inTRACKtive — A Web-Based Tool for Interactive Cell Tracking Visualization

Teun A.P.M. Huijben[a], Ashley G. Anderson III[b], Andrew Sweet[b], Erin Hoops[b], Connor Larsen[b], Kyle Awayan[a], Jordão Bragantini[a], Merlin Lange[a], Chi-Li Chiu[b], Loïc A. Royer[a]

[a] Chan Zuckerberg Biohub, San Francisco, United States
[b] Chan Zuckerberg Initiative, Redwood City, United States

correspondence to teun.huijben@czbiohub.org, loic.royer@czbiohub.org

We introduce inTRACKtive, an innovative web-based tool for interactive visualization and sharing of large 3D cell tracking datasets, eliminating the need for software installations or data downloads. Built with modern web technologies, inTRACKtive enables researchers to explore cell-tracking results from terabyte-scale microscopy data, conduct virtual fate-mapping experiments, and share these results via simple hyperlinks. The platform powers the Virtual Embryo Zoo, an online resource showcasing cell-tracking datasets from state-of-the-art light-sheet embryonic microscopy of six model organisms. inTRACKtive's open-source code allows users to visualize their own data or host customized viewer instances. By providing easy access to complex tracking datasets, inTRACKtive offers a versatile, interactive, collaborative tool for developmental biology.

The field of developmental biology is undergoing a profound transformation, driven by groundbreaking advances across multiple disciplines.[1] In particular, the convergence of state-of-the-art light-sheet microscopy and cutting-edge image analysis algorithms is redefining our ability to study and understand embryogenesis *in vivo*.[2] Light-sheet microscopy enables rapid, large-scale, multicolor imaging of developing embryos in unprecedented detail, allowing scientists to capture dynamic processes as they unfold in real-time.[3–5] Simultaneously, the advent of advanced cell tracking technologies, such as Ultrack,[6] Linajea,[7,8] Elephant,[9] TGMM,[10] Trackmate,[11] Mastodon[12] and TrackAstra,[13] has made it possible to follow individual cells throughout embryonic development with remarkable precision and speed.[14] These innovations, which have emerged in the past two decades, are only now realizing their full potential for embryology, offering the capability to track almost all cells in an embryo, from the earliest stages of development to fully gastrulated embryos and beyond. This unprecedented level of detail opens the door to answering fundamental questions about tissue development, organ formation, and the intricate orchestration of the cell behaviors that govern the entire embryo.[15,16]

However, the size and complexity of these microscopy datasets — often reaching tens of terabytes and containing up to tens of thousands of cells across hundreds of time points — pose significant challenges. While these imaging datasets hold the potential to reveal novel biological insights, accessing and interacting with such complex data typically demands highly specialized technical expertise and substantial computational resources. Consequently, only a limited subset of researchers can fully leverage these data, creating a barrier to broader adoption. Could a more user-friendly solution be developed that enables researchers to explore and analyze cell tracks of these datasets, regardless of their computational expertise or available hardware? Existing software tools allow visualization and analysis of cell tracking data (*e.g.*, *napari*,[17] TrackMate's TrackScheme,[11] Mastodon,[12] and CellTracksColab[18]), However, while very powerful in their respective application domains, most of these approaches require native software, have limited 3D support, or are intended for the curation of tracks; not the interactive exploration of large cell tracking datasets.

We introduce inTRACKtive, a web-based tool designed to address this challenge by providing an intuitive platform for visualizing and interacting with large 2D and 3D cell tracking datasets. With inTRACKtive, users can seamlessly explore large tracking datasets in an interactive, three-dimensional environment (Fig. 1a-b and Supplementary Video 1). One can select specific (groups of) cells and trace their lineages through time, which we validate by tracking tailbud cells during zebrafish development (Fig. S1); recapitulating previous findings that tailbud cells end up in the tail of the zebrafish embryo, particularly in the mesoderm, neural tube and tailbud itself.[16] The platform operates entirely in a web browser, requiring no software installation or manual data downloads, making it easily accessible to any user. In addition, users can share their exact viewing configuration - complete with view settings, zoom levels, and selected cells - simply by sharing a link. This allows for effortless collaboration and dissemination of results, when the dataset is located on a publicly accessible server.

We used inTRACKtive to create the Virtual Embryo Zoo (embryozoo.org), an online resource that showcases the highest-quality tracked light-sheet microscopy datasets of

developing embryos (Fig. 1c and Supplementary Video 2). This platform allows researchers to investigate the early embryogenesis of six widely studied model organisms (Table 1), Drosophila,[10] zebrafish,[16] C. elegans,[19] ascidian,[20] mouse,[4] and Tribolium[21] — via a web-based interface. Not only can users visualize these datasets, but they can also download the complete tracking data, which can be easily imported into *napari* and Python for further analysis, such as quantitative morphological analysis of track shapes, cell densities, and movement directions. In addition, the Virtual Embryo Zoo offers researchers the opportunity to contribute and showcase their own datasets (Fig. 1d), with the vision of creating a growing collaborative resource for the community.

inTRACKtive is an ideal tool for crafting custom cell-tracking visualizations. In addition to selecting and tracing individual cells, the platform allows users to color cells based on any provided attribute. For example, we demonstrate its application in Drosophila development, where cells are colored according to their velocity (Fig. S2 and Supplementary Videos 3 and 4). This functionality is not limited to velocity; it can represent any attributes of interest such as cell size, gene expression, or track length, offering extensive flexibility for exploring complex datasets.

Technically, inTRACKtive is built using modern web technologies, including TypeScript, React, and three.js, with Vite as the bundler and Vercel for deployment. The platform is optimized for performance, allowing users to interact with large-scale datasets. The app runs entirely client-side and only relies on static hosting for both the application and data. This makes it easy and inexpensive to self-host the application and make your own data accessible. One key innovation is the use of a specialized cell tracking format based on Zarr[22] (see Fig. S3 and Supplementary Methods), which enables asynchronous, lazy data loading. This ensures that tracking results from large imaging datasets can be explored seamlessly without significant delays or computational overhead. The tool works in any browser, including tablets and smartphones, offering flexibility in how users interact with the data.

Looking ahead, there is potential to expand inTRACKtive's capabilities. One exciting possibility is the incorporation of imaging data alongside cell-tracking results. Another area for development is further customization of track visualization and analysis of the selected tracks. While this functionality is not yet available directly in inTRACKtive, the platform allows the export of a user-selected subset of data to a `csv` format that can be further analyzed and visualized in tools like *napari*[17]. Thus, inTRACKtive is an ideal platform for sharing and disseminating complex cell tracking results, combining ease of use with the flexibility to integrate with more specialized software for advanced analyses.

In summary, inTRACKtive offers a powerful, versatile tool for the interactive visualization of cell tracking data, eliminating the need for complex software installations or large data downloads. Beyond visualization, inTRACKtive enables users to select individual cells, trace their lineages, and investigate both their ancestors and descendants, facilitating deeper insights into developmental processes. The platform serves multiple use cases: as a gateway for exploring datasets in the Virtual Embryo Zoo, as a tool for visualizing a researcher's own cell tracking data (easily shared via a link), and as a customizable viewer for users who want to host their own instances. Conversion scripts for exporting cell tracking results into the Zarr format used by inTRACKtive are provided. Furthermore, we provide an intuitive command line and Python API, including a *napari* widget and a Jupyter Notebook widget, enabling users to launch inTRACKtive directly from a terminal, Jupyter Notebook or *napari* (Supplementary Video 5). This eliminates the need to spin up a local host or convert data to Zarr manually, allowing for seamless visualization of custom tracking datasets.

Importantly, inTRACKtive is freely available and open-source (github.com/royerlab/inTRACKtive). By enabling the creation of the Virtual Embryo Zoo, inTRACKtive has the potential to become a valuable resource to the cell and developmental biology community and beyond. Moreover, inTRACKtive can be used to visualize tracking data acquired with any 2D (example[6]) or 3D microscopy modality from which tracking data can be obtained, from organoids down to single molecules. inTRACKtive offers a unique, browser-based playground for interacting with state-of-the-art microscopy tracking datasets.

Supplementary Videos

1. inTRACKtive web-viewer;
2. Virtual Embryo Zoo website;
3. Color cell labeling in inTRACKtive based on any attribute;
4. Visualizing velocity, density, and structure tensor in inTRACKtive;
5. Using inTRACKtive from the command line, from napari, from a Jupyter notebook, and *within* a Jupyter notebook;

Data availability statement

The tracking data presented in this study are available for download from the Virtual Embryo Zoo website (embryozoo.org). The datasets are either sourced from the original authors or tracked by us using the Ultrack[6,23] algorithm (see Table 1). All data has been converted and normalized to a common (see Fig. S3) format compatible with inTRACKtive and *napari*, ensuring seamless interaction and visualization.

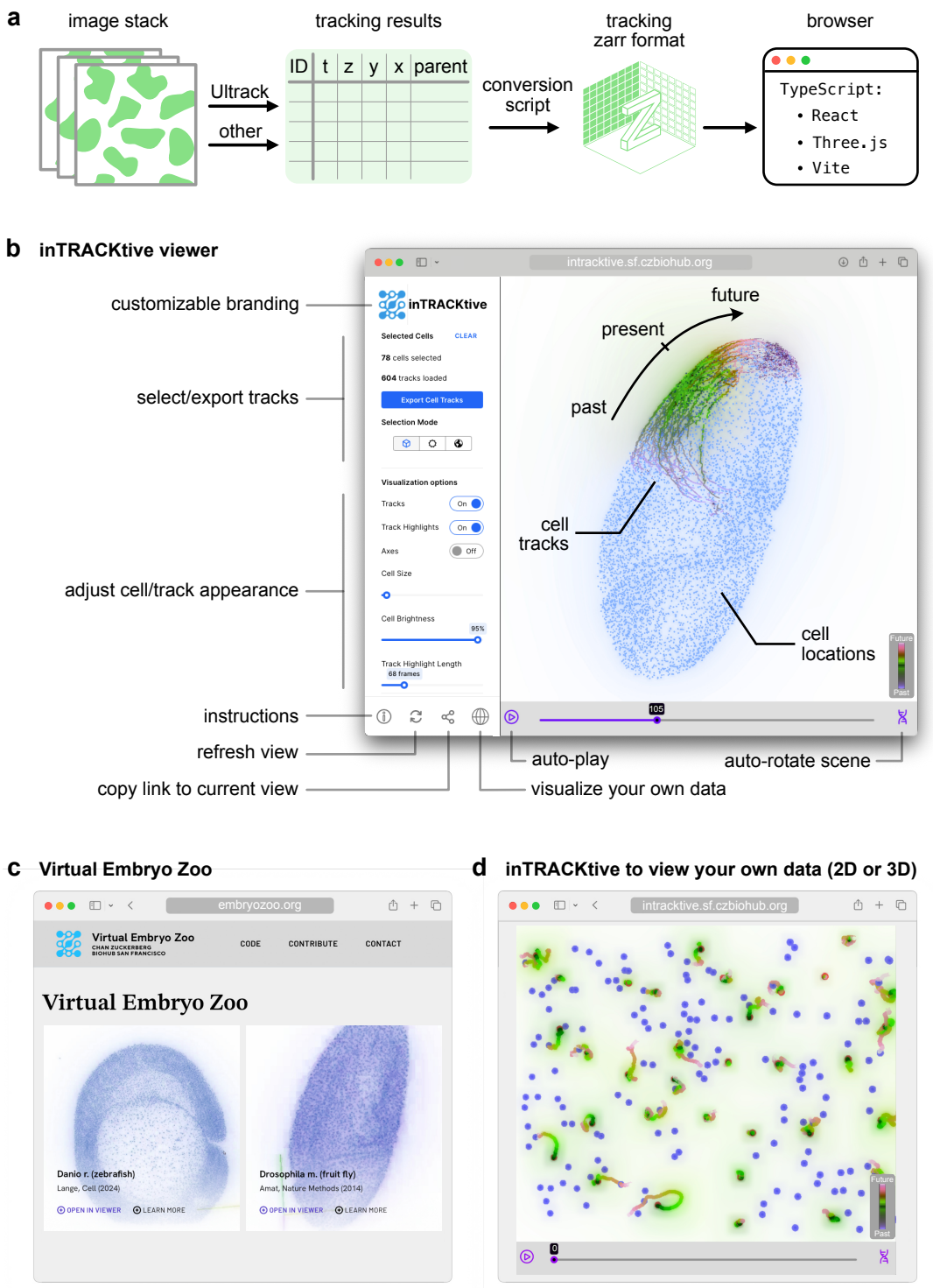

Fig. 1 | inTRACKtive cell tracking viewer and its applications. a inTRACKtive data pipeline. After users track their datasets with their preferred cell tracking software, the tracking results are converted to our custom Zarr format using a provided conversion script, and can be interactively visualized in the web-based tool. **b** The inTRACKtive web-viewer of cell tracking results. Users can upload their own data, select cells of interest, trace their lineages, adjust the appearance of the cells and tracks, and copy the state of the viewer into a sharable link. **c**, The Virtual Embryo Zoo encompasses six light-sheet microscopy datasets of developing embryos. The website provides a video graphical overview of the datasets, with one-click access to inTRACKtive viewer for visualizing the cell tracking data (see **b**) and the details of the data (paper, organism, authors, tracking algorithm, etc.). **d** inTRACKtive can be used to visualize one's own 2D or 3D cell tracking data in the browser.

Dataset	Paper	Tracking
zebrafish (Danio rerio)	M. Lange, *Cell* (2024)[16]	Ultrack[6,23]
fruit fly (Drosophila melanogaster)	F. Amat, *Nature Methods* (2014)[10]	TGMM[10]
mouse (Mus musculus)	K. McDole, *Cell* (2018)[4]	Linajea[7,8]
ascidian (Phallusia mammillata)	L. Guignard, *Science* (2020)[20]	Astec[20]
worm (Caenorhabditis elegans)	M. W. Moyle, *Nature* (2021)[19]	Linajea[7,8]
beetle (Tribolium castaneum)	A. Jain, (2018)[21]	Ultrack[6,23]

Table 1 | Details of the six embryo datasets portrayed in the Virtual Embryo Zoo.

Code availability statement

The repository for inTRACKtive, including documentation, examples code, and videos, can be found at github.com/royerlab/inTRACKtive.

Acknowledgements

We extend our gratitude to all the original authors who kindly provided data and guidance: Philipp Keller, Hari Shroff, Kate McDole, Akanksha Jain, Robert Haase, and Grégoire Malandain. We thank the Royer group at CZ Biohub SF and Estibaliz Gómez-de-Mariscal for feedback on inTRACKtive and the Virtual Embryo Zoo website. We appreciate Sandy Schmid for her constructive feedback on the manuscript. We thank CZI for the Science Design System (SDS) UI component library. Chan Zuckerberg Biohub San Francisco (CZB SF) funded this work. We thank the CZB SF donors P. Chan and M. Zuckerberg for their generous support.

Competing interests

The authors declare no competing interests.

Author Contributions

J.B. and L.A.R. conceived the research. A.A., A.S., E.H, C.L. and C-L.C developed the initial version of inTRACKtive. T.A.P.M.H. collected the light-sheet datasets from the original authors, converted them into the applicable formats, and further developed inTRACKtive (guided by A.A. and A.S.). K.A. and T.A.P.M.H. built the Virtual Embryo Zoo website. J.B. performed tracking of the Tribolium and zebrafish data using Ultrack. M.L. provided insight into the zebrafish tailbud migration. T.A.P.M.H. and L.A.R. wrote the paper. T.A.P.M.H. led the project. L.A.R. supervised the research. All authors contributed to editing the manuscript.

References

[1] P. Liberali and A. F. Schier, "The evolution of developmental biology through conceptual and technological revolutions," *Cell*, 2024.

[2] P. J. Keller, "Imaging morphogenesis: technological advances and biological insights," *Science*, vol. 340, no. 6137, p. 1234168, 2013.

[3] 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.

[4] K. McDole, L. Guignard, F. Amat, A. Berger, G. Malandain, 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.

[5] B. Yang, M. Lange, A. Millett-Sikking, X. Zhao, J. Bragantini, S. VijayKumar, 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.

[6] J. Bragantini, I. Theodoro, X. Zhao, T. A. Huijben, E. Hirata-Miyasaki, S. VijayKumar, A. Balasubramanian, T. Lao, R. Agrawal, S. Xiao, *et al.*, "Ultrack: pushing the limits of cell tracking across biological scales," *bioRxiv*, pp. 2024–09, 2024.

[7] 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*, vol. 41, no. 1, pp. 44–49, 2023.

[8] P. Hirsch, C. Malin-Mayor, A. Santella, S. Preibisch, D. Kainmueller, and J. Funke, "Tracking by weakly-supervised learning and graph optimization for whole-embryo c. elegans lineages," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 25–35, Springer, 2022.

[9] K. Sugawara, Ç. Çevrim, and M. Averof, "Tracking cell lineages in 3d by incremental deep learning," *Elife*, vol. 11, p. e69380, 2022.

[10] F. Amat, W. Lemon, D. P. Mossing, K. McDole, Y. Wan, K. Branson, E. W. Myers, and P. J. Keller, "Fast, ac-

curate reconstruction of cell lineages from large-scale fluorescence microscopy data," *Nature methods*, vol. 11, no. 9, pp. 951–958, 2014.

[11] 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.

[12] J.-Y. Tinevez, "Mastodon." https://github.com/mastodon-sc/mastodon, 2024.

[13] B. Gallusser and M. Weigert, "Trackastra: Transformer-based cell tracking for live-cell microscopy," *arXiv preprint arXiv:2405.15700*, 2024.

[14] M. Maška, V. Ulman, P. Delgado-Rodriguez, 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*, vol. 20, no. 7, pp. 1010–1020, 2023.

[15] Y. Wan, K. McDole, and P. J. Keller, "Light-sheet microscopy and its potential for understanding developmental processes," *Annual review of cell and developmental biology*, vol. 35, no. 1, pp. 655–681, 2019.

[16] M. Lange, A. Granados, S. VijayKumar, J. Bragantini, S. Ancheta, Y.-J. Kim, S. Santhosh, M. Borja, H. Kobayashi, E. McGeever, *et al.*, "A multimodal zebrafish developmental atlas reveals the state-transition dynamics of late-vertebrate pluripotent axial progenitors," *Cell*, vol. 187, no. 23, pp. 6742–6759, 2024.

[17] 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."

[18] E. Gómez-de Mariscal, H. Grobe, J. W. Pylvänäinen, L. Xénard, R. Henriques, J.-Y. Tinevez, and G. Jacquemet, "Celltrackscolab is a platform that enables compilation, analysis, and exploration of cell tracking data," *Plos Biology*, vol. 22, no. 8, p. e3002740, 2024.

[19] M. W. Moyle, K. M. Barnes, M. Kuchroo, A. Gonopolskiy, L. H. Duncan, T. Sengupta, L. Shao, M. Guo, A. Santella, R. Christensen, *et al.*, "Structural and developmental principles of neuropil assembly in c. elegans," *Nature*, vol. 591, no. 7848, pp. 99–104, 2021.

[20] L. Guignard, U.-M. Fiúza, B. Leggio, J. Laussu, E. Faure, G. Michelin, K. Biasuz, L. Hufnagel, G. Malandain, C. Godin, *et al.*, "Contact area–dependent cell communication and the morphological invariance of ascidian embryogenesis," *Science*, vol. 369, no. 6500, p. eaar5663, 2020.

[21] A. Jain, "Molecular, cellular and mechanical basis of epithelial morphogenesis during tribolium embryogenesis," *PhD Thesis*, 2018.

[22] A. Miles, J. Kirkham, M. Durant, J. Bourbeau, T. Onalan, J. Hamman, Z. Patel, shikharsg, M. Rocklin, raphael dussin, V. Schut, E. S. de Andrade, R. Abernathey, C. Noyes, sbalmer, pyup.io bot, T. Tran, S. Saalfeld, J. Swaney, J. Moore, J. Jevnik, J. Kelleher, J. Funke, G. Sakkis, C. Barnes, and A. Banihirwe, "zarr-developers/zarr-python: v2.4.0," Apr. 2020.

[23] J. Bragantini, M. Lange, and L. Royer, "Large-scale multi-hypotheses cell tracking using ultrametric contours maps," in *European Conference on Computer Vision*, 2024.

Supplementary Methods

The cell tracking data is structured as a table with six columns: `track_id`, `t`, `z`, `y`, `x`, and `parent_track_id`, where each row represents a cell at a single time point (Fig. S3a). The `track_id` column provides the identifier for each track, while `t`, `z`, `y`, and `x` represent the time point and spatial coordinates of a specific cell (providing `z` is optional). Note that the order of the columns is not important, since they are accessed by name. The `parent_track_id` column specifies the `track_id` of the parent track, with a value of -1 indicating no parent. To enable fast and asynchronous loading in inTRACKtive, this data is stored in a specialized Zarr[22] format optimized for efficient interaction. The Zarr format allows users to select a single cell at a specific time point and visualize its trajectory, as well as those of its ancestors and descendants (Fig. S3b). The cell tracking data is organized into a Zarr store comprising four Zarr arrays: `points`, `points_to_tracks`, `tracks_to_tracks`, and `tracks_to_points` (Fig. S3c-d). Within this context, cells are referred to as `points`. Below, we detail these four arrays and their roles before describing their storage in the Zarr format.

Four matrices

The `points` matrix (Fig. S3c) is a dense, ragged array of 32-bit floats with dimensions (`n_timepoints`, `num_values_per_point` $\times$ `max_points_per_timepoint`). The number of rows (`n_timepoints`) corresponds to the total time points. Each row represents a single time point, containing the coordinates (`z`, `y`, `x`) of all observed points. Rows are padded with `-9999.9` to reach the `max_points_per_timepoint`. Each point is assigned a unique identifier based on its time point (`t`) and index (`n`) within that time point: $\texttt{point_ID} = \texttt{t} \times \texttt{max_points_per_timepoint} + \texttt{n}$. This matrix allows efficient retrieval of point coordinates for a given time frame.

The `points_to_tracks` matrix (Fig. S3c) maps point IDs to their associated `track_id`. Its shape is (`n_points`, `n_tracks`). Each row represents a point, while columns denote tracks. A nonzero element (value = 1) in the matrix indicates that a specific point belongs to the corresponding track. Although values are not meaningful in this context, the column indices encode the `track_id`. This matrix serves as an adjacency matrix to associate points with tracks.

The `tracks_to_tracks` matrix (Fig. S3c) describes relationships between tracks and has dimensions (`n_tracks`, `n_tracks`). This matrix allows the determination of ancestors and descendants of a track. Rows represent tracks, while columns indicate the related tracks belonging to the same lineage. All tracks present in a row will be loaded by the application when a respective track is selected. As related tracks, we consider all the ancestors and descendants of the respective track. A value of -1 signifies a track without an ancestor, which happens when a cell was present in the first timepoint, or emerged and did not have a parent (no `parent_track_id`). A nonzero element in the matrix indicates a direct parent-child relationship, and the value itself is the `parent_track_id`. The values of the nonzero elements in the matrix are not essential to the application, since the column index already signifies the related `track_id`, the value is only used to efficiently export the tracking data of the selected cells.

The `tracks_to_points` matrix (Fig. S3c) maps tracks to their constituent points. It has dimensions (`n_tracks`, `n_points`) and is essentially the transpose of the `points_to_tracks` matrix. Additionally, this array includes the point coordinates, which creates a redundancy, but allows for optimized queries to fetch point coordinates.

Zarr format

The four matrices are stored as Zarr arrays in a single Zarr store (Fig. S3d). The `points` array is chunked by time, ensuring that only the data for a single time point is loaded during visualization. The remaining arrays (`points_to_tracks`, `tracks_to_tracks`, and `tracks_to_points`) are stored as compressed sparse row (CSR) matrices using `scipy.sparse`. The CSR format was specifically chosen for efficient row-retrieval, so each matrix is intended to be indexed by row. The CSR format encodes matrices efficiently with three arrays: `indices`, which contains the column indices of nonzero elements; `indptr`, which contains pointers to the start of rows; and `data`, which contains the values of nonzero elements. Our CSR arrays are essentially Zarr groups with minimal added metadata and 2-3 dense Zarr arrays. For `points_to_tracks`, only `indices` and `indptr` are stored, as the matrix values are irrelevant.

The final outline of the Zarr store (for the example zebrafish dataset) will look like this:

```
tracks_bundle.zarr (~550M)
|-- points (198M)
|-- points_to_tracks (62M)
|   |-- indices (61M)
|   |-- indptr (1M)
|-- tracks_to_points (259M)
|   |-- data (207M)
|   |-- indices (50M)
|   |-- indptr (2M)
```

```
|-- tracks_to_tracks (28M)
    |-- data (19M)
    |-- indices (7M)
    |-- indptr (2M)
```

Note the relatively small size of the `indptr` arrays provides an opportunity for efficient caching and reduces the total number of row-wise requests when fetching tracks.

This structure includes minimal redundancy to optimize queries and reduce latency during data fetching. Despite the redundancy, the Zarr store has a size of only 46% of the original `CSV` file, since it is binary encoded.

Fetching of tracks

To display tracks in the inTRACKtive viewer, the following steps retrieve the required coordinates (Fig. S3e). The first query to `points_to_tracks` identifies the `track_id` of the selected point. The second query to `tracks_to_tracks` finds all related tracks, including ancestors and descendants. The third query to `tracks_to_points` retrieves the coordinates of all points in the identified tracks. These queries run asynchronously using `async/await` in TypeScript, ensuring efficient data retrieval.

Adding cell size and color attributes

To enhance visualization, additional cell attributes can be incorporated. Firstly, each cell can be given a different size. A `radius` column in the input `tracks.csv` file and the `--add_radius` flag in the conversion script will add cell radii to the `points` array as a fourth coordinate (`z`, `y`, `x`, `radius`), which is automatically read by inTRACKtive as the cell size in the rendering. Secondly, every cell can optionally be given a color based on provided attributes (see Fig. S2 for an example). Additional extra columns in `tracks.csv` and the `--add_all_attributes` flag during conversion will generate a fifth Zarr array named `attributes`. This array stores custom attributes for each cell as a dense, ragged array with dimensions (`n_timepoints`, `num_attributes` $\times$ `max_points_per_timepoint`), and is chunked by time and attribute.

Supplementary Figures

See next page

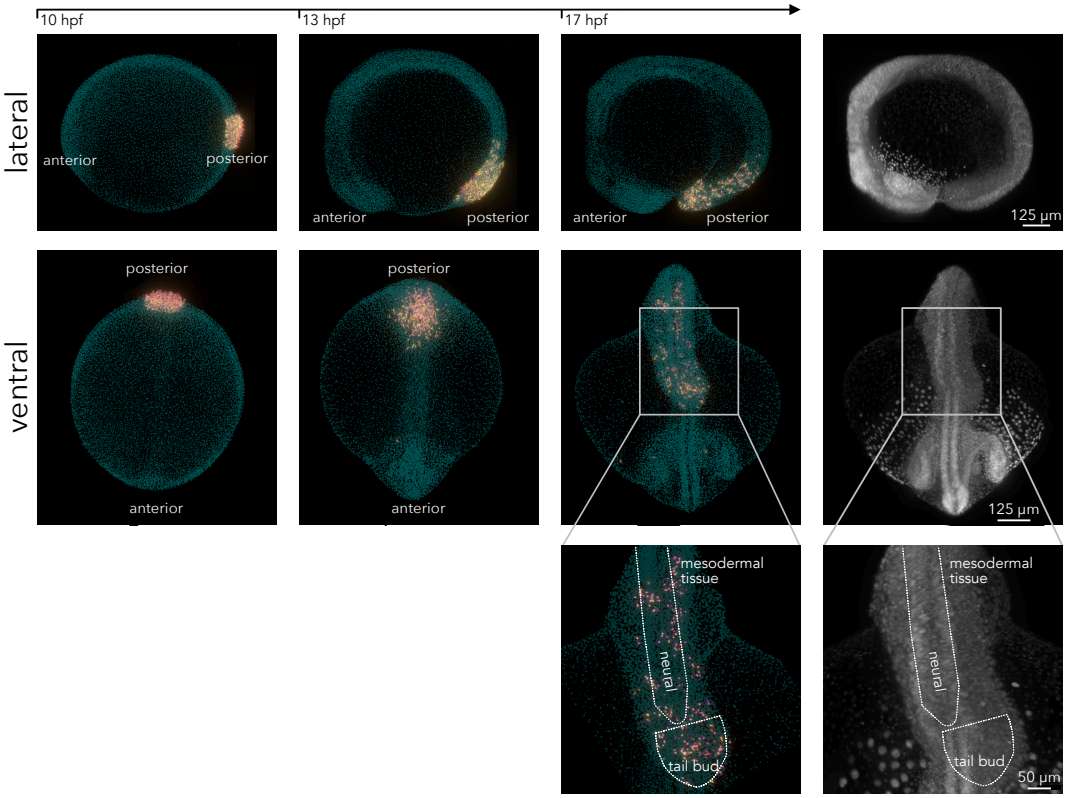

Fig. S1 | In-silico fate mapping of zebrafish tailbud cells using inTRACKtive. The fully-tracked zebrafish embryo visualized with inTRACKtive. The tailbud cells are selected within inTRACKtive at 10 hours post fertilization (hpf) and followed during development. Shown are inTRACKtive snapshots at 13 hpf (second column) and 17 hpf (third column), alongside the imaging data at 17 hpf. The zoomed panel shows that the selected tailbud cells end up in the tail of the zebrafish embryo, in particular, the mesoderm, neural tube, and tailbud itself – agreeing with previous findings.[16]

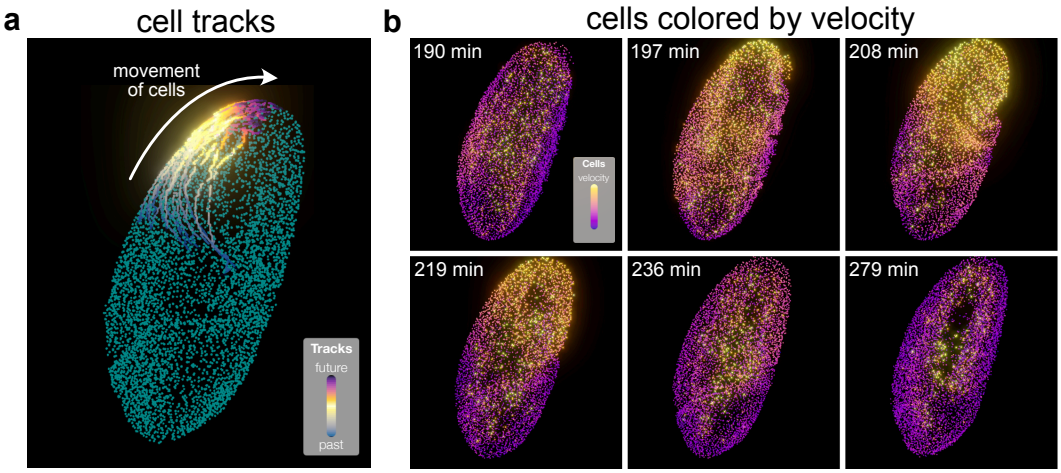

Fig. S2 | inTRACKtive allows coloring of cells based on a given attribute. a Fully tracked developing Drosophila melanogaster[10] embryo, visualized in inTRACKtive. When selecting several cells, the visualized tracks show the global movement of cells during gastrulation. **b** Same embryo visualized in inTRACKtive, but with the cells colored based on their velocity. The color of the cells (yellow means high velocity) clearly displays how the wave of migration propagates through the developing embryo. Timestamps represent minutes post-fertilization.

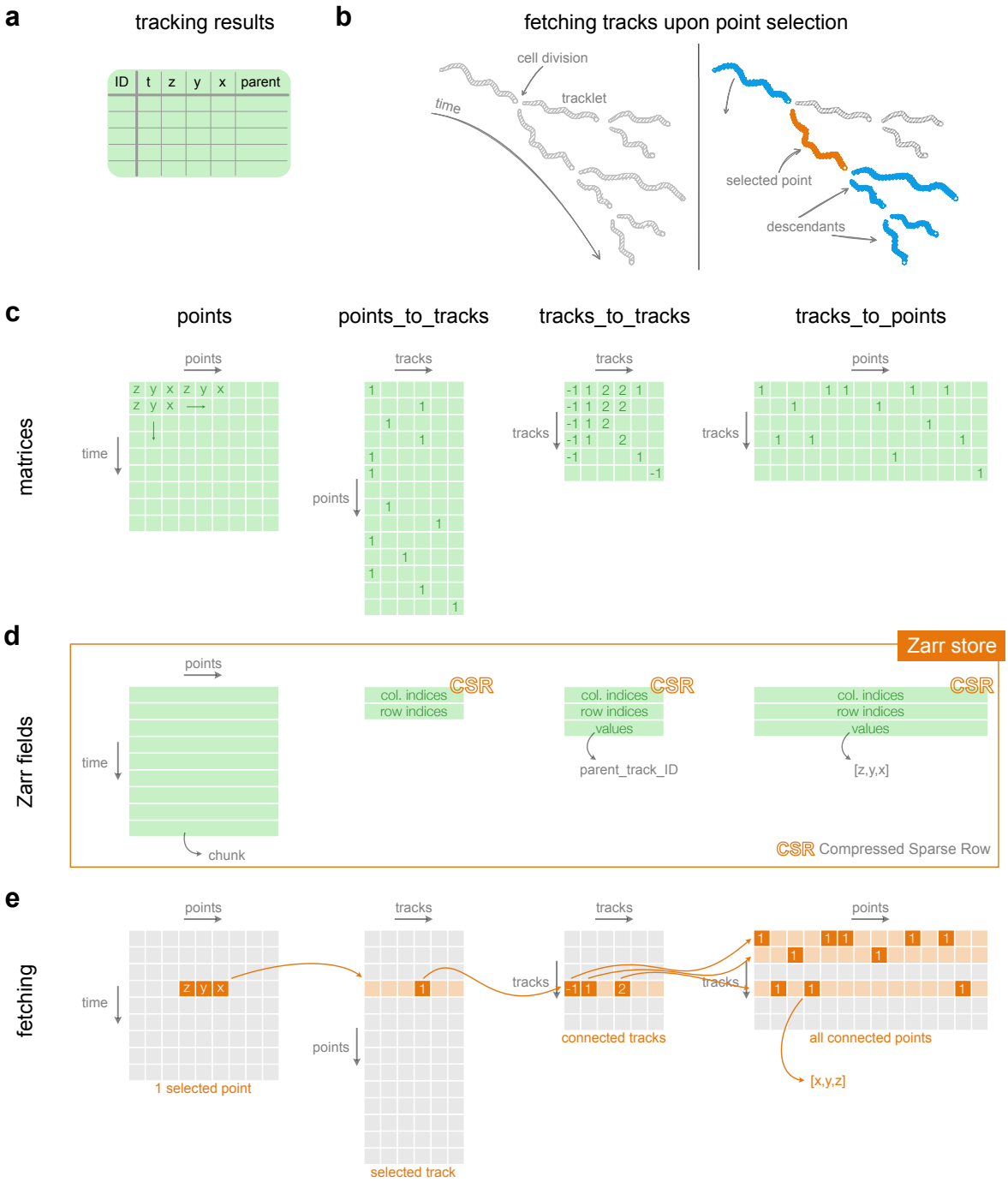

Fig. S3 | **The tracking Zarr data format. a** The start point is a table with cell tracking results. **b** The Zarr format is designed for fast and efficient fetching of tracks (ancestors and descendants) when the user selects a cell (point). **c** From the tracking results table, four matrices are constructed, one with the coordinates of all points, and three for the connections between points and tracks. **d** The four matrices of **c** are saved as Zarr arrays in one Zarr store. The `points` array is saved as a dense Zarr array, chunked per time. The other three are saved in the CSR (compressed sparse row) format, with fields `indices` (column indices), `indptr` (row indices), and `data` (values). **e** When the user selects a point, a series of three queries sequentially fetches the points of all the related tracks.