_v^+|\Theta) \prod_{v \in V} P(x_v^-|\Theta) \\ &= \arg \max_{\mathbf{x} \in X_G \cap X_V} \sum_{uv \in E} \log P(x_{uv}|\Theta) + \sum_{v \in V} \log P(x_v^+|\Theta) + \sum_{v \in V} \log P(x_v^-|\Theta) , \end{aligned} \quad (\text{A.1})$$

where we find the feasible set  $\mathbf{x} \in X_G \cap X_V$  to be identical to the feasible set of the MLTP. In order to arrive at a linear objective function, we substitute the bilinear representation for the pseudo-boolean functions  $P(x|\Theta) = P(x=1|\Theta)^x P(x=0|\Theta)^{1-x}$  for each of the remaining terms, apply the logarithm to each of them and then turn maximization into minimization by multiplying with  $-1$ ,

$$\begin{aligned} (\text{A.1}) &= \arg \max_{\mathbf{x} \in X_G \cap X_V} \sum_{uv \in E} x_{uv} \log \frac{P(x_{uv}=1|\Theta)}{P(x_{uv}=0|\Theta)} \\ &\quad + \sum_{v \in V} x_v^+ \log \frac{P(x_v^+=1|\Theta)}{P(x_v^+=0|\Theta)} + \sum_{v \in V} x_v^- \log \frac{P(x_v^-=1|\Theta)}{P(x_v^-=0|\Theta)} \quad (\text{A.2}) \\ &= \arg \min_{\mathbf{x} \in X_G \cap X_V} \sum_{uv \in E} x_{uv} \underbrace{\left( -\log \frac{P(x_{uv}=1|\Theta)}{P(x_{uv}=0|\Theta)} \right)}_{c_{uv}} \\ &\quad + \sum_{v \in V} x_v^+ \underbrace{\left( -\log \frac{P(x_v^+=1|\Theta)}{P(x_v^+=0|\Theta)} \right)}_{c_v^+} + \sum_{v \in V} x_v^- \underbrace{\left( -\log \frac{P(x_v^-=1|\Theta)}{P(x_v^-=0|\Theta)} \right)}_{c_v^-} , \end{aligned} \quad (\text{A.3})$$

750 where we dropped all constant terms with respect to  $\mathbf{x}$  to arrive at (A.2). Finally, we identify the coefficients  $c_{uv}$ ,  $c_v^+$  and  $c_v^-$  of the MLTPs objective function.

## Appendix B. Additional Results

Further qualitative views complementary to Fig. 8 can be found in Fig. B.11–  
B.15. Each figure shows both fluorescence and LFM image, and the detectors  
755 cell probability map as overlay on the LFM image. Detail views show a zoomed  
version of the regions indicated by the black rectangle, once in the same form as  
the main image and once with either annotations or detections indicated by red  
circles. The depicted annotations originate from the fluorescence image only,  
but are also drawn onto the LFM image in order to facilitate relating visual  
760 appearance in these images to actual cells.

For completeness, we report performance metrics for each individual se-  
quence of the fluorescence-annotated dataset in Table B.3. The reported mini-  
mum and maximum cell count per sequence is calculated on the annotation.

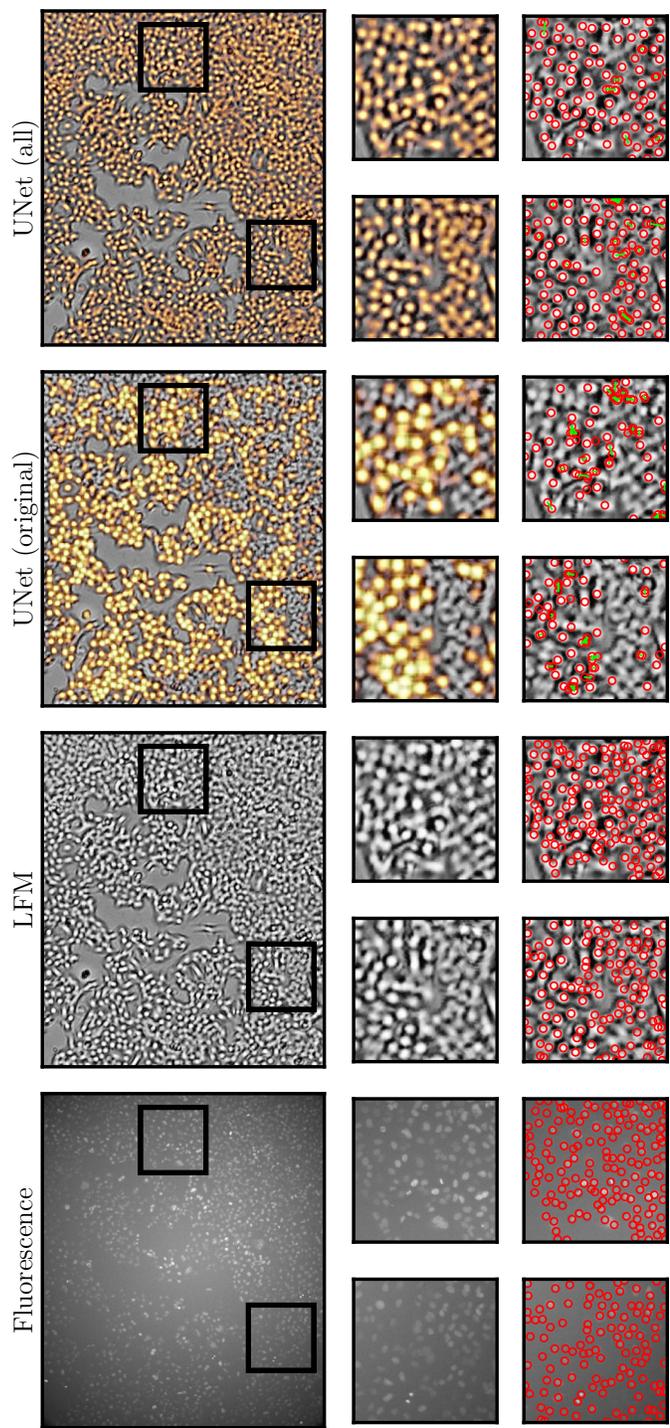


Figure B.11: Qualitative comparison on a sequence with a *dense* cell culture, corresponding to Fig. 8. From left to right: Fluorescence microscopy image, raw LFM image, LFM image with heatmap as overlay from a UNet trained on the original dataset (*original*) and retrained on fluorescence-based annotations (*all*). Detail views corresponding to the drawn rectangles are presented in the two lower rows. Red circles in the bottom row localize cells, originating from fluorescence-based annotations for fluorescence and LFM or generated from the detectors heatmap. For the latter two, green edges connect detections that were clustered together in the solution of the MLTP. Similar to the Resnet-23, the original version struggles with the varying sharpness in dense regions, while the retrained UNet handles this better and is able to distinguish closer cells.

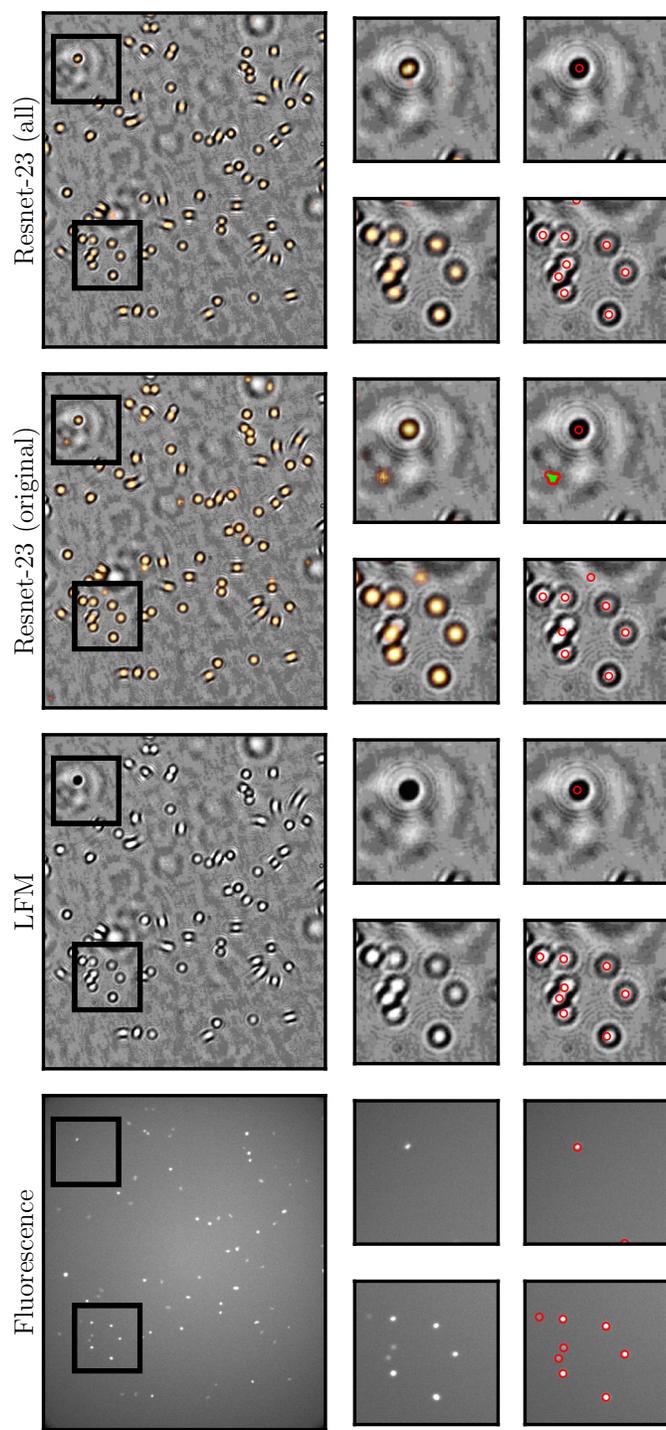


Figure B.12: Qualitative comparison on a sequence with a low cell culture density. The bottom rows show detail views corresponding to the rectangles, once containing raw images with cell probability maps as overlays (middle row) and once with annotations/detections drawn as red circles (bottom row). As before, annotations were created from the fluorescence images. We observe that the retrained Resnet-23 distinguishes the two close cells (left detail) and handles the larger artifacts better (right detail).

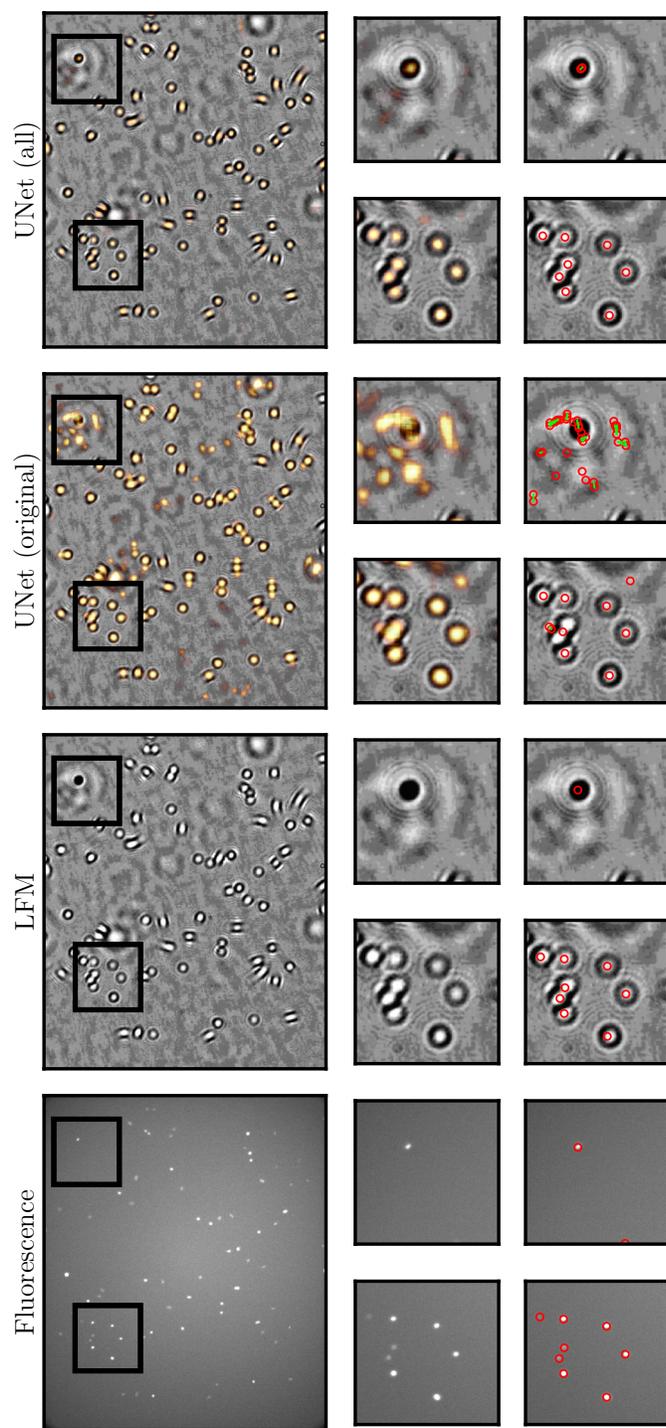


Figure B.13: Qualitative comparison on a sequence with a low cell culture density, corresponding to Fig. B.12 but with the UNet as detector. The bottom rows show detail views corresponding to the rectangles, once containing raw images with cell probability maps as overlays (middle row) and once with annotations/detections drawn as red circles (bottom row). As before, annotations were created from the fluorescence images. We observe the same trend as with Resnet-23 in Fig. B.12, only that the UNet does even worse in the presence of the large artifacts before retraining.

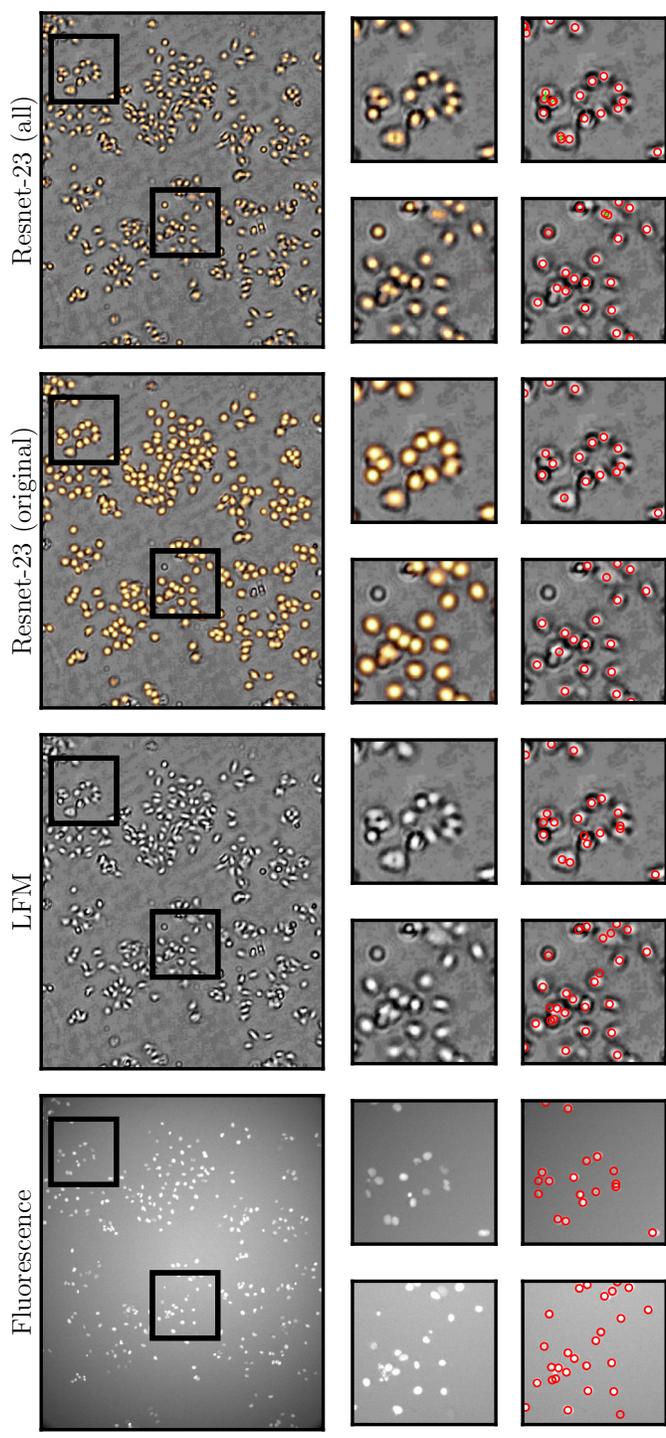


Figure B.14: Qualitative comparison on a sequence with medium cell culture density. The bottom rows show detail views corresponding to the rectangles, once containing raw images with cell probability maps as overlays (middle row) and once with annotations/detections drawn as red circles (bottom row). As before, annotations were created from the fluorescence images. Retraining the Resnet-23 on sequences with fluorescence-based annotations (all) improves its capabilities to discriminate several cells in close proximity. Nonetheless, a few are still missed, e.g. in the top-left cluster of the first detail view, where the interference patterns of five cells interact and appear in highly ambiguous form.

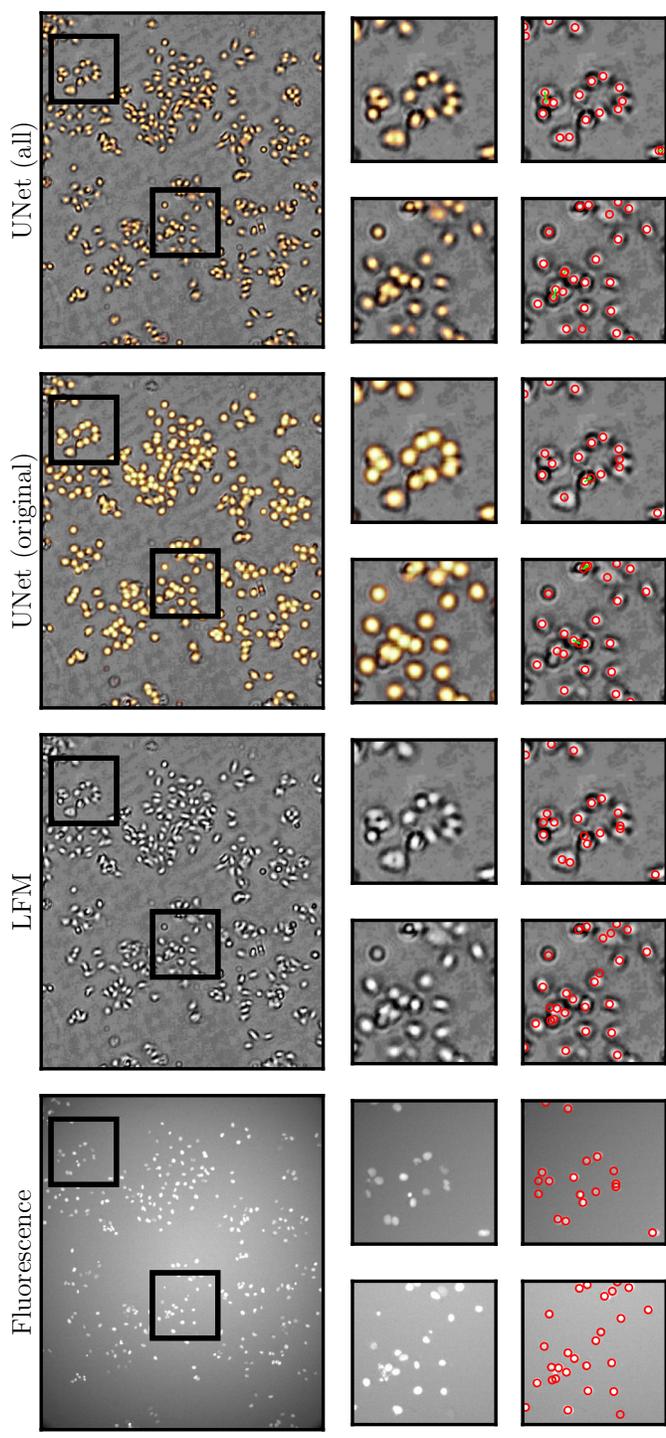


Figure B.15: Qualitative comparison on a sequence with medium cell culture density, corresponding to Fig. B.14 using the UNet as detector. The bottom rows show detail views corresponding to the rectangles, once containing raw images with cell probability maps as overlays (middle row) and once with annotations/detections drawn as red circles (bottom row). As before, annotations were created from the fluorescence images. We observe again that the detector trained on the smaller, original dataset does not distinguish cells that cluster together that well, while retraining it on the larger dataset with fluorescence-based annotations helps. Interestingly, the UNet errs on almost identical locations as the Resnet-23 after retraining.

Table B.3: Overview of performance metrics obtained on individual sequences of the fluorescence-annotated dataset with our method using Resnet-23 and MLTP.  $\#_{\min}$  and  $\#_{\max}$  denote the minimum and maximum number of cells in the ground truth annotation of the particular sequence. All metrics are shown in %. Averages are computed over all sequences with unit weight and reported in the last row.

| Cell line | $[\#_{\min}, \#_{\max}]$ | Prec. | Recall | F1    | MODA  | MOTA  | TRA   |
|-----------|--------------------------|-------|--------|-------|-------|-------|-------|
| A549      | [ 840, 2780]             | 98.10 | 68.07  | 80.20 | 98.73 | 69.82 | 54.81 |
|           | [ 581, 3436]             | 98.33 | 55.29  | 70.69 | 98.95 | 70.16 | 48.98 |
|           | [ 359, 1186]             | 94.98 | 79.50  | 86.54 | 95.70 | 65.22 | 57.88 |
|           | [ 390, 2079]             | 97.76 | 61.49  | 75.34 | 98.71 | 67.55 | 51.38 |
|           | [ 579, 1671]             | 94.53 | 73.61  | 82.76 | 95.41 | 67.29 | 57.30 |
| 3T3       | [ 826, 1072]             | 92.02 | 63.01  | 74.77 | 94.58 | 81.42 | 67.47 |
|           | [ 303, 537]              | 95.50 | 82.64  | 88.58 | 96.21 | 81.92 | 76.13 |
|           | [ 410, 732]              | 93.94 | 83.07  | 88.13 | 94.93 | 79.85 | 75.01 |
|           | [ 75, 149]               | 87.42 | 91.91  | 89.52 | 87.62 | 75.85 | 78.70 |
|           | [ 71, 139]               | 84.76 | 73.61  | 78.73 | 87.24 | 76.53 | 70.35 |
|           | [ 1730, 2481]            | 97.17 | 33.55  | 49.86 | 99.10 | 86.63 | 56.83 |
| HuH7      | [ 113, 200]              | 91.19 | 87.79  | 89.43 | 91.85 | 69.45 | 68.44 |
|           | [ 297, 741]              | 92.77 | 89.36  | 90.98 | 93.05 | 64.82 | 64.44 |
|           | [ 97, 171]               | 86.77 | 90.21  | 88.43 | 86.68 | 64.70 | 67.84 |
|           | [ 270, 573]              | 95.78 | 88.16  | 91.79 | 96.37 | 72.55 | 69.53 |
|           | [ 1533, 1897]            | 88.36 | 61.43  | 72.46 | 92.27 | 76.05 | 63.03 |
| Average   |                          | 93.09 | 73.92  | 81.14 | 94.21 | 73.11 | 64.26 |