

Ultrack: pushing the limits of cell tracking across biological scales

Jordão Bragantini^a, Ilan Theodoro^{a,b}, Xiang Zhao^a, Teun A.P.M. Huijben^a, Eduardo Hirata-Miyasaki^a, Shruthi VijayKumar^a, Akilandeswari Balasubramanian^a, Tiger Lao^a, Richa Agrawal^c, Sheng Xiao^a, Jan Lammerding^{c,d}, Shalin Mehta^a, Alexandre X. Falcão^b, Adrian Jacobo^a, Merlin Lange^a, Loïc A. Royer^a

^a Chan Zuckerberg Biohub, San Francisco, United States

^b Institute of Computing - State University of Campinas, Campinas, Brazil

^c Weill Institute for Cell and Molecular Biology - Cornell University, Ithaca, United States

^d Meinig School of Biomedical Engineering - Cornell University, Ithaca, United States

correspondence to {jordao.bragantini, loic.royer}@czbiohub.org

Tracking live cells across 2D, 3D, and multi-channel time-lapse recordings is crucial for understanding tissue-scale biological processes. Despite advancements in imaging technology, achieving accurate cell tracking remains challenging, particularly in complex and crowded tissues where cell segmentation is often ambiguous. We present Ultrack, a versatile and scalable cell-tracking method that tackles this challenge by considering candidate segmentations derived from multiple algorithms and parameter sets. Ultrack employs temporal consistency to select optimal segments, ensuring robust performance even under segmentation uncertainty. We validate our method on diverse datasets, including terabyte-scale developmental time-lapses of zebrafish, fruit fly, and nematode embryos, as well as multi-color and label-free cellular imaging. We show that Ultrack achieves state-of-the-art performance on the Cell Tracking Challenge and demonstrates superior accuracy in tracking densely packed embryonic cells over extended periods. Moreover, we propose an approach to tracking validation via dual-channel sparse labeling that enables high-fidelity ground truth generation, pushing the boundaries of long-term cell tracking assessment. Our method is freely available as a Python package with Fiji and napari plugins and can be deployed in a high-performance computing environment, facilitating widespread adoption by the research community.

Advancements in live imaging technologies¹⁻⁷ have enabled the visualization of cellular dynamics with unprecedented spatiotemporal resolution⁸ over large fields of view^{6,7} encompassing thousands of cells,^{5,9} producing vast amounts of multi-dimensional data.¹⁰ However, one key challenge remains: the accurate reconstruction of cell trajectories and lineages in complex biological systems.¹¹ This ability unlocks deeper insights into cellular state and behavior,^{9,12,13} tissue mechanics,¹⁴ morphogenesis,^{2,15,16} and regeneration.¹⁷

Cell segmentation and tracking have been persistent challenges in bioimage analysis.¹¹ While cell segmentation has advanced rapidly with the advent of deep-learning-based methods,¹⁸⁻²⁰ cell tracking continues to be an open problem.¹¹ The primary challenge stems from the accumulation of segmentation errors over time, hindering long-term tracking, especially in dense and dynamic cellular environments. Most automatic tracking methods employ a two-step approach: first segmenting cells, then linking them across time frames.²¹⁻²⁴ While computationally efficient, this approach struggles with the compounding of errors over time, particularly in dense tissues²⁵ or when cells divide rapidly.

Simultaneous segmentation and tracking offers a promising alternative,²⁶⁻³⁰ wherein multiple candidate segments and/or inter-segment links are generated, and then optimal candidates are selected by satisfying biological constraints (*i.e.* cells divide but do not merge) and

optimizing specific criteria (*i.e.* only keep links best supported by the data). However, existing methods in this category are constrained by their inability to utilize arbitrary segmentation inputs,^{28,31} as well as limitations in data scale or dimensionality due to high computational costs.^{29,30} These factors have limited their applicability to tissue-scale problems.

To address these challenges, we developed Ultrack, a robust and scalable method for large-scale cell tracking that excels even under segmentation uncertainty. Ultrack can track cells (or nuclei) in 2D, 3D, and multi-channel datasets, accommodating a wide range of biological contexts. Ultrack seamlessly integrates with various segmentation algorithms, including state-of-the-art deep learning-based cell segmentation tools^{18-20,32-34} (Fig. 1a). By jointly evaluating candidate segmentations and tracks, Ultrack employs temporal consistency to select the most accurate segments (Fig. 1b). This approach leverages information from adjacent time points to resolve cell segmentation and tracking ambiguities, enhancing performance, particularly in complex, densely packed tissues where cell boundaries are often uncertain (Supp. Video 1).

We demonstrate the utility and robustness of Ultrack through a comprehensive set of benchmarks and real-world applications, including top-scoring performance on the Cell Tracking Challenge.¹¹ Our method shows improved tracking performance across various use cases and scales, from multi-color and label-free 2D datasets

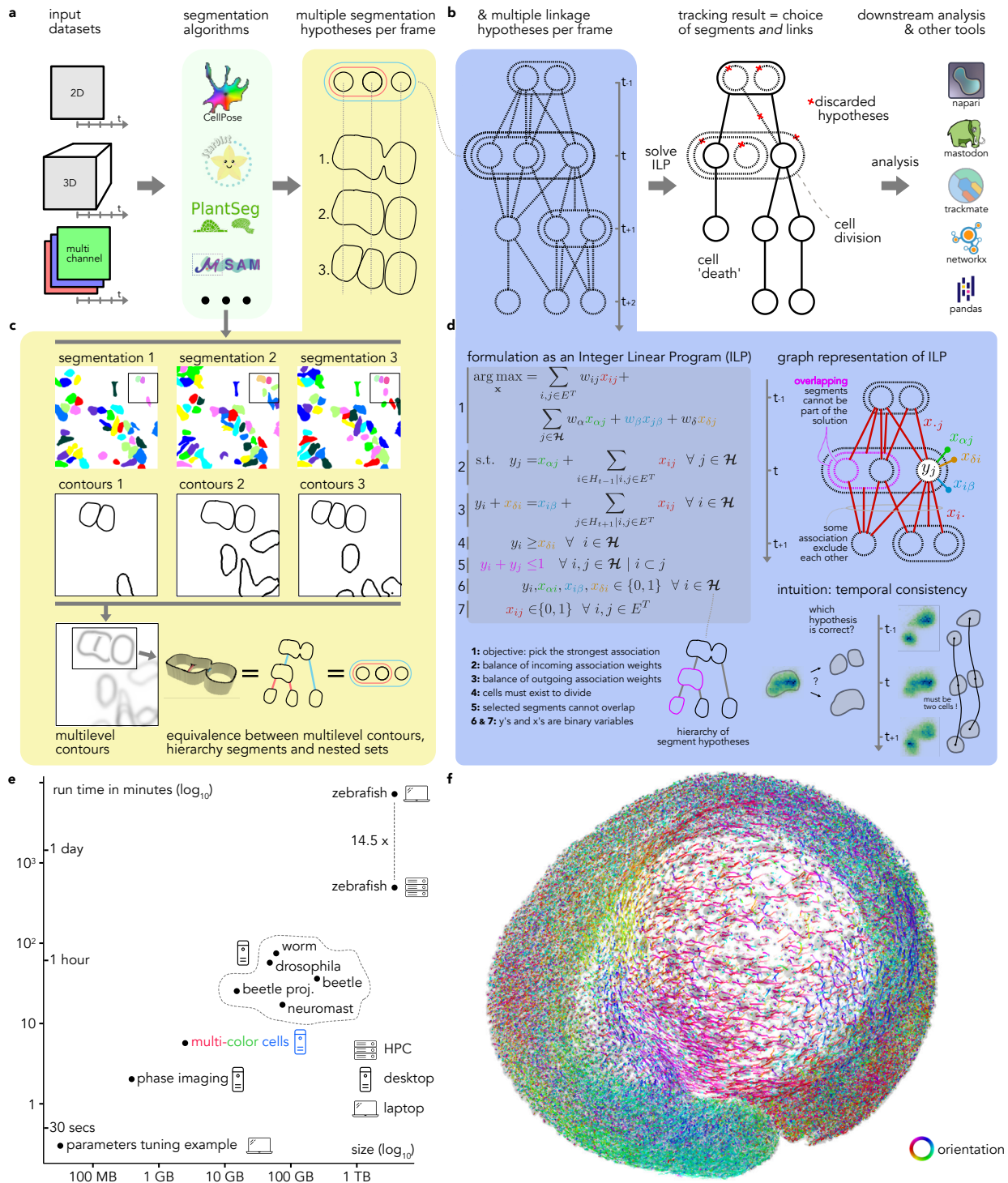


Fig. 1 | Ultrack overview. **a**, The Ultrack pipeline can process a variety of input images, including 2D, 3D, and multi-color datasets. These images are then processed by any segmentation algorithm or a combination of them. **b** Ultrack builds segmentation hypotheses between frames for tracking and solves an Integer Linear Programming (ILP) problem to identify cell segments and their trajectories. The resulting tracks can be exported in various formats compatible with multiple platforms. **c**, Illustration of how segmentation hypotheses are built using ultrametric contours: multiple segmentations are provided (first row), their binary contours are extracted (second row), and these are combined into a multilevel contour, equivalent to a hierarchy (third row). **d**, Joint segmentation and tracking is performed using an Integer Linear Programming (ILP) formulation, modeling cell behaviors (e.g. cell division, cell death, or cells leaving the field of view) while finding non-overlapping cells with maximal association over time. **e**, Ultrack can process arbitrarily sized datasets and scales from a laptop to a High-Performance Computing (HPC) cluster. **f**, Projection of a 3D image of a developing zebrafish embryo, with cell tracks overlaid and colored to indicate track orientation in the xy-plane.

to large-scale 3D time-lapse sequences of developing embryos.

Additionally, we introduce a novel dual-channel sparse labeling approach that enables the generation of high-fidelity ground truth for tracking validation. This method combines ubiquitous fluorescence labeling with sparse, random labeling at a distinct wavelength, allowing for the creation of annotations that would be impossible to produce manually. This innovative validation strategy pushes the limits of cell tracking assessment, enabling the evaluation of tracking performance over extended time periods and in increasingly complex tissue environments.

To ensure broad accessibility and ease of use, we provide intuitive interfaces in Fiji,³⁵ napari,³⁶ Python, and command line, catering to researchers with diverse computational backgrounds. We also made efforts to ensure interoperability with existing popular cell tracking frameworks such as TrackMate³⁷ and Mastodon³⁸ (see Fig. 1b). Thus, by enabling more accurate, scalable, and accessible cell tracking, Ultrack can accelerate research in developmental biology, cancer research, and regenerative medicine, offering new insights into cellular behavior and tissue dynamics.

RESULTS

Avoiding premature mistakes with multiple segmentation hypotheses.

Cell tracking accuracy is compromised when cell segmentation (or detection) fails. To address this challenge, Ultrack considers multiple segment hypotheses instead of prematurely committing to a particular choice of segments (see Fig. 1a). These hypotheses can be obtained from the original imaging data pixel intensities, multiple algorithms (e.g. watershed plus Cellpose), or from one or several algorithms configured with different parameters. This approach allows the integration of various segmentation methods, maintaining compatibility with the growing array of novel off-the-shelf segmentation models.^{39,40} However, the main challenge of this approach and previous joint segmentation and tracking methods, as well as the main obstacle for their practical application, is the potentially large number of possible segmentation hypotheses, especially for large multi-dimensional datasets.

Efficient representation of segmentation hypotheses.

To address the challenge of managing multiple segmentation hypotheses, we use ultrametric contour maps (UCMs),⁴¹⁻⁴³ which provide a compact representation of these hypotheses. Intuitively, UCMs are *multi-level contour maps* that represent a hierarchy of possible cell boundaries, where the strength of each boundary reflects its likelihood of being a true cell edge (Fig. 1c). This hierarchical structure allows for the efficient encoding of multi-

ple possible segmentations within a single map. More formally, UCMs partition space into nested segments, where valid segments do not overlap, or are contained within one another. This property enables Ultrack to efficiently consider multiple segmentation possibilities while tracking cells, even in terabyte-scale datasets (see Fig. 1e), that are challenging for existing joint segmentation and tracking solutions.^{26,28-30}

Hence, the canonical Ultrack input consists of two maps for each time frame: (i) a foreground map distinguishing potential cells from the background and (ii) a grayscale image representing the multilevel contour map (i.e., UCM), Fig. 1c. These two input maps can be generated from any method, making Ultrack compatible with a wide range of existing workflows (see Methods for technical details). Therefore, the same algorithm is always applied in every application presented here, and the varying factor is the process of obtaining the multilevel contour map.

Tracking by maximizing temporal consistency.

Once all candidate segments and links between segments are determined, tracking involves solving a complex combinatorial problem of selecting and associating the correct cell segments across time from the set of multiple segmentation and linkage hypotheses (see Fig. 1b and Supp. Video 1). As shown in Fig. 1d, we formulate this combinatorial problem as an Integer Linear Program (ILP)^{28,44} that simultaneously solves for the optimal selection of temporally consistent segments while adhering to biological constraints. These constraints include cell division, cells entering (e.g. moving into the frame), or exiting (e.g. cell death or moving out of the field of view), and ensuring that a pixel is not assigned to more than one cell.⁴⁵ This efficient and versatile mathematical formulation leverages highly optimized solvers (Gurobi,⁴⁶ Coin⁴⁷) and thus can handle tens of millions of segments from terabyte-scale datasets (see Fig. 1e). Once solved, the ILP formulation not only selects the most appropriate segments but also encodes the linkage data necessary to reconstruct the cell lineages from the optimal association between the selected segments (see Methods for associations score definition and ILP formulation details). Recapitulating cell lineages and migration patterns, as shown in the developing zebrafish embryo in Fig 1f and Supp. Video Supp. Video 1

Optimal parameter selection by temporal consistency.

Image segmentation techniques, from classical methods like watershed to advanced deep-learning models, face significant challenges in parameter tuning due to several factors: (i) computational: extensive resources are required to explore a large parameter space; (ii) practical: users often lack the expertise to tune parameters effectively; and (iii) technical: a single optimal parameter set

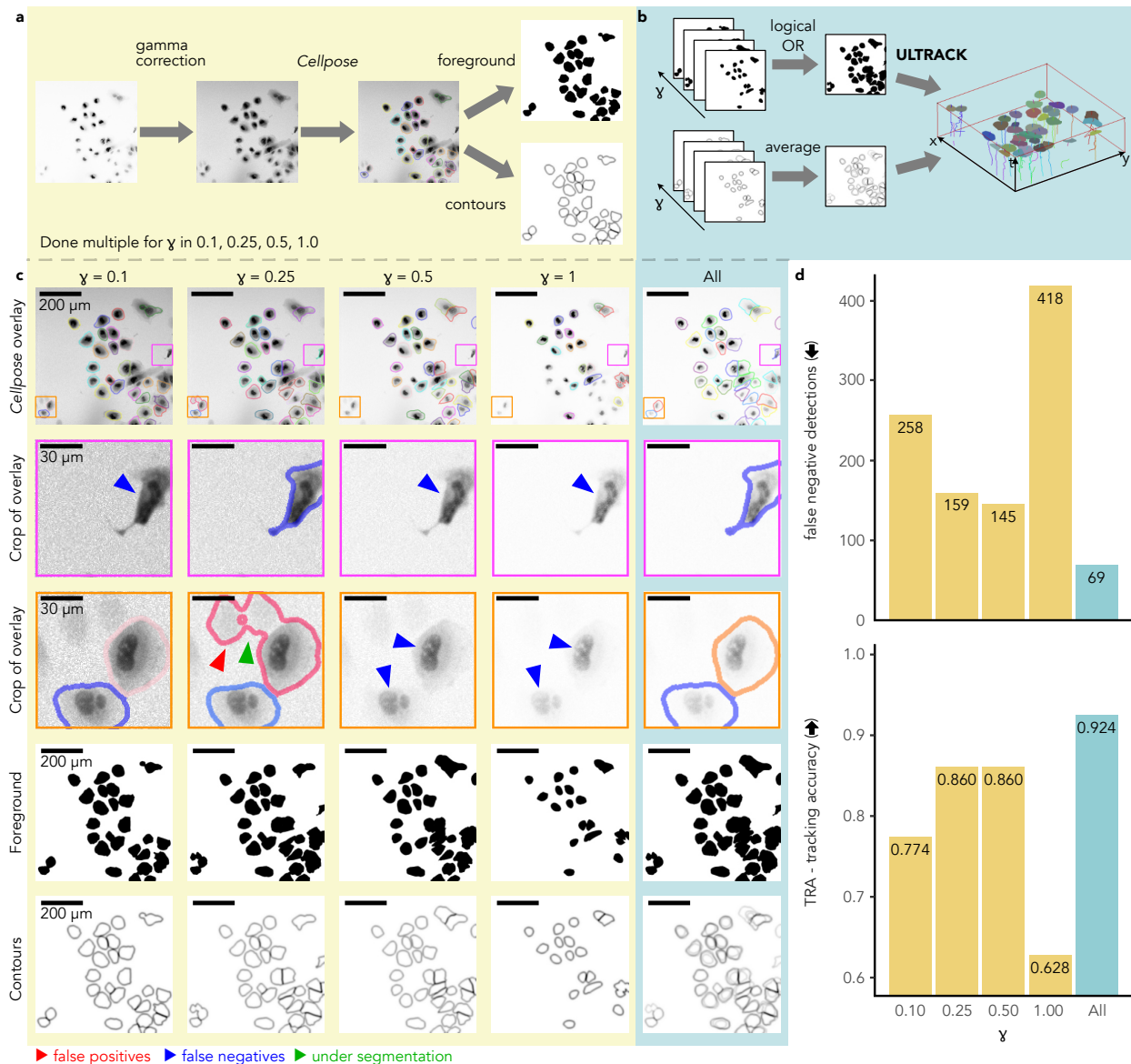


Fig. 2 | Multiple segmentation hypotheses alleviate the curse of parameter tuning. **a**, Segmentation pipeline using gamma correction with varying settings ($\gamma = 0.1, 0.25, 0.5$, and 1.0) to generate diverse segmentation hypotheses. **b**, Ultrack's approach: integrating multiple foreground and contour maps for joint segmentation and tracking. **c**, Impact of gamma correction on segmentation quality: original images (top row), segmentation results (second and third rows) with errors highlighted by colored arrows, foreground masks (fourth row), and contour maps (fifth row). **d**, Quantitative comparison of false negative cell detections and tracking accuracy (TRA) for individual gamma settings and Ultrack's combined approach.

may not fit all aspects of a diverse dataset, such as when segmenting cells of varying sizes.

Ultrack addresses these challenges by integrating multiple segmentation labels derived from different parameter settings into a single multilevel contour map. This approach leverages the temporal consistency of the tracking process to select the most appropriate segments for each cell (see Fig. 1d), resulting in more effective segmentation than what could be achieved with any single parameter

setting — even the optimal one.

To demonstrate this key feature of Ultrack, we tracked human hepatocarcinoma-derived cells expressing YFP-TIA-1⁴⁸ in their nucleus and cytoplasm from the Cell Tracking Challenge (CTC)¹¹ using the Cellpose "cyto2" weights³² without any fine-tuning. In this dataset, cells have significantly varying intensities, impairing the model's ability to consistently segment every cell correctly. For example, using original image intensities ($\gamma = 1.0$), the segmenta-

tion model fails to detect dimmer cells (see Fig. 2c – second column from right). Conversely, altering the dynamic range by taking the power of the image intensities with values closer to zero ($\gamma = 0.1, 0.25, 0.5$) improved the segmentation of dimmer cells but led to incorrect segmentation or non-detection of now-saturated bright cells (Fig. 2c – right most column).

To segment and track the combination of all parameterizations of gamma, we extracted the binary foreground and contour maps from the Cellpose labels, Fig. 2a. Next, we combined them into a single binary foreground and a multilevel contour map for tracking, Fig. 2b. Further displayed in Supp. Video 2.

We evaluated the tracking and segmentation performance for each individual gamma setting, as well as for the combined contour map approach. To assess performance, we compared the results against the manually curated ground truth provided by the Cell Tracking Challenge (CTC). Figure 2d summarizes the average results obtained from the two publicly available datasets. We used two key metrics for evaluation: the false negative detection rate and the tracking accuracy (TRA).⁴⁹ For false negative detection, lower scores indicate better performance, while for TRA, higher scores are better (see Methods for details).

These results demonstrate that Ultrack achieves more accurate tracking by leveraging multiple segmentation hypotheses, even outperforming the optimal single parameter setting ($\gamma = 0.5$). This indicates that combining diverse segmentation inputs is often more effective in practice, thus eliminating the need for precise parameter tuning and offering a more robust approach to cell tracking.

Enhancing cell tracking accuracy through multi-channel integration and color-aware associations.

Multi-color labeling has been shown to enhance cell segmentation and tracking capabilities, particularly in complex cellular environments.^{50,51} However, applying state-of-the-art deep-learning models to such data often requires training on dataset-specific annotations, as pre-trained weights are typically optimized for single-channel data. Ultrack addresses this challenge by combining multiple segmentation inputs in two ways: by varying parameters, as demonstrated earlier, and by integrating outputs from different segmentation algorithms applied to separate color channels.

We demonstrate this key feature of Ultrack on a three-channel "multi-color" dataset of metastatic breast adenocarcinoma cells (MDA-MB-231) with RGB-markings from a lentiviral gene ontology (LeGO) vector system,⁵² and show that we can effectively use models like Cellpose's "cyto2"³² alongside classical algorithms by applying them independently to each channel and then combining the outputs into a single multilevel contour representation.

This approach circumvents the need for extensive re-training or fine-tuning typically required by other approaches.^{37,53} We combined six segmentation outputs from three image channels — three from Cellpose and three from Otsu⁵⁴ with watershed^{55,56} — into a single contour and detection map for Ultrack (Fig. 3a). While classical image processing approaches like watershed are generally less precise at splitting cell instances, they provide greater control and allow for the detection of dimmer cells, complementing the more sophisticated Cellpose output. Furthermore, Ultrack can use the three color channels as a feature when associating segments between frames, which helps connect the right segments across time (Fig. 3b).

We assessed the effectiveness of our multi-channel multi-algorithm approach using progressively more sophisticated strategies: (i) Individual pipelines using Cellpose and watershed algorithms on a grayscale version (maximum intensity across multiple channels); (ii) Each algorithm applied independently on multiple channels; (iii) Combination of outputs from both algorithms in the multi-color configuration (Fig. 3a); (iv) Integration of color features into the association scores to improve linking of segments.

We manually curated 413 cell tracks, creating a gold-standard dataset for benchmarking our results. Demonstrating several key improvements. Incorporating color channels enhances cell distinction compared to grayscale versions, notably reducing under-segmentation (green arrow, Fig.3b, first two columns). The combination of watershed and Cellpose algorithms proves particularly effective, improving the detection of dimmer cells while maintaining Cellpose's accurate segmentation. Furthermore, color-feature linking improves tracking in ambiguous scenarios (Fig.3b, bottom center). Overall, Ultrack steadily increased accuracy as additional features (*i.e.*, segmentation labels, color information) were incorporated. This demonstrates its effectiveness in creating valid lineages (Fig. 3c and Supp. Video 3) from a large pool of segmentation hypotheses without requiring a fine-tuned segmentation model, highlighting its versatility and robustness in multi-color cell tracking scenarios.

Intensity-based tracking from label-free virtual staining.

As demonstrated in the previous section, off-the-shelf segmentation models face challenges when datasets deviate significantly from the training distribution. Users then often resort to time-consuming tasks such as annotating multiple segmentation labels to fine-tune a new model or implementing custom algorithms based on their expertise. This is particularly common because of the high diversity of imaging modalities and datasets typically encountered in Biology. Here, we show that our approach

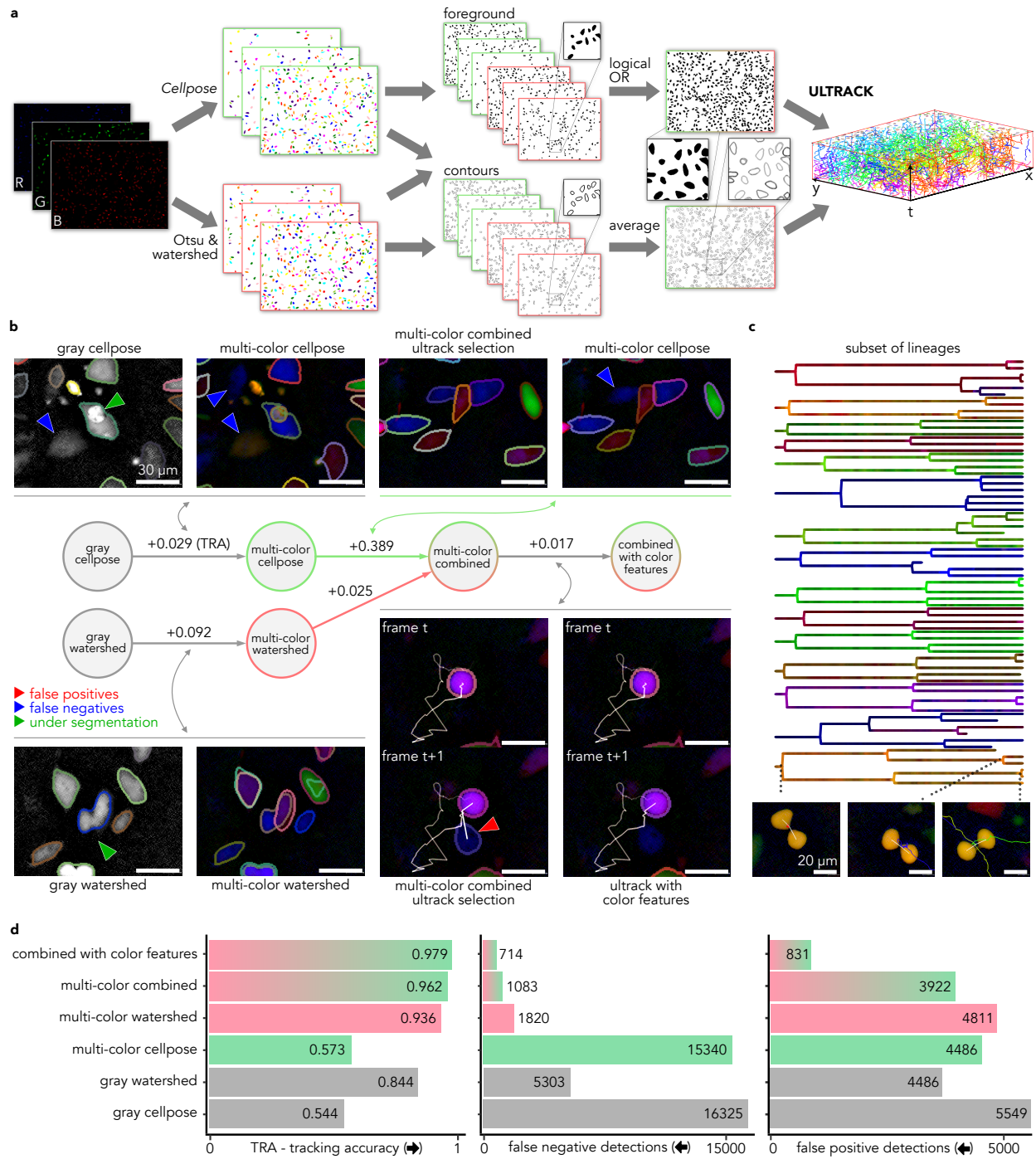


Fig. 3 | Ultrack enhances multi-color cell tracking by integrating diverse segmentation algorithms. **a**, Multi-channel, multi-algorithm tracking pipeline: Individual labels are generated for each color channel, from which foregrounds and contours are derived and subsequently integrated for joint segmentation and tracking. **b**, Stepwise improvement of segmentation strategies (left to right): (i) individual algorithms applied to grayscale images, (ii) applications to multi-color data, (iii) combination of labels from both algorithms and colors, and (iv) enhancement with color features in association scores. The central diagram illustrates the progression of methods, with numbers indicating improvements in the TRA metric. Colored arrows highlight specific error types. **c**, Representative subset of lineages recovered by Ultrack, showcasing cells' respective colors and examples of captured division events. **d**, Quantitative evaluation of tracking accuracy for the approaches described in **b**.

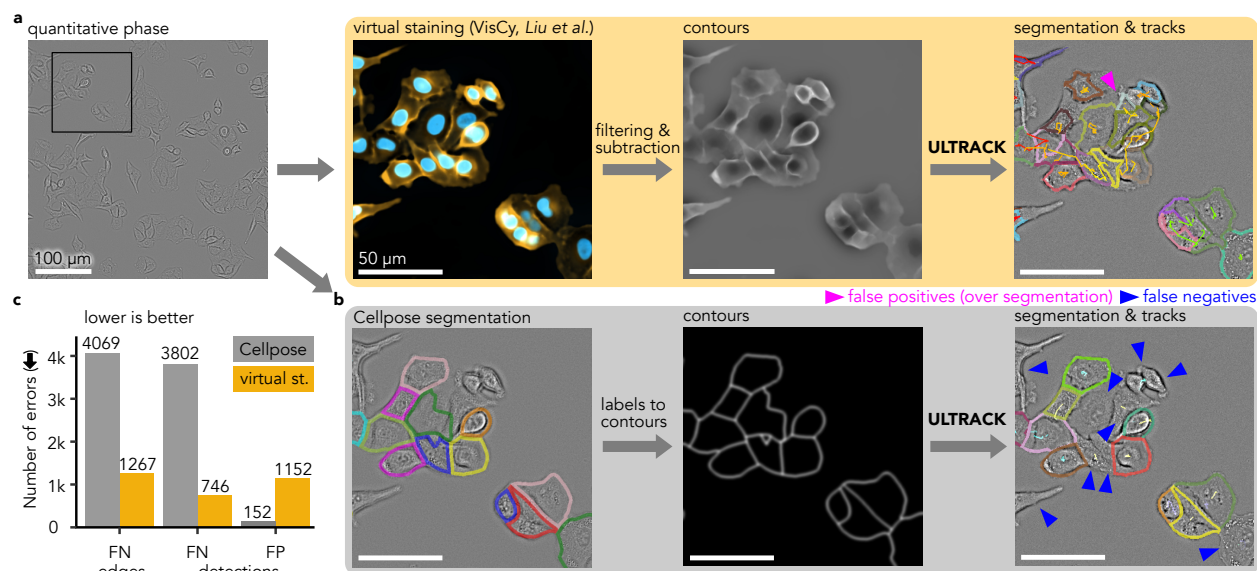


Fig. 4 | Ultrack improves cell tracking in label-free imaging by leveraging virtual staining. **a**, Pipeline for intensity-based joint segmentation and tracking from label-free quantitative phase imaging: Quantitative phase image (top left) is processed by VisCy⁵⁷ to generate virtual staining of nuclei and membrane (second from left). These virtual stains are used to derive contours (third from left) and foreground masks (not shown) through filtering, subtraction, and thresholding. Ultrack uses these inputs to perform tracking (right). **b**, We compare the above against first applying a Cellpose segmentation to the Quantitative phase image (bottom left), then deriving contours (and foreground mask, not shown), and then applying Ultrack (right). False positives (over-segmentation) are indicated in magenta, and false negatives in blue. **c**, Quantitative comparison of tracking errors using Cellpose⁵⁸ segmentations (gray) versus virtually stained membranes (orange) as input. The bar chart shows the number of false negative (FN) edges, false negative (FN) detections, and false positive (FP) detections for each method.

can circumvent these requirements by directly tracking image intensities as if they were multilevel contours, thus eliminating the need for segmentation labels as input.

Our method relies on a single assumption: the intensity map must highlight cell boundaries in some way. This could be achieved through membrane labeling, which delineates cells as a contour map, or through nuclear labeling, which highlights cell interiors. By inverting the latter, we obtain an image where boundaries have higher intensity values than cell interiors.

To evaluate this approach, we tracked A549 cells using quantitative phase imaging (QPI) data acquired with recOrder⁵⁹ on a widefield transmission microscope. Traditional segmentation models like Cellpose perform poorly on this data because QPI was not part of their training set.⁵⁷ We obtained the intensity map using VSCyto2D,⁵⁷ a convolutional neural network for virtual staining (VS) that predicts membrane and nuclear markers from label-free data.

To create the multilevel contour map, we combined both the nuclear and membrane channels by subtracting the normalized and filtered nuclear channel from the filtered membrane channel (Fig. 4a). The foreground was computed by manually thresholding the membrane channel and applying morphological filtering to close holes and

remove small objects. We then applied Ultrack to the foreground and the intensity map encoding cell contours (see Supp. Video 4).

For benchmarking, we tracked the virtual stained nuclear channel and manually curated a set of 247 tracklets of lineages starting in the first frame to the last frame. The cell centroids of these tracks were used to create reference gold-standard datasets following the Cell Tracking Challenge (CTC) format.

We compared against Cellpose with "livecell_cp3" weights⁵⁸ applied to the QPI channel, followed by tracking. With Ultrack applied to the combined membrane and nuclear VS channels, we detected most of the cells, Fig. 4b-c, while the off-the-shelf model failed to recognize them. We missed 2802 fewer associations between cells (*i.e.* FN edges) and 3014 cell detections (*i.e.* FN detections) than VS with Ultrack, at the cost of detection of 1000 false-positive cells.

This shows that tracking from image intensity directly, in this case from virtual staining, offers a competitive alternative to supervised segmentation and tracking methods, which often fail when presented with unseen imaging modalities, such as label-free quantitative polarization, Zernike phase contrast, DIC, or for different cell types. Thus eliminating the need for extensive human annota-

tion to train segmentation models.

Improving tracking by temporal registration.

Imaging systems must balance multiple factors, including the field of view size, spatial resolution, imaging speed, and depth.⁶⁰ A critical limitation often encountered is acquisition speed (*i.e.*, temporal resolution), which hampers cell tracking performance. This is especially problematic when cells move rapidly and have similar appearances, complicating the assignment of cells between consecutive frames.

Registration of adjacent frames often compensates for limited time resolution. However, collective cell motion within growing or deforming tissues frequently demonstrates local coherence that does not conform to simple linear or affine transformations. This is due to the complex dynamics of some biological processes. In such cases, non-linear registration, such as movement vector fields,^{61,62} is more appropriate. This approach effectively minimizes apparent motion when cells present distinct migration patterns.

Computing non-linear registration can be extremely time-consuming or require complex, specialized algorithms. To address this, Ultrack provides its own graphics-hardware-accelerated (GPU) routines to compute vector fields for 2D and 3D time-lapses. These routines are based on automatic differentiation from deep-learning models,⁶³ offering efficient computation without sacrificing accuracy (see Methods for details). The movement vector fields can be integrated into the tracking step, applying local movement to each candidate segmentation. This approach avoids altering the original image data. Moreover, Ultrack accepts any vector fields represented as a grid, allowing users to utilize their preferred registration methods.

We evaluated the effectiveness of using vector fields with Ultrack on the *Tribolium Castaneum* cartography dataset⁶⁴ from the Cell Tracking Challenge (CTC). This dataset features nuclear-labeled embryo cells imaged using selective plane illumination microscopy (SPIM). The epithelial cells' ellipsoidal surface was transformed into planar cartography using the ImSAnE toolbox⁶⁵ to facilitate downstream analysis. However, this cartographic transformation introduced distortion and excessive motion on the polar boundaries, as shown in Fig. 5a and Supp. Video 5.

We compared the tracking accuracy with and without flow registration (Fig. 5b) at various temporal resolutions, starting by skipping ten frames ($\Delta t = 10$) up to the original framerate ($\Delta t = 1$), as illustrated in Fig. 5c-d. The results show: (i) At $\Delta t = 10$ tracking accuracy improved from 0.443 to 0.623 with flow registration; (ii) At $\Delta t = 5$ accuracy increased from 0.571 to 0.756; (iii) At original

time resolution ($\Delta t = 1$) similar results were obtained (0.927 vs. 0.929). Fig. 5c provides a specific example where the flow field achieves the correct result, while a cell identity switch occurs in the unregistered case.

These results demonstrate that Ultrack's temporal registration capability enhances tracking accuracy, particularly when dealing with complex cell movements or lower temporal resolution datasets.

All these features — including its ability to handle multiple segmentation hypotheses, integrate multi-channel data, leverage intensity-based tracking, and apply temporal registration—collectively enable robust cell tracking in complex biological contexts. These capabilities are particularly valuable when confronting the challenges posed by dense, three-dimensional tissues with dynamic cellular behaviors. To demonstrate the power and versatility of this approach, we next evaluated Ultrack's performance on one of the most demanding applications in bioimage analysis: tracking cells in developing embryos.

Ultrack achieves state-of-the-art cell tracking in developing embryos.

The Cell Tracking Challenge (CTC) is a long-standing benchmark for cellular segmentation and tracking methods.¹¹ It provides participants with training datasets to tune and train their algorithm parameters; the test sets are concealed and only accessible to the organizers. This structure prevents information leakage between the training and testing phases and offers an unbiased evaluation of each algorithm's performance. We assessed our method using their three 3D whole-embryo datasets: the *C. elegans* (worm),⁶⁶ *D. melanogaster* (fly),³¹ and *T. castaneum* (beetle) embryos.⁶⁴

The worm dataset includes manually curated lineage data and cell coordinates, as well as silver-standard segmentation annotations derived from a consensus among prior challenge submissions.⁶⁷ Using these annotations, we trained a U-Net model^{45,68} that predicts two outputs per voxel: (i) whether the voxel is on a cell contour and (ii) whether that voxel is in the foreground. We optimized these predictions using three dice losses:⁶⁹ (i) binary foreground prediction, (ii) binary cells' contour prediction, and (iii) U-Net's intermediate cell contour predictions.⁷⁰ The cells' contours and foreground were then used as inputs for Ultrack. As shown in Table 1, Ultrack achieves a top combined segmentation and tracking score of 0.844.

In general, training data for segmentation is not always available nor easily obtained, which is a typical situation users face. This is the case for the fly and beetle datasets, which lack ground truth to train segmentation models. Instead, we asked whether we could forgo machine learning and use classical image-processing techniques for segmentation. We opted for a straightforward

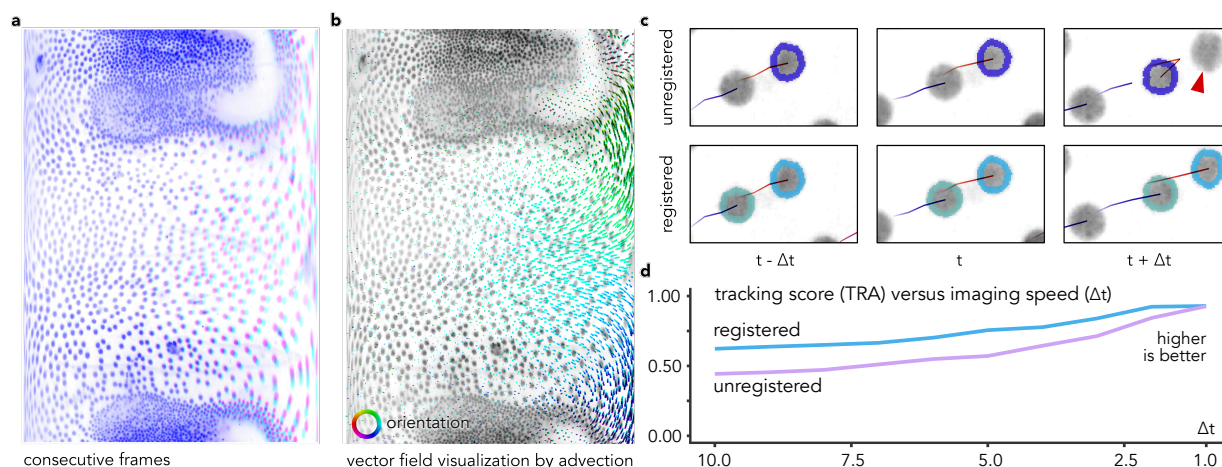


Fig. 5 | Enhancing tracking accuracy through temporal registration. **a**, Consecutive frames, t (magenta) and $t + 1$ (cyan), highlighting divergent motion along the right boundary. **b**, Vector field generated by non-linear registration, colored by orientation. **c**, Qualitative comparison of cell identity preservation: top row shows a cell identity switch in unregistered frames, identity switch indicated by red arrow, bottom row demonstrates improved consistency with flow registration. **d**, Comparative analysis of tracking accuracy for registered and unregistered data across different cell velocities (Δt), illustrating enhanced performance with registration.

method. First, we detected the foreground using a Difference of Gaussians filter and applied Otsu's threshold⁵⁴ to select the foreground voxels. Next, we used image intensity as a proxy for cell boundaries by inverting and normalizing these intensities between 0 and 1, so dimmer voxels, which are more likely to be the cells' contours, represent the highest values.

Combining this simple image-processing approach with Ultrack's handling of multiple hypotheses implicit in the contour representation yields noteworthy results compared to competing approaches that require training data. As shown in Table 1, for the fly dataset, Ultrack achieved a combined segmentation and tracking score of 0.708, compared to 0.617 for the next best method. Similarly, for the beetle dataset, Ultrack scored 0.841, while the next best result was 0.804 (see Methods for additional details). These outcomes demonstrate Ultrack's capability to perform well even when deep-learning models are unavailable. See Supp. Video 6 for recordings of the results.

High-quality cell tracking ground-truth via sparse fluorescence labeling.

While the Cell Tracking Challenge (CTC) provides a valuable unbiased benchmark for evaluating segmentation and tracking, it relies primarily on human-generated annotations. These annotations, while crucial, are inherently limited in scale and potentially in quality due to the challenge of manually annotating the latest-generation 3D live microscopy datasets.^{1, 2, 6, 7, 9, 15, 16} Such datasets often span hundreds of gigabytes to terabytes, recording thousands of cells over hundreds or more time points, making

manual annotation a task that is not only time-consuming but often beyond human capability. Moreover, in densely labeled or multi-dimensional datasets, distinguishing individual cells can be highly uncertain and sometimes impossible.^{15, 16, 24, 31, 64}

Encouraged by Ultrack's performance on the CTC, we developed a higher-fidelity ground truth dataset that could overcome these limitations and potentially provide a more challenging benchmark of a crowded and dynamic cellular environment from a developing zebrafish embryo. We applied a dual labeling protocol that combines ubiquitous fluorescence nuclear labeling (H2B-mNeonGreen) with sparse random nuclear labeling at a distinct wavelength (pMTB-ef1-H2B-mCherry). Sparse labeling was accomplished by early-stage microinjection of a marker (Tol2 transposase-mediated integration, see Methods for details) that propagates to daughter cells upon division, effectively recapitulating their lineage (Fig. 6a).

The sparsity of this second channel significantly simplifies tracking and allows for rapid curation into a set of high-quality, or "platinum-level," annotations (Fig. 6b). Cells in the sparse channel were automatically tracked using Ultrack and then manually curated to ensure the highest quality standards (Fig. 6b-top). This curation process involved selecting long, green-overlapping, high-quality lineages, conducting visual inspections, and making necessary corrections, resulting in 152 annotated tracklets spanning from 85 to 521 frames. This methodology not only simplifies the manual curation process but also reveals cell trajectories that are challenging to annotate manually using conventional methods on dense datasets,

Dataset	Rank	combined score (CTB)	tracking score (TRA)	segmentation score (SEG)
Worm <i>C. elegans</i>	1 st	0.844 ^{ultrack}	0.987 ^{MPI}	0.759 ^{MU}
	2 nd	0.829 ^{KTH}	0.979 ^{JAN}	0.729 ^{KIT}
	3 rd	0.808 ^{KIT}	0.975 ^{IGFL}	0.722 ^{KTH}
	4 th	-	0.967 ^{ultrack}	0.722 ^{ultrack}
Fly <i>D. melanogaster</i>	1 st	0.708 ^{ultrack}	0.802 ^{ultrack}	0.613 ^{ultrack}
	2 nd	0.617 ^{KTH}	0.785 ^{JAN}	0.613 ^{KTH}
	3 rd	0.591 ^{JAN}	0.785 ^{MPI}	0.397 ^{JAN}
Beetle <i>T. castaneum</i>	1 st	0.841 ^{ultrack}	0.955 ^{MPI}	0.746 ^{ultrack}
	2 nd	0.804 ^{MPI}	0.936 ^{ultrack}	0.684 ^{RWTH}
	3 rd	0.785 ^{RWTH}	0.886 ^{RWTH}	0.654 ^{MPI}

Table 1 | Segmentation and tracking scores for 3D embryo datasets of the Cell Tracking Challenge. Ultrack results, as of August 2024, are highlighted in bold. Ultrack obtains the highest combined segmentation and tracking score (CTB) for all three datasets and is the method with the best balance between segmentation (SEG) and tracking (TRA) metrics.

see Supp. Video 7.

As evident in Fig. 6c, tracking is easier at earlier stages of development when cells are less numerous, less densely packed, and when the optical properties of the tissue are more favorable for imaging. Despite these challenges, cell nuclei in the sparse channel are consistently easier to track compared to the dense channel (Fig. 6c, first row).

We used these high-quality sparse-channel-derived tracking labels to evaluate Ultrack’s accuracy on the ubiquitously labeled channel. We used two segmentation approaches as in:⁴⁵ (i) image processing to produce foreground masks and nuclei boundaries directly and (ii) a convolutional neural network to generate foreground and contour images. Hence, the Ultrack setup was kept the same, and only the input foreground and contour maps were modified. Figure 6c compares the tracking results using the sparse channel (red) and the ubiquitous channel (green) using the deep-learning-based contour map, approach (ii), for three frames (150, 400, and 500), showcasing the disproportionate difficulty of tracking cells in the dense versus sparse channel.

To quantitatively assess the performance, we evaluated our automatic dense tracking results against the manually curated sparse tracking over two time periods, Fig 6d: the first 150 frames as the previous work²⁴ and an extended period of 500 frames, nearly covering the entire time-lapse. Assessing tracking performance with five metrics,²⁴ all expressed as proportions relative to the total number of annotated edges: (i) FN: false-negative edges (*i.e.* missing connections between cells), (ii) IS: identity switches (*i.e.* incorrect association between cells), (iii) FP-D: false-positive divisions, (iv) FN-D: false-negative divisions, and (v) the sum error rate that combines the first four metrics.

As expected, tracking accuracy drops for longer imaging periods (500 versus 150 frames, see Fig. 6d), reflect-

ing the growing complexity of the developing organism already apparent in Fig. 6c. Notably, Ultrack with a deep-learning-based input, which benefits from available training data, consistently outperforms plain image processing contour and foreground maps, maintaining comparable accuracy over 500 frames to what the image intensity-based approach achieves in just 150 frames. This performance demonstrates the robustness of the deep-learning method in maintaining high tracking accuracy over extended periods of embryonic development, even as the tracking task becomes increasingly challenging.

To contextualize our results within recent advances in cell tracking for embryo-scale 3D datasets, we compared our findings to those reported by Malin *et al.*²⁴ Their study, which used a similar multi-view light-sheet microscope setup,⁷¹ established a baseline sum error rate of 0.078 for tracking neuronal cells in zebrafish embryos during 150 frames. As shown in Fig. 6d, Ultrack achieves a lower proportion of sum errors for both deep learning and intensity-image-based contour maps on 150 frames (0.049 and 0.070, respectively). Moreover, when using deep learning, we maintain accuracy (0.073) comparable to our baseline (0.078) even over a much more challenging 500-frame period. This extended performance is particularly noteworthy, as it underscores the robustness of our method in maintaining high tracking accuracy over prolonged periods of embryonic development despite increasing tissue complexity, cell density, and imaging artifacts. These results not only validate the effectiveness of our dual-channel approach but also highlight Ultrack’s potential to push the boundaries of long-term cell tracking in complex biological systems.

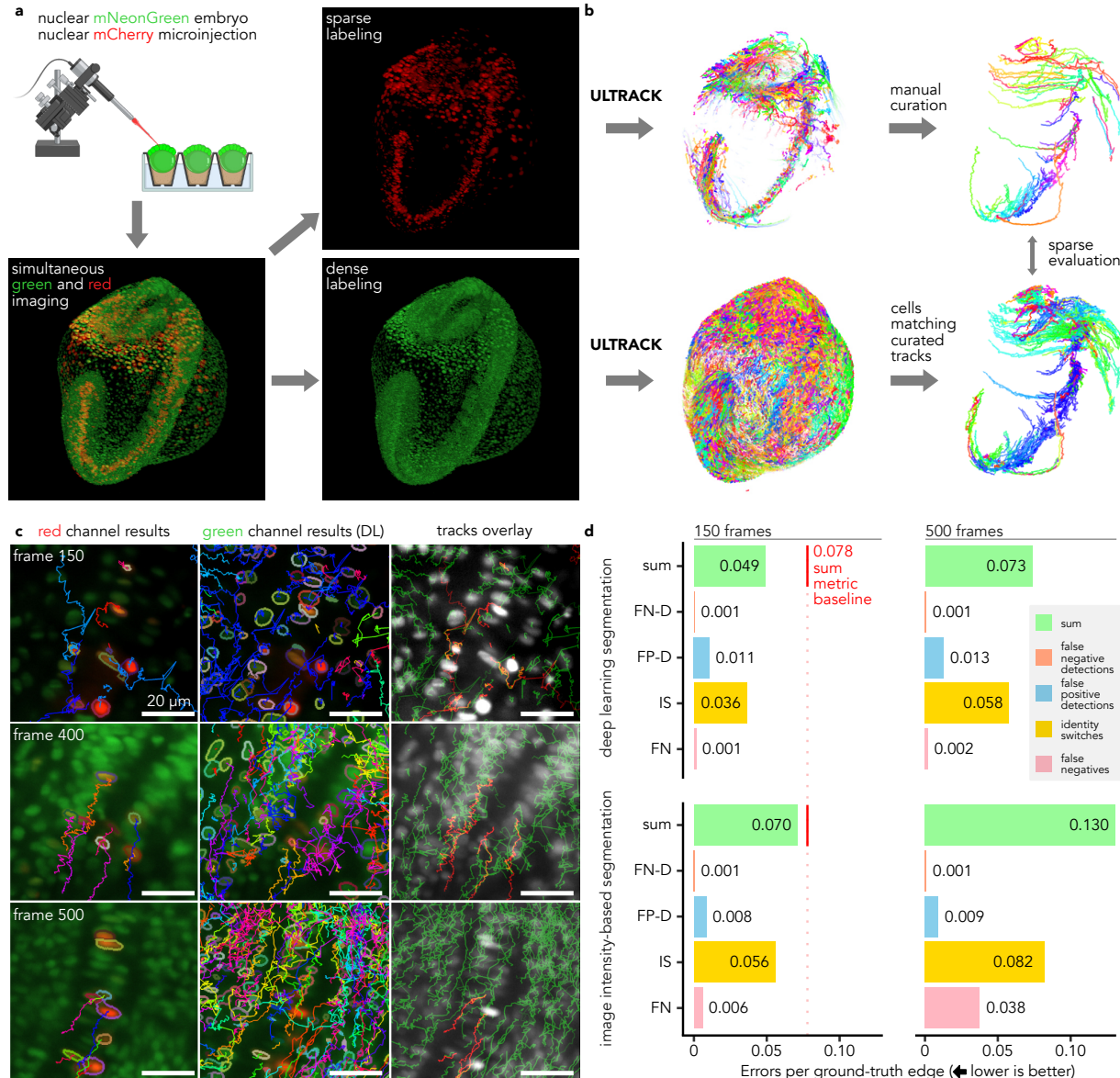


Fig. 6 | Sparse fluorescence labeling enables high-fidelity tracking validation over extended time-lapses. **a**, Validation pipeline using sparse fluorescence labeling: (i) Embryos with ubiquitous green nuclear labeling (H2B-mNeonGreen) are injected with DNA plasmids expressing red fluorescent protein pMTB-ef1-H2B-mCherry; a single embryo is then selected for imaging. (ii) Simultaneous imaging of red and green channels. **b** (iii) Independent tracking of cell nuclei in both channels. (iv) Selection and manual curation of a subset of cell nuclei tracks from the red channel. (v) Comparison of curated tracks to those obtained from ubiquitous labeling. **c**, Comparative visualization of red (first column) and green (second column) channel segmentation and tracking results at different time points, and combined results overlaid grayscale intensity colored in red and green (last column); cell density and tracking difficulty increase over time (top to bottom). **d**, Quantitative evaluation of tracking accuracy under different conditions and durations: using deep learning for cell contour estimation (first row) vs. direct image-intensity-based segmentation (second row); measurements taken up to the first 150 frames (first column) and 500 frames (second column). Metrics shown are proportions of false-negative edges (FN), identity switches (IS), false-positive divisions (FP-D), false-negative divisions (FN-D), and their sum. The red vertical line indicates the baseline sum error rate for 150 frames of 0.078 for comparison with Malin *et al.*²⁴

Ultrack scales to terabyte-scale high-resolution large-field-of-view light-sheet datasets.

To demonstrate Ultrack's scalability and performance on massive datasets, we employed it to recover cell lineages

of nuclear-labeled zebrafish embryos⁹ acquired using the fast (1 stack per minute), high-resolution (20x magnifica-

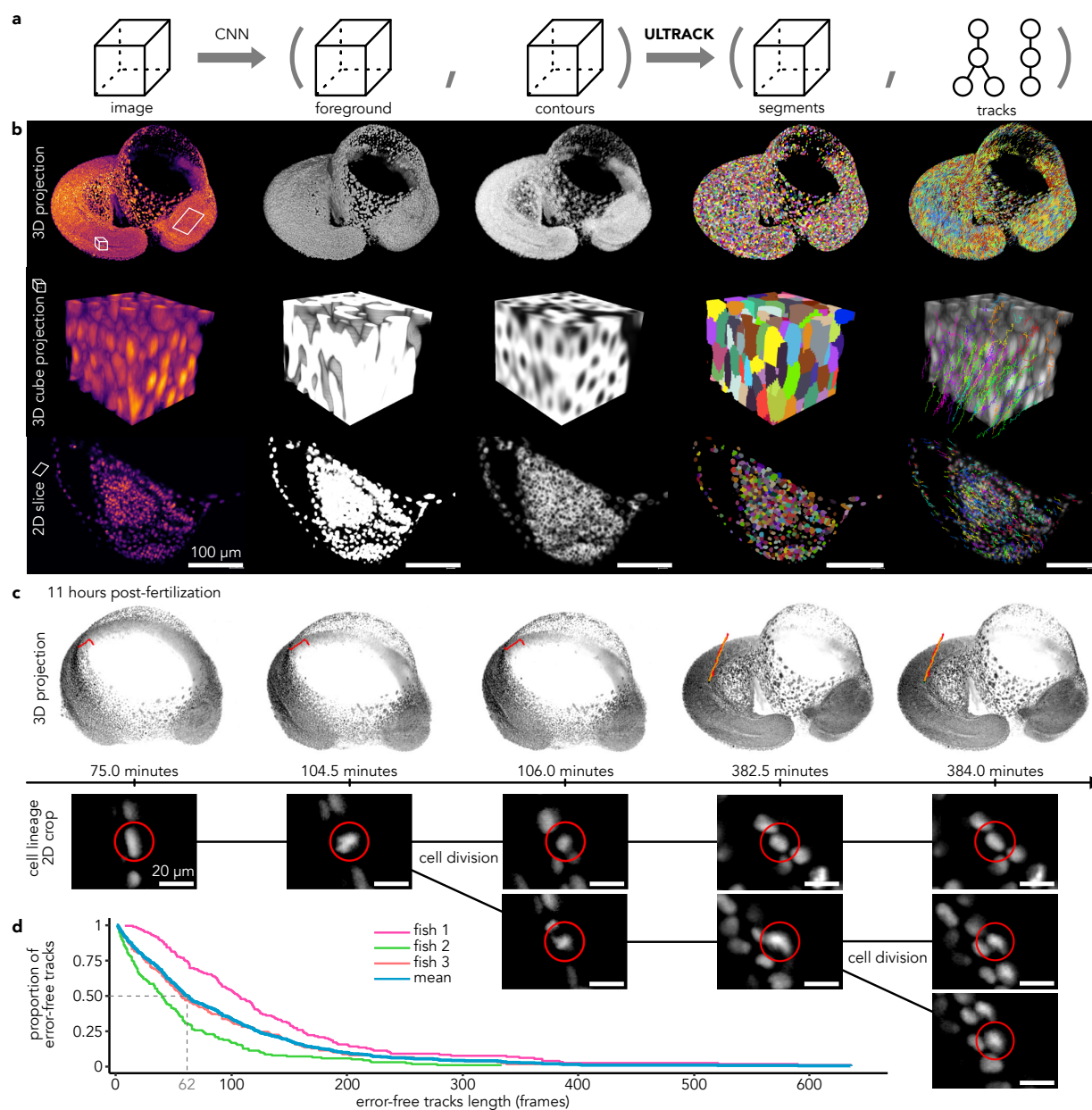


Fig. 7 | Multi-terabyte cell tracking of zebrafish embryo. a, Pipeline for joint segmentation and tracking using a convolutional neural network and Ultrack. **b**, Intermediate results at each stage: original image (first column); predicted foreground (second column) and contour probabilities (third column); Ultrack segmentation (fourth column) and tracking results (fifth column). **c**, Detailed views of fully automated lineage reconstructions, showcasing two rounds of cell divisions and their corresponding 2D slices. **d**, Distribution of the proportion of error-free tracks for varying lengths across three embryos. Indicating that, on average, 50% of the lineages are error-free for at least 62 frames.

tion and 1.0 NA Olympus objective), and large-imaging-volume ($\geq 1 \text{ mm}^3$) DaXi light-sheet microscope,⁶ producing uncompressed time-lapse data ranging from 1.7 to 3.7 terabytes, capturing 8.6 to 13.2 hours of zebrafish development (Fig. 7a,b).

Our workflow began with training a U-Net to predict nuclei pixels (foreground) and their contours (Fig. 7a,b).

We then applied Ultrack to these network predictions. Leveraging Ultrack's command-line interface (CLI) and SLURM,⁷² we processed these massive datasets within hours using distributed computing in a high-performance computing (HPC) environment. For example, a 3.7 TB dataset ($791 \times 448 \times 2174 \times 2423$ voxels, $T \times Z \times Y \times X$) was processed in approximately 8.2 hours (Fig. 1e).

Ultrack's computational scaling extends across computing environments, from high-performance clusters to off-the-shelf laptops. Ultrack supports out-of-core processing: it is able to process arbitrarily large datasets, even those significantly larger than available memory (RAM) — as long as a single image (2D or 3D) can fit into memory (see Methods for details). We could process the same 3.7 TB dataset on a 64 GB RAM laptop in 5 days, accessing data remotely via a 1 Gbps connection (Fig. 1e).

The tracking resulted in millions of segments across hundreds of time frames. To assess performance, we manually evaluated time-lapses until the first tracking mistake, recording the duration of error-free tracks. We employed a stratified sampling strategy, manually annotating distinct regions (head, body, tail bud, and presomitic mesoderm) at various time points and image qualities. From these strata, we randomly sampled approximately 140 lineages per embryo ($n = 3$). Figure 7d shows each fish's cumulative proportion of error-free tracks and their mean. On average, it took 62 frames for a tracking error to appear in 50% of the lineages, meaning that for a period of approximately 1 hour of development (60 frames), half of the tracks are completely error-free, demonstrating Ultrack's robust performance over extended periods. Figure 7c and Supp. Video 8 provides a detailed view of an automatically reconstructed lineage, capturing two rounds of cell divisions and illustrating Ultrack's ability to maintain accurate tracking through multiple cell generations.

These results underscore Ultrack's capacity to efficiently process and accurately track cells in multi-terabyte, high-resolution datasets utilizing HPC resources or standard hardware. This scalability and versatility make Ultrack a powerful tool for analyzing large-scale developmental processes at unprecedented resolution and scale.

Ultrack attains near-perfect cell tracking of organ-scale 3D datasets.

Cell tracking is often crucial in scenarios with fewer cells, where perfect lineage reconstruction is necessary to uncover precise details of cell behavior, such as division, migration, and death.^{73,74} To evaluate Ultrack's performance in such a context, we chose the zebrafish neuromast as our model system, given its well-defined structure and importance in studying sensory organ development. Previous efforts in segmenting and tracking cells in this organ included manual tracking of cells⁷⁵ or semi-automated segmentation of single frames.⁷⁶ To our knowledge, no other study has reported the fully automatic segmentation and tracking of cells in this complex 3D organ for extended time-periods.

We imaged a zebrafish neuromast for 42 hours using membrane and nuclei markers. Imaging was per-

formed on an Olympus IX83 microscope equipped with a microlens-based, super-resolution spinning disk confocal system (VT iSIM, VisiTech International), using a 60x 1.3 NA silicone-oil objective. This setup yielded a dataset of 500 frames in two colors, with volumes of $73 \times 1024 \times 1024$ voxels, at a resolution of $250 \times 76.6 \times 76.6 \mu\text{m}$ per voxel. Frames were captured every 5 minutes.

To ensure optimal segmentation accuracy, we fine-tuned a 3D Cellpose model³² to segment cells using both color channels. We then converted the Cellpose predictions into foreground and contour maps for input into the Ultrack algorithm (Fig. 8a). This approach allowed us to reconstruct lineages spanning the entire time-lapse, capturing diverse cellular events such as migration within the organ, cell division, and cell death (Fig. 8b-c and Supp. Video 9).

To assess Ultrack's accuracy, we visually inspected the generated lineages and manually corrected any errors, creating ground-truth reference lineages. Using the Cell Tracking Challenge (CTC) evaluation routines, we quantified Ultrack's performance (Fig. 8d). Ultrack achieved a TRA (tracking accuracy) score of 0.9989, demonstrating its high precision in this challenging scenario.

We compared Ultrack's performance against TrackMate,³⁷ using identical Cellpose segmentations as input. To ensure a fair comparison, we first optimized both Ultrack's and TrackMate's parameters using a different 50-frame dataset. Key parameters tuned included minimum and maximum cell size and the maximum movement. We then applied these optimized settings to the full 500-frame dataset reported here.

In this case, Ultrack's tracking results exhibit a more than 2-fold reduction in errors (279 vs. 721), with 442 fewer mistakes needing manual correction when compared to TrackMate. The primary source of errors in TrackMate's results stemmed from imperfect Cellpose segmentations (Fig. 8d). Despite TrackMate's segmentation filtering options, it struggled to correct issues such as over-segmentation and false merges. In contrast, Ultrack's approach naturally incorporates these imperfect segments into its hypotheses, allowing for more accurate merging and tracking (Fig. 8e).

These results demonstrate Ultrack's ability to maintain high tracking accuracy in this complex, long-term imaging scenario and, thus, its utility for detailed studies of cellular behavior and tissue dynamics.

Democratizing high-performance cell tracking at all skill levels.

Cell tracking software often represents one of the most complex components in image analysis pipelines, primarily due to the necessity of handling the time dimension — a factor frequently overlooked in segmentation and

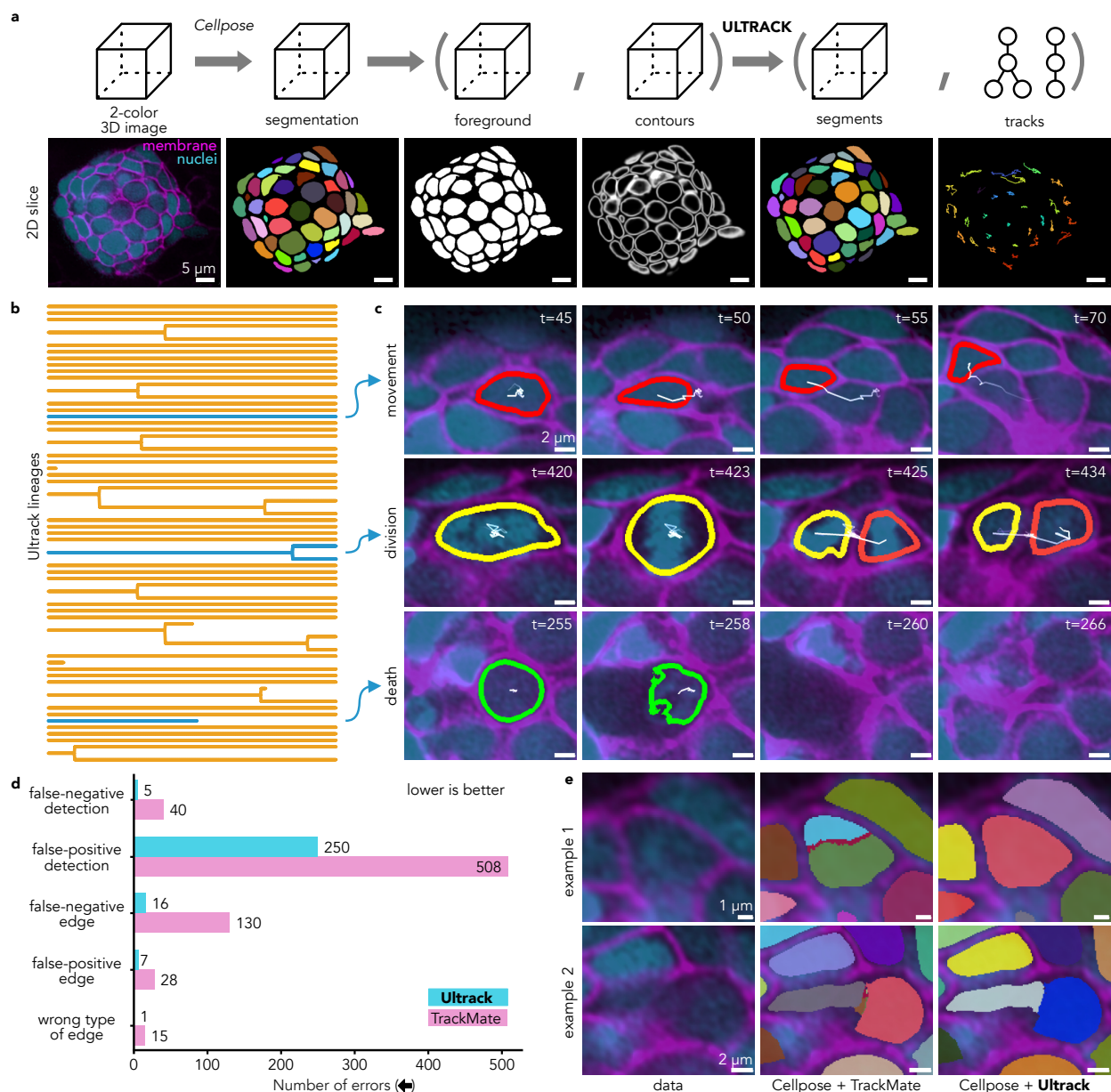


Fig. 8 | Near-perfect nuclear- and membrane-based 3D tracking of zebrafish neuromast cells. **a**, Pipeline for joint segmentation and tracking using Cellpose and Ultrack, showing 3D schematics (top) and 2D slice examples (bottom) at each stage: original 2-color image, Cellpose segmentation, foreground and contour probabilities, Ultrack segmentation, and tracking results. **b**, Dendrogram recovered by Ultrack for 71 cells over 500 frames (41.7 hours), with orange lines representing cell lineages and blue highlights indicating examples detailed next: **c**, Time-lapse images illustrating cell movement (red outlines), division (yellow/orange outlines), and death (green outlines). **d**, Quantitative comparison between Ultrack (blue) and TrackMate (pink). The bar chart shows error counts for various metrics, with Ultrack obtaining 442 fewer mistakes than TrackMate. **e**, Two examples demonstrating Ultrack's robustness to over-segmentation, showing the original image (left), fragmented TrackMate results (middle), and accurate Ultrack segmentation and tracking (right) despite imperfect Cellpose input.

other image processing operations. Moreover, it operates on non-matrix data (*i.e.* track lineages), where efficient data processing routines are less prevalent. This complexity has historically restricted accessibility to a broad user base, with only a few longstanding solutions like Track-

Mate³⁷ and others,^{21,22,31} alongside specialized solutions for handling larger datasets.²⁴

Ultrack addresses these challenges by offering a versatile, user-friendly solution that caters to a wide range of users, from biologists with minimal programming ex-

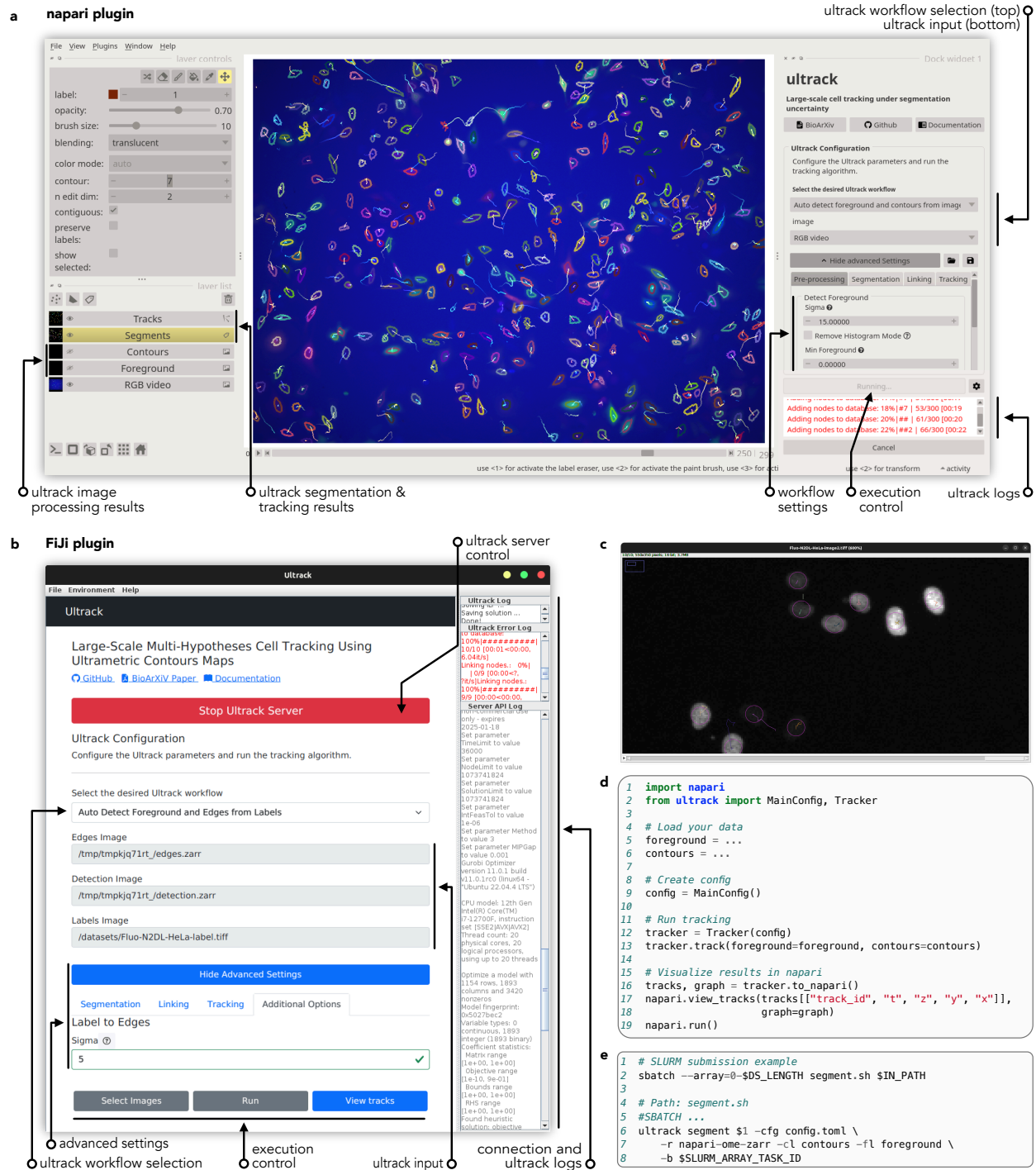


Fig. 9 | Ultrack's multi-interface approach enables cell tracking for diverse user needs and computational environments. a, Napari plugin interface showcasing multi-color automatic cell tracking results. The interface includes workflow selection, input configuration, image processing results, segmentation and tracking visualization, workflow settings, and execution control. **b,** Fiji plugin interface guiding users through workflow selection, input specification, and advanced settings. It also provides execution control and displays Ultrack server logs. **c,** Example of image data overlaid with cell detections and tracks as viewed in the Fiji interface. **d,** A concise Python API example (18 lines) demonstrating data loading, configuration, tracking execution, and result visualization in *napari*. **e,** Sample SLURM submission script for Ultrack's CLI, illustrating its integration with high-performance computing environments for distributed processing.

perience to machine learning experts requiring customizable inputs. By providing multiple interfaces and scalable processing capabilities, Ultrack democratizes access to advanced cell tracking technology, enabling efficient analysis of datasets of various sizes across different computational resources (see Fig. 1e).

At the heart of Ultrack's accessibility is its multi-interface approach. Ultrack offers two options for users who prefer a graphical interface: *napari* and Fiji. The *napari* plugin (Fig. 9a) integrates transparently with the Python ecosystem, providing a comprehensive set of features for non-coding users. This interface offers graphics-hardware-accelerated pre-processing, vector-field flow computation, and full configuration of Ultrack parameters. Complementing this, the Fiji plugin³⁵ (Fig. 9b) exposes Ultrack's algorithm to Fiji's mature ecosystem and broad user base, allowing seamless integration with existing Fiji workflows and ensuring compatibility with popular tools like TrackMate³⁷ and Mastodon.³⁸ Supp. Video 10 presents them in action.

For users comfortable with programming, Ultrack provides a user-friendly Python API (Fig. 9c,d) that exposes all of its functionalities, allowing for easy customization and reproducible workflows. This API accepts various array formats (numpy, zarr, dask, etc.), and can export data into multiple file formats, including networkx,⁷⁷ tables (i.e., data frames),⁷⁸ Cell Tracking Challenge,¹¹ and TrackMate³⁷ formats, enhancing its interoperability with other tools and workflows. Additionally, Ultrack offers a command line interface (CLI, see Fig. 9e) designed for batch processing and streamlined workflows, which supports distributed processing in high-performance computing environments. This CLI accommodates any file format supported by existing *napari* reader plugins, eliminating the need for data conversion — a crucial feature for handling large datasets.

DISCUSSION

Ultrack advances cell tracking by offering a robust, scalable solution for analyzing complex cellular dynamics across various scales and dimensions. Its integration of segmentation and tracking, leveraging multilevel contour maps (UCMs), enables effective analysis in challenging scenarios where previous methods often struggled.

The method's versatility is demonstrated through its top-scoring performance in 3D embryonic cell tracking challenges, even when using non-learned classical image processing algorithms. This ability to perform well without training data or fine-tuning addresses a critical challenge in the microscopy field, where 3D annotations are often scarce.¹¹ Furthermore, Ultrack's capacity to combine results from parameter sweeps eliminates the common problem of identifying optimal segmentation parameters. This flexibility extends to working directly with im-

age intensities thus circumventing the need for explicit segmentation in certain scenarios, for example, virtual staining of label-free images. This approach is not limited to label-free imaging such as quantitative phase imaging but extends to virtually any imaging method — whether conventional fluorescence microscopy, label-free techniques, or virtual staining approaches — that can produce intensity maps highlighting cellular structures such as nuclei or membranes. By leveraging these intensity-based representations and employing direct tracking, Ultrack offers an adaptable and efficient solution that bypasses the need for extensive annotation or fine-tuning of segmentation models. Ultrack also excels in processing multi-channel data by combining multiple single-channel labels, a feature that, when paired with technologies for engineering multi-colored cells,^{50,51} promises to enable a new generation of high-accuracy lineage reconstruction. Finally, Ultrack addresses challenges associated with low temporal resolution through GPU-accelerated flow-field registration between frames, broadening its applicability to more experiment designs.

The introduction of a dual-channel sparse labeling approach for validation offers a more challenging and heterogeneous benchmark compared to previous studies focused on specific cell subsets or biological contexts.^{15,16,31,64} This approach paves the way for more robust validation of tracking algorithms over extended periods in large datasets.

Ultrack's scalability, tested on both high-performance computing clusters and laptops, addresses the growing demand for tools capable of handling increasingly large and complex datasets produced by advanced imaging technologies.^{1,6,7} This flexibility in computational requirements democratizes access to high-quality cell tracking across diverse research settings.

Ultrack's integration with popular platforms like Fiji³⁵ and compatibility with established tracking frameworks such as TrackMate³⁷ and Mastodon³⁸ positions it for widespread adoption. Its multiple interfaces, ranging from graphical user interfaces to command-line tools for high-performance computing environments, align with community efforts to make complex computational tools more accessible to biologists.^{19,20,33,34,36,79}

Despite all of this, Ultrack has its limitations, as the segmentation hypotheses are only as good as the provided foreground and contour maps. Therefore, despite its robustness to spurious candidate segmentations, when provided with an overwhelming number of incorrect segmentations or systematic errors that are consistent over time, the ILP optimization might favor erroneous segments.

Looking ahead, several promising avenues for future development emerge: (i) Integration of advanced deep-learning techniques, particularly transformer-based models for sequence prediction,⁸⁰ could further improve long-

term tracking accuracy. (ii) Expansion of the sparse labeling approach to generate large-scale, high-quality datasets for training and benchmarking future tracking algorithms. (iii) Development of interactive correction tools leveraging Ultrack's multiple segmentation hypotheses to facilitate efficient manual curation of tracking results. (iv) Exploration of Ultrack's potential applications beyond developmental biology, including cancer research, immunology, and regenerative medicine.

In conclusion, Ultrack's innovative approach for joint segmentation and tracking, combined with its scalability and accessibility, has the potential to accelerate research across a wide range of biological disciplines by enabling more accurate, efficient, and accessible cell tracking. Ultrack opens new avenues for investigating complex cellular behaviors, tissue dynamics, and developmental processes at unprecedented scales and resolutions.

Supplementary Videos

1. [Ultrack intuition and overview](#);
2. [Using multiple segmentation hypotheses with temporal consistency helps reduce the challenges of parameter tuning](#);
3. [Enhanced multi-color cell tracking by integrating segmentation results from multiple algorithms](#);
4. [Improved cell tracking in label-free imaging by leveraging virtual staining](#);
5. [Enhancing tracking accuracy through temporal registration](#);
6. [Visualizing Ultrack's Cell Tracking Challenge benchmark results](#);
7. [Sparse fluorescence labeling enables high-fidelity tracking validation over extended time-lapses](#);
8. [Multi-terabyte cell tracking of zebrafish embryo](#);
9. [Near-perfect nuclear- and membrane-based 3D tracking of zebrafish neuromast cells](#);
10. [Ultrack's guided user interfaces across multiple platforms facilitate cell tracking, catering to diverse user needs and computational environments](#).

Code and data availability

The repository for the Ultrack python package and its updated versions are available at <https://github.com/royerlab/ultrack>. The Fiji plugin can be found at <https://github.com/royerlab/ultrack-imagej>. The imaging data is accessible at <https://public.czbiohub.org/royerlab/ultrack> and <https://public.czbiohub.org/royerlab/zebrahub/imaging/single-objective>

METHODS

Animals All zebrafish care and experimental procedures complied with protocols approved by the Institutional Animal Care and Use Committee at the University of California San Francisco (UCSF). Adult fish were maintained at

28.5 °C and fed twice daily using an automatic feeder.⁸¹ Embryos were raised at 28.5°C, and staged based on hours post fertilization (hpf). For light-sheet microscopy and injection experiments, we used two transgenic lines: tg(h2afva:h2afva-mCherry) (a gift from J. Huisken, University of Göttingen) and tg(eef1a1/1:H2b-mNeonGreen) (a gift from D. Wagner, UCSF). Zebrafish neuromasts were imaged using the transgenic line tg(cldnb:lyn-mscarlet) x tg(she:h2b-EGFP).^{82,83}

Multi-colored cell line generation and imaging. MDA-MB-231 metastatic breast adenocarcinoma cells (ATCC) were maintained in Dulbecco's Modified Eagle Medium (DMEM, VWR) with 10% (v/v) fetal bovine serum (FBS, Seradigm VWR), and penicillin and streptomycin (50 U/mL, ThermoFisher Scientific). Cells were stably modified with a variety of RGB-markings with lentiviral gene ontology (LeGO) vector system, as described in Weber *et al.*⁵² Lentivirus was prepared as described in Shah *et al.*⁸⁴ In brief, pseudoviral particles were generated using 293TN cells (System Biosciences, SBI) co-transfected with lentiviral packaging and envelope plasmids (psPAX and pMD2.G, gifts from Didier Trono), and one plasmid of interest containing fluorophore eGFP, mCherry, or EBFP2 (Addgene, #25917, #27339, #85213). Supernatants containing lentivirus were collected, passed through a 0.45 µm filter, concentrated by ultracentrifugation at 200,000 × g for 4 hours at 4°C, resuspended in phosphate-buffered saline (PBS, VWR), and stored at -80 °C. MDA-MB-231s were transduced simultaneously with all three constructs at equal viral titer and a range of multiplicity of infections (MOI). To optimize color diversity, a fraction of cells expressing fluorophore was assessed at five days post-transduction using fluorescent microscopy, and a condition with ≈ 50% transduction rate of all fluorophores (MOI ≈ 0.7) was selected for subsequent experiments. Cells were seeded 12 h before imaging at a density of ≈ 10,000/well on a glass-bottom plate coated with 50 µg/mL rat tail type I collagen (Corning) in 0.02N glacial acetic acid (Sigma) overnight at 4°C. Imaging was performed on a Keyence BZ-X800 imaging system equipped with temperature, humidity, and CO2 control, and images were acquired at 10X magnification at ten-minute intervals for 2 days.

Microinjection for zebrafish sparse labeling. DNA plasmids (pMTB-ef1-H2B-mCherry2 gifted by D. Wagner, UCSF) were extracted and purified with a QIAprep Spin Miniprep Kit (Cat. No. 27104). Purified DNA was diluted to 100 ng/l in water (Invitrogen™ UltraPure™ DNase/RNase-Free Distilled Water, Cat. No. 10977015) and saved in -20. Zebrafish Tol2 mRNA was prepared using pT3TS-Tol2 plasmids shared by Ekker lab.⁸⁵ pT3TS-Tol2 plasmid DNA was linearized by the restriction enzyme BamHI digestion. Linearized plasmids were gel-purified

with a QIAquick Gel Extraction Kit (Cat. No. 28704). The 5'-capped mRNAs were synthesized using Invitrogen™ mMACHINE™ T3 Transcription Kit and purified by lithium included in the kit (ThermoFisher, Cat. No. AM1348). The purified RNA was dried and dissolved with 20 l DNase/RNase-Free water. Dilute it to 200 ng/l and save it as stock in -70 degrees. Before the experiment, the stocked DNA and RNA were thawed, and 1 µL DNA plus 1 µL RNA was mixed with 3 µL of DNase/RNase-Free water containing 0.05% phenyl red. For each microinjection, a 4 nL mixture was injected into a single cell of a 4-8 cell stage embryo from the Tg [ef1-H2B-mNeonGreen] strain. All injections were done with an injector (WPI PV830 Pneumatic Pico Pump), and a micromanipulator (Narishige, Tokyo, Japan). 50 to 100 embryos were injected for each experiment. The injected embryos are ready for mounting and imaging at approximately 4.5 hpf.

Sample preparation for single objective light-sheet microscopy (DaXi). Embryos are first dechorionated and then gently embedded into a 1% solution of low-gelling-temperature agarose (Sigma, A0701). The samples are then transferred in a glass-bottom cell culture dish (35mm Cell Culture Dish with Glass Bottom 20mm – Stellar Scientific) and positioned at the correct imaging angle using a custom-made capillary. When solid, the agarose surrounding the tail was cut off and removed using a dissection knife and forceps to permit full development and tail elongation. During the time lapses, the embryos are placed in an embryo-medium solution with 0.016% tricaine to prevent embryo movement.

Quantitative phase imaging and virtual staining The A549 cells were cultured at 37°C and 5% CO₂ and maintained between 20% and 90% confluency. The cells were seeded on a 12-well glass bottom plate (Cellvis, P12-1.5H-N) and imaged using a wide-field microscope (Leica Dmi8). The quantitative phase volumes are virtually stained using the VSCyto2D model to highlight landmark channel organelles such as the nuclei and cell plasma.⁵⁷

Simultaneous sparse and ubiquitous labels imaging. Four hours post-microinjection, injected embryos (4 hpf) were screened using a Thunder Imager Model Organism (Leica Microsystems). An embryo with 20%-30% H2B-mCherry2 expressing cells (compared to ubiquitous H2B-mNeonGreen cells) was selected and transferred to a glass-bottom petri dish (35mm Cell Culture Dish with Glass Bottom 20mm, Stellar Scientific) containing 0.3x Danieau's embryo medium. The embryo's chorion was removed using sharp forceps under the same microscope. The dechorionated embryo was then gently transferred to a petri dish containing 0.1% low-temperature melting agarose (Sigma, A0701) using a glass pipette. Subsequently, the embryo was transferred to FEP tubes (Valco-TFEP130, OD 1/16", ID 0.03", Vici Metronics) with

agarose via micropipette through a 200 µl tip (Eppendorf). Both ends of the FEP tube were sealed with 2% low-temperature melting agarose to secure the embryo. The tube was placed in the chamber of an adaptive multi-view multi-color light-sheet platform⁹ using a customized holder insertion tool filled with low-toxicity silicone (Kwik-Sil™, World Precision Instrument). The mounting system was similar to that previously described in ZebraHub,⁹ but with an upside-down mounting direction. The chamber was filled with embryo water, which was circulated with 0.016% tricaine during imaging. For whole-embryo imaging, the multi-color-interleaved mode was used with an exposure time of 20 ms per channel with the Open-SimView⁹ microscope. The imaging volume comprised 493 z-planes with 1.625 µm z-steps. Images were captured at 90-second intervals, starting at the shield stage (6 hpf), and ending after 24 hpf.

DaXi microscope parameters. We used our DaXi imaging platform⁶ to perform high-resolution tail development imaging (1.0 NA). We use two orthogonal, oblique light sheets for each time point to improve the image field of view coverage. During imaging, the exposure was set to 20ms, the z-step to 1.24µm, the imaging volume was adapted to each sample, and the time step was around 60 seconds. Time-lapse sessions are about 10 hours long and usually cover a developmental window from 1 or 2 somites to 27 somites.

Software. The Ultrack package, including its REST API, and additional image analysis routine and plotting were implemented using Blosc, CuPy, FastAPI, imagecodecs, iohub, httpx, ggplot2, napari, networkx, Numba, NumPy, Pandas, PyTorch, scikit-image, SciPy, SLURM, tifffile, traccuracy, uvicorn, websockets, and zarr.^{36, 63, 72, 77, 78, 86-92}

Datasets Table 2 presents the datasets evaluated in this paper. While not a requirement, some images of the Cell Tracking Challenge were upsampled for improved accuracy.

Light-sheet imaging data pre-processing. Upon acquisition, the imaging data are converted to the ome-zarr format^{93,94} using the iohub⁸⁷ Python package. Next, the raw multi-view images are fused. Phase-cross correlation techniques are applied to correct inter-frame movements to ensure stability throughout the time-lapse sequence. In cases where deep-learning methodologies are utilized, image normalization is performed, scaling pixel intensities to the range between 0 and 1 using a user-specified lower and upper quantile, and their background is subtracted using area white top-hat transform. These pre-processing routines and other image processing operations are implemented and accessible via <https://github.com/royerlab/dexpv2>.

Dataset	T, C, Z, Y, X	Size (GB)	Usage	Ref.
YFP-TIA-1 (CTC)	30, 1, 1, 1024, 1024	0.063*	Fig. 2	48
MDA-MB-231	300, 3, 1, 1440, 1920	2.5	Fig. 3	this work
Label-free	48, 1, 8, 2046, 2003	5.85 [†]	Fig. 4	57
Beetle proj. (CTC)	135, 13, 2447, 1719	14.76*	Fig. 5	64
Worm (CTC)	165, 31, 512, 712	3.73*	Tab. 1	66
Fly (CTC)	50, 125, 603, 1272	9.59*	Tab. 1	31
Beetle (CTC)	105, 983, 1846, 983	376*	Tab. 1	64
Sparse zebrafish	522, 2, 505, 2217, 2170	5072	Fig. 6	this work
Late zebrafish	791, 1, 448, 2174, 2423	3733	Fig. 7	9
Neuromast	500, 1, 73, 1024, 1024	71.3	Fig. 8	this work

Table 2 | List and references of datasets. †: size before 3D to 2D virtual staining. * indicates averaged size and shape. The size presented in Fig. 1f is regarding the input contour map, which can include projections and upsampling.

GPU-accelerated flow field estimation. Flow field estimation of large 3D volumes can be extremely challenging and computationally intensive, often following particle image velocimetry strategies that estimate local translation using phase cross-correlation for small patches of the data.⁶¹ Instead of maximizing cross-correlation, which minimizes the mean-squared error between frames given a field in the coordinate space, we employ gradient descent to optimize the coordinate space directly. This choice, justified by its enhanced computational efficiency and simplicity, allows for the integration of any differentiable function within the coordinate space. The loss function is defined as:

$$L(\theta) = \|I_{t-1} - \text{grid_sample}(I_t, C + \theta)\|_1 + TV(\theta) \quad (1)$$

where I_t denotes the frame at time t , C represents the identity coordinates (*i.e.* no movement), θ indicates the movement flow field we optimize, and TV denotes the L1 total variation loss on the flow field. The `grid_sample` function, a differentiable data sampling method from PyTorch,⁶³ enables this process. This routine is tailored for specialized GPU hardware, greatly enhancing processing speed. The θ optimization is performed for a fixed number of iterations over multiple image resolutions, starting from the lowest resolution, which is then used to initialize the next θ optimization stage.

Label conversion to contour and foreground map. The conversion of multiple segmentation labels into foreground and contour maps involves two main processes. For the foreground map generation: (i) each label is transformed into a binary map, with non-zero pixels assigned a value of one (foreground) and zero pixels remaining as background; (ii) these individual maps are merged into a single foreground map by taking the maximum value across corresponding pixels from all segmentation maps; (iii) this logical OR operation ensures that any pixel belonging to at least one segment is included in the final foreground map. For the contour map generation: (i) to

preserve each label and its dual contour, we process each label map to create a binary contour map, where pixels are set to one if their neighboring pixels have differing labels, indicating a boundary; (ii) the final contour map is then created by averaging these individual binary contour maps; (iii) this averaging approach, as opposed to summation, keeps pixel values within the 0 to 1 range.

Candidate hierarchical (nested) segmentation. The goal of hierarchical segmentation is to compute a set of nested partitions (*i.e.* segmentations) from a multilevel contour map, where each level corresponds to a different segmentation threshold. While a naive approach resorts to binarizing the contour map at each level and computing the connected components of the binary images, Ultrack employs a more sophisticated and computationally efficient method: the hierarchical watershed algorithm.^{43,95} This algorithm represents the image as a graph $G = (V, E)$, where V are the pixels or voxels, and E denotes the edges connecting adjacent voxels. It defines a non-decreasing order of edge weights:

$$w(p, q) = \frac{I(p) + I(q)}{2} \quad (2)$$

where $p, q \in E$ and $I(p)$ is the image intensity at a pixel p of a contour map. The algorithm then recovers nested segmentations using a region merging approach based on Kruskal's minimum-spanning tree⁹⁶ and Tarjan's union-find algorithms.⁹⁷ The process begins with an initial segmentation where each pixel forms its own segment. It then iteratively: (i) selects the minimum unsewn edge (p, q) ; (ii) merges the segments of p and q if they are not already in the same segment; (iii) continues until all edges have been processed. Importantly, this segmentation process operates within each connected component of the binary foreground mask, allowing the algorithm to ignore background regions and thereby greatly speeding up the process. For further details on hierarchical segmentation processing and implementation, see.⁴⁵

Candidate segmentation association. For each candidate segmentation, we compute the potential temporal associations (*i.e.*, tracks). Let \mathcal{H} represent the set of all segmentations derived from hierarchical segmentations. We define E^T as the set of pairs representing potential associations between these segmentations, with $w(\cdot, \cdot)$ denoting their respective weights. The association process involves the following steps: (i) For each segmentation $i \in \mathcal{H}$, where t denotes the time point of segment i , we compute its k -nearest neighbors within a predefined radius using their centroid coordinates at time $t - 1$; (ii) These pairs are then included in E^T ; (iii) The association score between any two segmentations i and j is calculated as $w(i, j) = \text{IoU}(i, j)^\gamma$. Where γ is consistently set to four throughout our study and IoU represents the intersection over union between the segmentation masks of i and j .

Segmentation selection and tracking ILP. Given \mathcal{H} and E^T , as defined previously, we compute the joint segmentation and tracking by solving an integer linear programming (ILP) optimization problem. This ILP framework aims to find the optimal pairing between segments that maximizes their association while adhering to biological constraints. The ILP is formulated as follows:

$$\arg \max_{\mathbf{x}} = \sum_{i,j \in E^T} w_{ij} x_{ij} + \sum_{j \in \mathcal{H}} w_{\alpha} x_{\alpha j} + w_{\beta} x_{j\beta} + w_{\delta} x_{\delta j} \quad (3)$$

$$\text{s.t. } y_j = x_{\alpha j} + \sum_{i \in H_{t-1} | i, j \in E^T} x_{ij} \quad \forall j \in \mathcal{H} \quad (4)$$

$$y_i + x_{\delta i} = x_{i\beta} + \sum_{j \in H_{t+1} | i, j \in E^T} x_{ij} \quad \forall i \in \mathcal{H} \quad (5)$$

$$y_i \geq x_{\delta i} \quad \forall i \in \mathcal{H} \quad (6)$$

$$y_i + y_j \leq 1 \quad \forall i, j \in \mathcal{H} \mid i \subset j \quad (7)$$

$$y_i, x_{\alpha i}, x_{i\beta}, x_{\delta i} \in \{0, 1\} \quad \forall i \in \mathcal{H} \quad (8)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in E^T \quad (9)$$

In this formulation, y_i indicates whether a segment is selected, and x_{ij} indicates the selected associations, $x_{\alpha i}$, $x_{i\beta}$ and $x_{\delta i}$ are slack variables indicating an appearing, disappearing and dividing cell, respectively. Eq. 3 is the objective function to maximize the association between segmentations. The first term in the objective function represents the weight of segmentation associations, whereas the subsequent terms account for appearance, disappearance, and division events, which are user-defined non-positive penalization weights. Eq. 4 and Eq. 5 ensure flow conservation, stipulating that an appearance, division, or incoming association must correspond to at least an outgoing association or a cell disappearance. Eq. 6 restricts cell division to pre-existing cells,

and Eq. 7 prevents selecting overlapping segmentations, thereby enforcing that each pixel is uniquely selected.

Tracking evaluation metrics. The evaluation scores were calculated using official binaries from the Cell Tracking Challenge (CTC) for datasets submitted to the competition. Additional calculations were performed using the `traccuracy` Python package.⁹⁸ The TRA metric quantifies the cost-effectiveness of editing a predicted lineage into the reference lineage, divided by the cost of constructing the lineage from scratch, $AOGM_0$. This is mathematically defined as:

$$AOGM = w_{NS} n_{NS} + w_{FN} n_{FN} + w_{FP} n_{FP} + w_{ED} n_{ED} + w_{EA} n_{EA} + w_{EC} n_{EC} \quad (10)$$

$$TRA = 1 - \frac{\min(AOGM, AOGM_0)}{AOGM_0} \quad (11)$$

where n_{NS} represents the number of node splits (*i.e.* undersegmented cells), n_{FN} and n_{FP} are false negative and false positive nodes respectively, n_{ED} and n_{EA} denote false positive and negative edges, and n_{EC} represents nodes with incorrect semantics. The weights are set as follows: $w_{NS} = 5$, $w_{FN} = 10$, $w_{FP} = 1$, $w_{ED} = 1$, $w_{EA} = 1.5$, and $w_{EC} = 1$, emphasizing the greater complexity in correcting missing units compared to deleting extra. For further details, refer to Matula et al.⁴⁹ The combined CTC score averages the TRA and SEG scores. The SEG score calculates the average intersection over union (Jaccard index) across all segmentation instances. A ground-truth (*gt*) segmentation instance, G_j , is paired to a predicted instance P_i if:

$$|P_i \cup G_j| > 0.5 |G_j| \quad (12)$$

This criterion ensures that the overlap between the predicted and *gt* instances is at least half the size of the *gt* instance, guaranteeing a single match per *gt* instance. The same criterion matches nodes in the $AOGM$ metric calculation.

On the multi-color and label-free cell culture experiments where not all lineages are annotated, we applied the recommended CTC protocol, where the first frame of the ground truth is provided to indicate which lineages should be kept, all the remaining lineages are removed, and the CTC metric is computed between this subset of lineages and the annotated data.

Zebrafish sparse tracking evaluation metrics. The sparse tracking data evaluation uses implementation from the Python package `linajea`.²⁴ The metrics are the proportion of different types of mistakes over the total number of edges (*i.e.* connections between cells of adjacent frames) of the ground-truth data. Because the data is sparse, only a subset of metrics are valid estimates for the remaining unlabeled data, them being:

- FN: Proportion of false negative (*i.e.* missing edges);
- IS: Proportion of identity switches (*i.e.* edges associating incorrect cells);

- FP-D: Proportion of false positive divisions (*i.e.* fake division) within a 1-frame tolerance.
- FN-D: Proportion of false negative divisions (*i.e.* missing division) within a 1-frame tolerance.
- sum: the sum of all the above.

Color features association. The image intensity information can be used to filter out segments between adjacent frames that are unlikely to represent the same cell. To do that, we compute a normalized distance between the average intensities of segments; if this distance exceeds a predefined threshold z , their association is excluded from the tracking Integer Linear Programming (ILP) formulation. Specifically, the relationship must satisfy the following conditions:

$$\max \left| \frac{\mu_i - \mu_j}{\sigma_j} \right| \leq z \quad (13)$$

where μ_j is the mean intensity vector within the mask of a segment S_j at time t and σ_j the standard deviation, the operations are done element-wise and \max results in a scalar from it. Here, compared to a neighboring segment S_i at $t - 1$. The threshold z was set to 3.0 throughout the multi-color experiments.

Fiji plugin and Python interoperability. Fiji³⁵ is a Java-based software that can be extended through plugins developed in the same language. To integrate Ultrack with Fiji, we developed a RESTful API⁹⁹ using the FastAPI framework,⁸⁸ facilitating communication between Java and Python. This API employs websockets¹⁰⁰ for persistent communication, enabling real-time event logging and output streaming from Ultrack to Fiji. We created a graphical user interface (GUI) in Java that serves as a Fiji plugin, leveraging the API provided with the Ultrack Python package. This interface grants access to features and functionalities of both systems. The GUI components and user experience are implemented using the Bootstrap 5¹⁰¹ front-end framework. To promote broader interoperability, Ultrack adheres to API specifications detailed in our documentation (https://royerlab.github.io/ultrack/rest_api.html), allowing integration with various programming languages beyond Java and Python.

Napari plugin. Napari³⁶ is a Python library for n-dimensional image visualization, annotation, and analysis. It features a GUI built with Qt,¹⁰² which is extensible with Qt primitives in Python. We developed a custom plugin for Napari integration that still scales with large datasets. In addition to the tracking standard workflow, the plugin offers basic preprocessing operations to fit most users' needs, which can benefit a significant speedup from GPUs, if any are available. Ultrack's parameters can be tuned through the GUI. The processing runs in the background in a separate thread, and for that reason, the intermediate results can be visualized using the native Napari viewer. By the tracking execution finish, the results can be exported to several formats, such as Track-

mate,³⁷ NetworkX graph,⁷⁷ and Zarr array.

Acknowledgements

We extend our gratitude to Dan Wagner and Jan Huisken for generously sharing their zebrafish lines. We thank Ahmed Abbas and Paul Swoboda for their insightful discussions on Integer Linear Programming and optimization. We are grateful to Bruno Moretti, Veronika Pfannenstill, and Jacky Ko for their valuable feedback and for being early adopters of Ultrack. We appreciate Ivo Sbalzarini and Sandy Schmid for their mentorship and constructive feedback on the manuscript. A.X.F. and I.T. were funded by CNPq 304711/2023-3 and FAPESP 2022/16491-2, 2022/07877-4, and 2023/14427-8. J.L. was funded by NIH Awards R01GM137605 and R35GM153257. This work was funded by the Chan Zuckerberg Biohub San Francisco (CZB SF). We are deeply grateful to CZB SF donors Priscilla Chan and Mark Zuckerberg for their generous support, which made this research possible.

Author Contributions

J.B., M.L., and L.A.R. conceived the research. J.B. wrote the first version of the Ultrack software, performed the computational experiments, and curated the data except for neuromast and label-free images. I.T. wrote the user interfaces, implemented the REST backend, maintained the software, and trained the model for the sparse/dense labeled embryo. X.Z. prepared and imaged the sparse/dense labeled embryo. M.L. prepared and imaged the zebrahub embryo. T.L. prepared and imaged the neuromast. T.H. and A.B. analyzed the neuromast data. A.J. and A.B. curated the neuromast data. E.H. curated and analyzed the label-free data. S.X. provided support for the microscopy. R.A. prepared and imaged the multicolor data. J.B., I.T., and L.A.R. wrote the paper. J.L., S.M., A.X.F., and A.J. provided guidance. L.A.R. supervised the research. All authors contributed to editing the manuscript.

References

- [1] P. J. Keller, A. D. Schmidt, J. Wittbrodt, and E. H. Stelzer, "Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy," *Science*, vol. 322, no. 5904, pp. 1065-1069, 2008.
- [2] G. Shah, K. Thierbach, B. Schmid, J. Waschke, A. Reade, M. Hlawitschka, I. Roeder, N. Scherf, and J. Huisken, "Multi-scale imaging and analysis identify pan-embryo cell dynamics of germlayer formation in zebrafish," *Nature Communications*, vol. 10, no. 1, p. 5753, 2019.
- [3] B.-C. Chen, W. R. Legant, K. Wang, L. Shao, D. E. Milkie, M. W. Davidson, C. Janetopoulos, X. S. Wu, J. A. Hammer III, Z. Liu, *et al.*, "Lattice light-sheet

- microscopy: imaging molecules to embryos at high spatiotemporal resolution,” *Science*, vol. 346, no. 6208, p. 1257998, 2014.
- [4] B. Chen, B.-J. Chang, P. Roudot, F. Zhou, E. Sapoznik, M. Marlar-Pavey, J. B. Hayes, P. T. Brown, C.-W. Zeng, T. Lambert, *et al.*, “Resolution doubling in light-sheet microscopy via oblique plane structured illumination,” *Nature Methods*, vol. 19, no. 11, pp. 1419–1426, 2022.
- [5] Y. Shi, J. S. Tabet, D. E. Milkie, T. A. Daugird, C. Q. Yang, A. T. Ritter, A. Giovannucci, and W. R. Legant, “Smart lattice light-sheet microscopy for imaging rare and complex cellular events,” *Nature Methods*, pp. 1–10, 2024.
- [6] B. Yang, M. Lange, A. Millett-Sikking, X. Zhao, J. Bragantini, S. Vijay Kumar, M. Kamb, R. Gómez-Sjöberg, A. C. Solak, W. Wang, *et al.*, “Daxi—high-resolution, large imaging volume and multi-view single-objective light-sheet microscopy,” *Nature Methods*, vol. 19, no. 4, pp. 461–469, 2022.
- [7] M. B. Bouchard, V. Voleti, C. S. Mendes, C. Lacefield, W. B. Grueber, R. S. Mann, R. M. Bruno, and E. M. Hillman, “Swept confocally-aligned planar excitation (scape) microscopy for high-speed volumetric imaging of behaving organisms,” *Nature Photonics*, vol. 9, no. 2, pp. 113–119, 2015.
- [8] T.-L. Liu, S. Upadhyayula, D. E. Milkie, V. Singh, K. Wang, I. A. Swinburne, K. R. Mosaliganti, Z. M. Collins, T. W. Hiscock, J. Shea, *et al.*, “Observing the cell in its native state: Imaging subcellular dynamics in multicellular organisms,” *Science*, vol. 360, no. 6386, p. eaaq1392, 2018.
- [9] M. Lange, A. Granados, S. VijayKumar, J. Bragantini, S. Ancheta, S. Santhosh, M. Borja, H. Kobayashi, E. McGeever, A. C. Solak, *et al.*, “Zebrahub—multimodal zebrafish developmental atlas reveals the state-transition dynamics of late-vertebrate pluripotent axial progenitors,” *bioRxiv*, pp. 2023–03, 2023.
- [10] F. Amat, B. Höckendorf, Y. Wan, W. C. Lemon, K. McDole, and P. J. Keller, “Efficient processing and analysis of large-scale light-sheet microscopy data,” *Nature Protocols*, vol. 10, no. 11, pp. 1679–1696, 2015.
- [11] M. Maška, V. Ulman, P. Delgado-Rodríguez, E. Gómez-de Mariscal, T. Nečasová, F. A. Guerrero Peña, T. I. Ren, E. M. Meyerowitz, T. Scherr, K. Löffler, *et al.*, “The cell tracking challenge: 10 years of objective benchmarking,” *Nature Methods*, pp. 1–11, 2023.
- [12] D. Feldman, A. Singh, J. L. Schmid-Burgk, R. J. Carlson, A. Mezger, A. J. Garrity, F. Zhang, and P. C. Blainey, “Optical pooled screens in human cells,” *Cell*, vol. 179, no. 3, pp. 787–799, 2019.
- [13] I. E. Ivanov, E. Hirata-Miyasaki, T. Chandler, R. C. Kovilakam, Z. Liu, C. Liu, M. D. Leonetti, B. Huang, and S. B. Mehta, “Mantis: high-throughput 4d imaging and analysis of the molecular and physical architecture of cells,” *bioRxiv*, pp. 2023–12, 2023.
- [14] S. Kim, M. Pochitaloff, G. A. Stooke-Vaughan, and O. Campàs, “Embryonic tissues as active foams,” *Nature Physics*, vol. 17, no. 7, pp. 859–866, 2021.
- [15] K. McDole, L. Guignard, F. Amat, A. Berger, G. Ma-landain, L. A. Royer, S. C. Turaga, K. Branson, and P. J. Keller, “In toto imaging and reconstruction of post-implantation mouse development at the single-cell level,” *Cell*, vol. 175, no. 3, pp. 859–876, 2018.
- [16] Y. Wan, Z. Wei, L. L. Looger, M. Koyama, S. Druckmann, and P. J. Keller, “Single-cell reconstruction of emerging population activity in an entire developing circuit,” *Cell*, vol. 179, no. 2, pp. 355–372, 2019.
- [17] J. D. Currie, A. Kawaguchi, R. M. Traspas, M. Schuez, O. Chara, and E. M. Tanaka, “Live imaging of axolotl digit regeneration reveals spatiotemporal choreography of diverse connective tissue progenitor pools,” *Developmental cell*, vol. 39, no. 4, pp. 411–423, 2016.
- [18] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, “Cell detection with star-convex polygons,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 265–273, Springer, 2018.
- [19] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, “Cellpose: a generalist algorithm for cellular segmentation,” *Nature Methods*, vol. 18, no. 1, pp. 100–106, 2021.
- [20] A. Archit, S. Nair, N. Khalid, P. Hilt, V. Rajashekar, M. Freitag, S. Gupta, A. Dengel, S. Ahmed, and C. Pape, “Segment anything for microscopy,” *bioRxiv*, pp. 2023–08, 2023.
- [21] K. Ulicna, G. Vallardi, G. Charras, and A. R. Lowe, “Automated deep lineage tree analysis using a bayesian single cell tracking approach,” *Frontiers in Computer Science*, vol. 3, p. 734559, 2021.
- [22] K. Sugawara, Ç. Çevrim, and M. Averof, “Tracking cell lineages in 3d by incremental deep learning,” *eLife*, vol. 11, p. e69380, 2022.

- [23] K. Löffler and R. Mikut, “Embed-track—simultaneous cell segmentation and tracking through learning offsets and clustering bandwidths,” *IEEE Access*, vol. 10, pp. 77147–77157, 2022.
- [24] C. Malin-Mayor, P. Hirsch, L. Guignard, K. McDole, Y. Wan, W. C. Lemon, D. Kainmueller, P. J. Keller, S. Preibisch, and J. Funke, “Automated reconstruction of whole-embryo cell lineages by learning from sparse annotations,” *Nature Biotechnology*, pp. 1–6, 2022.
- [25] V. Ulman, M. Maška, K. E. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, *et al.*, “An objective comparison of cell-tracking algorithms,” *Nature Methods*, vol. 14, no. 12, pp. 1141–1152, 2017.
- [26] M. Schiegg, P. Hanslovsky, B. X. Kausler, L. Hufnagel, and F. A. Hamprecht, “Conservation tracking,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2928–2935, 2013.
- [27] F. Jug, T. Pietzsch, D. Kainmüller, J. Funke, M. Kaiser, E. van Nimwegen, C. Rother, and G. Myers, “Optimal joint segmentation and tracking of escherichia coli in the mother machine,” in *Bayesian and Graphical Models for Biomedical Imaging: First International Workshop, BAMBI 2014, Cambridge, MA, USA, September 18, 2014, Revised Selected Papers*, pp. 25–36, Springer, 2014.
- [28] E. Türetken, X. Wang, C. J. Becker, C. Haubold, and P. Fua, “Network flow integer programming to track elliptical cells in time-lapse sequences,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 4, pp. 942–951, 2016.
- [29] F. Jug, E. Levinkov, C. Blasse, E. W. Myers, and B. Andres, “Moral lineage tracing,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5926–5935, 2016.
- [30] M. Rempfler, J.-H. Lange, F. Jug, C. Blasse, E. W. Myers, B. H. Menze, and B. Andres, “Efficient algorithms for moral lineage tracing,” in *IEEE International Conference on Computer Vision*, pp. 4695–4704, 2017.
- [31] F. Amat, W. Lemon, D. P. Mossing, K. McDole, Y. Wan, K. Branson, E. W. Myers, and P. J. Keller, “Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data,” *Nature Methods*, vol. 11, no. 9, pp. 951–958, 2014.
- [32] M. Pachitariu and C. Stringer, “Cellpose 2.0: how to train your own model,” *Nature Methods*, vol. 19, no. 12, pp. 1634–1641, 2022.
- [33] A. Wolny, L. Cerrone, A. Vijayan, R. Tofanelli, A. V. Barro, M. Louveaux, C. Wenzl, S. Strauss, D. Wilson-Sánchez, R. Lymbouridou, *et al.*, “Accurate and versatile 3d segmentation of plant tissues at cellular resolution,” *eLife*, vol. 9, p. e57613, 2020.
- [34] A. Vijayan, T. A. Mody, Q. Yu, A. Wolny, L. Cerrone, S. Strauss, M. Tsiantis, R. S. Smith, F. Hamprecht, A. Kreshuk, *et al.*, “A deep learning-based toolkit for 3d nuclei segmentation and quantitative analysis in cellular and tissue context,” *bioRxiv*, pp. 2024–02, 2024.
- [35] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, *et al.*, “Fiji: an open-source platform for biological-image analysis,” *Nature Methods*, vol. 9, no. 7, pp. 676–682, 2012.
- [36] N. Sofroniew, T. Lambert, K. Evans, J. Nunez-Iglesias, G. Bokota, P. Winston, G. Peña-Castellanos, K. Yamauchi, M. Bussonnier, D. Doncila Pop, A. Can Solak, Z. Liu, P. Wadhwa, A. Burt, G. Buckley, A. Sweet, L. Migas, V. Hilsenstein, L. Gaifas, J. Bragantini, J. Rodríguez-Guerra, H. Muñoz, J. Freeman, P. Boone, A. R. Lowe, C. Gohlke, L. Royer, A. Pierré, H. Har-Gil, and A. McGovern, “napari: a multi-dimensional image viewer for Python.”
- [37] D. Ershov, M.-S. Phan, J. W. Pylvänäinen, S. U. Rigaud, L. Le Blanc, A. Charles-Orszag, J. R. Conway, R. F. Laine, N. H. Roy, D. Bonazzi, *et al.*, “Trackmate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines,” *Nature Methods*, vol. 19, no. 7, pp. 829–832, 2022.
- [38] T. Pietzsch and J.-Y. Tinevez, “Mastodon.”
- [39] J. Ma, R. Xie, S. Ayyadhury, C. Ge, A. Gupta, R. Gupta, S. Gu, Y. Zhang, G. Lee, J. Kim, *et al.*, “The multimodality cell segmentation challenge: toward universal solutions,” *Nature Methods*, pp. 1–11, 2024.
- [40] C. Edlund, T. R. Jackson, N. Khalid, N. Bevan, T. Dale, A. Dengel, S. Ahmed, J. Trygg, and R. Sjögren, “Live-cell—a large-scale dataset for label-free live cell segmentation,” *Nature Methods*, vol. 18, no. 9, pp. 1038–1045, 2021.
- [41] P. Arbelaez, “Boundary extraction in natural images using ultrametric contour maps,” in *Conference on Computer Vision and Pattern Recognition Workshop*, pp. 182–182, IEEE, 2006.

- [42] L. Najman and M. Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1163–1173, 1996.
- [43] L. Najman, "On the equivalence between hierarchical segmentations and ultrametric watersheds," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 3, pp. 231–247, 2011.
- [44] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.
- [45] J. Bragantini, M. Lange, and L. Royer, "Large-scale multi-hypotheses cell tracking using ultrametric contours maps," *European Conference on Computer Vision (to appear)*, 2024.
- [46] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023.
- [47] J. Forrest, T. Ralphs, S. Vigerske, H. G. Santos, J. Forrest, L. Hafer, B. Kristjansson, jpfasano, Edwin-Straver, M. Lubin, Jan-Willem, rlougee, jpgoncal1, S. Brito, h-i gassmann, Cristina, M. Saltzman, tostost, B. Pitrus, F. Matsushima, and to st, "coin-or/cbc: Release releases/2.10.11," Oct. 2023.
- [48] A. Ruggieri, E. Dazert, P. Metz, S. Hofmann, J.-P. Bergeest, J. Mazur, P. Bankhead, M.-S. Hiet, S. Kallis, G. Alvisi, *et al.*, "Dynamic oscillation of translation and stress granule formation mark the cellular response to virus infection," *Cell host & microbe*, vol. 12, no. 1, pp. 71–85, 2012.
- [49] P. Matula, M. Maška, D. V. Sorokin, P. Matula, C. Ortiz-de Solórzano, and M. Kozubek, "Cell tracking accuracy measurement based on comparison of acyclic oriented graphs," *PloS one*, vol. 10, no. 12, p. e0144959, 2015.
- [50] D. Cai, K. B. Cohen, T. Luo, J. W. Lichtman, and J. R. Sanes, "Improved tools for the brainbow toolbox," *Nature Methods*, vol. 10, no. 6, pp. 540–547, 2013.
- [51] K. Y. Chan, C.-C. S. Yan, H.-Y. Roan, S.-C. Hsu, T.-L. Tseng, C.-D. Hsiao, C.-P. Hsu, and C.-H. Chen, "Skin cells undergo asynthetic fission to expand body surfaces in zebrafish," *Nature*, vol. 605, no. 7908, pp. 119–125, 2022.
- [52] K. Weber, M. Thomaschewski, D. Benten, and B. Fehse, "Rgb marking with lentiviral vectors for multicolor clonal cell tracking," *Nature protocols*, vol. 7, no. 5, pp. 839–849, 2012.
- [53] J. Hayashida, K. Nishimura, and R. Bise, "Mpm: Joint representation of motion and position map for cell tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3823–3832, 2020.
- [54] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.
- [55] R. d. A. Lotufo, A. X. Falcão, and F. A. Zampirolli, "IFT-watershed from gray-scale marker," in *Conf. on Graphics, Patterns and Images (SIBGRAPI)*, pp. 146–152, IEEE, 2002.
- [56] A. X. Falcão, J. Stolfi, and R. A. de Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, 2004.
- [57] Z. Liu, E. Hirata-Miyasaki, S. Pradeep, J. Rahm, C. Foley, T. Chandler, I. Ivanov, H. Woosley, T. Lao, A. Balasubramanian, *et al.*, "Robust virtual staining of landmark organelles," *bioRxiv*, pp. 2024–05, 2024.
- [58] C. Stringer and M. Pachitariu, "Cellpose3: one-click image restoration for improved cellular segmentation," *bioRxiv*, pp. 2024–02, 2024.
- [59] M. Lab, "recOrder."
- [60] J. B. Pawley, "Fundamental limits in confocal microscopy," in *Handbook of biological confocal microscopy*, pp. 20–42, Springer, 2006.
- [61] M. Pereyra, A. Drusko, F. Krämer, F. Strobl, E. H. Stelzer, and F. Matthäus, "Quickpiv: Efficient 3d particle image velocimetry software applied to quantifying cellular migration during embryogenesis," *BMC bioinformatics*, vol. 22, pp. 1–20, 2021.
- [62] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "Voxelmorph: a learning framework for deformable medical image registration," *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1788–1800, 2019.
- [63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, pp. 8026–8037, 2019.
- [64] A. Jain, *Molecular, Cellular and Mechanical basis of Epithelial Morphogenesis during Tribolium Embryogenesis*. PhD thesis, Technische Universität Dresden, 2018.

- [65] I. Heemskerk and S. J. Streichan, "Tissue cartography: compressing bio-image data by dimensional reduction," *Nature Methods*, vol. 12, no. 12, pp. 1139–1142, 2015.
- [66] J. I. Murray, Z. Bao, T. J. Boyle, M. E. Boeck, B. L. Mericle, T. J. Nicholas, Z. Zhao, M. J. Sandel, and R. H. Waterston, "Automated analysis of embryonic gene expression with cellular resolution in *c. elegans*," *Nature Methods*, vol. 5, no. 8, pp. 703–709, 2008.
- [67] C. E. Akbaş, V. Ulman, M. Maška, F. Jug, and M. Kozubek, "Automatic fusion of segmentation and tracking labels," in *Computer Vision-ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part VI* 15, pp. 446–454, Springer, 2019.
- [68] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [69] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu, "Learning to predict crisp boundaries," in *European Conference on Computer Vision*, pp. 562–578, 2018.
- [70] S. Xie and Z. Tu, "Holistically-nested edge detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1395–1403, 2015.
- [71] R. Tomer, K. Khairy, F. Amat, and P. J. Keller, "Quantitative high-speed imaging of entire developing embryos with simultaneous multiview light-sheet microscopy," *Nature Methods*, vol. 9, no. 7, pp. 755–763, 2012.
- [72] A. B. Yoo, M. A. Jette, and M. Grondona, "SLURM: Simple linux utility for resource management," in *Job Scheduling Strategies for Parallel Processing: 9th International Workshop*, pp. 44–60, Springer, 2003.
- [73] P. K. Shah, A. Santella, A. Jacobo, K. Siletti, A. Hudspeth, and Z. Bao, "An in toto approach to dissecting cellular interactions in complex tissues," *Developmental cell*, vol. 43, no. 4, pp. 530–540, 2017.
- [74] A. Erzberger, A. Jacobo, A. Dasgupta, and A. Hudspeth, "Mechanochemical symmetry breaking during morphogenesis of lateral-line sensory organs," *Nature physics*, vol. 16, no. 9, pp. 949–957, 2020.
- [75] O. Viader-Llangués, V. Lupperger, L. Pola-Morell, C. Marr, and H. López-Schier, "Live cell-lineage tracing and machine learning reveal patterns of organ regeneration," *eLife*, vol. 7, p. e30823, Mar. 2018. Publisher: eLife Sciences Publications, Ltd.
- [76] M. N. Hewitt, I. A. Cruz, T. H. Linbo, and D. W. Raible, "Spherical harmonics analysis reveals cell shape-fate relationships in zebrafish lateral line neuro-masts," *Development*, vol. 151, p. dev202251, Jan. 2024.
- [77] A. Hagberg, P. Swart, and D. S. Chult, "Exploring network structure, dynamics, and function using networkx," tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
- [78] W. McKinney and P. Team, "Pandas-powerful python data analysis toolkit," 2015.
- [79] L. von Chamier, R. F. Laine, J. Jukkala, C. Spahn, D. Krentzel, E. Nehme, M. Lerche, S. Hernández-Pérez, P. K. Mattila, E. Karinou, *et al.*, "Democratising deep learning for microscopy with zero-costdl4mic," *Nature Communications*, vol. 12, no. 1, p. 2276, 2021.
- [80] B. Gallusser and M. Weigert, "Trackastra: Transformer-based cell tracking for live-cell microscopy," *arXiv preprint arXiv:2405.15700*, 2024.
- [81] M. Lange, A. Solak, S. Vijay Kumar, H. Kobayashi, B. Yang, and L. A. Royer, "Zaf, the first open source fully automated feeder for aquatic facilities," *Elife*, vol. 10, p. e74234, 2021.
- [82] J. Peloggia, D. Münch, P. Meneses-Giles, A. Romero-Carvajal, M. E. Lush, N. D. Lawson, M. McClain, Y. A. Pan, and T. Piotrowski, "Adaptive cell invasion maintains lateral line organ homeostasis in response to environmental changes," *Developmental Cell*, vol. 56, no. 9, pp. 1296–1312.e7, 2021.
- [83] D. E. Dalle Nogare, N. Natesh, H. D. Vishwasrao, H. Shroff, and A. B. Chitnis, "Zebrafish Posterior Lateral Line primordium migration requires interactions between a superficial sheath of motile cells and the skin," *eLife*, vol. 9, p. e58251, 2020.
- [84] P. Shah, C. M. Hobson, S. Cheng, M. J. Colville, M. J. Paszek, R. Superfine, and J. Lammerding, "Nuclear deformation causes dna damage by increasing replication stress," *Current Biology*, vol. 31, no. 4, pp. 753–765, 2021.
- [85] D. Balciunas, K. J. Wangensteen, A. Wilber, J. Bell, A. Geurts, S. Sivasubbu, X. Wang, P. B. Hackett, D. A. Largaespada, R. S. McIvor, *et al.*, "Harnessing a high cargo-capacity transposon for genetic applications in vertebrates," *PLoS genetics*, vol. 2, no. 11, p. e169, 2006.

- [86] R. Nishino and S. H. C. Loomis, "Cupy: A numpy-compatible library for nvidia GPU calculations," *Advances in Neural Information Processing Systems Workshop*, vol. 151, no. 7, 2017.
- [87] Z. Liu, I. Ivanov, J. Bragantini, T. Chandler, A. C. Solak, E. Hirata-Miyasaki, C. Foltz, L.-H. Yeh, and S. Mehta, "iohub," February 2024.
- [88] S. Ramirez, "FastAPI."
- [89] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.
- [90] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, 2020.
- [91] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.
- [92] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A llvm-based python jit compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6, 2015.
- [93] J. Moore, D. Basurto-Lozada, S. Besson, J. Bogovic, J. Bragantini, E. M. Brown, J.-M. Burel, X. Casas Moreno, G. de Medeiros, E. E. Diel, *et al.*, "Ome-zarr: a cloud-optimized bioimaging file format with international community support," *Histochemistry and Cell Biology*, vol. 160, no. 3, pp. 223–251, 2023.
- [94] J. Moore, C. Allan, S. Besson, J.-M. Burel, E. Diel, D. Gault, K. Kozlowski, D. Lindner, M. Linkert, T. Manz, *et al.*, "Ome-ngff: a next-generation file format for expanding bioimaging data-access strategies," *Nature Methods*, vol. 18, no. 12, pp. 1496–1498, 2021.
- [95] L. Najman, J. Cousty, and B. Perret, "Playing with kruskal: algorithms for morphological trees in edge-weighted graphs," in *Mathematical Morphology and Its Applications to Signal and Image Processing*, pp. 135–146, Springer, 2013.
- [96] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.
- [97] R. E. Tarjan, "Efficiency of a good but not linear set union algorithm," *Journal of the ACM (JACM)*, vol. 22, no. 2, pp. 215–225, 1975.
- [98] T. Gals, "traccuracy."
- [99] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.
- [100] A. Melnikov and I. Fette, "The WebSocket Protocol." RFC 6455, Dec. 2011.
- [101] M. Otto and J. Thornton, "Bootstrap 5."
- [102] The Qt Company, "Qt - cross-platform software libraries and apis." <https://www.qt.io/>, 2024. Accessed: 2024-07-19.