Genome Biology

METHOD

Open Access

Mapping lineage-traced cells across time points with moslin

Marius Lange[1,2,3†], Zoe Piran[4†], Michal Klein[5†], Bastiaan Spanjaard[6,7†], Dominik Klein[2,3], Jan Philipp Junker[6,8,9], Fabian J. Theis[2,3,10*] and Mor Nitzan[4,11,12*]

†Marius Lange, Zoe Piran, Michal Klein, and Bastiaan Spanjaard contributed equally to this work.

†Michal Klein's work was partially done while at Helmholtz Center Munich, Germany.

*Correspondence: fabian.theis@helmholtz-muenchen.de; mor.nitzan@mail.huji.ac.il

[2] Department of Mathematics, Technical University of Munich, Munich, Germany
[4] School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel
Full list of author information is available at the end of the article

Abstract

Simultaneous profiling of single-cell gene expression and lineage history holds enormous potential for studying cellular decision-making. Recent computational approaches combine both modalities into cellular trajectories; however, they cannot make use of all available lineage information in destructive time-series experiments. Here, we present moslin, a Gromov-Wasserstein-based model to couple cellular profiles across time points based on lineage and gene expression information. We validate our approach in simulations and demonstrate on *Caenorhabditis elegans* embryonic development how moslin predicts fate probabilities and putative decision driver genes. Finally, we use moslin to delineate lineage relationships among transiently activated fibroblast states during zebrafish heart regeneration.

Keywords: Fate decisions, Lineage tracing, Cellular dynamics, Optimal transport, Regeneration

Background

Many central biological processes like development, disease, or regeneration play out as complex changes on multiple levels of biological hierarchy, including the cellular level. Due to their transforming nature, these changes are best captured by time-resolved measurements. Single-cell assays, including single-cell RNA-sequencing (scRNA-seq), probe cellular heterogeneity at unprecedented resolution and scale at different time points but destroy cells in the process. Thus, previous work introduced computational approaches that link cells across time based on similar gene expression profiles [1–3]. While these approaches successfully uncovered trajectories and fate decisions for in vitro systems [1, 4] and some in vivo systems [5, 6], they require dense temporal sampling and remain limited to simpler processes where expression similarity faithfully represents lineage relationships [7].

To improve the accuracy of trajectory inference, scRNA-seq has recently been combined with heritable barcodes that record clonal relationships over long time scales in single-cell lineage tracing (scLT) assays [8–13]. For in vitro systems, we can sample

from the same cell population several times and use matching barcodes to relate cells across time points [14]. In this clonal resampling setting, an earlier method, CoSpar [15] (coherent sparse optimization), learns an early-to-late transition matrix by iteratively filtering low-probability transitions, enforcing transitions within clones, and promoting overall coherence by smoothing transitions over cells with similar gene expression profiles. However, such strategies do not naturally generalize to in vivo lineage-traced systems, as each time point corresponds to a different individual, and barcodes are not comparable across individuals. Current analysis strategies [13, 16–20] mostly focus on analyzing isolated lineage-traced time points. Thus, most methods cannot use the available lineage-tracing information to connect ancestors to putative descendants at later time points.

A notable exception, LineageOT [21], has been introduced to link cells across time points for the challenging in vivo lineage tracing setting. However, LineageOT ignores lineage information from the earlier time point and constructs the final mapping based on expression similarity between early and lineage-smoothed late-stage cells. Similarly, CoSpar includes a variant which makes use of gene expression at both time points, but lineage information only at a single time point. Thus, the comprehensive integration of lineage and gene expression information from all available time points to estimate cellular state-change trajectories remains an open computational problem.

Here, we present multi-omic single-cell optimal transport for lineage data (moslin), a computational method to embed in vivo clonal dynamics in their temporal context. Moslin uses expression similarity and lineage concordance to reconstruct cellular state-change trajectories for complex biological processes. Moslin uses lineage information from all available time points and includes the effects of cellular growth and stochastic cell sampling. Our algorithm is based on a variant of optimal transport (OT) [22], which allows us to compare cell pairs (as opposed to individual cells) across time points for their lineage history, thus overcoming the limitation of incompatible lineage information.

Our approach outperforms LineageOT, CoSpar, and OT-baselines on simulated data where ground truth is available. Further, on a scRNA-seq dataset of *Caenorhabditis elegans* embryogenesis, containing gene expression profiles across seven time points with known lineage relationships, we combine moslin with CellRank 2 [23], a trajectory inference framework, to uncover differentiation trajectories and putative decision-driver genes. Finally, in zebrafish heart regeneration, barcoded in single cells using evolving CRISPR/Cas9 recording [10] across four time points, we use moslin to predict lineage relationships between recently discovered activated fibroblast states that emerge after injury. We implemented moslin as a user-friendly Python package with documentation and tutorials, available at github.com/theislab/moslin.

Results

Moslin combines lineage and state information to link cells across time

Moslin is an algorithm to reconstruct molecular trajectories of complex cellular state changes from time-series single-cell lineage tracing [8, 24, 25] (scLT) studies. Using gene expression and lineage information from all available time points, moslin computes probabilistic mappings between cells in early and late time points. Using the computed

mappings, we infer ancestor and descendant probabilities for rare or transient cell states and interface with CellRank 2 [23] to visualize gene expression trends, uncover activation cascades, and pinpoint potential regulators of key decision events (Fig. 1a).

We designed moslin for time-series scLT studies ("Methods"). These record evolving clonal relationships using a variety of approaches, including Cas9/12-induced insertions and deletions (indels) [10–13, 26–29] and naturally occurring mutations [30, 31]. Importantly, such evolving lineage tracing systems record hierarchical clonal structure,

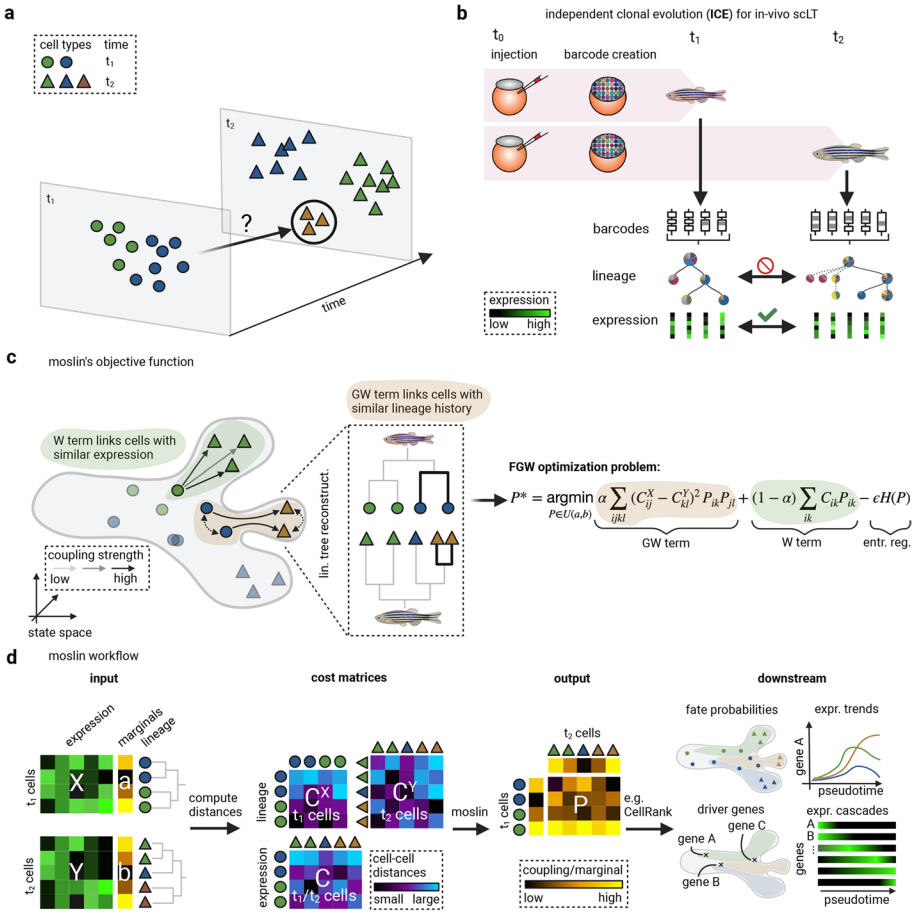

Fig. 1 Moslin maps lineage-traced single cells across time points. **a** Schematic of scRNA-seq time-course experiment with time points t_1 (circles) and t_2 (triangles). Cells are destroyed upon sequencing; this makes it difficult to study the trajectories of early cells giving rise to late cells. We highlight a rare population (brown triangles) that only appears at t_2, with uncertain origin at t_1. **b** Illustration of independent clonal evolution (ICE) experimental design for scLT studies, adjusted from ref [32]. ICE samples cells from different individuals at different time points and is applicable to in vivo settings. **c** Overview of moslin's optimal-transport (OT)-based objective function for in vivo scLT. The gray outline shows a simplified state manifold; shapes and colors as in **a**. The dashed inset highlights lineage trees reconstructed independently for each time point [16]; these trees may be used in moslin to quantify lineage similarity. We use Wasserstein (W) and Gromov-Wasserstein (GW) terms to compare cells in terms of gene expression and lineage similarity, respectively. The combination of W and GW terms gives rise to moslin's Fused Gromov-Wasserstein (FGW) objective function on the right ("Methods"). **d** The moslin workflow; based on gene expression matrices X and Y, marginals a and b, and lineage information across time points, we compute distance matrices C^X, C^Y, and C, and use moslin to reconstruct a coupling matrix P, probabilistically matching early to late cells. The marginals may be used to quantify measurement uncertainty or cellular growth and death. The coupling matrix P may be analyzed directly or passed to CellRank 2 [23] to compute fate probabilities, driver genes and expression trends or cascades. Figure created in BioRender.com

which moslin uses to estimate fine-grained lineage distances ("Methods"). We refer to the entirety of any such genomic lineage information in a single cell as a "barcode" and demonstrate moslin here over both simulated data and scLT experiments using Cas9-induced indels.

Applying scLT to in vivo systems usually requires that each time point corresponds to a different individual. We relate to this experimental design as "independent clonal evolution" (ICE), as barcode generation proceeds independently in each individual. In contrast to the clonal resampling setting introduced above, barcodes in ICE can be directly compared within one individual to estimate lineage trees [10–13, 16, 18, 19], but they are incompatible across different individuals and hence time points. However, gene expression continues to be comparable across time points, giving rise to a hybrid setting where we may relate lineage and gene expression within and across time points, respectively (Fig. 1b and "Methods").

To link cells from an early (t_1) to a late (t_2) time point, we make two assumptions. First, we assume that cells change their molecular state gradually, a common assumption that forms the basis of many established pseudotime algorithms [1, 33–37]. Second, if lineage relationships shape molecular states [10, 13], and molecular states gradually evolve across time points (first assumption), then we may assume that there exists (potentially imperfect) "lineage concordance" across time points, even if these time points correspond to different individuals. Thus, according to the second assumption, cell pairs at t_1 should be mapped to cell pairs at t_2 with similar relative lineage distances. By lineage distance, we mean the degree to which two cells have diverged on the lineage tree. We designed moslin using the flexible framework of optimal transport [38, 39] (OT), which allows us to combine both assumptions into a single cost function (Fig. 1c, "Methods," and Additional file 2: Note S1).

We include the first assumption in moslin using a Wasserstein (W) term, which encourages links between cells with similar gene expression. Briefly, the W term sums over all combinations of early and late cells, aiming to find a probabilistic mapping that minimizes the overall cost of transporting cells [1] ("Methods"). We include the second assumption in moslin using a Gromov-Wasserstein [22] (GW) term ("Methods" and Additional file 2: Note S1). Briefly, the GW term sums over all pairwise combinations of early and late cells, aiming to find a probabilistic mapping that minimizes the discrepancy between pairwise lineage distances ("Methods"). Cells are allowed to violate either assumption at the cost of incurring a penalty in the objective function.

We balance both terms with an α parameter between 0 and 1, corresponding to W and GW terms, respectively [40]. This parameter allows us to tune the weight given to gene expression and lineage information. Further, we add entropic regularization at weight ϵ to our objective function to speed up the optimization and to improve the statistical properties of the solution [39, 41, 42]. Thus, moslin solves a Fused Gromov-Wasserstein [40] (FGW) problem with hyperparameters α and ϵ, jointly optimizing lineage concordance and gene expression similarity (Fig. 1c, "Methods," and Additional file 2: Note S1).

The FGW problem is non-convex and our optimization routine will in general converge to a local minima of the optimization landscape. Numerous previous publications demonstrated that local minima of the FGW objective function give good performance in practice, for example, when mapping dissociated single cells to physical space [43],

aligning tissue slides or computing consensus slides [44, 45], integrating data across modalities [46], inferring cell–cell communication [47], or mapping spatial representations across time [48].

Inputs to the moslin workflow are gene expression matrices X at t_1 and Y at t_2, as well as lineage information (Fig. 1d and "Methods"). In the first step, we compute cost matrices C and C^X, C^Y, representing expression and lineage distances, respectively. We quantify expression distance across time points using squared Euclidean distance in a latent space [1], computed using PCA or scVI [49]. To quantify lineage distance within each time point, we either work with Hamming distance among raw barcodes or with the shortest path distance among reconstructed lineage trees [10–13, 16, 18, 19] ("Methods"). The choice of lineage distance metric depends on the structure of the lineage information, the expressibility of the barcodes, and the quality of tree reconstruction ("Methods"). In a second step, moslin solves the FGW problem to find an optimal coupling matrix P, relating cells at t_1 and t_2. The coupling simultaneously minimizes expression distances according to C and maximizes lineage concordance between C^X and C^Y, using the W and GW terms, respectively. For each cell i at t_1, the vector $P_{i,:}$ quantifies lineage and state-informed transition probabilities towards any cell j at t_2. Finally, we use the coupling matrix P to compute ancestor and descendant probabilities [1] directly in moslin and pass it to CellRank 2 for further analysis.

Following previous approaches that link cells across time points using OT [1, 21] or related approaches [2], we optionally include prior information about cellular growth and death into our objective function. We accomplish this by adjusting the marginal distributions passed to moslin, such that cells likely to proliferate or die can distribute more or less probability mass, respectively (Fig. 1d). We calculate growth and death rates based on prior knowledge or curated marker gene sets [1]. Our implementation additionally includes an unbalanced formulation [39, 50, 51] governed by a hyperparameter τ_a, which accounts for uncertain growth and death rates as well as for varying cell type proportions across time points when sampling cells from different individuals (Additional file 2: Note S1 and "Methods").

Moslin accurately reconstructs simulated trajectories

We start by assessing moslin's performance in two simulated settings, with a total of five different datasets. As an initial verification, we consider simulated single-cell transcriptome trajectories using a previously suggested setting [21]. In this simplified setting, all meaningful dynamics occur in two dimensions, representing two genes. A biologically plausible trajectory structure is prescribed via a vector field that cells follow through diffusion and occasional cell division. A lineage barcode, including random mutations, is assigned to each cell and inherited by its descendants.

We consider four different trajectories of increasing complexity: (i) *bifurcation* (B), where a single progenitor cell type splits into two descendant cell types, (ii) *partial convergent* (PC), where two initial clusters split independently, and following the split, two of the resulting four clusters merge for a total of three clusters, (iii) *convergent* (C), where two initial clusters converge to a single final cell type, and (iv) *mismatched clusters* (MC), where two initial clusters each split into two late-time clusters and cells from two of the

resulting late-time clusters are transcriptionally closer to early cells that are not their ancestors (Fig. 2a).

We benchmark the performance of moslin against LineageOT [21] and CoSpar [15] ("Methods"). We also test two extreme cases of moslin: (i) using only gene expression information in a W term (W; $\alpha = 0$) and (ii) using only lineage information in a GW term (GW; $\alpha = 1$). We test all methods with two types of lineage-distance computation:

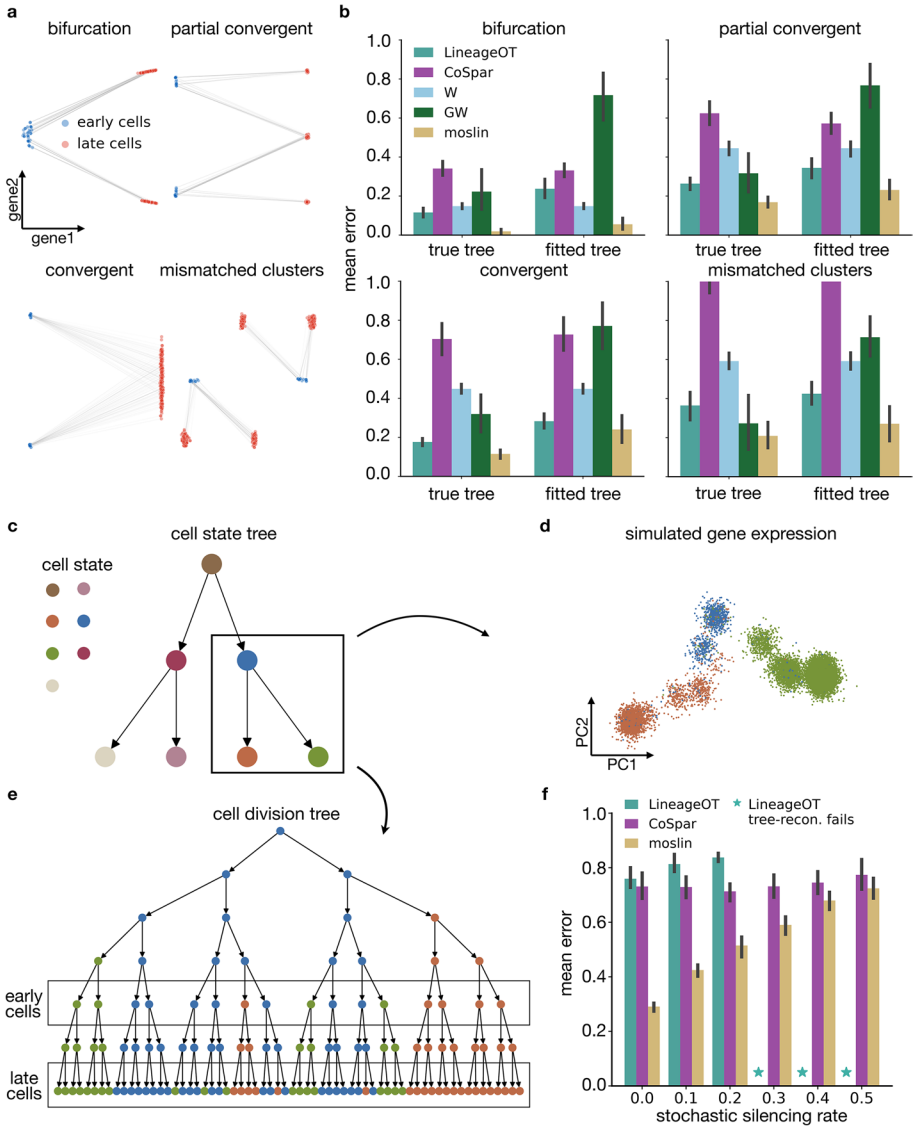

Fig. 2 Moslin obtains accurate couplings for simple and complex trajectory topologies. **a** Visualization of the four different kinds of simulated trajectories in gene expression space for the 2D setting. **b** Each subplot presents the evaluation of a different simulated trajectory. Per trajectory, the mean error (the mean value of the ancestors and descendants error) is evaluated for the true tree or a reconstructed fitted tree for all methods, LineageOT, CoSpar, W, GW, and moslin ("Methods"). Error bars depict the 95% confidence interval across 10 random simulations. **c**–**e** Simulated tree and expression using TedSim [52]. The cell state tree (**c**) defines the underlying trajectories of cell differentiation. TedSim simulations yield gene expression (**d**) and a cell division tree (**e**), which represents the true lineage and barcode for each cell. **f** Mean prediction error of moslin compared to CoSpar and LineageOT, as a function of the *stochastic silencing rate*. Error bars depict the 95% confidence interval across 10 random simulations

(i) using the ground truth tree and (ii) using a fitted tree based on the simulated barcodes (Additional file 1: Fig. S1a, b, " Methods"). We perform a grid search for each case over the relevant hyperparameters (Additional file 1: Fig. S1c, d, "Methods"). To quantify method accuracy, we compare gene expression of predicted and ground-truth ancestors and descendants in terms of Wasserstein distance [21] ("Methods"). We normalize this value by the Wasserstein distance we obtain from an uninformative coupling, given by the marginal-outer product, to obtain ancestor and descendant errors. Each value lies between 0 (ground truth) and 1 (uninformative). Finally, to obtain a single number quantifying method accuracy, we average over ancestor and descendant errors to obtain the "mean error."

Across all tested settings, moslin achieves the lowest mean error across all trajectory structures and distance variants (Fig. 2b). On average across all trajectories, moslin attains an improvement of 10% (true trees) and 12% (fitted trees) over LineageOT and 56% (true trees) and 52% (fitted trees) over CoSpar ("Methods"). Of note, GW performs well using ground-truth tree distances, outperforming W in three of four cases and demonstrating the value of ground-truth lineage information. However, as expected, pure GW is heavily affected by noise in tree distances and shows the largest mean error across all trajectories on more realistic, fitted tree distances.

These results demonstrate the power of the moslin approach: while pure GW is heavily affected by noisy lineage information, moslin compensates for this noise using gene expression information. Thus, the manner of interpolation between gene expression and lineage information in moslin allows it to perform well on realistically fitted tree distances (Fig. 2b).

Next, we consider a more complex simulation using TedSim [52], which simulates cell division events from root to terminal cells. It generates two data modalities for each cell, gene expression and a lineage barcode, defining a much more complex setting than the two-dimensional regime considered above. The cell lineage tree is simulated as a binary tree that encodes cell division events, where a predefined cell state tree dictates the allowed transitions towards terminal cell states. We cut the lineage tree at an intermediate depth to simulate an early time point and use leaf nodes for the late time point (Fig. 2c–e and "Methods"). We map cells from the early to the late time point, providing only lineage relationships within time points to moslin and using the lineage relationships across time points to score the quality of our reconstructed mapping.

scLT datasets often suffer from barcode detection issues, and it is, therefore, crucial to assess the performance of computational pipelines on partially detected barcodes. In our simulations, we introduce a stochastic silencing rate (ssr), the rate at which individual elements of the barcode remain undetected ("Methods"). In this example, we test an alternative to lineage tree reconstruction and directly use the scaled Hamming distance between barcodes to measure lineage distances in moslin and to construct clonal relations for CoSpar (Additional file 1: Fig. S2a, b and "Methods"). We perform a grid search for each case over the relevant hyperparameters (Additional file 1: Fig. S2c, "Methods"). As expected, as we increase the ssr, the lineage information becomes less reliable hence the optimal GW scaling parameter, α, decreases.

For lower ssr values, performance is substantially improved by moslin compared to LineageOT and CoSpar. Namely, at ssr values 0.0, 0.1, and 0.2, moslin improves

LineageOT's mean error by 0.46, 0.39, and 0.32 and CoSpar's mean error by 0.44, 0.30, and 0.20, respectively (Fig. 2f). At larger ssr's the improvement over CoSpar reduces, though still visible, while LineageOT struggles with tree reconstruction in that regime ("Methods").

At last, using the ground truth coupling, we validate that moslin faithfully captures transitions to emergent states that appear only at the later time point and that it is robust to subsampling of cells at the later time point (Additional file 1: Fig. S2e–f, "Methods").

Mapping gene expression across *C. elegans* embryonic development

To showcase moslin's performance in a realistic setting where ground truth is available, we consider *C. elegans* embryonic development. The adult animal consists of only 959 somatic cells [53, 54], generated following a sequence of deterministic lineage decisions. This species' ground truth lineage tree is known [54] and available to assess moslin's reconstruction performance. Further, this well-studied system is a good test case to validate biological insights gained by combining moslin with CellRank 2 for fate mapping, gene dynamics, and driver gene prediction.

Previous work mapped time-series gene expression profiles of approx. 86k single cells to individual tree-nodes [7], providing a setting where joint lineage, state, and time information is available. Not all cells in this study could be mapped unambiguously. Thus, we focus on the well-annotated ABpxp lineage, which produces mostly ciliated and non-ciliated neurons, glia, and excretory cells [55]. AB is one of the founding lineages of *C. elegans*; "p" ("a") indicates the posterior (anterior) ancestor, and "x" replaces "l" (left) or "r" (right), indicating a left/right symmetry [7, 55] (Additional file 1: Fig. S3a). The dataset consists of 6476 ABpxp cells across 7 time points from 170 to 510 min past fertilization (Fig. 3a, Additional file 1: Fig. S3b, c, and "Methods"). We treat the original author's mapping [7] of cells to the *C. elegans* lineage tree as ground truth and use it to evaluate our algorithm.

We benchmark the performance of moslin, its two extreme variants, W ($\alpha = 0$), GW ($\alpha = 1$), and LineageOT across time points on the ABpxp lineage using a similar setup as for the TedSim [52] data, and as suggested in ref [21]. (Fig. 2c–f and Additional file 1: Fig. S4a). Specifically, we provide within time point lineage distance to those methods that account for them (moslin, GW, and LineageOT). We evaluate performance by comparing predictions with ground-truth lineage relations across time points, using the mean prediction error over ancestor and descendant states ("Methods").

For all time point pairs, moslin outperforms other methods and achieves a lower mean error (Fig. 3b). As expected from the high resolution lineage information in this dataset, we find that moslin performs best for large values of α, reflecting strong influence of the lineage term (Additional file 1: Fig. S4a). Considering the same setup, we test method performance on another, distinct subset of *C. elegans* cells, for which complete lineage information is available (Additional file 1: Fig. S3a). Note that cells from the ABpxp sublineage do not have complete lineage information because "x" can replace either "r" or "l" ("Methods"). Moslin performs second best, only outperformed by the extremal GW, on all time point pairs but 330/390 min (Additional file 1: Fig. S4b and Additional file 1: Fig. S5a). On this pair of time points, moslin outperforms GW by a large margin. We hypothesize that GW might get stuck in a local minima, which moslin can avoid

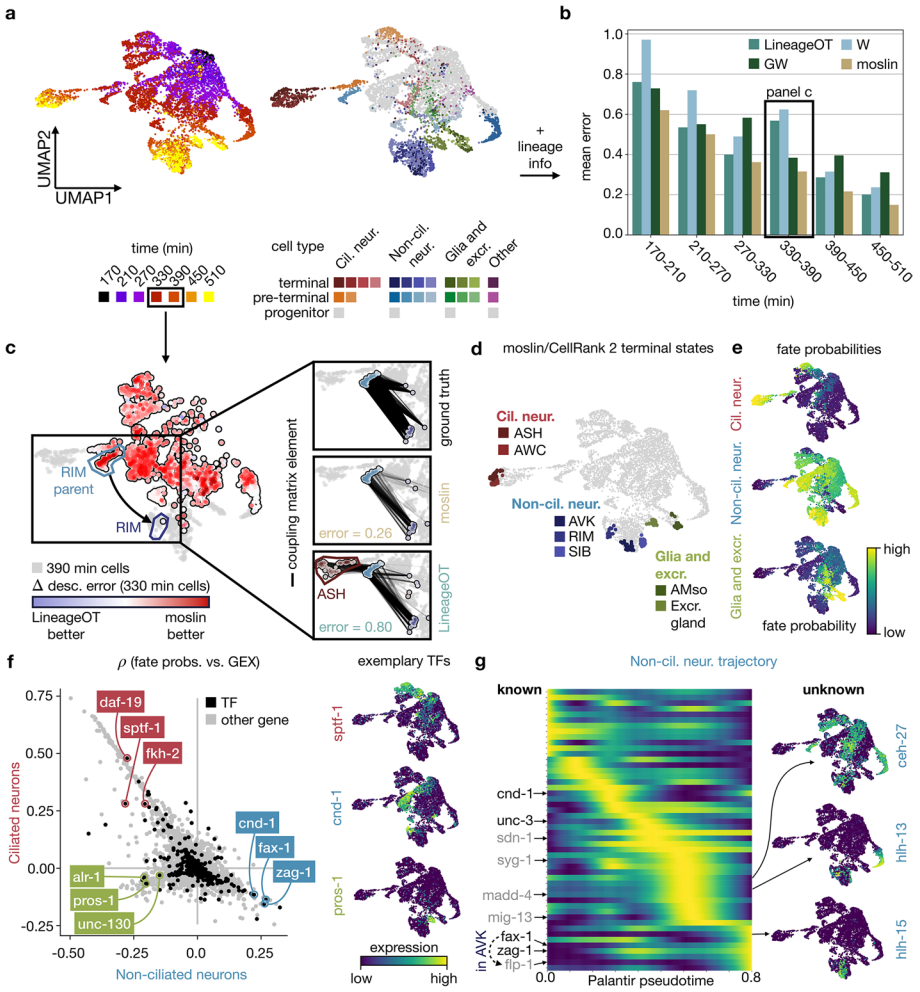

Fig. 3 Moslin accurately captures *C. elegans* embryogenesis. **a** UMAP [56] of approx. 6.5k *C. elegans* ABpxp cells, colored by time point (left) and cell type (right) [7]. **b** Bar chart of the mean error for different methods across time points ("Methods"). **c** Left: UMAP of 330–390 min cells, colored in gray (390 min cells) or by the difference in descendant error between moslin and LineageOT (330 min cells). Black inset highlights RIM parent cells, which transition towards RIM cells [7]. Right: ground-truth, moslin and LineageOT couplings for the RIM parent population; "error" indicates the aggregated descendant error over this population (Additional file 1: Fig. S6 and "Methods") **d** UMAP, showing the top 30 cells per moslin/CellRank 2 (ref [23]) computed terminal state. **e** UMAPs of aggregated fate probabilities towards ciliated neurons, non-ciliated neurons, and glia and excretory cells (Additional file 1: Fig. S11 and "Methods"), computed via absorption probabilities in CellRank 2 ("Methods"). **f** Scatter plot, showing the correlation of gene expression (GEX) with non-ciliated (x-axis) and ciliated (y-axis) neuronal fate probabilities. Annotated TFs are known to be involved in the developmental trajectory they correlate with (Additional file 3: Table S1). Right: UMAPs, showing expression of exemplary TFs. **g** Left: heatmap showing expression values for the top 50 predicted driver genes of non-ciliated neurons (all gene names shown in Additional file 1: Fig. S15). Each row corresponds to a gene, smoothed using fate probabilities (**e**) and the Palantir pseudotime [57] (x-axis; Additional file 1: Fig. S9). We annotate a few TFs, including cnd-1 [58, 59], fax-1 [60], and zag-1 [61–63] (black), and other genes, including syg-1 [64–66], madd-4 [67–69], and flp-1 [60, 70] (gray), which are known to play important roles in establishing non-ciliated neurons (Additional file 3: Table S1). Right: UMAPs, showing expression of previously unknown predicted driver TFs

due to gene expression regularization. Indeed, when we use the W solution to initialize GW, its mean error decreases by a large margin (Additional file 1: Fig. S5d). This example illustrates that even with perfect within-time point lineage information, using

moslin with gene expression regularization is more robust than relying purely on lineage information.

The number of cells per time point and the difference in cell number between adjacent time points are not main factors determining moslin's performance (Additional file 1: Fig. S5b, c). Focusing on moslin's robustness to inaccuracies in lineage-tree distances, we find that the mean error typically increases by less than 0.1 when we permute up to 20% of non-diagonal cost matrix elements (Additional file 1: Fig. S5d and "Methods").

On the ABpxp lineage, the mean error difference between moslin and LineageOT is largest on the 330/390 min pair of time points. To illustrate this point, we zoom in on the difference between moslin and LineageOT per 330 min cell (Fig. 3c and Additional file 1: Fig. S6a, b). As an example, we pick a pre-terminal population of RIM (non-ciliated) neurons for which moslin's descendant error is much smaller compared to LineageOT's. We find that moslin and GW correctly link these cells to RIM neurons, while LineageOT and W predict many erroneous connections with ASH (ciliated) neurons (Fig. 3c, Additional file 1: Fig. S6c, and "Methods"). The case of pre-terminal RIM cells is an example where only methods that consider lineage information at both time points (moslin and GW) are able to predict descendants correctly. Note that moslin achieves a lower descendant prediction error compared to GW, and moslin's predicted RIM-ancestor distribution is more similar to ground-truth (Additional file 1: Fig. S6c, d).

Going beyond a single pair of time points, we combine moslin's couplings across all time points to study *C. elegans* embryogenesis using CellRank 2 [23], a computational fate mapping tool. To account for developmental asynchrony within each time point, CellRank 2 computes, for each time point, a transition matrix reflecting undirected gene expression similarity. These within-time point transition matrices are combined with moslin's across time point coupling matrices to yield the final transition matrix, reflecting cellular dynamics within and across time points ("Methods"). When we use the final transition matrix to simulate 500-step random walks from the 170 min time point, we find that these terminate in the known terminal cell types, recapitulating the established developmental hierarchy (Additional file 1: Fig. S7a, b). This result remains robust when we vary the length of random walks (Additional file 1: Fig. S7c).

Using this transition matrix, we set out to study gene dynamics and fate choice among ABpxp cells. As a first step, we use moslin/CellRank 2 to compute seven terminal states and recover known ciliated-neuronal, non-ciliated-neuronal, glia, and excretory subtypes [7] (Fig. 3d). The terminal states we identify are among the best-resolved cell types for ciliated-neuronal, non-ciliated-neuronal, glia, and excretory groups in terms of cell number (Additional file 1: Fig. S3d). Thus, we capture representative candidates of each group. As expected, predicted terminal states mostly consist of late-stage cells, and each only contains cells from a single cell type (Additional file 1: Fig. S8a). When we vary the number of terminal states from five to ten, these terminal states continue to consist mostly of late-stage cells from a single cell type (except for one terminal state when we compute 10 terminal states).

Next, using the flexibility of CellRank 2, we turn to compare moslin with general trajectory inference methods which rely solely on gene expression, Palantir [57] and CytoTRACE [71, 72] (Supplementary Fig. 7a–c). As expected, pseudotime assignments in both approaches are correlated with experimental time (Supplementary Fig. 7d) [72].

We supply CellRank 2 with W, GW, and LineageOT couplings, as well as with the Palantir [57] and CytoTRACE [71, 72] pseudotimes. For each approach, we compute seven terminal states, as described above, and calculate the mean time point assignment of cells in terminal states and the mean state purity [72] ("Methods"). We expect both the mean time point assignment and the mean state purity to be high, as terminal states should consist mostly of late-stage cells, and terminal states should each represent one defined phenotypic state, respectively. We find that approaches which have access to lineage information (moslin, GW, LineageOT) or time point information (W) perform much better in terms of the time point metric as they are biased to assign late-stage cells to terminal states (Additional file 1: Fig. S10). Out of these approaches, only moslin and LineageOT achieve perfect state purity, with moslin achieving a higher mean real time (475 vs. 466 min; Additional file 1: Fig. S8b and Additional file 1: Fig. S10).

We aggregate the seven terminal states into three groups: ciliated neurons, non-ciliated neurons, and glia and excretory cells and use CellRank 2 to compute fate probabilities towards these groups (Fig. 3e and Additional file 1: Fig. S11). In agreement with known biology, moslin/CellRank 2 predict that most progenitors in the ABpxp lineage transition towards non-ciliated neurons [7] (Additional file 1: Fig. S12a, b). For each of the three terminal cell groups, our approach correctly assigns the largest mean fate probability to the corresponding pre-terminal group of cells (Additional file 1: Fig. S12c, d). We repeat this analysis for competing OT-based (W, GW, and LineageOT) and general trajectory inference (Palantir, CytoTRACE) approaches and find that only moslin can uniquely identify the correct pre-terminal population for all of our three terminal cell groups (Additional file 1: Fig. S12). We also find that moslin/CellRank 2 aggregated fate probabilities are robust with respect to changes in the number of terminal states from five to ten (Additional file 1: Fig. S8).

Using our moslin results, we correlate fate probabilities with gene expression to identify putative driver genes for each of the three trajectories. Focusing our attention on *C. elegans* transcription factors [73] (TFs), we automatically recover known drivers for each trajectory, including sptf-1 for ciliated neurons [74], cnd-1 for non-ciliated neurons [58, 59], and pros-1 for glia and excretory cells [75–77] (Fig. 3f, Additional file 1: Fig. S14, Additional file 3: Table S1, and "Methods").

Finally, to study the temporal dynamics of fate decisions during *C. elegans* embryogenesis, we combine moslin's predictions with the Palantir-derived pseudotime (Additional file 1: Fig. S9a, b). Focusing on the non-ciliated neuron trajectory, we compute the 50 top-correlated genes with non-ciliated fate probabilities. For each of these genes, we combine the Palantir pseudotime with moslin/CellRank 2 fate probabilities to compute smooth expression trends ("Methods" and Additional file 3: Table S1). Sorting expression trends by their pseudotime-peak and plotting them in a heatmap reveals a sequential activation pattern (Fig. 3g and Additional file 1: Fig. S15). Our results show that some TFs with known function in non-ciliated neuron generation, including cnd-1 [58, 59] or unc-3 [78, 79], are activated before others, including fax-1 [60] and zag-1 [61–63] (Additional file 3: Table S1). In particular, our activation pattern predicts that fax-1 is activated before flp-1, a known regulatory interaction in (non-ciliated) AVK cells [60].

While many moslin/CellRank 2 predicted driver genes had known functions in non-ciliated neuron generation, we also identify candidate driver genes that are novel, to the

best of our knowledge. In particular, our results predict ceh-27, hlh-13, and hlh-15 as putative drivers (Fig. 3g). ceh-27 is a homeobox TF, a class of TFs known to be crucial for *C. elegans* neurogenesis [80, 81]. While previous work [80] reported ceh-27 expression in non-ciliated neurons, the TF has no known function in fate specification towards these neurons. hlh-13 and hlh-15 are basic helix-loop-helix TFs; hlh-15 is known to be involved in *C. elegans* aging [82].

In summary, we find that moslin accurately recovers ancestor and descendant relationships and may be combined with CellRank 2 to obtain biologically meaningful terminal states, fate probabilities, and driver genes. While competing approaches perform well in individual comparisons, moslin is the only approach to consistently perform well across all of them.

Moslin determines the dynamics of transient fibroblasts in heart regeneration

The zebrafish heart regenerates after injuries, such as ventricular resections [83] or cryoinjuries [84–86]. A previous study used the integrated lineage-tracing and transcriptome profiling technique LINNAEUS [10] to generate a dataset of approximately 200,000 single cells in the zebrafish heart across four time points: before injury (control), 3 days after injury (3 dpi), 7 days after injury (7 dpi), and 30 days after injury (30 dpi). This dataset includes inferred lineage trees and cell type annotations for each time point [32] (Fig. 4a, Supplementary Fig. 15).

One key result from this study was the emergence of several transcriptionally distinct fibroblast substates during regeneration. Analysis of lineage trees created with LINNAEUS showed that some transient states originate from the endocardial layer and others from the epicardial layer. The persistent constitutive fibroblasts share a lineage with the epicardial layer as well. One state from the epicardial layer, a fibroblast subtype characterized by a high col12a1a-expression, called col12a1a fibroblasts, was shown to be essential for regeneration: ablation of col12a1a fibroblasts strongly reduces the regenerative capacity of the zebrafish heart. Another epicardial-based transient state, the col11a1a fibroblast state, characterized by high col11a1a expression, only occurs at 3 dpi, and its role is unclear. This state could lead to col12a1a fibroblasts, or it could be independent. Since the original analysis was restricted to individual time points, this question could previously not be resolved, which precluded further analysis of the underlying regulatory interactions. We reasoned that we could characterize this relationship by combining time points using moslin.

We apply moslin on all single cells in this dataset with lineage information—approximately 44,000 single cells from 20 individual animals across ctrl, 3 dpi, and 7 dpi. We embed the transcriptomic readout of all single cells with lineage information into a joint latent space using scVI [49], retaining the original cluster annotations. We calculate lineage distances as shortest path distances along the original reconstructed trees and use moslin to calculate couplings between cells at consecutive time points for a grid of hyperparameters values for hyperparameters α, ϵ, and τ_a. For method comparison, we also calculate those couplings using W and LineageOT ("Methods").

Initially, we validate the performance of moslin and other methods in this challenging regeneration setting. We design a test around the assumption that most persistent cell states should be their own precursor; for example, precursors of atrial endocardial cells

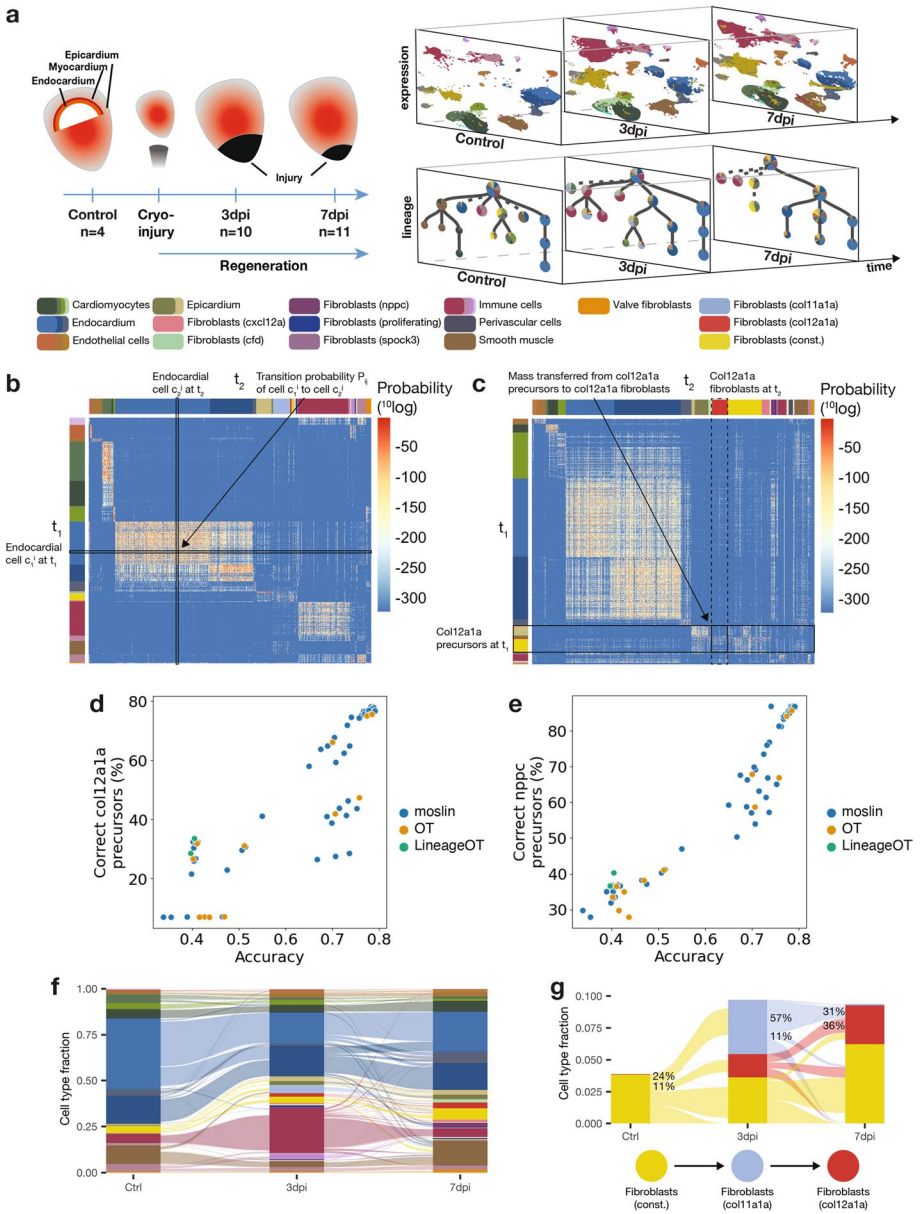

Fig. 4 Moslin recovers lineage relations among transient fibroblast subsets. **a** Underlying data describes zebrafish heart regeneration, measured through single-cell transcriptomic and lineage profiling before injury ($n=4$), at 3 dpi ($n=9$), and 7 dpi ($n=7$) [32]. Right-hand side projections show transcriptomic data over time (top) and a representative lineage tree for each time point (bottom). **b** Cell type persistence test: for each cell at t_2, determine if the t_1 cell with the highest transition probability to it is of the same cell type ("Methods"). Annotation colors indicate cell types as in **a**. **c** Transient fibroblast test: calculate proportion of ground truth ancestor cell types for transient col12a1a and nppc fibroblasts. Annotation colors indicate cell types as in **a**. Performance on transient fibroblast tests correlates with cell type persistence accuracy: col12a1a (**d**) and nppc (**e**). **f** Flow diagram of cell type transitions. Colors indicate cell types as in **a**. **g** Flow diagram of transient epicardial fibroblasts corroborates col11a1a fibroblasts as an intermediary state between constitutive and col12a1a fibroblasts. Colors indicate cell types as in **a**

at 7 dpi should be atrial endocardial cells at 3 dpi. Thus, while ground-truth lineage relationships across time points are unknown in this setting, we initially restrict our attention to a subset of persistent cell states, which we assume to evolve into themselves. This

methodology is applicable in human and model organisms such as mouse and zebrafish, where lineage relationships are not deterministic as they are in *C. elegans*. We used this test to select optimal hyperparameter values. Briefly, for each cell at t_2 and for each method, we select the most probable t_1 ancestor and calculate the accuracy between the t_2 and t_1 cell types, jointly for cells at 3 and 7 dpi (Fig. 4b and "Methods"). For moslin, the maximal cell type persistence accuracy is 0.79, for W, 0.79, and for LineageOT, 0.41 (Additional file 1: Fig. S17). The low performance of LineageOT may reflect an inability to correct for strong cell type frequency imbalances between time points. For moslin, we observed robustness in accuracy values across the bulk of hyperparameters values (Additional file 1: Fig. S18).

To test moslin's ability to predict complex temporal relationships between cell types outside its training regime of persistent cell types, we evaluated its performance in a setting where cell types are not identical between time points. This contrasts to the cell type persistence test used for hyperparameter tuning. Earlier research has shown that the transient col12a1a fibroblasts are of epicardial origin, and the transient nppc fibroblasts are of endocardial origin [32]. We calculated the percentage of mass transferred to transient fibroblasts from their ground truth origins (Fig. 4c and "Methods"). For the optimal hyperparameter values determined by the cell type persistence test, we found that moslin predicted that 77% of col12a1a fibroblasts are of epicardial origin, and 87% of nppc fibroblasts originate from endocardial cell types. On nppc fibroblasts, moslin outperformed W by a small but significant margin of 1.2% ($p = 0.00027$) (Additional file 1: Fig. S19 and "Methods"). Over the range of hyperparameters tested, we observed a strong correlation between method performance on transient fibroblasts and on cell type persistence (Fig. 4d, e). This suggests that the cell type persistence test, which relies on minimal prior knowledge of the biological system, can be used to find optimal hyperparameter values for moslin.

We next investigate the origins of transient fibroblast substates, including col11a1a and col12a1a fibroblasts. In particular, the previously published analysis [32] had left room for two hypotheses: either col11a1a fibroblasts are an intermediary state between constitutive and col12a1a fibroblasts, or these two fibroblast states arise from constitutive fibroblasts independently. To approach this, we calculated couplings with moslin, took weighted averages of cell type frequencies over separate organisms, and aggregated couplings between cell types to quantify cell type transitions during regeneration (Additional file 1: Fig. S20 and "Methods"). As expected from the cell type persistence test, we observe that cell types have strong aggregated couplings with themselves between time points (Fig. 4f).

Furthermore, we observe that constitutive fibroblasts preferentially generate col11a1a fibroblasts, and that most col12a1a fibroblasts originate from col11a1a fibroblasts: 24% (95% confidence interval: 19–32%) of the mass generated by constitutive fibroblasts at control goes towards col11a1a fibroblasts, whereas only 11% (95% confidence interval: 6–17%) goes directly towards col12a1a fibroblasts. At 3 dpi, 57% (95% confidence interval: 30–69%) of the mass generated by col11a1a fibroblasts goes towards col12a1a fibroblasts, which constitutes 31% (95% confidence interval: 18–41%) of the mass col12a1a fibroblast receive at 7 dpi (Fig. 4g). Confidence intervals for the frequencies and couplings were constructed by subsampling ("Methods").

Taken together, this suggests that the majority of col12a1a fibroblasts is generated by constitutive fibroblasts that transition through a col11a1a-expressing state (Fig. 4g). We hypothesize that the 3 dpi col12a1a fibroblasts that seem to originate directly from constitutive fibroblasts have actually transitioned through a col11a1a fibroblast state between injury and 3 dpi. Our findings demonstrate the added value of temporal lineage models like moslin in analyzing scLT time-course data.

Discussion

We introduce moslin, an approach that combines intra-individual lineage similarity with inter-individual gene-expression similarity for trajectory reconstruction from single-cell lineage-traced data. Moslin outperforms competing methods on simulated and *C. elegans* data by interpolating between Wasserstein and Gromov-Wasserstein regimes and by using lineage information from all available time points. We highlight in simulations that moslin compensates for noisy lineage relations through gene expression information, an important property for real scLT data. We illustrate moslin's capability to recover cell-state trajectories from real scLT data in zebrafish heart regeneration [32], where we predict a new origin for regenerative activated fibroblast states. We anticipate moslin to enable similar discoveries in the future for accumulating time-structured lineage-tracing datasets.

In contrast to many previous analysis paradigms for in vivo scLT data, moslin relates cells across time points rather than focusing on individual, isolated time points. While tree reconstruction from a single time point of lineage-traced cells can uncover shared lineage ancestry [10–13, 16, 18, 19], it falls short of characterizing the molecular properties of these ancestors. Moslin links putative ancestors to their descendants based on lineage and gene expression information; this enables us to relate the different activated fibroblast states as a function of the time past injury, a hypothesis that remains to be validated experimentally. Cell states undergo far-reaching changes over time in many situations such as cancer, cardiovascular, and neurodegenerative diseases. To understand the gene regulatory events that underlie these changes, it is crucial to identify the corresponding sequence of state transitions. As engineered mouse lines with built-in lineage recorders enable more labs to perform in vivo lineage tracing [28, 29], moslin serves as an easy-to-use toolkit to uncover these sequential state transitions.

We have tested moslin in scenarios with and without a known ground truth. An advantage of moslin is that we can easily interpret the role of each hyperparameter: α weights the importance of gene expression compared to lineage information, ε controls the strength of entropic regularization, and τ_a (only used for the zebrafish dataset) determines the level of unbalancedness at the source marginal, i.e., the earlier time point. Importantly, this knowledge provides guidelines for the directionality of adapting these per setting but does not assist in selecting the actual value, which is dataset dependent. We have addressed this in the analysis of the zebrafish dataset, where we lack ground truth. Specifically, we suggest setting initial marginals using growth rates and selecting the hyperparameters α, ε, and τ_a using a grid search, evaluating over a proxy task "cell type persistency." Performance on this proxy task strongly correlates with performance on the actual target task, the transient fibroblast precursor predictions, suggesting the cell type persistence test can be applied to other datasets without a known ground truth.

Through our analyses, we have identified guiding principles on the choice of hyperparameters: First of all, the entropic regularization parameter ϵ should be chosen small as long as this does not lead to convergence issues or overfitting. The GW scaling parameter α should have higher values if the datasets have a high degree of lineage information, and lower values if the lineage information is of lower resolution. The exact value of the unbalancedness parameter τ_a does not seem to be of strong impact on the performance in our analysis; however, this particular parameter was only used in one dataset.

Under the hood, moslin is based on moscot [48], a versatile framework for OT applications in single-cell genomics. As such, it benefits from moscot's interoperability with the scverse [87, 88] ecosystem and can take advantage of future moscot improvements concerning scalability and usability. Moslin's interface with CellRank 2 [23] grants it access to a range of established, constantly growing downstream-analysis functions. We demonstrate the power of combining moslin with CellRank 2 on the *C. elegans* data, where their combination reveals long-range state-change trajectories, driver genes, and temporal dynamics. Moslin's couplings could further be employed to regularize the inference of gene regulatory networks [89, 90], or to improve perturbation predictions [91].

In this study, we focus on the independent clonal evolution experimental design because it allows us to apply our method to in vivo scLT data. In this setting, lineage relationships are not directly comparable across time points, and we revert to assuming lineage concordance for pairs of cells. In contrast, for in vitro experiments and some regenerative or transplantation in vivo experiments, cells from the same population can be sampled at different time points, rendering their lineage information directly compatible across time points ("Methods"). Previously, OT-like approaches [14, 15] have been suggested for this clonal resampling experimental design [25]. Moslin could be extended towards this setting by adjusting the cost matrix definition.

While moslin consistently outperformed competing approaches in two simulation setups as well as on *C. elegans* embryogenesis data, we only achieved a small performance gain on zebrafish regeneration compared to a gene-expression-only baseline. We hypothesize this result to be a consequence of the lower lineage resolution offered in this experiment, compared to our simulations and to the ground-truth lineage tree used for *C. elegans* analysis. Recent innovations, including mitochondrial lineage tracing [30, 31, 92, 93] and base/prime editing [94–97], represent compelling use cases for moslin that might reveal the full potential of combining lineage with gene expression information. Improved lineage resolution will allow our method to yield highly accurate trajectory reconstructions in challenging disease contexts like cancer or inflammation.

Currently, moslin is limited to one replicate per time point. In the zebrafish data [32], where several replicates per time point are available, we address this by computing pairwise replicate linkages across time points and aggregating our insights across these. With the increasing popularity of scLT approaches, we expect more complex, multi-replicate time series to become available. For these, as an alternative to the aggregation approach above, we envisage a two-step process, first computing a consensus lineage representation per time point across replicates, and second, linking the consensus representations across time points.

Moslin could further be extended towards multi-modal scLT data [98, 99] to link molecular layers across time. For example, this could reveal how epigenetic changes

manifest in altered gene expression dynamics [100, 101]. Additionally, spatially resolved lineage tracing data [102–104] would enable moslin to regularize the coupling computation further using spatial neighborhoods. In this setting, moslin's inferred trajectories could be used to interrogate the relative contribution of internal state versus external signals towards observed fate decisions. scLT is a fast-moving field; we believe that development of novel simulation frameworks, which are capable of modeling diverse temporal dynamic regimes, will allow major advancements and further validation of suggested analysis tools. We further anticipate computational tools like moslin to play a crucial role in analyzing and interpreting novel lineage-traced datasets.

Conclusions

Moslin is a method for inferring differentiation trajectories from time-series single-cell studies, jointly making use of gene expression and lineage-tracing information at all available time points. These trajectories reveal terminal states of cellular differentiation, fate probabilities, lineage-correlated genes, and their temporal activation, and they allow pinpointing ancestor and descendant relationships in complex cellular processes like regeneration. Across two simulation studies, *C. elegans* embryogenesis and zebrafish regeneration, moslin either outperformed or was on-par with alternative approaches. Thus, we anticipate moslin to play an important role in unlocking the full potential of upcoming lineage-tracing technologies by combining their readout with molecular information across time points.

Methods

The moslin algorithm

Introduction and model overview

Moslin is an algorithm aimed at linking single-cell profiles across experimental time points. Computational linkage is required as sequencing is destructive; moslin thus allows linking molecular differences among cells at early time points with their eventual fate outcome at later time points. Critically, moslin uses molecular similarities and lineage tracing information to solve this challenging reconstruction problem. Specifically, moslin is applicable to dynamic, CRISPR-Cas-based approaches [10–13, 20, 105] that record lineage relationships in vivo. While previous analysis approaches for this type of lineage tracing data remained limited to individual, isolated time points [16–18, 20, 106, 107], moslin embeds clonal dynamics in their temporal context.

Moslin's inputs. The input to moslin are pairs of state matrices and lineage information $(X \in R^{N\times G}, \xi)$ and $(Y \in R^{M\times G}, \zeta)$ corresponding to N and M observed cells at early (t_1) and late (t_2) time points. State matrices X and Y typically represent gene expression (scRNA-seq) across G genes; however, moslin can also be applied to modalities like chromatin accessibility through adapted cost function definitions. The lineage information arrays ξ and ζ contain the lineage tracing outcome for every cell; their exact nature depends on the lineage tracing technology ("In vivo single-cell lineage tracing (scLT)" section). Optionally, moslin takes marginal distributions $a \in \Delta_N$ and $b \in \Delta_M$ over cells at t_1 and t_2 for probability simplex $\Delta_N := \{a \in R_+^N | \sum_{i=1}^N a_i = 1\}$. These marginals can represent any cell-level prior information; we use them to incorporate the effects of cellular growth and death.

Moslin's outputs. The output of moslin is a coupling matrix $P \in U(a,b)$ where $U(a,b)$ is the set of feasible coupling matrices given by

$$U(a,b) := \{P \in R_+^{N\times M} | P1_M = a,\ P^\top 1_N = b\},$$

for constant one vector $1_N = [1,...,1]^\top \in R^N$. The coupling matrix P links cells at t_1 with cells at t_2; the i th row $P_{i,:}$ tells us how cell i from t_1 distributes its probability mass across cells at t_2 and the j th column $P_{:,j}$ tells us how much probability mass cell j at t_2 receives from cells at t_1. The set $U(a,b)$ contains all matrices P which are compatible with the prescribed marginals a at t_1 and b at t_2.

With these definitions at hand, we can formalize the aim of moslin: we seek to find the coupling matrix $P \in U(a,b)$ which simultaneously minimizes the distance cells have to travel in phenotypic space between t_1 and t_2 while respecting lineage relationships. We explain how we find such a matrix in the "Moslin's objective function for in vivo ICE" section.

In vivo single-cell lineage tracing (scLT)

Moslin uses lineage tracing data to guide the reconstruction of a coupling matrix P between t_1 and t_2 cells. Early methods for lineage tracing were labor-intensive, limited to transparent organisms, and relied on manual observation of individual cells in time-lapse microscopy [25, 54]; recent approaches are sequencing-based and use heritable genetic barcodes [8, 108–110]. While a multitude of such techniques exists, moslin is geared towards those that achieve single-cell resolution, yield joint lineage and gene expression readout, and can be applied in vivo.

Clonal resampling (CR) versus independent clonal evolution (ICE). Critically, moslin is able to describe non-steady state biological processes like development or regeneration that require time-series experimental designs to capture cell-state trajectories. Experimentally, this can be achieved using either clonal resampling (CR) or independent clonal evolution (ICE) designs, which assay cells from the same or different clones across several time points, respectively.

In clonal resampling (CR), the aim is to observe the same clone (cells sharing the same barcode) across several time points, i.e., for a single phylogenetic tree, we aim to observe some ancestral nodes besides the leaf nodes. As this approach relies on the repeated sampling of clonally related cells, it applies primarily to in vitro settings [9, 19, 111], in vivo transplantation settings [112], or in vivo regenerative systems like human PBMC and CD34+samples [92, 113] or the zebrafish fin [11]. Beyond these transplantation and regenerative settings, applying time-series scLT in vivo requires independent clonal evolution (ICE), i.e., different individuals, sequenced at different time points with independent clonal evolution proceeding in each animal. This represents an additional challenge since the lineage of cells in different individuals cannot be compared directly. We designed moslin for the challenging ICE setting that allows us to model in vivo systems.

Moslin's objective function for in vivo ICE

With the definition of ICE at hand, we return to moslin's key task: finding a coupling matrix $P \in U(a,b)$ which simultaneously minimizes the distance cells have to travel in phenotypic space while respecting lineage relationships. Mathematically, we cast

this task as an optimal transport (OT) problem [39]; in particular, we use a Fused Gromov-Wasserstein [40] (FGW) formulation which allows us to include terms for across and within time point similarity (Additional file 2: Note S1). Previous single-cell methods successfully used OT to map cells across time points without lineage information [1, 3, 48], impute gene expression in spatial data [43, 48], predict perturbation response [114], learn patient manifolds [115, 116], integrate data across modalities [46], and infer cell–cell communication [47]. In particular, we make the following assumptions (A):

- A1: cells change their state gradually; overall, they minimize the distance traveled in phenotypic space between t_1 and t_2.
- A2: on average, lineage relations are concordant across time points; cells with similar lineage history at t_1 are likely to transition into cells with similar lineage history at t_2.

All three assumptions may be challenged in practice:

- Batch effects and incomplete molecular information challenge A1.
- Rapid transcriptional convergence and divergence, as well as noisy or incomplete lineage readout, challenge A2.

Thus, rather than enforcing A1 and A2 exactly, we design custom cost functions to balance them in our FGW objective function; individual cells may violate any combination of assumptions at the cost of incurring a penalty.

Note that A2 does not require deterministic development, where each individual develops according to the exact same lineage tree. Instead, we require a relaxed version of this setup, where different individuals develop according to different lineage trees which may be related at the level of pairwise distances.

A combined approach for in vivo scLT data. In ICE, gene expression information is comparable across time points but lineage information is not ("In vivo single-cell lineage tracing (scLT)" section). Our FGW setting allows us to define terms that handle both type of information:

- A linear Wasserstein (W) term for comparable features, encouraging A1. This term quantifies gene expression similarity.
- A quadratic Gromov-Wasserstein (GW) term for incomparable features, encouraging A2. This term quantifies lineage distance concordance.

The W term for individual comparisons. To encourage A1, we consider a W term [39] which compares individual cells in the source (t_1) and target (t_2) distributions in terms of their gene expression vectors. Given gene expression vectors $\left(x_i, y_j\right) \in X \times Y$, we construct a cost matrix, $C \in R_+^{N \times M}$ with $C_{ij} = c\left(x_i, y_j\right)$ for cost function c. An entry in the cost matrix, C_{ij}, depicts the distance between cells i and j according to the cost function c. We define the cost function to represent squared euclidean distance in a joint latent space over X and Y, computed using PCA or scVI [49]. Formally, the mapping problem is defined as

$$P^* := argmin_{P \in U(a,b)} \langle C, P \rangle = argmin_{P \in U(a,b)} \sum_{ij}^{N,M} C_{ij} P_{ij},$$

for optimal coupling matrix P^*. This objective function defines a convex linear program; the optimal P^* will be the one accumulating the lowest cost according to C when transporting cells from t_1 to t_2.

The GW term for pairwise comparisons. To encourage A2, we consider a GW term [22] which compares cell pairs in the source (t_1) and target (t_2) distributions in terms of their lineage information. Given lineage information at two time points, we define two independent cost matrices, $C^X \in R_+^{N \times N}$ and $C^Y \in R_+^{M \times M}$ with $C_{ij}^X = c^X\left(x_i, x_j\right)$ and $C_{kl}^Y = c^Y\left(y_k, y_l\right)$ for cost functions c^X and c^Y.

Focusing on the early time point, consider lineage information ξ_i. Define the t_1-cost function

$$c^X\left(\xi_i, \xi_j\right) = c_l\left(f^X(\xi_i), f^X\left(\xi_j\right)\right),$$

for mapping function f^X, providing a representation of the lineage information at t_1, and lineage distance function c_l. Moslin supports two ways of representing lineage information:

- Barcode representation: f^X is the identity and c_l quantifies hamming distance between raw barcodes.
- Lineage tree representation: f^X is a lineage-tree reconstruction computed using a method like Cassiopeia [16] or LINNAEUS [10] and c_l quantifies shortest path distance along reconstructed trees.

We employ an analogous set of definitions for the t_2-cost function c^Y. We apply these cost functions to all (pairs of) cells to yield the cost matrices $C^X \in R^{N \times N}$ and $C^Y \in R^{M \times M}$. With the cost matrices at hand, we define a quadratic GW term that compares pairwise distances across time points,

$$P^* := argmin_{P \in U(a,b)} \sum_{ijkl}^{N,N,M,M} L\left(C_{ij}^X, C_{kl}^Y\right) P_{ik} P_{jl},$$

for some distance metric L that compares cost-matrix entries. By default, we use the l_2 distance in moslin. Intuitively, this term encourages similar cells at t_1 to be matched to similar cells at t_2.

Moslin's Fused Gromov-Wasserstein (FGW) approach. To simultaneously encourage A1 and A2, we combine the W with the GW term to yield moslin's objective function for in vivo ICE data,

$$P^* = argmin_{P \in U(a,b)} \alpha \sum_{ijkl}^{N,N,M,M} L\left(C_{ij}^X, C_{kl}^Y\right) P_{ik} P_{jl} + (1-\alpha) \sum_{ik}^{N,M} C_{ik} P_{ik} \quad (1)$$

which is known as a Fused Gromov-Wasserstein (FGW) problem [40] (Additional file 2: Note S1). The parameter $\alpha \in (0,1)$ controls the interpolation between the W and GW

terms. Using this interpolation, we jointly optimize the coupling with respect to gene expression and lineage information.

Entropic regularization and optimization. The combined objective of Eq. 1 defines a quadratic programming problem; to introduce a notion of uncertainty and to speed up the optimization, we follow previous approaches [1, 41] and include and entropy regularization term,

$$H(P) = -\sum_{ij}^{N,M} P_{ij}\left(\log P_{ij} - 1\right),$$

and the regularized FGW objective reads

$$P^* := argmin_{P \in U(a,b)}\ \alpha \sum_{ijkl}^{N,N,M,M} L\left(C^X_{ij}, C^Y_{kl}\right) P_{ik} P_{jl} + (1-\alpha) \sum_{ik}^{N,M} C_{ik} P_{ik} - \epsilon H(P),$$

for regularization strength ϵ. Intuitively, the entropy term $H(P)$ favors probabilistic over deterministic couplings. We optimize the entropy-regularized FGW objective function using a mirror descent scheme; each inner iteration of the algorithm reduces to well-known Sinkhorn iterations [41] (Additional file 2: Note S1). To determine convergence, we check whether the current and previous regularized OT costs are close using $jax.numpy.isclose(..., rtol = R\_TOL)$, with $R\_TOL = 1e-3$ by default.

Marginals encode prior biological information. If additional information about sampled cells is available, e.g., growth and death rates and uncertainty, we incorporate them via the marginals a and b. If no additional information is available, we assign them uniformly. By default, in moslin, we choose the right marginal b uniformly, $b_j = 1/M\ \forall j \in \{1, ..., M\}$, and adjust the left marginal to accommodate cellular growth and death between t_1 and t_2,

$$a_i = \frac{g(x_i)^{t_2 - t_1}}{\sum_{j=1}^{N} g\left(x_j\right)^{t_2 - t_1}}\ \forall i \in \{1, ..., N\},$$

where $g : R^D \to R$ is modeled as the expected value of a birth–death process with proliferation at rate $\beta(x)$ and death at rate $\delta(x)$, thus $g(x) = e^{\beta(x) - \delta(x)}$ for $\beta(x)$ and $\delta(x)$ estimated from curated marker gene sets for proliferation and apoptosis, respectively [1].

Accommodating uncertainty in the inputs. As we estimate growth and death rates from marker genes, they represent a noisy estimate of the underlying ground truth growth and death rates. In addition, we randomly sample cells from a population, which leads to deviations from the ground-truth cell-type proportions. In our case, different time points typically correspond to different individuals, which amplifies differences in cell type proportions across time points.

Accordingly, we allow small deviations from the exact marginals a and b in an unbalanced FGW framework [51] where we replace the hard constraint $P \in U(a, b)$ with soft Kullback–Leibler (KL) divergence penalties, giving rise to moslin's final objective function for time-series scLT data. To control the weight given to left (a) and right (b) marginal constraints, we use two parameters $\tau_a, \tau_b \in (0,1)$(Additional file 2: Note S1). For

the optimization, we employ the algorithm presented by ref [51]. which is based on a bi-convex relaxation leading to alternate Sinkhorn iterations.

Implementation. Moslin is available at https://github.com/theislab/moslin. Under the hood, moslin is based on moscot, our open-source framework for multi-omic single-cell optimal transport. moscot is a scalable, easy-to-use, open-source solution for OT-based analysis in single-cell genomics; it interfaces with optimal transport tools [117] (OTT) in the backend to support GPU acceleration and just-in-time compilation via JAX [118].

Downstream usage of coupling matrices

Once we have identified the optimal coupling matrix P, we use it to link observed cells between t_1 and t_2. Note that the coupling matrix P combines the information from molecular similarity and lineage history; thus, all downstream analysis is lineage and state informed.

Consider a t_1 cell state P of interest. This state could represent, e.g., a rare or transient population with unknown position in the differentiation hierarchy. Define the corresponding normalized indicator vector,

$$p_{t_1}(x) := \{\frac{1}{|P|} x \in P, 0 \; else,$$

where x is a cell from t_1 and $|P|$ corresponds to the number of cells in state P. Following ref [1]., we compute t_2 descendants of cell state P by a push-forward operation of p_{t_1},

$$p_{t_2} = P^\top p_{t_1}, \tag{2}$$

where $p_{t_2}(x)$ is the probability mass that cell state P distributes to cell x at t_2. Similarly, to compute ancestors of a cell state Q at t_2, consider the corresponding normalized indicator vector q_{t_2}. To compute the ancestor distribution, we use a pull-back operation,

$$q_{t_1} = P q_{t_2}, \tag{3}$$

where $q_{t_1}(x)$ is the probability mass that cell x contributes towards cell state Q at t_2. For further downstream analysis, e.g., to identify initial and terminal states, driver genes of fate decisions, and gene expression trends, we interface with CellRank 2 (ref [23].), a fate mapping toolkit that analyzes our coupling matrices using a Markov framework.

Coupling cells across more than two time points. Moslin relates cells across more than two time points; consider a time-series experiment with sequencing at time points $\{t_1, ..., t_T\}$. Following ref [1]., we solve for individual pairwise couplings between adjacent time points; this yields coupling matrices $\{P^{t_1,t_2}, ..., P^{t_{T-1},t_T}\}$. We construct longer-range couplings by matrix-multiplying individual couplings. For example, to couple initial-day cells to final-day cells, we obtain

$$P^{t_1,t_T} = P^{t_1,t_2} P^{t_2,t_3} ... P^{t_{T-1},t_T}.$$

We compute ancestors and descendants for multi-day couplings in the same way as above (Eqs. 2 and 3).

Datasets

2-gene simulations

We use a simulation setting suggested by ref [21]. which constructs a vector field to recreate a biologically plausible trajectory structure. Under the simulation, cells follow the vector field with diffusion and occasional cell division. The simulation assigns a heritable lineage barcode that is randomly mutated, to each cell. Four different types of trajectories, of increasing complexity, are considered in this simulated setting:

1. Bifurcation (B): a simple bifurcation of a single progenitor cell type into two descendant cell types.
2. Partial convergent (PC): two initial clusters split independently, following the split, two of the resulting four clusters merge together for a total of three clusters.
3. Convergent (C): two initial clusters converge to a single final cell type.
4. Mismatched clusters (MC): two initial clusters both split into two late-time clusters, and cells from two of the resulting clusters are transcriptomically closer to early cells that are not their ancestors.

The simulated data provides us with what ref [21]. defines as an *embedded lineage tree*, referring to the collection of branching paths due to cell divisions within a population (whereas a lineage tree denotes the coordinate-free tree structure). For each of the trajectories, we simulate 10 different data sets with a different random seed and measure the *embedded lineage tree* at two time points (with 64 and 1024 cells, respectively). All simulations were performed using the default settings provided in the LineageOT code package: https://github.com/aforr/LineageOT.

Given the simulated data, which consists of gene expression, barcodes, and the true lineage tree, we compute couplings between time points in two manners, considering the *true tree* or a *fitted tree*. For the latter, the tree is inferred using the neighbor-joining algorithm as implemented in LineageOT [21]. We compare the performance of moslin to LineageOT, CoSpar [15], and two extreme cases of the moslin formulation: using only gene expression in a W term ($\alpha = 0$) and using only lineage information in a GW term ($\alpha = 1$). For CoSpar, we test two settings, one relying only on gene expression and one which includes lineage information as well. We quantify method performance using the ancestor and descendant errors introduced in LineageOT. Lineage information is incorporated differently by each method:

1. Moslin: we set the lineage costs by computing distances between cells along the tree. The distance is defined as the length of a weighted shortest path found using Dijkstra's algorithm with weights associated to edges according to "time" between two nodes.
2. LineageOT: the tree (true or fitted) is used directly to compute the couplings.
3. CoSpar: given the tree (true or fitted) the clonal assignment of cells at the later time point is done based on their ancestor at the earlier time point. That is, given cell i from the earlier time point, its descendants at the later time point are associated with clone i.

For ground truth coupling P^* and predicted coupling P, we compare their predicted ancestors and descendants per cell using a Wasserstein-2 distance (Additional file 2: Note S1). To obtain the descendant error $E_D(P)$, we compute

$$E_D(P) = \sum\nolimits_{i=1}^{N} a_i \, W_2^2\left(P^*_{i,:}, P_{i,:}\right),$$

for squared Wasserstein-2 distance W_2^2 (Additional file 2: Note S1) and right marginal $a_i = \sum_j P_{ij}$. Similarly, to obtain the ancestor error $E_A(P)$, we compute

$$E_A(P) = \sum_{j=1}^{M} b_j W_2^2\left(P^*_{:,j}, P_{:,j}\right),$$

for left marginal $b_j = \sum_i^N P_{ij}$. Note that we compare rows for $E_D(P)$ and columns for $E_A(P)$, scaled by the corresponding marginal to adapt the weight we give to each cell. Thus, a value of zero in either metric means that we are on par with the ground-truth coupling. Additionally, we independently normalize ancestor and descendant errors using the outer product of the marginals, $\widehat{P} = ab^\top$, corresponding to an uninformative coupling with the same marginals as the predicted coupling P. Specifically, we compute $E_D(P)/E_D\left(\widehat{P}\right)$ and $E_A(P)/E_A\left(\widehat{P}\right)$, such that a value of one corresponds to an uninformative result. Our final error metric is given by the mean of the two quantities [21]. Crucially, this distance takes the geometry of the underlying phenotypic landscape into account. Couplings to cells that are not the actual ancestors or descendants of the reference cell incur a larger penalty in mean error if they are further away from the true ancestors or descendants in terms of their gene expression states.

We perform a grid search to find the optimal parameters for each data set and method independently. For moslin and LineageOT, the entropy parameter is optimized over 15 values of ϵ log-spaced between $1e-4$ and $1e+1$. For moslin, we also perform a grid search for the interpolation parameter, $\alpha \in \{0.1,0.2,0.3,0.4,0.5,0.7,0.8,0.9,0.95,0.98,0.999\}$. To run CoSpar, we use "cospar.tmap.infer_Tmap_from_state_info_alone()" (only gene expression) and "cospar.tmap.infer_Tmap_from_one_time_clones()" (with lineage information) using hyperparameters reported in the tutorial Transition map inference. For proper evaluation, as the obtained transition matrix is not a valid transport map we row (column) normalize it and ensure a total mass of one for descendant (ancestor) error evaluation.

Of note, we hypothesize CoSpar's poor performance can be explained by the simplicity of the simulated trajectories and CoSpar's reliance on state information when applied using clonal information only from the latest time point. In this setting CoSpar first constructs a coupling based only on state heterogeneity, obtained using optimal transport (similarly to the moslin extreme using gene expression only, $\alpha = 0$, W). The initial coupling is used to back-propagate the initial state likelihood and infer initial clones used as input to run CoSpar assuming "full" clonal knowledge. With this, the limited performance of the W solution over these trajectories (Fig. 2b) hints on an error which is propagated in the couplings inferred by CoSpar.

TedSim simulated data

We utilize TedSim [52] (single-cell temporal dynamics simulator), which simulates cell division events from root cells to present-day cells, simultaneously generating two data modalities for each cell, gene expression, and a lineage barcode. The cell lineage tree is simulated as a binary tree that models the cell division events. In order to simulate diverse cell types, the notion of asymmetric divisions [119–121] is used. The asymmetric divisions allow cells to divide into cells with different cellular fates. One cell evolves into a new state and the other preserves the ancestor state. The evolution of cells is governed by a *cell state tree*. Two user-defined parameters control this simulation process:

1. *step_size*: defines the distance between two adjacent sampled states on the cell state tree. Larger *step_size* implies more distinct cell states along the tree.
2. p_a: the probability for a division in the sampled tree to be asymmetric. Larger p_a implies rapid transitions in the sampled tree.

In accordance with the original publication [52], we noticed that these parameters have a small effect on the mapping accuracy hence report results for $p_a = 0.4$ and $step\_size = 0.4$.

For the lineage information, barcodes are simulated as an accumulation of CRISPR/Cas9-induced scars along the paths from the root to all the leaf cells. Here, we add to the TedSim simulated barcodes a stochastic silencing rate, corresponding to the rate at which entire segments (cassettes) are removed from the barcode. In the TedSim simulation, each cassette has 4 characters and there are 8 cassettes per barcode. With this, we aim to simulate the expected dropout due to low sensitivity of assays.

To obtain the datasets, we follow the TedSim published tutorial, Simulate-data-multi.Rmd. Setting $p_a = 0.4$ and $step\_size = 0.4$ and creating 10 different data sets using different random seeds.

Given the simulated gene expression and barcodes, we define moslin's lineage costs as the scaled hamming distance between the barcodes, as defined by ref [21].. The scaling is defined such that (i) the number of sites where both cells were measured is taken into account and (ii) the distance between two scars is twice the distance from scarred to unscarred sites. To benchmark moslin, we ran a grid search over $\alpha \in \{0.1,0.25,0.5,0.75,0.9,1\}$ and $\epsilon \in \{1e-3,1e-4\}$. For LineageOT, we tested with $\epsilon \in \{1e-1,1\}$.

Similarly to the previous setting, for LineageOT, the barcodes are used internally to construct a fitted tree. At high *ssr* values ($ssr > 0.2$), LineageOT fails at the tree reconstruction procedure. Specifically, the failure occurs once it encounters a cell with a completely nullified barcode.

For CoSpar, we rely on barcode distances to construct the clonal information. Formally we cluster the barcode distance matrix to define n clones, where n is chosen to be the number of cells in the early time point. Of note, we use the barcode distance to limit dependency on external reconstructions tools (as these may fail at high *ssr* as observed for LineageOT), whereas using barcode distance moslin attains good performance. We have validated this choice by comparing CoSpar's performance using ground truth tree (for $ssr = 0$) and using LineageOT's reconstruction (for $ssr \leq 0.2$, Supplementary

Fig. 1e). Fitting is done using provided functions as in the previous setting. Again, for proper evaluation, as the obtained transition matrix is not a valid transport map we row (column) normalize it and ensure a total mass of 1 for descendant (ancestor) error evaluation.

Judging the performance of LineageOT and CoSpar over all *ssr* s, in comparison to moslin's performance. We hypothesize that both fail to extract valuable structure from the provided lineage information, regardless of *ssr* as their performance is not harmed as the *ssr* increases and is comparable to moslin's at largest *ssr* at which lineage information is noisy.

Robustness analysis

- Mapping of emergent states: We evaluate method's ability to accurately identify ancestors of emergent states. That is cell states that appear only in the late time point. For a cell at a late time point, an accurate mapping is a mapping in which the most probable ancestor corresponds to its ancestor in the ground truth tree.
- Subsampling: To preserve the notion of ground truth, we limit subsampling to the removal of a fraction f of cells in the late time point. In this setting, considering a range of $f \in \{0.1,0.2,0.4,0.6\}$ we evaluate the mean error of the obtained mapping.

C. elegans embryonic development

The *C. elegans* development dataset [7] contains gene expression for approx. 86k single cells, sequenced using $10\times$ genomics. The original authors [7] mapped these cells towards the known *C. elegans* lineage tree [54] and obtained lineage information for a subset of cells. Additionally, they mapped their data towards a bulk time-series dataset [122] to estimate the developmental stage of individual cells. Binning these estimated cell times gave rise to several pseudo-experimental time points, spanning 150–580 min past fertilization.

Pre-processing. To evaluate moslin's performance, we required ground-truth lineage information. The original study's mapping inferred partial lineage information for a subset of approx. 46k cells. To obtain complete lineage information, we implemented two suggestions by ref [21].:

1. Strategy 1: subsetting to the ABpxp lineage. This is a symmetric lineage where "x" indicates either the right ("r") or the left ("l") cell.
2. Strategy 2: subsetting to all cells with complete lineage information.

As the lineage for cells obtained from strategy 1 is not fully specified due to "x," the two strategies lead to disjoint subsets of cells, allowing us to test moslin's performance in two different scenarios.

For either cell subset, we pre-processed the data using SCANPY [87] and used default parameters if not indicated otherwise. In particular, we normalized total counts, log-transformed the data, annotated the top 3k highly variable genes using the "seurat" flavor [123], and computed 50 principal components in the space of highly variable genes.

To have a sufficient number of cells per time point, we removed time points that contained less than 100 cells. This left us with the following 7 time points: 170, 210, 270, 330, 390, 450, and 510 min past fertilization.

Embedding and cell-type labels. Using the top 10 principal components, we computed a k-nearest neighbor (kNN) graph for 30 nearest neighbors and visualize it by computing a UMAP embedding [56]. To reduce complexity and focus on the main groups of terminal cell states, we aggregated original cluster annotations slightly to arrive at the annotations we show in Fig. 2 and Supplementary Fig. 2. Our aggregation entailed the following steps:

- Summarize AIM, AIY, AVB, DB, PVP, RIB, RIC, SIA, and RIV as "other terminal non-ciliated neurons."
- Summarize Neuroblast_PVC_LUA and Parents_of_U_F_B_DVA as "other pre-terminal non-ciliated neurons."
- Summarize pm7, DVA, GLR, DA, and Pharyngeal_neuron as "other terminal cells."
- Summarize AIN_parent, M1_parent, PVQ_parent, RME_LR_parent, Parents_of_Y_DA6_DA7_DA9, Parent_of_tail_spike_and_hyp10, and Parents_of_PHsh_hyp8_hyp9 as "other pre-terminal cells."

The vast majority of cells we labeled "other terminal cells" are pharyngeal neurons (24/30 cells), and the vast majority of cells we labeled "other pre-terminal cells" are pre-terminal hypodermis cells (Parent_of_tail_spike_and_hyp10 with 53/90 cells and Parents_of_PHsh_hyp8_hyp9 with 25/90 cells). We show the original cluster annotations, prior to aggregation, in Supplementary Fig. 2.

We labeled cells that had neither terminal nor pre-terminal cell-type label (but lineage annotation) as "progenitors." These correspond to earlier cells in the lineage tree, for which terminal identity has not been established yet.

Benchmarking descendant and ancestor reconstructions Unless stated otherwise, we use default method parameters.

Shared method parameters and settings. We benchmarked moslin with LineageOT, and the two extreme cases of our method W (just gene expression in a Wasserstein term, corresponding to $\alpha = 0$) and GW (just lineage information in a Gromov-Wasserstein term, corresponding to $\alpha = 1$) on the two cell subsets (strategies 1 and 2), using the pre-processing described above. We use the marginals a and $b$to capture the effects of cellular growth and death, and calculate them using the lineage tree following ref [21]. Gene-expression distances among cells from different time points were measured using squared Euclidean distance in the PCA space and passed to all methods in the mean-scaled cost matrix C.

Additional moslin, W, and GW parameters. We did not allow for deviations from the marginals via unbalancedness in this application, as the marginals are lineage-informed and thus more accurate compared to other applications. To construct the lineage cost matrices C^X and C^Y for moslin and GW, we compute distances between same-time point cells along the lineage tree. The distance is defined as the length of a weighted

shortest path found using Dijkstra's algorithm. The weights represent the temporal difference between a node and its parent. Additionally, we mean-scaled the C^X and C^Y cost matrices. We set a maximum iteration budget of 30k (inner) Sinkhorn iterations for moslin, W, and GW.

Additional LineageOT parameters. We run LineageOT following the original authors' reproducibility repository. LineageOT runs the Sinkhorn algorithm as implemented in Python optimal transport (POT) [124] under the hood; their convergence criterion checks that the constraints imposed by the marginal distributions are satisfied within a certain threshold. We set this threshold to 10^{-3}.

Grid search. To identify the best hyperparameters for either method per time point pair, we run a grid search over the following parameter grid:

- W and GW:
  - $\epsilon \in$ [0.001,0.01,0.05,0.1,0.5].
- Moslin:
  - ϵ as above.
  - $\alpha \in$ [0.01,0.1,0.25,0.5,0.75,0.9,0.95,0.98].
- LineageOT:
  - ϵ as above.

For each method, the performance we report corresponds to the best performance found across this grid.

Mean error computation. To quantify method performance per time point, we computed the ancestor and descendant errors over the PCA space, as described above for our simulation study. We used the mean over ancestor and descendant errors as our final accuracy metric.

Studying the effect of errors in lineage distances on moslin's performance. We wanted to evaluate how inaccuracies in lineage distance information affect moslin's performance in terms of the mean error. Separately for a set of time point pairs from the two data subsets (ABpxp and cells with complete lineage information), we used ground-truth lineage information to compute symmetric lineage-distance cost matrices C^X and C^Y, corresponding to cells at early and late time points, respectively. Throughout this analysis, we used optimal moslin hyperparameters as identified in our grid search.

In order to simulate tree reconstruction errors, we perturbed a certain fraction of the information in C^X and C^Y, separately for early and late cells. In particular, given a target percentage c, we extracted the indices corresponding to the upper matrix triangular and picked c% of these indices. Next, we randomly permuted lineage distance information for these c% of indices by sampling without replacement. As some cells might by chance receive the same lineage information through sampling, we computed the actual percentage of permuted cost matrix elements c'%. To maintain cost matrix symmetry, we mirrored the perturbed upper matrix triangular to the lower matrix triangular. We repeated these calculations for cells at early and late time points and averaged their

corresponding percentages of actual perturbed matrix elements c'% to arrive at a final measure for the degree of lineage-distance cost matrix perturbation. Separately for each pair of time points and for each data subset, we iterated over target permutation percentages c between 0 and 100%, applied moslin, and recorded the mean error.

Zoom in to the 330/390 min time point pair. To visualize the predicted transitions for RIM_parent cells, we selected 330 min RIM_parent cells and further restricted our attention to those cells assigned to the ABpxppaapa lineage; these cells represented the vast majority (80/85) of the RIM_parent population. We considered the corresponding rows in predicted coupling matrices. To focus on the most confident predicted links, we only retained matrix elements exceeding 10% of the maximum coupling value, i.e., we required $P_{ij} > 0.1\, max_{ij} P_{ij}$, separately for all predicted couplings and for the ground-truth coupling. We visualized the remaining matrix elements in a UMAP embedding by connecting each RIM_parent cell to its confidently predicted descendants. To quantify method performance over the RIM_parent population, independent of the UMAP embedding and of any thresholding scheme, we computed the descendant error for RIM_parent cells, as described in our simulation study. Additionally, we computed the ancestor error for RIM cells at the later time point.

Combining different methods with CellRank 2 for comparative fate mapping analysis We focused on the ABpxp lineage (strategy 1) and ran moslin, W, GW, and LineageOT with the optimal hyperparameters identified in our grid search. We filtered out cells assigned a zero value in the marginal distributions to arrive at 6476 cells used for this analysis.

Computing pseudotimes with Palantir and CytoTRACE. Additionally, we included two state-of-the-art methods for general trajectory inference that do not make use of lineage tracing information: Palantir [57] and CytoTRACE [71].

Palantir computes a pseudotime based on iteratively reweighted random walks in the space of multi-scale diffusion components. Using Palantir, we computed a pseudotime from a randomly selected cell from the earliest embryo stage in our data. We used 30-nearest neighbors and sampled 1200 waypoint cells.

CytoTRACE computes a measure of developmental potential based on the number of genes expressed per cell and refines this score through k-NN smoothing. The algorithm is based on the assumption that less mature cells, on average, express more genes because they regulate their chromatin less tightly compared to more mature cells. In the original study [71], the authors validated this assumption across different species, experimental technologies, and developmental stages. While the original CytoTRACE score S captures developmental potential and is thus high for naive cells and low for mature cells, we scale S to the [0,1] range and employ $1 - S$ as a pseudotime. As the original CytoTRACE implementation does not scale to large cell numbers, we used the CellRank 2 implementation [23] with k-NN imputation over a 10-dimensional PCA space with 30 neighbors.

Interfacing different methods with CellRank 2. CellRank [125] is a fate mapping framework originally designed for RNA velocity [126, 127] data. In version 2 (ref [23].), it has been extended towards other data modalities, including time-series and pseudotime information. We make use of these extensions here to compare different methods in a

systematic way, with consistent downstream processing enabled by CellRank 2. In the following, we used CellRank 2 with default parameters if not indicated otherwise.

Transition matrix construction with the RealTimeKernel. For LineageOT, W, GW, and moslin, we use CellRank 2's RealTimeKernel to compute a joint transition matrix T across all time points for downstream CellRank analysis. Starting from an all-zero matrix T, containing cells from all time points, we execute the following steps:

1. First, we place coupling matrices on the superdiagonal of T for transporting cells from early to late time points.
2. Second, we compute transition matrices within each time point based on gene expression similarity. We place these matrices on the diagonal of T.
3. Third, we compute a global transition matrix $T\prime$ across all time points based on gene expression similarity. We combine T with $T\prime$ with weights 0.9 and 0.1, respectively. This step improves matrix conditioning and yields the matrix T''.

We row-normalize T'' to arrive at the final CellRank 2 transition matrix, which we interpret as a Markov chain. For moslin's CellRank 2 transition matrix, we simulated 200 random walks, each containing 500 steps, to visualize the predicted cell dynamics, starting from randomly selected 170 min cells. We repeated this analysis for 200, 300, 400, 600, and 700 steps to demonstrate method robustness.

Transition matrix construction with the PseudotimeKernel. For Palantir and CytoTRACE, we use CellRank 2's Pseudotime and CytoTRACEKernels, respectively, to compute a joint transition matrix T across all time points for downstream CellRank analysis. Both kernels combine their corresponding pseudotime with a $k = 30$ nearest neighbor graph, computed over 10-dimensional PCA space, to direct graph edges into the direction of increasing pseudotime. Thus, both kernels yield directed transition matrices, which reflect the developmental dynamics encoded through pseudotime and gene-expression similarity in the k-NN graph.

Identifying terminal states and computing aggregated fate probabilities. For all methods, we used CellRank 2's GPCCA estimator [128, 129] to compute 7 terminal states. We represented each terminal state by the 30 cells most confidently assigned to it. We aggregated individual terminal states to represent ciliated neurons, non-ciliated neurons, and glia and excretory cells, by combining the 30 cells identified per state. We computed absorption probabilities on the Markov chain towards these combined cell sets per terminal state group and interpreted these as fate probabilities. In other words, for each non-terminal cell, we initialized several random walks and recorded the terminal cell set they reached. Taking the number of random walks to infinity, these "arrival frequencies" converge to absorption probabilities, which can be computed efficiently in CellRank 2 [23].

Comparing terminal states across methods. We used two methods to compare terminal states across methods:

1. The mean time point assignment over cells assigned to terminal states.
2. The mean macrostate purity.

For the first metric, we compute a mean time point assignment per terminal state by averaging over the time point assigned to each cell within that terminal state, and we further average this quantity over all terminal states. For the second metric, we follow the implementation in CellRank 2 and compute, for each terminal state, the fraction of cells assigned to the largest cell population within that terminal state. If all cells within one terminal state come from the same underlying cluster, this quantity would be 1. We further average this quantity over all terminal states.

Comparing fate probabilities across methods. We compare different methods by averaging their fate probabilities towards ciliated neurons, non-ciliated neurons, and glia and excretory cells, over groups of pre-terminal and progenitor populations.

Predicting driver genes. Using CellRank 2's transition matrix for moslin, we correlated each gene's expression with the computed fate probabilities across all cells and subsetted to known *C. elegans* transcription factors [73] (TFs). We focused on the top 20 most strongly correlated TFs per terminal cell group and treated these as predicted driver TFs.

Visualizing expression trends in a heatmap. To visualize expression trends towards the non-ciliated neuron terminal state group, we selected the top 50 genes most strongly correlated with the corresponding fate probabilities (not subsetting to TFs). We imputed gene expression using MAGIC [130] and fitted generalized additive models (GAMs) to each gene's imputed expression as a function of the Palantir pseudotime, supplying non-ciliated neuron fate probabilities as cell-level weights to the loss function. Specifically, we used a spline basis and fitted GAMs with the mgcv package, through the CellRank 2 interface.

Zebrafish heart regeneration (LINNAEUS)

The zebrafish heart regeneration dataset [32] consists of hearts from 25 organisms: four uninjured hearts (ctrl), nine at 3 days after injury (3 dpi), and seven at 7 days after injury (7 dpi). We use moslin to calculate couplings P_{ik}^{ab}, with a and b denoting datasets at consecutive time points. For ease of reading, we will suppress indices a and b in the following unless necessary.

Mapping datasets We embed the transcriptomic readout of all single cells with lineage information into a joint latent space using scVI [49], retaining the original cluster annotations. We calculate tree distances as shortest path distances along the original reconstructed trees. We use the moslin unbalanced FGW setting to calculate couplings between cells at consecutive time points for a hyperparameter grid where $\alpha \in \{0.01,0.1,0.15,0.5\}$, $\epsilon \in \{0.01,0.05,0.1,1\}$, and $\tau_a \in \{0.4,0.5,0.6,0.9,1\}$. For method comparison, we calculate the same couplings using just gene expression in a W term (W) (with hyperparameters $\epsilon \in \{0.01,0.1,1\}$ and $\tau_a \in \{0.4,0.6,0.9,1\}$) and LineageOT (with hyperparameter $\epsilon \in \{0.001,0.01,0.1\}$).

In our calculations, we provide growth rates as initial marginals. To calculate growth rates, we use cell cycle marker genes typically used in single cell data [123] and the GSEA Hallmark apoptosis geneset (https://www.gsea-msigdb.org/gsea/msigdb/human/geneset/HALLMARK_APOPTOSIS.html).

These are converted to their zebrafish orthologs using orthologs from Alliance, as previously described [32]. Next, we use these two gene sets to calculate growth rates [1]. For cells at 3 dpi that are in the regeneration process, we use the growth rates as calculated. However, cells at control are not in a regenerating heart and the calculated growth rates may not correlate with the actual injury response. Instead, we use cell type average growth rates as an approximation of the tendency of cell types to proliferate.

Cell type persistence test We constructed a cell type persistence test to select optimal hyperparameter values. We expect that cells of the same, non-transient, type are persistent over time; cells c_2^j of a non-transient (more than ten cells of type A in a single t_1 and t_2 dataset, see below) type A at time t_2 should, for the most part, stem from cells c_1^i of type A at time t_1. This means that for every cell at t_2 of type A, the t_1 cell with the highest coupling should also be of type A. In other words, for every cell at t_2 we have both a real cell type and a "predicted" cell type: the cell type of its maximally coupled t_1 ancestor. We then calculate the accuracy of this prediction and select the combination of hyperparameters that leads to the highest accuracy. For moslin, the maximal cell type persistence accuracy is 0.793 ($\alpha = 0.01$, $\epsilon = 0.05$, $\tau_a = 0.4$), for W, 0.785 ($\epsilon = 0.01$, $\tau_a = 0.4$), and for LineageOT, 0.406 ($\epsilon = 0.01$).

We calculate confidence intervals on these accuracies in two ways. For Fig. 4c, we calculate the standard error of the mean accuracy over all dataset combinations, weighted by the number of cells w_i in the N_{t_2} t_2 datasets:

$$\underset{-}{\sigma} = \sigma \sqrt{\sum_{i=1}^{N_{t_2}} w'_i},$$

where $w'_i = \frac{w_i}{\sum_{i=1}^{N_{t_2}} w_i}$ the normalized weights. For Supp. Figure 14a, we subsampled 25% of the total number of t_2 cells 100 times and calculated the 95% confidence interval based on that.

We call a cell type non-transient if it has more than ten cells in at least one dataset at t_1 and t_2. In particular, that means that the cell type does not have to be present in all t_1 datasets. However, we feel this constitutes a conservative but fair way of testing persistence: due to sampling noise, low-frequent cell types can be missed in single cell sequencing, and without thorough system knowledge it is hard to distinguish between low-frequency and absent cell types.

Transient fibroblast test We used transient fibroblasts to compare method performances on non-equal cell types. The transient col12a1a fibroblasts are of epicardial origin and could originate from the epicardium (ventricle), epicardium (atrium), fibroblasts (const.), fibroblasts (cfd), fibroblasts (cxcl12a), and fibroblasts (proliferating) cell types, and the transient nppc fibroblasts are of endocardial origin and could originate from the endocardium (ventricle), endocardium (atrium), and fibroblasts (spock3) cell types.

We tested the significance of the performance difference between moslin and W by calculating the difference in ground truth ancestors per dataset combination, and then

used a t-test to calculate the significance of the mean moslin performance being greater than the mean W performance.

Calculating cellular flows Given a coupling P_{ik} between a t_1 dataset a and a t_2 dataset b, cell type transitions from type A to type B can be quantified as

$$P_{AB}^{ab} = \sum_{i \in A, k \in B} P_{ik}^{ab},$$

which satisfies $\sum_{AB} P_{AB} = 1$ since $\sum_{ik}^{N,M} P_{ik} = 1$. We construct weighted averages of these cell type transitions over all dataset combinations, weighing by the product of $\#a$ and $\#b$, with $\#a$ the number of cells in a:

$$\widetilde{P}_{AB} := \sum_{ab} \left(P_{AB}^{ab} \frac{\#a * \#b}{\sum_a \#a * \sum_b \#b} \right).$$

This definition satisfies $\sum_{AB} \widetilde{P}_{AB} = 1$.

We similarly obtain cell type frequencies at every time point by a weighted average of cell type frequencies f_A^a in each dataset a with weights $\#a$:

$$\widetilde{f}_A := \sum_a \frac{\#a}{\sum_a \#a} f_A^a.$$

Again, $\sum_A \widetilde{f}_A = 1$ since $\sum_A f_A^a = 1$ for each a.

To calculate the proportion s_{AB} of cells of type A becoming cells of type B, we divide $\widetilde{P}_{AB}$ by the total mass outgoing from A:

$$s_{AB} := \frac{\widetilde{P}_{AB}}{\sum_C \widetilde{P}_{AC}},$$

while the proportion t_{AB} of cells type B being generated by cells of type A is similarly calculated as

$$t_{AB} = \frac{\widetilde{P}_{AB}}{\sum_C \widetilde{P}_{CB}}.$$

Finally, we subsampled the datasets used to calculate the proportions s_{AB}, and then used the range of obtained values to determine confidence intervals. To reduce the amount of data roughly by half, we randomly selected three out of four control datasets, six out of nine 3 dpi datasets, and five out of seven 7 dpi datasets, meaning 18 instead of 36 couplings between control and 3 dpi datasets, and 30 instead of 63 couplings between 3 and 7 dpi datasets. This method of random selection allows for a total of 7056 combinations:

$$\binom{4}{3} * \binom{9}{6} * \binom{7}{5} = 7056.$$

We explicitly calculated s_{AB} for all 7056 combinations to determine confidence intervals.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-024-03422-4.

Additional file 1: Supplementary figures.

Additional file 2: Supplementary note describing the moslin algorithm in detail.

Additional file 3: Supplementary table containing TFs from the *C. elegans* analysis.

Additional file 4: Review history.

Acknowledgements
We would like to thank Adan Forrow for helpful discussions about LineageOT, Xinhai Pan as well as Xiuwei Zhang for advice regarding their simulation tool, TedSim, Shou-Wen Wang for clarifications regarding CoSpar, and Marco Cuturi for input concerning optimal transport. We would further like to thank Matthew G. Jones, Manuel Gander, Philipp Weiler, and all members of Nitzan, Theis, and Treutlein labs for the great discussions.

Review history
The review history is available as Additional File 4.

Peer review information
Veronique van den Berghe and Kevin Pang were the primary editors of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Authors' contributions
M.L. conceived the project, guided by F.J.T. and M.N. J.P.J., F.J.T., and M.N. supervised the research. M.L. designed the algorithm with contributions by Z.P. and M.K., and guided by M.N. M.K. implemented the algorithm, with contributions by D.K. M.K. and Z.P. benchmarked the method across datasets. Z.P. conducted the simulations studies, with contributions by M.K. M.L. analyzed the C. elegans data, with contributions by M.K. B.S. analyzed the zebrafish data, with contributions by Z.P. M.L., Z.P., B.S., F.J.T., and M.N. wrote the manuscript. All authors read and approved the final manuscript.

Authors' X handles
X handles: @MariusLange8 (Marius Lange); @zoe_piran (Zoe Piran); @fabian_theis (Fabian J. Theis); @mor_nitzan (Mor Nitzan).

Funding
This work was supported by the BMBF (grant No. 01IS18036B and grant No. 01IS18053A), by the Helmholtz Association (Incubator grant sparse2big, grant No. ZT-I-0007), by the European Union (M.N.: ERC, DecodeSC, 101040660; J.P.J.: ERC, 715361; F.J.T.: ERC, DeepCell, 101054957), by the Wellcome Trust, Wellcome Leap, Delta Tissue [9E8E84F7-8991-4D4A-A9EC], by the DZHK (German Centre for Cardiovascular Research), by the Israel Science Foundation (grant No. 1079/21), and by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A532B, 031A533A, 031A533B, 031A534A, 031A535A, 031A537A, 031A537B, 031A537C, 031A537D, 031A538A). M. Lange further acknowledges financial support by the DFG through the Graduate School of QBM (GSC 1006), by the Joachim Herz Foundation, and through an EMBO Postdoctoral Fellowship. Z.P. is supported by a scholarship for outstanding doctoral students in data-science by the Israeli council for higher education and the Clore Scholarship for Ph.D. students. M.N. is supported by an Early Career Faculty Fellowship by the Azrieli Foundation. For all support coming via EU funding, the views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Data availability
Raw published data for the ***C. elegans*** [7] and zebrafish [32] examples are available from the Gene Expression Omnibus under accession codes GSE126954 [131] and GSE159032 [132], respectively. Processed data is available from figshare [133] (https://doi.org/https://doi.org/10.6084/m9.figshare.c.6533377.v1). The moslin software is implemented in Python and can be freely accessed via GitHub [134] (https://github.com/theislab/moslin) and Zenodo [135] (https://zenodo.org/records/13890587), including documentation, tutorials, and examples. Jupyter notebooks and Python scripts to reproduce our results are available via the same GitHub repository. Moslin is released under a BSD-3 Clause License.

Declarations

Ethics approval and consent to participate
Not applicable. Ethical approval was not needed for the study as we relied on publicly available datasets.

Consent for publication
Not applicable. No private, confidential, or sensitive information pertaining to individuals was utilized.

Competing interests
F.J.T. consults for Immunai Inc., Singularity Bio B.V., CytoReason Ltd, Cellarity, and Omniscope Ltd, and has ownership interest in Dermagnostix GmbH and Cellarity. The remaining authors declare no competing interests.

Author details
[1]Department of Biosystems Science and Engineering, ETH Zürich, Basel, Switzerland. [2]Department of Mathematics, Technical University of Munich, Munich, Germany. [3]Institute of Computational Biology, Helmholtz Center Munich,

Munich, Germany. [4]School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel. [5]Apple, Cupertino, USA. [6]Berlin Institute for Medical Systems Biology, Max Delbrück Center for Molecular Medicine, Berlin, Germany. [7]Department of Paediatric Oncology/Hematology, Charité-Universitätsmedizin Berlin, Berlin, Germany. [8]Charité-Universitätsmedizin Berlin, Berlin, Germany. [9]DZHK (German Centre for Cardiovascular Research), Partner Site Berlin, Berlin, Germany. [10]TUM School of Life Sciences Weihenstephan, Technical University of Munich, Munich, Germany. [11]Racah Institute of Physics, The Hebrew University of Jerusalem, Jerusalem, Israel. [12]Faculty of Medicine, The Hebrew University of Jerusalem, Jerusalem, Israel.

Received: 29 February 2024 Accepted: 10 October 2024
Published online: 21 October 2024

References

1. Schiebinger G, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. Cell. 2019;176:1517.
2. Fischer DS, et al. Inferring population dynamics from single-cell RNA-sequencing time series data. Nat Biotechnol. 2019;37:461–8.
3. Tong A, Huang J, Wolf G, van Dijk D, Krishnaswamy S. TrajectoryNet: a dynamic optimal transport network for modeling cellular dynamics. Proc Mach Learn Res. 2020;119:9526–36.
4. Guan J, et al. Chemical reprogramming of human somatic cells to pluripotent stem cells. Nature. 2022;605:325–31.
5. Pijuan-Sala B, et al. A single-cell molecular map of mouse gastrulation and early organogenesis. Nature. 2019;566:490–5.
6. Guibentif C, et al. Diverse routes toward early somites in the mouse embryo. Dev Cell. 2021;56:141–153.e6.
7. Packer JS, et al. A lineage-resolved molecular atlas of C. elegans embryogenesis at single-cell resolution. Science. 2019;365:eaax1971. https://doi.org/10.1126/science.aax1971. Preprint at.
8. Wagner DE, Klein AM. Lineage tracing meets single-cell omics: opportunities and challenges. Nat Rev Genet. 2020;21:410–27.
9. Biddy BA, et al. Single-cell mapping of lineage and identity in direct reprogramming. Nature. 2018;564:219–24.
10. Spanjaard B, et al. Simultaneous lineage tracing and cell-type identification using CRISPR–Cas9-induced genetic scars. Nat Biotechnol. 2018;36:469–73.
11. Alemany A, Florescu M, Baron CS, Peterson-Maduro J, van Oudenaarden A. Whole-organism clone tracing using single-cell sequencing. Nature. 2018;556:108–12.
12. Raj B, et al. Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. Nat Biotechnol. 2018;36:442–50.
13. Chan MM, et al. Molecular recording of mammalian embryogenesis. Nature. 2019;570:77–82.
14. Prasad N, Yang K, Uhler C. Optimal transport using GANs for lineage tracing. arXiv preprint arXiv:2007.12098. 2020.
15. Wang SW, Herriges MJ, Hurley K, Kotton DN, Klein AM. CoSpar identifies early cell fate biases from single-cell transcriptomic and lineage information. Nat Biotechnol. 2022;40(7):1066–74.
16. Jones MG, et al. Inference of single-cell phylogenies from lineage tracing data using Cassiopeia. Genome Biol. 2020;21:92.
17. Gong W, et al. Benchmarked approaches for reconstruction of in vitro cell lineages and in silico models of C. elegans and M. musculus developmental trees. Cell Syst. (2021). https://doi.org/10.1016/j.cels.2021.05.008.
18. Konno N, et al. Deep distributed computing to reconstruct extremely large lineage trees. Nat Biotechnol. 2022;40:566–75.
19. Weinreb C, Klein AM. Lineage reconstruction from clonal correlations. Proc Natl Acad Sci U S A. 2020;117:17041–8.
20. Wagner DE, et al. Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo. Science. 2018;360:981–7.
21. Forrow A, Schiebinger G. LineageOT is a unified framework for lineage tracing and trajectory inference. Nat Commun. 2021;12:4940.
22. Peyré G, Cuturi M, Solomon J. Gromov-Wasserstein averaging of kernel and distance matrices. in Proceedings of the 33rd international conference on machine learning (eds. Balcan, M. F. & Weinberger, K. Q.) vol. 48 2664–2672 (PMLR, New York, New York, USA, 2016).
23. Weiler P, Lange M, Klein M, Pe'er D, Theis F. CellRank 2: unified fate mapping in multiview single-cell data. Nat Methods. (2024).https://doi.org/10.1038/s41592-024-02303-9.
24. Haghverdi L, Ludwig LS. Single-cell multi-omics and lineage tracing to dissect cell fate decision-making. Stem Cell Reports. 2023;18:13–25.
25. VanHorn S, Morris SA. Next-generation lineage tracing and fate mapping to interrogate development. Dev Cell. 2021;56:7–21.
26. Quinn JJ, Jones MG, Okimoto RA, Nanjo S, Chan MM, Yosef N, et al. Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts. Science. 2021;371(6532):eabc1944.
27. Hughes NW, et al. Machine-learning-optimized Cas12a barcoding enables the recovery of single-cell lineages and transcriptional profiles. Mol Cell. 2022;82:3103–3118.e8.
28. Li L, et al. A mouse model with high clonal barcode diversity for joint lineage, transcriptomic, and epigenomic profiling in single cells. Cell. 2023. https://doi.org/10.1016/j.cell.2023.09.019.
29. Bowling S, et al. An engineered CRISPR-Cas9 mouse line for simultaneous readout of lineage histories and gene expression profiles in single cells. Cell. 2020;181:1410–1422.e27.
30. Ludwig LS, et al. Lineage tracing in humans enabled by mitochondrial mutations and single-cell genomics. Cell. 2019;176:1325–1339.e22.

31. Miller TE, et al. Mitochondrial variant enrichment from high-throughput single-cell RNA sequencing resolves clonal populations. Nat Biotechnol. 2022. https://doi.org/10.1038/s41587-022-01210-8.
32. Hu B, et al. Origin and function of activated fibroblast states during zebrafish heart regeneration. Nat Genet. 2022;54:1227–37.
33. Bendall SC, et al. Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. Cell. 2014;157:714–25.
34. Setty M, et al. Wishbone identifies bifurcating developmental trajectories from single-cell data. Nat Biotechnol. 2016;34:637–45.
35. Haghverdi L, Büttner M, Wolf FA, Buettner F, Theis FJ. Diffusion pseudotime robustly reconstructs lineage branching. Nat Methods. 2016;13:845–8.
36. Trapnell C, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat Biotechnol. 2014;32:381–6.
37. Wolf FA, et al. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. Genome Biol. 2019;20:59.
38. Villani C. Optimal transport: old and new (Vol. 338). Berlin: Springer; 2009. p. 23.
39. Peyré G, Cuturi M. Computational optimal transport Preprint at. 2019. https://doi.org/10.1561/9781680835519.
40. Vayer T, Chapel L, Flamary R, Tavenard R, Courty N. Fused Gromov-Wasserstein distance for structured objects. Algorithms. 2020;13:212.
41. Cuturi M. Sinkhorn distances: lightspeed computation of optimal transport. Adv Neural Inf Process Syst. 2013;26.
42. Genevay A, Chizat L, Bach F, Cuturi M, Peyré G. Sample complexity of sinkhorn divergences. in Proceedings of the twenty-second international conference on artificial intelligence and statistics (eds. Chaudhuri, K. & Sugiyama, M.) vol. 89 1574–1583 (PMLR, 16--18 Apr 2019).
43. Nitzan M, Karaiskos N, Friedman N, Rajewsky N. Gene expression cartography. Nature. 2019;576:132–7.
44. Zeira R, Land M, Strzalkowski A, Raphael BJ. Alignment and integration of spatial transcriptomics data. Nat Methods. 2022;19:567–75.
45. Liu X, Zeira R, Raphael BJ. PASTE2: partial alignment of multi-slice spatially resolved transcriptomics data. bioRxiv. (2023). https://doi.org/10.1101/2023.01.08.523162.
46. Demetci P, Santorella R, Sandstede B, Noble WS, Singh R. SCOT: single-cell multi-omics alignment with optimal transport. J Comput Biol. 2022;29:3–18.
47. Cang Z, Nie Q. Inferring spatial and signaling relationships between cells from single cell transcriptomic data. Nat Commun. 2020;11:2084.
48. Klein D, et al. Mapping cells through time and space with moscot. bioRxiv. 2023. 2023.05.11.540374. https://doi.org/10.1101/2023.05.11.540374.
49. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods. 2018;15:1053–8.
50. Chizat L, Peyré G, Schmitzer B, Vialard FX. Scaling algorithms for unbalanced optimal transport problems. Math Comput. 2018;87(314):2563–609.
51. Séjourné, Vialard & Peyré. The unbalanced Gromov Wasserstein distance: conic formulation and relaxation. Adv Neural Inf Process Syst.
52. Pan X, Li H, Zhang X. TedSim: temporal dynamics simulation of single-cell RNA sequencing data and cell division history. Nucleic Acids Res. 2022. https://doi.org/10.1093/nar/gkac235.
53. Sulston JE, Horvitz HR. Post-embryonic cell lineages of the nematode. Caenorhabditis elegans Dev Biol. 1977;56:110–56.
54. Sulston JE, Schierenberg E, White JG, Thomson JN. The embryonic cell lineage of the nematode Caenorhabditis elegans. Dev Biol. 1983;100:64–119.
55. Riddle DL, Blumenthal T, Meyer BJ, Priess JR. Specification of cell fates in the AB lineage. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press; 1997.
56. McInnes L, Healy J, Melville J. Umap: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426. 2018.
57. Setty M, et al. Characterization of cell fate probabilities in single-cell data with Palantir. Nat Biotechnol. 2019;37:451–60.
58. Aquino-Nunez W, et al. cnd-1/NeuroD1 functions with the homeobox gene ceh-5/Vax2 and hox gene ceh-13/labial to specify aspects of RME and DD neuron fate in Caenorhabditis elegans. G3. 2020;10:3071–85.
59. Hallam S, Singer E, Waring D, Jin YT, The C. elegans NeuroD homolog cnd-1 functions in multiple aspects of motor neuron fate specification. Development. 2000;127:4239–52.
60. Wightman B, Ebert B, Carmean N, Weber K, Clever S. The C. elegans nuclear receptor gene fax-1 and homeobox gene unc-42 coordinate interneuron identity by regulating the expression of glutamate receptor subunits and other neuron-specific genes. Dev Biol. 2005;287:74–85.
61. Clark SG, Chiu CC. elegans ZAG-1, a Zn-finger-homeodomain protein, regulates axonal development and neuronal differentiation. Development. 2003;130:3781–94.
62. Ramakrishnan K, Okkema PG. Regulation of C. elegans neuronal differentiation by the ZEB-family factor ZAG-1 and the NK-2 homeodomain factor CEH-28. PLoS One. 2014;9:e113893.
63. Wacker I, Schwarz V, Hedgecock EM, Hutter H. zag-1, a Zn-finger homeodomain transcription factor controlling neuronal differentiation and axon outgrowth in C. elegans. Development. 2003;130:3795–805.
64. Tucker DK, Adams CS, Prasad G, Ackley BD. The immunoglobulin superfamily members syg-2 and syg-1 regulate neurite development in C. elegans. J Dev Biol. 2022;10(1):3.
65. Shen K, Bargmann CI. The immunoglobulin superfamily protein SYG-1 determines the location of specific synapses in C. elegans. Cell. 2003;112:619–30.
66. Shen K, Fetter RD, Bargmann CI. Synaptic specificity is generated by the synaptic guidepost protein SYG-2 and its receptor, SYG-1. Cell. 2004;116:869–81.

67. Maro GS, et al. MADD-4/punctin and neurexin organize C. elegans GABAergic postsynapses through neuroligin. Neuron. 2015;86:1420–32.
68. Platsaki S, et al. The Ig-like domain of punctin/MADD-4 is the primary determinant for interaction with the ectodomain of neuroligin NLG-1. J Biol Chem. 2020;295:16267–79.
69. Seetharaman A, et al. MADD-4 is a secreted cue required for midline-oriented guidance in Caenorhabditis elegans. Dev Cell. 2011;21:669–80.
70. Buntschuh I, et al. FLP-1 neuropeptides modulate sensory and motor circuits in the nematode Caenorhabditis elegans. PLoS ONE. 2018;13:e0189320.
71. Gulati GS, et al. Single-cell transcriptional diversity is a hallmark of developmental potential. Science. 2020;367:405–11.
72. Weiler P, Lange M, Klein M, Pe'er D, Theis, F. J. Unified fate mapping in multiview single-cell data. bioRxiv. 2023. 2023.07.19.549685. https://doi.org/10.1101/2023.07.19.549685.
73. Shen WK, et al. AnimalTFDB 4.0: a comprehensive animal transcription factor database updated with variation and expression annotations. Nucleic Acids Res. 2023;51:D39–45.
74. González-Barrios M, et al. Cis- and trans-regulatory mechanisms of gene expression in the ASJ sensory neuron of Caenorhabditis elegans. Genetics. 2015;200:123–34.
75. Wallace SW, Singhvi A, Liang Y, Lu Y, Shaham S. PROS-1/prospero is a major regulator of the glia-specific secretome controlling sensory-neuron shape and function in C. elegans. Cell Rep. 2016;15:550–62.
76. Kage-Nakadai E, et al. Caenorhabditis elegans homologue of Prox1/prospero is expressed in the glia and is required for sensory behavior and cold tolerance. Genes Cells. 2016;21:936–48.
77. Kolotuev I, Hyenne V, Schwab Y, Rodriguez D, Labouesse M. A pathway for unicellular tube extension depending on the lymphatic vessel determinant Prox1 and on osmoregulation. Nat Cell Biol. 2013;15:157–68.
78. Wang J, et al. The C. elegans COE transcription factor UNC-3 activates lineage-specific apoptosis and affects neurite growth in the RID lineage. Development. 2015;142:1447–57.
79. Prasad B, Karakuzu O, Reed RR, Cameron S. unc-3-dependent repression of specific motor neuron fates in Caenorhabditis elegans. Dev Biol. 2008;323:207–15.
80. Reilly MB, Cros C, Varol E, Yemini E, Hobert O. Unique homeobox codes delineate all the neuron classes of C. elegans. Nature. 2020;584:595–601.
81. Hobert O. A map of terminal regulators of neuronal identity in Caenorhabditis elegans. Wiley Interdiscip Rev Dev Biol. 2016;5:474–98.
82. Mansfeld J, et al. Branched-chain amino acid catabolism is a conserved regulator of physiological ageing. Nat Commun. 2015;6:10043.
83. Poss KD, Wilson LG, Keating MT. Heart regeneration in Zebrafish. Science. 2002;298:2188–90. https://doi.org/10.1126/science.1077857. Preprint at.
84. Schnabel K, Wu C-C, Kurth T, Weidinger G. Regeneration of cryoinjury induced necrotic heart lesions in zebrafish is associated with epicardial activation and cardiomyocyte proliferation. PLoS ONE. 2011;6:e18503.
85. González-Rosa JM, Martín V, Peralta M, Torres M, Mercader N. Extensive scar formation and regression during heart regeneration after cryoinjury in zebrafish. Development. 2011;138:1663–74.
86. Chablais F, Veit J, Rainer G, Jaźwińska A. The zebrafish heart regenerates after cryoinjury-induced myocardial infarction. BMC Dev Biol. 2011;11:21.
87. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19:15.
88. Virshup I, et al. The scverse project provides a computational ecosystem for single-cell omics data analysis. Nat Biotechnol. 2023. https://doi.org/10.1038/s41587-023-01733-8.
89. Kamimoto K, et al. Dissecting cell identity via network inference and in silico gene perturbation. Nature. 2023;614:742–51.
90. Fleck JS, et al. Inferring and perturbing cell fate regulomes in human brain organoids. Nature. 2022. https://doi.org/10.1038/s41586-022-05279-8.
91. Lotfollahi, M. et al. Learning interpretable cellular responses to complex perturbations in high-throughput screens. bioRxiv. 2021. 2021.04.14.439903. https://doi.org/10.1101/2021.04.14.439903.
92. Lareau CA, et al. Massively parallel single-cell mitochondrial DNA genotyping and chromatin profiling. Nat Biotechnol. 2021;39:451–61.
93. Weng C, et al. Deciphering cell states and genealogies of human hematopoiesis. Nature. 2024. https://doi.org/10.1038/s41586-024-07066-z.
94. Rodriguez-Fraticelli A, Morris SA. In preprints: the fast-paced field of single-cell lineage tracing. Development. 2022;149(11):dev200877.
95. Choi J, et al. A time-resolved, multi-symbol molecular recorder via sequential genome editing. Nature. 2022;608:98–107.
96. Choi J, et al. Precise genomic deletions using paired prime editing. Nat Biotechnol. 2022;40:218–26.
97. Loveless TB, et al. Lineage tracing and analog recording in mammalian cells by single-site DNA writing. Nat Chem Biol. 2021;17:739–47.
98. Mimitou EP, et al. Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. Nat Biotechnol. 2021. https://doi.org/10.1038/s41587-021-00927-2.
99. Jindal K, et al. Multiomic single-cell lineage tracing to dissect fate-specific gene regulatory programs. bioRxiv. 2022. 2022.10.23.512790. https://doi.org/10.1101/2022.10.23.512790.
100. Ma S, et al. Chromatin potential identified by shared single-cell profiling of RNA and chromatin. Cell. 2020;183:1103–1116.e20.
101. Kartha VK, Duarte FM, Hu Y, Ma S, Chew JG, Lareau CA, et al. Functional inference of gene regulation using single-cell multi-omics. Cell Genom. 2022;2(9).
102. Chow KHK, Budde MW, Granados AA, Cabrera M, Yoon S, Cho S, et al. Imaging cell lineage with a synthetic digital recording system. Science. 2021;372(6538):eabb3099.

103. Frieda KL, et al. Synthetic recording and in situ readout of lineage information in single cells. Nature. 2017;541:107–11.
104. Chadly, D. M. et al. Reconstructing cell histories in space with image-readable base editor recording. bioRxiv. 2024. https://doi.org/10.1101/2024.01.03.573434.
105. Yang D, et al. Lineage tracing reveals the phylodynamics, plasticity, and paths of tumor evolution. Cell. 2022;185:1905–1923.e25.
106. Seidel S, Stadler T. TiDeTree: a Bayesian phylogenetic framework to estimate single-cell trees and population dynamic parameters from genetic lineage tracing data. Proc Biol Sci. 2022;289:20221844.
107. Zafar H, Lin C, Bar-Joseph Z. Single-cell lineage tracing by integrating CRISPR-Cas9 mutations with transcriptomic data. Nat Commun. 2020;11:3055.
108. Baron CS, van Oudenaarden A. Unravelling cellular relationships during development and regeneration using genetic lineage tracing. Nat Rev Mol Cell Biol. 2019;20:753–65.
109. Moreno-Ayala R, Junker JP. Single-cell genomics to study developmental cell fate decisions in zebrafish. Brief Funct Genomics. 2021. https://doi.org/10.1093/bfgp/elab018.
110. Olivares-Chauvet P, Junker JP. Inclusion of temporal information in single cell transcriptomics. Int J Biochem Cell Biol. 2020;122:105745.
111. Hurley K, et al. Reconstructed single-cell fate trajectories define lineage plasticity windows during differentiation of human PSC-derived distal lung progenitors. Cell Stem Cell. 2020;26:593–608.e8.
112. Weinreb C, Rodriguez-Fraticelli A, Camargo FD, Klein AM. Lineage tracing on transcriptional landscapes links state to fate during differentiation. Science. 2020;367(6479):eaaw3381.
113. Penter L, et al. Longitudinal single-cell dynamics of chromatin accessibility and mitochondrial mutations in chronic lymphocytic leukemia mirror disease history. Cancer Discov. 2021. https://doi.org/10.1158/2159-8290.CD-21-0276.
114. Bunne C, et al. Learning single-cell perturbation responses using neural optimal transport. Nat Methods. 2023;20:1759–68.
115. Tong AY, Huguet G, Natik A, MacDonald K, Kuchroo M, Coifman R, et al. Diffusion earth mover's distance and distribution embeddings. In International Conference on Machine Learning: PMLR; 2021. p. 10336–46.
116. Chen WS, et al. Uncovering axes of variation among single-cell cancer specimens. Nat Methods. 2020;17:302–10.
117. Cuturi M, Meng-Papaxanthos L, Tian Y, Bunne C, Davis G, Teboul O. Optimal transport tools (ott): a jax toolbox for all things wasserstein. arXiv preprint arXiv:2201.12324. 2022.
118. Frostig R, Johnson M, Leary C. Compiling machine learning programs via high-level tracing. 2018.
119. Lin H, Schagat T. Neuroblasts: a model for the asymmetric division of stem cells. Trends Genet. 1997;13:33–9.
120. Morrison SJ, Kimble J. Asymmetric and symmetric stem-cell divisions in development and cancer. Nature. 2006;441:1068–74.
121. Knoblich JA. Mechanisms of asymmetric stem cell division. Cell. 2008;132:583–97.
122. Hashimshony T, Feder M, Levin M, Hall BK, Yanai I. Spatiotemporal transcriptomics reveals the evolutionary history of the endoderm germ layer. Nature. 2015;519:219–22.
123. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol. 2015;33:495–502.
124. Flamary R, et al. POT: Python optimal transport. J Mach Learn Res. 2021;22:1–8.
125. Lange M, et al. Cell Rank for directed single-cell fate mapping. Nat Methods. 2022;19:159–70.
126. La Manno G, et al. RNA velocity of single cells. Nature. 2018;560:494–8.
127. Bergen V, Lange M, Peidli S, Wolf FA, Theis FJ. Generalizing RNA velocity to transient cell states through dynamical modeling. Nat Biotechnol. 2020;38:1408–14.
128. Reuter B, Fackeldey K, Weber M. Generalized Markov modeling of nonreversible molecular kinetics. J Chem Phys. 2019;150:174103.
129. Reuter B, Weber M, Fackeldey K, Röblitz S, Garcia ME. Generalized Markov state modeling method for nonequilibrium biomolecular dynamics: exemplified on amyloid β conformational dynamics driven by an oscillating electric field. J Chem Theory Comput. 2018;14:3579–94.
130. van Dijk D, et al. Recovering gene interactions from single-cell data using data diffusion. Cell. 2018;174:716–729.e27.
131. Packer JS, et al. A lineage-resolved molecular atlas of C. elegans embryogenesis at single cell resolution. Datasets. Gene Expression Omnibus. 2019. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE126954.
132. Hu B, et al. Cellular drivers of injury response and regeneration in the adult zebrafish heart. Datasets. Gene Expression Omnibus. 2022. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE159032.
133. Lange M, Piran Z, Klein M, Spanjaard B, Klein D, Junker JP, Theis FJ, Nitzan M. Mapping lineage-traced cells across time points with moslin. Figshare. 2024. https://doi.org/10.6084/m9.figshare.c.6533377.v1.
134. Lange M, Piran Z, Klein M, Spanjaard B, Klein D, Junker JP, Theis FJ, Nitzan M. Mapping lineage-traced cells across time points with moslin. GitHub. 2024. https://github.com/theislab/moslin.
135. Lange M, Piran Z, Klein M, Spanjaard B, Klein D, Junker JP, Theis FJ, Nitzan M. Mapping lineage-traced cells across time points with moslin. Zenodo. 2024. https://doi.org/10.5281/zenodo.13890586.

Publisher's Note