Applying Deep Learning Algorithms To Track Cells In Time Lapse Microscopy

Introduction: *Salmonella enterica* Serovar Typhimurium (STm) is a Gram-negative bacterium and a major cause of foodborne illnesses worldwide[1]. STm can be ingested or phagocytosed by macrophages, an immune cell type. Following phagocytosis, the host cell either degrades the bacteria or the bacteria replicates to high, medium, or low levels. However, the basis for these different STm fates remains unclear. One method to improve our understanding of the factors involved is to follow individual macrophages over the entire duration of infection. Live-cell microscopy or imaging is an ideal method for this, but it requires reliably tracking, or following, the cells over many hours. Current cell tracking algorithms can be computationally expensive and face challenges when used over a long time course. Through my eight-week research, I attempted to apply newer tracking algorithms that incorporate elements of deep learning, such as Deep Cell, Trackmate7, and Btrack to resolve these tracking issues[2–4]. Moreover, I optimized an analysis pipeline to track infected macrophages over time by integrating these tracking softwares to use on existing microscopy data.

Background: Tracking infected macrophages requires proper cell segmentation, or identification, and tracking. Previous methods for cell segmentation and tracking are not scalable, prone to human errors, and often fail due to issues related to proper cell identification following events such as cell division or death (Figures 1-2) or due to macrophages moving in irregular directions over a long time course. Potential solutions to these tracking issues are the recent application of deep learning to cell segmentation and tracking.

Deep learning allows computers to develop a systematic method to generate predictions after analyzing a vast amount of data. Regarding biological image analysis, deep learning has led to the development of advanced segmentation tools for cell nuclei and cytoplasm, such as Cellpose and Stardist[5-6]. Deep learning's application to tracking is not yet as developed as for segmentation; however, some recent examples of tracking algorithms that incorporate deep learning include Deep Cell, Btrack, and Trackmate7[2–4]. Deep Cell uses a deep learning model with a linear programming approach to analyze and track cells[2]. Btrack is based on Object Linking using the probabilities outputs of the Bayesian Belief Matrix using a Kalman filter based on the Cell State Classifier and motion model[3]. Trackmate7 integrates several detectors, such as the Labeled Image Detector to detect cells given segmented images, and trackers that track detected objects using algorithms such as the LAP tracker, which is based on the Linear Assignment problem (LASP)[4]. Ultimately, the three algorithms were evaluated to determine the optimal tracking algorithm to set up a pipeline incorporating segmentation and tracking.

Methods/Results: The following is an overview of the analysis I conducted to construct a tracking pipeline based on segmented nuclei data. The pipeline integrates segmentation along with tracking to thereby produce an efficient way to automate image analysis of macrophages across time. To begin tracking, I segmented 119 positions each with 73 frames using Cellpose on Quest, a High-Performance Computing (HPC) cluster, or collection of many separate computers called connected nodes that allow fast and parallel launching of jobs via BashScript[7]. To evaluate the performance of the segmentation, Segmentation Measure (SEG) is used based on an Intersection over Union or Jaccard similarity index score ranging from zero to one, where one indicates an exact match[8-9]. Cellpose was modified across both the cyto2 and nuclei models as

well as different Mask and Flow Thresholds and evaluated based on Ground Truth data that I manually created on one frame with 15 positions. Cellpose with the cyto2 model and a flow threshold of 0.6 and a mask threshold of 0.0 ultimately achieved a SEG score of 0.56, meaning around 50 percent of the labeled image overlaps correctly with the Ground Truth. Issues that contributed to the score include over-segmentation and problems identifying cells with overlapping borders (Figure 3). However, Stardist[5], another Deep-Learning segmentation algorithm based on the U-net Neutral network and star-convex polygons for the localization of nuclei, performed significantly better with a SEG score of 0.78, meaning over 75 percent of the labeled image overlaps correctly with the Grouth Truth. These results contributed to my decision to switch from Cellpose, which was initially proposed, to Stardist for segmentation.

Tracking was mainly performed using Trackmate7 and Deepcell due to time constraints. Deepcell tracking was implemented with a Python script and default parameters on its pre-trained Graph Neural Network tracking model. Deepcell's issues were noticeable from its excessive amount of tracks, which was around three times as much as the total cells; this was mainly due to misidentification of cells when there is excessive cell movement across frames, and cell misidentification after division (Figure 4). Attempts to alter the default parameters to generate a better result proved to output the same results; subsequently, attempts were made to alter the adjacency matrix used in Deepcell and they proved to show an effect therefore demonstrating inputs are being accepted but the pre-trained model isn't applying the inputs supplied (Figure 5). For trackmate7, local tracking for each position of 73 frames took less than one minute on Fiji/ImageJ and through a Jython script with the LAP tracker; the parameters were linking-max-distance of 45 microns, max-frame-gap of 2 frames, gap-closing max-distance of 45 microns, and segment-splitting max-distance of 45 microns[10]. One issue with Trackmate7 is that parent cells are being recognized after divisions, which can be shown in its lineage detection tool—Trackscheme (Figure 6). Ultimately the final pipeline was constructed based on Stardist and Trackmate7, consisting of a bash script for segmentation via Stardist on Quest and a Jython script for tracking with Trackmate locally.

Future direction: Potential follow-up studies will include evaluating the pipeline through a quantitative measure like the Tracking (TRA) score, which shows the cell tracking results as an acyclic-oriented graph with nodes representing detected cells and edges being temporal connections between cells over different time points. This will give a better representation of the accuracy of the overall pipeline[8-9]. In regards to Deepcell, future exploration will focus on its default model not responding to input parameters and subsequently the identification of parameters to optimize DeepCell tracking. Issues with Trackmte7 will also be further explored with adjustments to the parameters associated with feature penalties, frames of gap-closing, and max-distance of gap-closing and segment splitting of the LAP tracker.

Personal Statement: Through this project, I gained experience working with unfamiliar programming languages such as Bash Script and Jython as well as with running scripts on Quest through different conda environments. I also gained more knowledge on the overall processes associated with implementing deep-learning-based segmentation and tracking algorithms. I plan to take Biology 399 and machine learning internships to further explore the future directions associated with this project.

Appendix

Figure 1. Tracking challenges associated with correct cell lineage identification.
Multiple possible connections of cells between time points(t = 0, t = 1) are possible. Left Panel shows a raw image of the cells' position; the upper right panel shows an option for a potential cell connection while the lower right panel shows an alternative option.

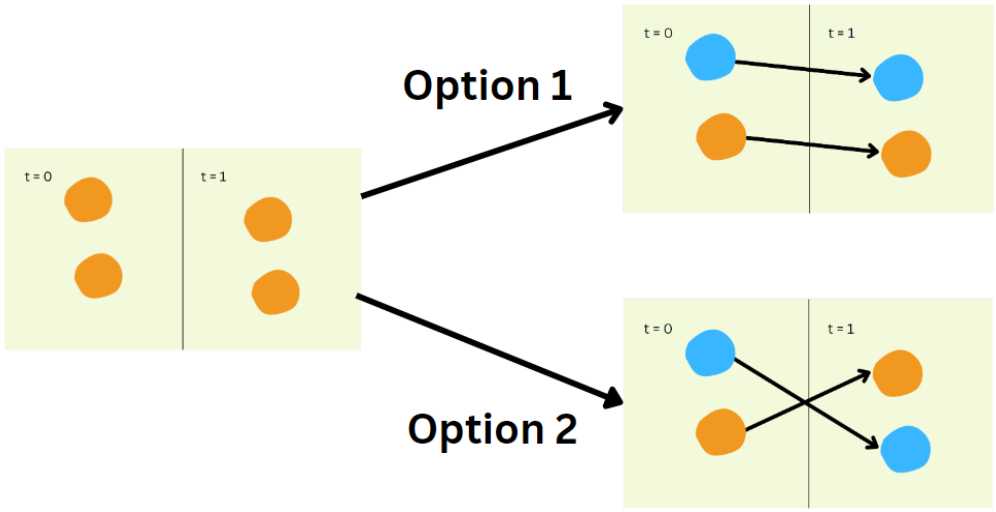
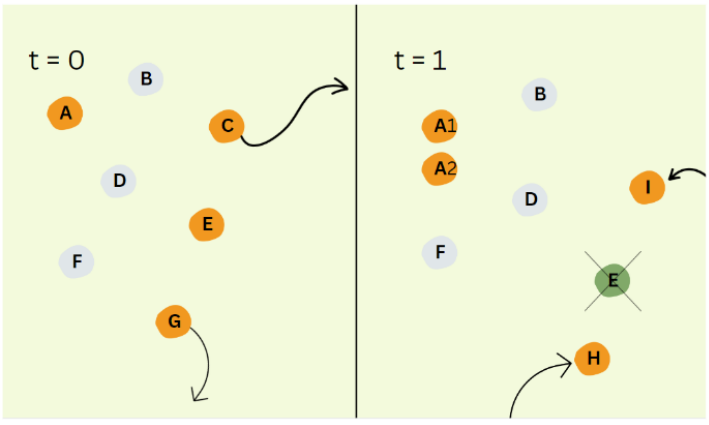


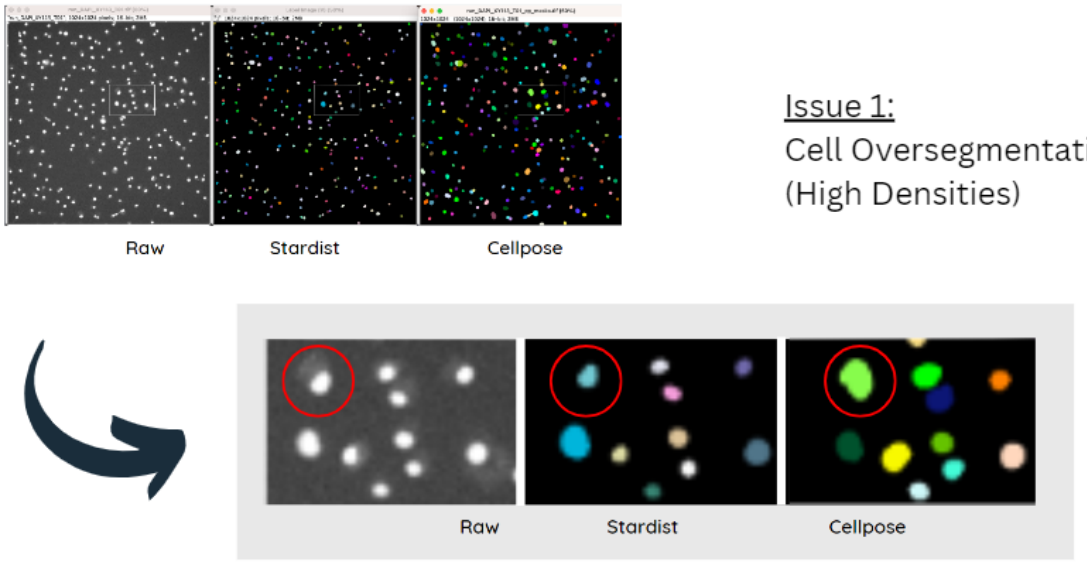
Figure 2. Tracking challenges associated with cell death, cell division, and cell entry and exit from the field of view on the microscope.
Panel shows cells between time points (t = 0, t = 1). Events that tracking needs to account for include cell division (A to A1, A2), cell death (E), cell entry (H, I), and cell exit (G, C). To identify these events correctly, predictions will be made using the intensity from the nuclear staining and distance from a previous cell.



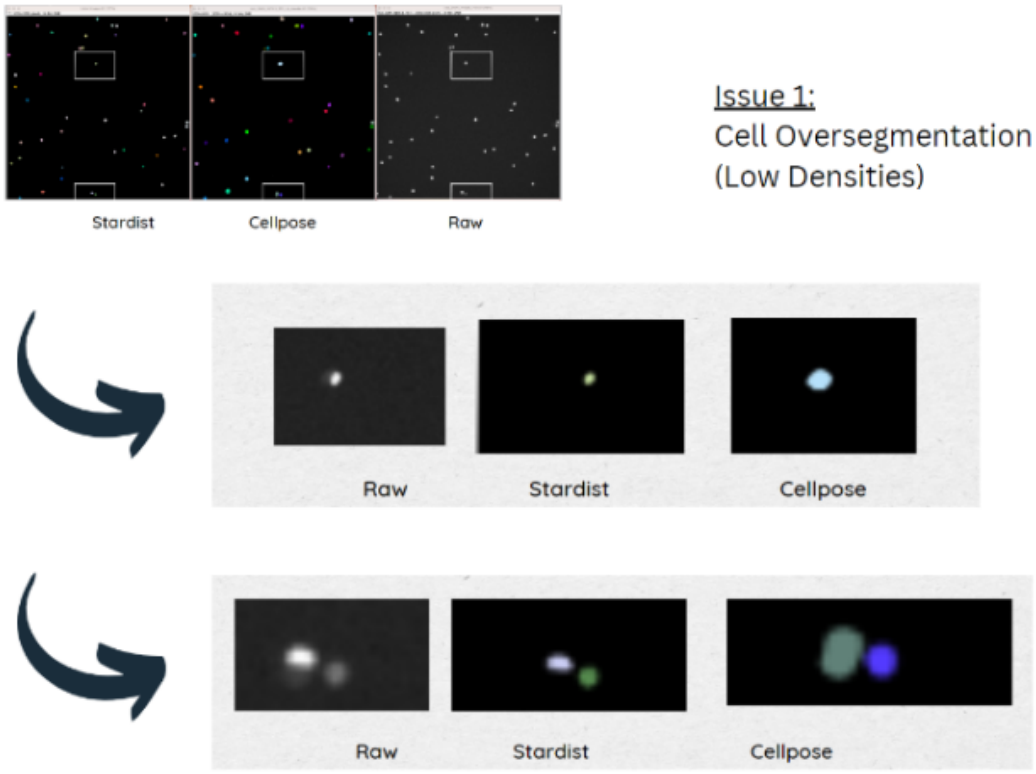
Challenges
Cell Division: A1, A2
Cell Death: E
Cell Entry: H, I
Cell Exit: G, C

Proposed Solutions
- Intensity from nuclear Staining
- Distance from previous cell

Figure 3. Segmentation issues associated with Cellpose.
(A) At high densities, the bottom right montage shows segmentation differences between the Raw image, Stardist segmented image, and Cellpose segmented image.



(B) At low densities, the bottom right montage shows segmentation differences between the Raw image, Stardist segmented image, and Cellpose segmented image.



(C) The bottom right montage shows segmentation differences between the Raw image, Stardust segmented image, and Cellpose segmented image.

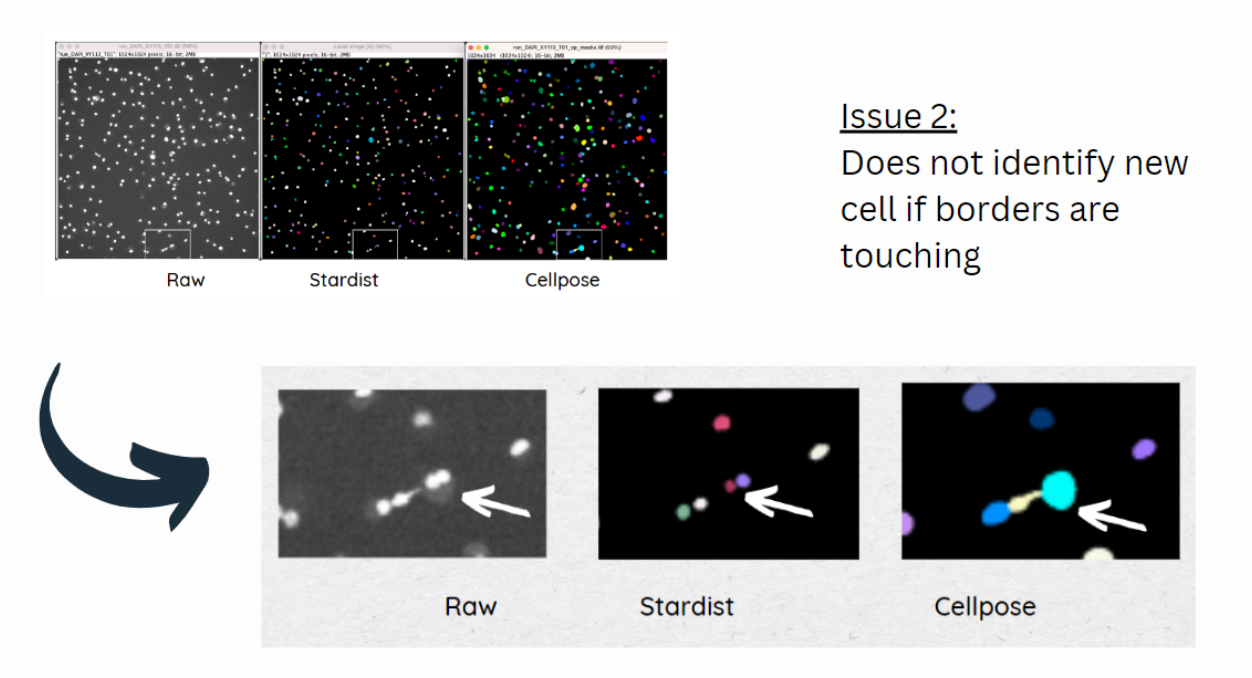
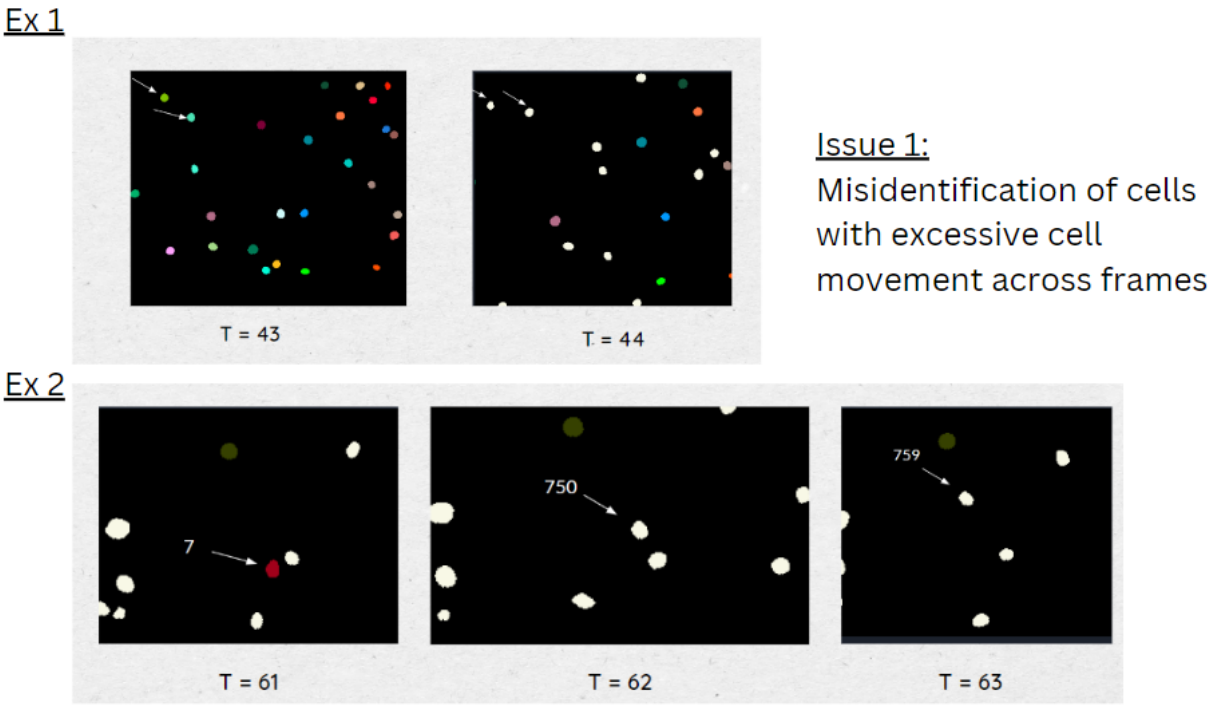


Figure 4. Tracking issues associated with Deepcell.
(A) Both examples of cell tracking done by Deepcell show changes in cell labeling over time after excessive cell movement, indicated by the change of colors across time (t) frames.



(B) Cell tracking done by Deepcell does not show changes in cell labeling after cell division, indicated by the same color across time(t) frames.

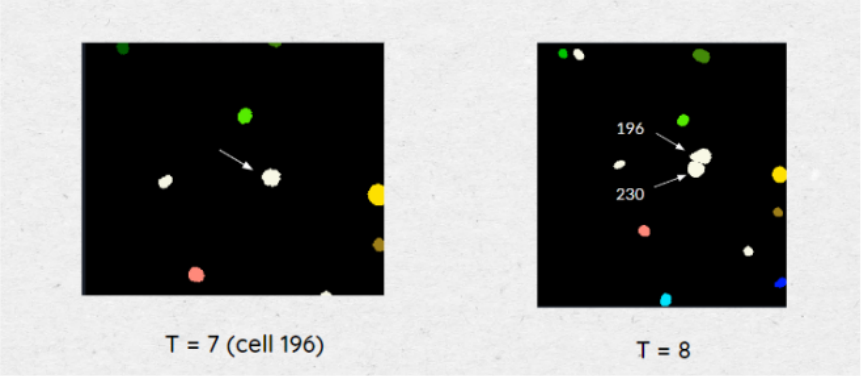


Figure 5. Attempts at Fixing Tracking issues associated with Deepcell.
Top panel shows the result of tracking after changing Distance Threshold and Disvison probability used in DeepCell's pre-trained GNN Cell Tracking model. Bottom panel shows visual representation of the Adjacency matrix used in LASP after changing the parameters to confirm input is being accepted.

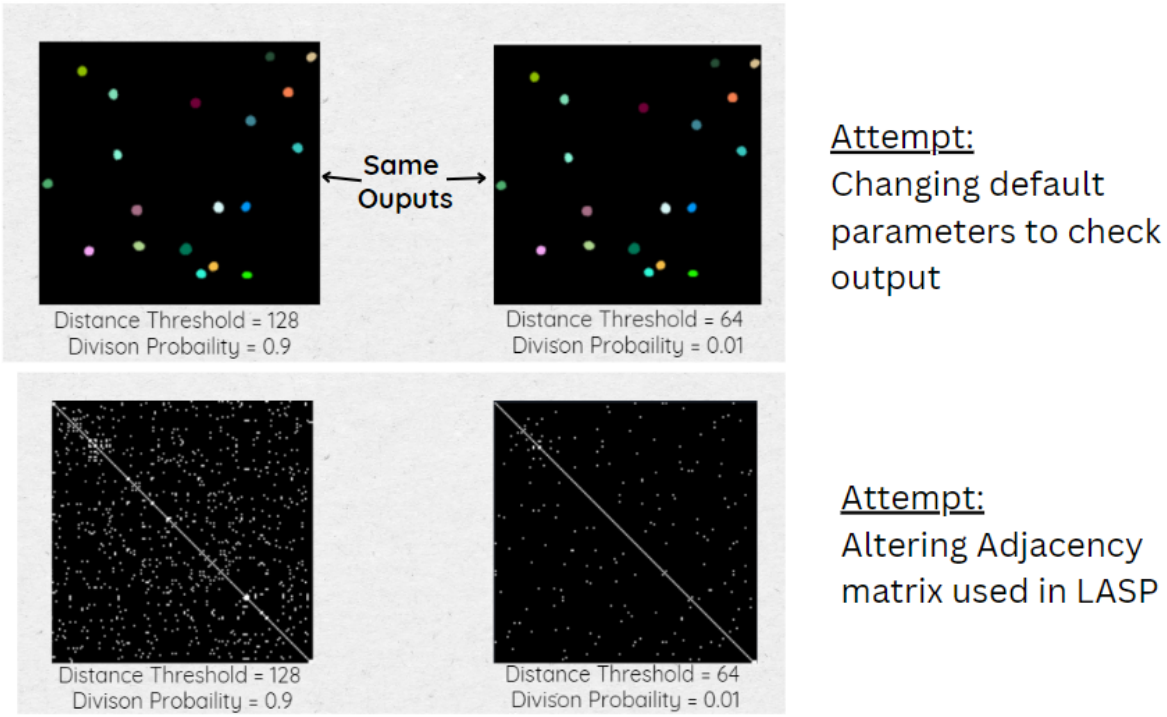
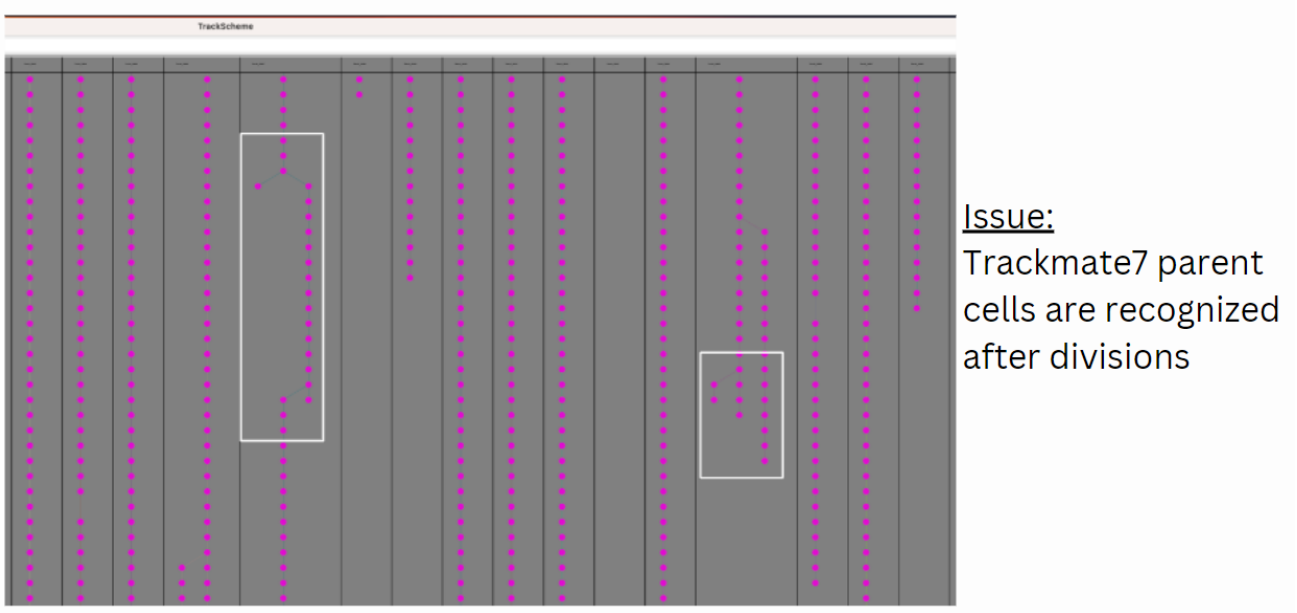


Figure 6. Tracking issues associated with Trackmate7.
Left Panel shows TrackScheme, Trackmate7's lineage detection tool. Each red dot represents a cell label and every vertical column of red dots represents the linkage of cell labels across time frames, where each new red dot in the vertical list is the detected cell label in the next frame. Boxed lineages show detections of parent cell labels immediately and given a certain frame gap after divisions.



References

1. Sun, H., Wan, Y., Du, P. & Bai, L. The Epidemiology of Monophasic Salmonella Typhimurium. *Foodborne Pathog. Dis.* **17**, 87–97 (2020).
2. Moen, E. *et al.* Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning. *bioRxiv* 803205 (2019) doi:10.1101/803205.
3. Ulicna, K., Vallardi, G., Charras, G. & Lowe, A. R. Automated deep lineage tree analysis using a bayesian single cell tracking approach. *Front. Comput. Sci.* **3**, (2021).
4. Ershov, D. *et al.* TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nat. Methods* **19**, 829–832 (2022).
5. Schmidt, U., Weigert, M., Broaddus, C. & Myers, G. Cell Detection with Star-Convex Polygons. in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II* (eds. Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C. & Fichtinger, G.) vol. 11071 265–273 (Springer International Publishing, 2018).
6. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods* **18**, 100–106 (2021).
7. Quest High-Performance Computing Cluster: Information Technology - Northwestern University. it.northwestern.edu/departments/it-services-support/research/computing/quest.
8. Maška, M. *et al.* A benchmark for comparison of cell tracking algorithms. *Bioinformatics* **30**, 1609–1617 (2014).
9. Matula, P. *et al.* Cell Tracking Accuracy Measurement Based on Comparison of Acyclic Oriented Graphs. *PLoS One* **10**, e0144959 (2015).
10. Schindelin, J. et al. Fiji: an open-source platform for biological-image analysis. *Nat.*

Methods **9**, 676–682 (2012).