JAN-US

Authors: Caroline Malin-Mayor, Peter Hirsch, Leo Guignard, Katie McDole, Yinan Wan, William C. Lemon, Philipp J. Keller, Stephan Preibisch, Jan Funke

Email: funkej@janelia.hhmi.org

Platform: Linux

Prerequisites: Python 3, MongoDB, pylp, zarr (and other standard python packages; we provide a singularity definition file with all dependencies included)

JAN-US: SUMMARY

We use sparse point annotations to train a convolutional neural network to predict at each pixel a cell indicator value that peaks at the center of each nucleus [1, 2], and a movement vector that points to the center of the same cell nucleus in the previous time frame [3, 4]. From these predictions, we generate a candidate graph in two steps: first, we place nodes at the local maxima of the cell indicator values to represent possible cell center locations, with a score to encode the network's confidence. Second, we locally connect nodes in adjacent frames with edges to represent the possibility that the nodes represent the same cell, and assign a score to each edge based on agreement with the predicted movement vector. Next, we solve a global constrained optimization problem on the candidate graph to select a subset of nodes and edges that form coherent lineage trees. We know that between time frames, cells can move, divide into two, enter or leave the field of view, or die, but not merge or split into more than two. Thus, we introduce hard constraints to prevent merging and divisions producing more than two progeny. The objective function incorporates prior knowledge that cell movement is much more common than division, death, and entering or leaving the field of view, encouraging long, continuous lineages by penalizing the start and end of tracks. These tree constraints and continuity costs are similar to those in previous work [5, 6, 7]; however, we also incorporate the node and edge scores generated by the neural networks into the objective function as learned costs. Thus, we optimize for valid lineages that are both continuous and supported by the learned cell location and movement features. Our Integer Linear Program (ILP) formulation of the optimization problem additionally allows solving piece-wise in parallel on large datasets by introducing additional constraints to ensure consistent solutions between adjacent regions. The method is described in detail in our preprint [7], together with results on other datasets.

JAN-US: PREPROCESSING

The training and testing volumes are converted into zarr containers. Gold-truth tracks for the training data (keeping only the centroid information provided by the challenge, not the segmentation) are stored

as CSV files. After training of the cell indicator and movement vector network, predictions on the testing volume are stored in zarr containers as well. Predictions outside the foreground are masked out. For **Fluo-N3DL-DRO** the foreground is determined automatically based on the raw image data. As this failed for **Fluo-N3DH-CE** a 2D polygon with about 10 points is drawn manually around a maximum intensity projection of each volume, this is extended in the *z* dimension and used as a mask.

JAN-US: TRACKING

We create a candidate graph $G = (V, E)$ where nodes represent possible cell center locations, and edges possible movements of cells across frames. The nodes are found as NMS detections of the cell indicator values. We construct the set of directed edges E by locally connecting nodes in adjacent frames with edges that point one frame backwards in time. For each candidate v at time t_v, we compute the predicted location $\hat{l}_v$ of the same cell in the previous frame: $\hat{t}_v = l_v + m_v$, where l_v is the position of the node and m_v is the predicted movement vector. Then, we add an edge from v to each node candidate u at time $t_v - 1$ where the distance between the predicted location and the actual location of u is less than a hyperparameter θ. The edge is further scored by this distance. A lineage tree is then found by optimizing an integer linear program to find a cost-minimal forest in the candidate graph. Costs are defined for the selection of nodes (based on the value of the cell indicator) and edges (based on the distance between prediction and actual location). Linear constraints ensure that the selected nodes and edges form a binary forest. See [7] for a detailed description of the optimization problem.

JAN-US: SEGMENTATION

Our method is designed as a detection plus tracking system and therefore does not natively provide segmentations. To overcome this, we utilize the cell indicator predictions. Our network predicts a map with peaks at the centers of predicted nuclei locations. To create instance segmentation masks we perform a seeded watershed with the detections selected by the ILP as seed points and the inverted cell indicator map as the watershed surface. We threshold this map at a value τ (hyperparameter). To further improve the segmentation we roughly estimate the nuclei size on the training set at up to 5 frames with varying nuclei size and count. We mask each instance with a sphere centered at the detection and with a size equal to the estimated nuclei size based on the next later estimated frame.

JAN-US: POST-PROCESSING

No post-processing step has been taken after segmentation.

REFERENCES

1. Höfener H, Homeyer A, Weiss N, Molin J, Lundström CF, Hahn HK. Deep learning nuclei detection: A simple approach can deliver state-of-the-art results. *Computerized Medical Imaging and Graphics* **70**, 43-52 (2018).
2. Kok RNU, Hebert L, Huelsz-Prince G, Goos YJ, Zheng X, Bozek K, Stephens GJ, Tans SJ, van Zon JS. OrganoidTracker: Efficient cell tracking using machine learning and manual error correction. *PLoS One* **15**, e0240802 (2020).
3. Hayashida J, Nishimura K, Bise R. MPM: Joint representation of motion and position map for cell tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3822-3831 (2020).
4. Sugawara K, Cevrim C, Averof M. Tracking cell lineages in 3D by incremental deep learning. bioRxiv:2021.02.26.432552, 2021.
5. Schiegg M, Hanslovsky P, Kausler BX, Hufnagel L, Hamprecht FA. Conservation tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 2928-2935 (2013).
6. Jug F, Levinkov E, Blasse C, Myers EW, Andres B. Moral lineage tracing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5926-5935 (2016).
7. Malin-Mayor C, Hirsch P, Guignard L, McDole K, Wan Y, Lemon WC, Keller PJ, Preibisch S, Funke J. Automated reconstruction of whole-embryo cell lineages by learning from sparse annotations. bioRxiv:2021.07.28.454016, 2021.