Mapping lineage-traced cells across time points with moslin

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

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

* Equal contribution
† Work partially done while at Helmholtz Center Munich, Germany
+ Corresponding authors: fabian.theis@helmholtz-muenchen.de and mor.nitzan@mail.huji.ac.il

Abstract

Simultaneous profiling of single-cell gene expression and lineage history holds enormous potential for studying cellular decision-making beyond simpler pseudotime-based approaches. However, it is currently unclear how lineage and gene expression information across experimental time points can be combined in destructive experiments, which is particularly challenging for in-vivo systems. Here we present moslin, a Fused Gromov-Wasserstein-based model to couple matching cellular profiles across time points. In contrast to existing methods, moslin leverages both intra-individual lineage relations and inter-individual gene expression similarity. We demonstrate on simulated and real data that moslin outperforms state-of-the-art approaches that use either one or both data modalities, even when the lineage information is noisy. On *C. elegans* embryonic development, we show how moslin, combined with trajectory inference methods, 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. We anticipate moslin to play a crucial role in deciphering complex state change trajectories from lineage-traced single-cell data.

Introduction

Important biological processes like development, disease, or regeneration play out as complex changes on 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 been combined with heritable barcodes that link clonally related cells 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 previous methods used this setting to relate cells across time points clonally[14,15]. However, such strategies do not generalize to in-vivo lineage-traced systems, as each time point corresponds to a different individual, and barcodes are not comparable across individuals. Most current analysis strategies[13,16–20] remain limited to analyzing isolated lineage-traced time points. Thus, they do not embed lineage relationships in the temporal context of cellular state changes.

While a previous method, LineageOT[21], represented an important step towards mapping lineage-traced cells, it cannot relate lineage information across time points and includes it only in the later time point. Further, the tool has only been demonstrated on simulated examples or examples with known ground truth. Thus, the comprehensive integration of lineage and gene expression information 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. To the best of our knowledge, moslin is the first method to use lineage information at two or more time points and to include the effects of cellular growth and stochastic cell sampling. Our approach outperforms LineageOT and optimal transport (OT)-baselines on simulated data where ground truth is available. Further, on *Caenorhabditis (C.) elegans* embryogenesis, we combine moslin with CellRank[22], a trajectory inference framework, to uncover differentiation trajectories and putative decision-driver genes. Finally, in zebrafish heart regeneration, we predict lineage relationships between recently discovered activated fibroblast states that emerge after injury using moslin. 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[8,23,24] (scLT) studies. Using gene expression and lineage information, moslin computes probabilistic mappings between cells in early- and late time points. Moslin distinguishes itself from previous approaches[21] by incorporating lineage information at both time points to guide the inference process. Using the computed mapping, we infer ancestor and descendant probabilities for rare or transient cell states and interface with CellRank[22] 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-induced scars[10–13,25] and naturally-occurring mutations[26,27]. We refer to the entirety of any such genomic lineage information in a single cell as a “barcode” and stress that moslin is applicable to any kind of barcode information.

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. While barcodes can be directly compared within one individual to estimate lineage trees[10–13,16,18,19], 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 or gene expression within or 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 major assumptions: (i) cells change their molecular state gradually, and (ii) lineage distances are, on average, conserved between time points. 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[28,29] (OT), which allows us to include both assumptions into a single cost function (Fig. 1c, Methods, and Supplementary Note 1).

The first assumption forms the basis of many successful pseudotime algorithms[1,30–34]; we include it 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). The second assumption implies a type of lineage concordance: cell pairs at t_1 should be mapped to cell pairs at t_2 with similar relative lineage distances. We include this assumption in moslin using a Gromov-Wasserstein[35] (GW)-term (Methods and Supplementary Note 1). 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).

We balance both terms with an α parameter between 0 and 1, corresponding to W and GW terms, respectively[36]. 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[29,37,38]. Thus, moslin solves a Fused Gromov-Wasserstein[36] (FGW) problem with hyperparameters α and ϵ (Fig. 1c, Methods, and Supplementary Note 1).

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[39]. 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. 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 according to C^X and C^Y, using the W and GW terms, respectively. For each t_1 cell i, the vector $P_{i,:}$ quantifies lineage and state-informed transition probabilities towards any t_2 cell j. Finally, we use the coupling matrix P to compute ancestor and descendant probabilities[1] directly in moslin and pass it to CellRank[22] for further analysis.

Following previous successful 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[29,40,41], which accounts for uncertain growth and death rates, as well as for stochastic cell sampling (Methods).

Moslin accurately reconstructs simulated trajectories

We assess moslin's performance on two simulated datasets. As an initial verification, we consider simulated single-cell transcriptome trajectories using a setting suggested by Forrow et al.[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 transcriptomically closer to early cells that are not their ancestors (Fig. 2a, see ref.[21]).

We benchmark the performance of moslin against the only competing method, LineageOT[21], which uses lineage information only at the later time point. We also test two extreme cases of our moslin approach: (i) using only gene expression information in a W-term ($\alpha = 0$), and (ii) using only lineage information in a GW-term ($\alpha = 1$) (Methods, for moslin we perform a grid search to set the interpolation parameter α). We test all methods with two types of lineage-distance computation: (i) using the ground truth tree and (ii) using a fitted tree based on the simulated barcodes (Supplementary Fig. 1a,b). We perform a grid search for each case to find the optimal hyperparameters (Methods and Supplementary Fig. 1c,d). 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”.

In agreement with the original publication[21], we find that LineageOT improves over the baseline OT setting in seven of eight cases (Fig. 2b). Moslin further improves on LineageOT, with an average improvement of 10% and 12% in the mean error across all trajectories using true and fitted trees, respectively. Across all tested methods, moslin achieves the lowest mean error across all trajectory structures and distance variants. Of note, GW performs well using ground-truth tree distances, outperforming OT 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. Importantly, the authors of LineageOT[21] reported that their tree reconstruction was only moderately accurate, implying that moslin outperforms the baseline OT approach in a setting reminiscent of real scLT data. Thus, the interpolation between gene expression and lineage information allows our approach to achieve excellent performance on realistically fitted tree distances (Fig. 2b).

Next, we consider a more complex simulation using TedSim[42], which simulates cell division events from root to present-day 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 toward 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. 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 (Methods).

We find that moslin outperforms LineageOT across our range of *ssr* values. In particular, moslin with maximal *ssr* achieves lower mean error than LineageOT on noise-free barcodes (Fig. 2f). Critically, moslin can be used robustly even for relatively high ssr, while LineageOT fails and does not provide any mapping beyond a certain threshold (*ssr* > 0.2).

Mapping gene expression across *C. elegans* embryonic development

To showcase moslin's performance in a realistic setting where ground truth is still available, we consider *C. elegans* embryonic development. The adult animal consists of only 959 somatic cells[43,44], generated following a sequence of deterministic lineage decisions. This species' ground truth lineage tree is known[44] 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 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[45]. 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[7,45] (Supplementary Fig. 2a). The dataset consists of 6,476 ABpxp cells across 7 time points from 170-510 min past fertilization (Fig. 3a, Supplementary Fig. 2b-d and Methods).

We benchmark the performance of moslin and LineageOT across time points on the ABpxp lineage using a similar set-up as for the TedSim[42] data, and as suggested in ref.[21] (Fig. 2c-f). Specifically, we only provide lineage distances within time points to both methods. We compare predictions with ground-truth lineage relations across time points by calculating the mean prediction error over ancestor and descendant states (Methods). For all time point pairs, moslin outperforms LineageOT and achieves a lower mean error (Fig. 3b). These results generalize to another, distinct subset of *C. elegans* cells with precise lineage information (Supplementary Fig. 3).

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 Supplementary Fig. 4). 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 correctly links these cells to RIM neurons, while LineageOT predicts many erroneous connections with ASH (ciliated) neurons (Fig. 3c and Methods).

Going beyond a single pair of time points, we combine moslin's couplings across all time points to study *C. elegans* embryogenesis using CellRank[22], a computational fate mapping tool. Embryogenesis, especially at later stages, is a loosely synchronized process where embryos of similar ages can represent slightly different developmental stages. This holds in particular for our *C. elegans* example, where developmental time per cell was estimated by comparing to bulk expression data[7]. Thus, we expect to find cells of slightly different maturity stages within each assigned time point. To account for developmental asynchrony, CellRank 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 (Supplementary Fig. 5a,b).

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 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 (Supplementary Fig. 2d). Thus, we successfully 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 (Supplementary Fig. 5c,d).

We aggregate the seven terminal states into three groups: Ciliated neurons, Non-ciliated neurons, and Glia and excretory cells. Next, we use CellRank to compute fate probabilities towards these groups (Fig. 3e and Supplementary Fig. 6). In agreement with known biology, moslin/CellRank predicts most progenitors in the ABpxp lineage to transition towards Non-ciliated neurons[7] (Supplementary Fig. 7a,b). For each of the three terminal cell groups, predicted fate probabilities are significantly higher among cells from the corresponding known pre-terminal populations (Supplementary Fig. 7c and Methods). 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[46] (TFs), we automatically recover known drivers for each trajectory, including sptf-1 for Ciliated neurons[47], cnd-1 for Non-ciliated neurons[48,49], and pros-1 for Glia and excretory cells[50–52] (Fig. 3f, Supplementary Fig. 8, Supplementary Table 1 and Methods).

Finally, to study the temporal dynamics of fate decisions during *C. elegans* embryogenesis, we compute a pseudotime using Palantir[53], starting in a 170 min-cell (Supplementary Fig. 9a). As expected, pseudotime values increase across time points (Supplementary Fig. 9b). 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 fate probabilities to compute smooth expression trends (Methods and Supplementary Table 1). Sorting expression trends by their pseudotime-peak and plotting them in a heatmap reveals a sequential activation pattern (Fig. 3g and Supplementary Fig. 10). Our results show that some TFs with known function in Non-ciliated neuron generation, including cnd-1[48,49] or unc-3[54,55], are activated before others, including fax-1[56] and zag-1[57–59] (Supplementary Table 1). In particular, our activation pattern predicts that fax-1 is activated before flp-1, a known regulatory interaction in (non-ciliated) AVK cells[56].

While many moslin/CellRank 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[60,61]. While previous work[60] 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[62].

Moslin determines the dynamics of transient fibroblasts in heart regeneration

The zebrafish heart regenerates after injuries, such as ventricular resections[63] or cryoinjuries[64–66]. 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), three days after injury (3dpi), seven days after injury (7dpi) and thirty days after injury (30dpi). This dataset includes inferred lineage trees and cell type annotations for each time point[67] (Fig. 4a).

One key result from this study was the emergence of several transcriptomically 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 3dpi, 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, 3dpi and 7dpi. We embed the transcriptomic readout of all single cells with lineage information into a joint latent space using scVI[39], 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, using the interpolation parameter $\alpha = 0.5$.

Initially, we validate the performance of moslin 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 at 7dpi should be atrial endocardial cells at 3dpi. We use a Welch-test-based framework to test whether cells of type A at t_2 are significantly coupled to cells of type A at t_1, or to any other cell type B at t_2 (Fig. 4b and Methods). Under the expectation that a significant coupling of A at t_1 to A at t_2 is a true positive and a significant coupling to any other cell type is a false positive, we visualize receiver operating characteristic (ROC) curves for control-3dpi and 3dpi-7dpi couplings. Areas under the ROC curve (AUCs) of 0.992 and 0.984, respectively, show that moslin can be used to determine cell state relationships across time (Fig. 4c and Methods).

In this framework, we test a cell type A at t_1 against all cell types at t_2; all but one of these tests (namely, the one where we test A at t_1 against A at t_2) is supposed to yield a negative result. To ensure AUCs are not inflated by this high amount of true negatives, we also test whether cells of type A at t_2 are significantly coupled to cells of type A at t_1 or to the ensemble of other cells. Here we find AUCs of 0.9999 and 1 (Supplementary Fig. 11 and Methods). Finally, we find moslin's performance decreases by 3% (from AUC 0.99 to 0.96 at both time points) if ϵ is increased to 0.1; variations in other hyperparameters yield performance changes below 1% (Supplementary Fig. 11), showing that moslin is robust to hyperparameter changes.

We next investigate the origins of transient fibroblast substates, including col11a1a and col12a1a fibroblasts. In particular, the previously published analysis 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. We calculate couplings with moslin, take weighted averages of cell type frequencies over separate organisms, and aggregate couplings between cell types to quantify cell type transitions during regeneration (Supplementary Fig. 12 and Methods). As expected, we observe that persistent cell types couple strongly to themselves (Fig. 4d).

Furthermore, we observe that constitutive fibroblasts preferentially generate col11a1a fibroblasts, and that most col12a1a fibroblasts originate from col11a1a fibroblasts: 21% (95% confidence interval: 15-28%) of the mass generated by constitutive fibroblasts at control goes towards col11a1a fibroblasts, whereas only 12% (95% confidence interval: 6-19%) goes directly towards col12a1a fibroblasts. At 3dpi, 42% (95% confidence interval: 22-56%) of the mass generated by col11a1a fibroblasts goes towards col12a1a fibroblasts, which constitutes 39% (95% confidence interval: 26-47%) of the col12a1a fibroblast mass at 7dpi (Fig. 4e). 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. 4e). We hypothesize that the 3dpi col12a1a fibroblasts that seem to originate directly from constitutive fibroblasts have actually transitioned through a col11a1a fibroblast state between injury and 3dpi. Our findings demonstrate the added value of temporal lineage models like moslin in analyzing scLT time-course data.

Discussion

We demonstrate that combining intra-individual lineage similarity with inter-individual gene-expression similarity improves trajectory reconstruction substantially for in-vivo single-cell lineage-tracing (scLT) data. moslin outperforms competing methods on simulated and real data by interpolating between Wasserstein and Gromov-Wasserstein regimes and using lineage information at both time points. Crucially, we highlight in simulations that moslin compensates for noisy lineage relations through gene expression information, rendering our method suitable for real scLT data. We illustrate moslin's capability to recover cell-state trajectories from real scLT data in zebrafish heart regeneration[67], where we predict a new origin for regenerative activated fibroblast states. Importantly, moslin is the first computational method with demonstrated success in this challenging real data setting.

Moslin's key advantage over previous analysis paradigms for in-vivo scLT data is that it 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 characterising 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. Moslin now provides a unified framework for this identification from time-resolved single-cell lineage tracing studies.

Under the hood, moslin is based on moscot, a robust and easy-to-use framework for OT applications in single-cell genomics. As such, it benefits from moscot's interoperability with the scverse[68,69] ecosystem and can take advantage of future moscot improvements concerning scalability and usability. Moslin's interface with CellRank[22] grants it access to a range of established, constantly growing downstream-analysis functions. We demonstrate the power of combining moslin with CellRank 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[70,71], or to improve perturbation predictions[72].

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 only comparable within one time point. In contrast, for in-vitro experiments, cells from the same population can be sampled at different time points, rendering their lineage information directly compatible across time points. Previously, OT-like approaches[14,15] have been suggested for this clonal resampling experimental design[24]. Moslin could be extended towards this setting by adjusting the cost-matrix definition.

While moslin is robust to noise in lineage information, it will benefit from improved experimental lineage tracing technologies. Recent innovations, including mitochondrial lineage tracing[26,27,73] and base/prime editing[74–77], represent compelling use cases for moslin. 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[67], 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[78,79] to link molecular layers across time. For example, this could reveal how epigenetic changes manifest in altered gene expression dynamics[80,81]. Additionally, spatially-resolved lineage tracing data would enable moslin to regularise the coupling computation further using spatial neighbourhoods. 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 anticipate computational tools like moslin to play a crucial role in analyzing and interpreting novel lineage-traced datasets.

References

1. Schiebinger, G. *et al.* Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming. *Cell* **176**, 1517 (2019).
2. Fischer, D. S. *et al.* Inferring population dynamics from single-cell RNA-sequencing time series data. *Nat. Biotechnol.* **37**, 461–468 (2019).
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* **119**, 9526–9536 (2020).
4. Guan, J. *et al.* Chemical reprogramming of human somatic cells to pluripotent stem cells. *Nature* **605**, 325–331 (2022).
5. Pijuan-Sala, B. *et al.* A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature* **566**, 490–495 (2019).
6. Guibentif, C. *et al.* Diverse Routes toward Early Somites in the Mouse Embryo. *Dev. Cell* **56**, 141–153.e6 (2021).
7. Packer, J. S. *et al.* A lineage-resolved molecular atlas of C. elegans embryogenesis at single-cell resolution. *Science* vol. 365 eaax1971 Preprint at https://doi.org/10.1126/science.aax1971 (2019).
8. Wagner, D. E. & Klein, A. M. Lineage tracing meets single-cell omics: opportunities and challenges. *Nat. Rev. Genet.* **21**, 410–427 (2020).
9. Biddy, B. A. *et al.* Single-cell mapping of lineage and identity in direct reprogramming. *Nature* **564**, 219–224 (2018).
10. Spanjaard, B. *et al.* Simultaneous lineage tracing and cell-type identification using CRISPR–Cas9-induced genetic scars. *Nat. Biotechnol.* **36**, 469–473 (2018).
11. Alemany, A., Florescu, M., Baron, C. S., Peterson-Maduro, J. & van Oudenaarden, A. Whole-organism clone tracing using single-cell sequencing. *Nature* **556**, 108–112 (2018).

12. Raj, B. *et al.* Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. *Nat. Biotechnol.* **36**, 442–450 (2018).

13. Chan, M. M. *et al.* Molecular recording of mammalian embryogenesis. *Nature* **570**, 77–82 (2019).

14. Wang, S.-W., Herriges, M. J., Hurley, K., Kotton, D. N. & Klein, A. M. CoSpar identifies early cell fate biases from single-cell transcriptomic and lineage information. *Nat. Biotechnol.* 1–9 (2022).

15. Prasad, N., Yang, K. & Uhler, C. Optimal Transport using GANs for Lineage Tracing. *arXiv [cs.LG]* (2020).

16. Jones, M. G. *et al.* Inference of single-cell phylogenies from lineage tracing data using Cassiopeia. *Genome Biol.* **21**, 92 (2020).

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) doi:10.1016/j.cels.2021.05.008.

18. Konno, N. *et al.* Deep distributed computing to reconstruct extremely large lineage trees. *Nat. Biotechnol.* **40**, 566–575 (2022).

19. Weinreb, C. & Klein, A. M. Lineage reconstruction from clonal correlations. *Proc. Natl. Acad. Sci. U. S. A.* **117**, 17041–17048 (2020).

20. Wagner, D. E. *et al.* Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo. *Science* **360**, 981–987 (2018).

21. Forrow, A. & Schiebinger, G. LineageOT is a unified framework for lineage tracing and trajectory inference. *Nat. Commun.* **12**, 4940 (2021).

22. Lange, M. *et al.* CellRank for directed single-cell fate mapping. *Nat. Methods* **19**, 159–170 (2022).

23. Haghverdi, L. & Ludwig, L. S. Single-cell multi-omics and lineage tracing to dissect cell fate decision-making. *Stem Cell Reports* **18**, 13–25 (2023).

24. VanHorn, S. & Morris, S. A. Next-Generation Lineage Tracing and Fate Mapping to Interrogate Development. *Dev. Cell* **56**, 7–21 (2021).

25. Quinn, J. J. *et al.* Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts. *Science* **371**, (2021).

26. Ludwig, L. S. *et al.* Lineage Tracing in Humans Enabled by Mitochondrial Mutations and Single-Cell Genomics. *Cell* **176**, 1325–1339.e22 (2019).

27. Miller, T. E. *et al.* Mitochondrial variant enrichment from high-throughput single-cell RNA sequencing resolves clonal populations. *Nat. Biotechnol.* (2022) doi:10.1038/s41587-022-01210-8.

28. Villani, C. *Optimal Transport*. (Springer Berlin Heidelberg).

29. Peyré, G. & Cuturi, M. Computational Optimal Transport. Preprint at https://doi.org/10.1561/9781680835519 (2019).

30. Bendall, S. C. *et al.* Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell* **157**, 714–725 (2014).

31. Setty, M. *et al.* Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nat. Biotechnol.* **34**, 637–645 (2016).

32. Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F. & Theis, F. J. Diffusion pseudotime robustly reconstructs lineage branching. *Nat. Methods* **13**, 845–848 (2016).

33. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32**, 381–386 (2014).

34. Wolf, F. A. *et al.* PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.* **20**, 59 (2019).

35. 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, 2016).

36. Vayer, T., Chapel, L., Flamary, R., Tavenard, R. & Courty, N. Fused Gromov-Wasserstein

Distance for Structured Objects. *Algorithms* **13**, 212 (2020).

37. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Adv. Neural Inf. Process. Syst.*
38. 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).
39. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
40. Chizat, L., Peyré, G., Schmitzer, B. & Vialard, F. X. Scaling algorithms for unbalanced optimal transport problems. *Math. Comput.* (2018).
41. Séjourné, Vialard & Peyré. The unbalanced Gromov Wasserstein distance: Conic formulation and relaxation. *Adv. Neural Inf. Process. Syst.*
42. Pan, X., Li, H. & Zhang, X. TedSim: temporal dynamics simulation of single-cell RNA sequencing data and cell division history. *Nucleic Acids Res.* (2022) doi:10.1093/nar/gkac235.
43. Sulston, J. E. & Horvitz, H. R. Post-embryonic cell lineages of the nematode, Caenorhabditis elegans. *Dev. Biol.* **56**, 110–156 (1977).
44. Sulston, J. E., Schierenberg, E., White, J. G. & Thomson, J. N. The embryonic cell lineage of the nematode Caenorhabditis elegans. *Dev. Biol.* **100**, 64–119 (1983).
45. Riddle, D. L., Blumenthal, T., Meyer, B. J. & Priess, J. R. *Specification of Cell Fates in the AB Lineage*. (Cold Spring Harbor Laboratory Press, 1997).
46. Shen, W.-K. *et al.* AnimalTFDB 4.0: a comprehensive animal transcription factor database updated with variation and expression annotations. *Nucleic Acids Res.* **51**, D39–D45 (2023).
47. González-Barrios, M. *et al.* Cis- and trans-regulatory mechanisms of gene expression in the

ASJ sensory neuron of Caenorhabditis elegans. *Genetics* **200**, 123–134 (2015).

48. 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* **10**, 3071–3085 (2020).
49. Hallam, S., Singer, E., Waring, D. & Jin, Y. The C. elegans NeuroD homolog cnd-1 functions in multiple aspects of motor neuron fate specification. *Development* **127**, 4239–4252 (2000).
50. Wallace, S. W., 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.* **15**, 550–562 (2016).
51. 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* **21**, 936–948 (2016).
52. 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.* **15**, 157–168 (2013).
53. Setty, M. *et al.* Characterization of cell fate probabilities in single-cell data with Palantir. *Nat. Biotechnol.* **37**, 451–460 (2019).
54. 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* **142**, 1447–1457 (2015).
55. Prasad, B., Karakuzu, O., Reed, R. R. & Cameron, S. unc-3-dependent repression of specific motor neuron fates in Caenorhabditis elegans. *Dev. Biol.* **323**, 207–215 (2008).
56. 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. **287**, 74–85 (2005).

57. Clark, S. G. & Chiu, C. C. elegans ZAG-1, a Zn-finger-homeodomain protein, regulates axonal development and neuronal differentiation. *Development* **130**, 3781–3794 (2003).
58. Ramakrishnan, K. & Okkema, P. G. Regulation of C. elegans neuronal differentiation by the ZEB-family factor ZAG-1 and the NK-2 homeodomain factor CEH-28. *PLoS One* **9**, e113893 (2014).
59. Wacker, I., Schwarz, V., Hedgecock, E. M. & Hutter, H. zag-1, a Zn-finger homeodomain transcription factor controlling neuronal differentiation and axon outgrowth in C. elegans. *Development* **130**, 3795–3805 (2003).
60. Reilly, M. B., Cros, C., Varol, E., Yemini, E. & Hobert, O. Unique homeobox codes delineate all the neuron classes of C. elegans. *Nature* **584**, 595–601 (2020).
61. Hobert, O. A map of terminal regulators of neuronal identity in Caenorhabditis elegans. *Wiley Interdiscip. Rev. Dev. Biol.* **5**, 474–498 (2016).
62. Mansfeld, J. *et al.* Branched-chain amino acid catabolism is a conserved regulator of physiological ageing. *Nat. Commun.* **6**, 10043 (2015).
63. Poss, K. D., Wilson, L. G. & Keating, M. T. Heart Regeneration in Zebrafish. *Science* vol. 298 2188–2190 Preprint at https://doi.org/10.1126/science.1077857 (2002).
64. 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* **6**, e18503 (2011).
65. González-Rosa, J. M., Martín, V., Peralta, M., Torres, M. & Mercader, N. Extensive scar formation and regression during heart regeneration after cryoinjury in zebrafish. *Development* **138**, 1663–1674 (2011).
66. Chablais, F., Veit, J., Rainer, G. & Jaźwińska, A. The zebrafish heart regenerates after cryoinjury-induced myocardial infarction. *BMC Dev. Biol.* **11**, 21 (2011).
67. Hu, B. *et al.* Origin and function of activated fibroblast states during zebrafish heart

regeneration. *Nat. Genet.* **54**, 1227–1237 (2022).

68. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).

69. Virshup, I. *et al.* The scverse project provides a computational ecosystem for single-cell omics data analysis. *Nat. Biotechnol.* (2023) doi:10.1038/s41587-023-01733-8.

70. Kamimoto, K. *et al.* Dissecting cell identity via network inference and in silico gene perturbation. *Nature* **614**, 742–751 (2023).

71. Fleck, J. S. *et al.* Inferring and perturbing cell fate regulomes in human brain organoids. *Nature* (2022) doi:10.1038/s41586-022-05279-8.

72. Lotfollahi, M. *et al.* Learning interpretable cellular responses to complex perturbations in high-throughput screens. *bioRxiv* 2021.04.14.439903 (2021) doi:10.1101/2021.04.14.439903.

73. Lareau, C. A. *et al.* Massively parallel single-cell mitochondrial DNA genotyping and chromatin profiling. *Nat. Biotechnol.* **39**, 451–461 (2021).

74. Rodriguez-Fraticelli, A. & Morris, S. A. In preprints: the fast-paced field of single-cell lineage tracing. *Development* **149**, (2022).

75. Choi, J. *et al.* A time-resolved, multi-symbol molecular recorder via sequential genome editing. *Nature* **608**, 98–107 (2022).

76. Choi, J. *et al.* Precise genomic deletions using paired prime editing. *Nat. Biotechnol.* **40**, 218–226 (2022).

77. Loveless, T. B. *et al.* Lineage tracing and analog recording in mammalian cells by single-site DNA writing. *Nat. Chem. Biol.* **17**, 739–747 (2021).

78. Mimitou, E. P. *et al.* Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. *Nat. Biotechnol.* (2021) doi:10.1038/s41587-021-00927-2.

79. Jindal, K. *et al.* Multiomic single-cell lineage tracing to dissect fate-specific gene regulatory

programs. *bioRxiv* 2022.10.23.512790 (2022) doi:10.1101/2022.10.23.512790.

80. Ma, S. *et al.* Chromatin Potential Identified by Shared Single-Cell Profiling of RNA and Chromatin. *Cell* **183**, 1103–1116.e20 (2020).
81. Kartha, V. K. *et al.* Functional inference of gene regulation using single-cell multi-omics. *Cell Genom* **2**, (2022).
82. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv [stat.ML]* (2018).
83. Tucker, D. K., Adams, C. S., Prasad, G. & Ackley, B. D. The Immunoglobulin Superfamily Members syg-2 and syg-1 Regulate Neurite Development in C. elegans. *J Dev Biol* **10**, (2022).
84. Shen, K. & Bargmann, C. I. The immunoglobulin superfamily protein SYG-1 determines the location of specific synapses in C. elegans. *Cell* **112**, 619–630 (2003).
85. Shen, K., Fetter, R. D. & Bargmann, C. I. Synaptic specificity is generated by the synaptic guidepost protein SYG-2 and its receptor, SYG-1. *Cell* **116**, 869–881 (2004).
86. Maro, G. S. *et al.* MADD-4/Punctin and Neurexin Organize C. elegans GABAergic Postsynapses through Neuroligin. *Neuron* **86**, 1420–1432 (2015).
87. 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.* **295**, 16267–16279 (2020).
88. Seetharaman, A. *et al.* MADD-4 is a secreted cue required for midline-oriented guidance in Caenorhabditis elegans. *Dev. Cell* **21**, 669–680 (2011).
89. Buntschuh, I. *et al.* FLP-1 neuropeptides modulate sensory and motor circuits in the nematode Caenorhabditis elegans. *PLoS One* **13**, e0189320 (2018).

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 and all members of Nitzan and Theis labs for the great discussions. 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) and by the Joachim Herz Foundation. 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.

Author 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.

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.

Data availability

Raw published data for the C. elegans[7] and Zebrafish[67] examples are available from the Gene Expression Omnibus under accession codes GSE126954 and GSE159032, respectively. Processed data is available from figshare under https://doi.org/10.6084/m9.figshare.c.6533377.v1

Code availability

The moslin software can be accessed via https://github.com/theislab/moslin, including documentation, tutorials and examples. Jupyter notebooks and Python scripts to reproduce our results are available via the same GitHub repository.

Supplementary materials

Supplementary Figures: Supplementary Figures 1 - 12
Supplementary Tables: Supplementary Table 1
Supplementary Notes: Supplementary Note 1

Figures

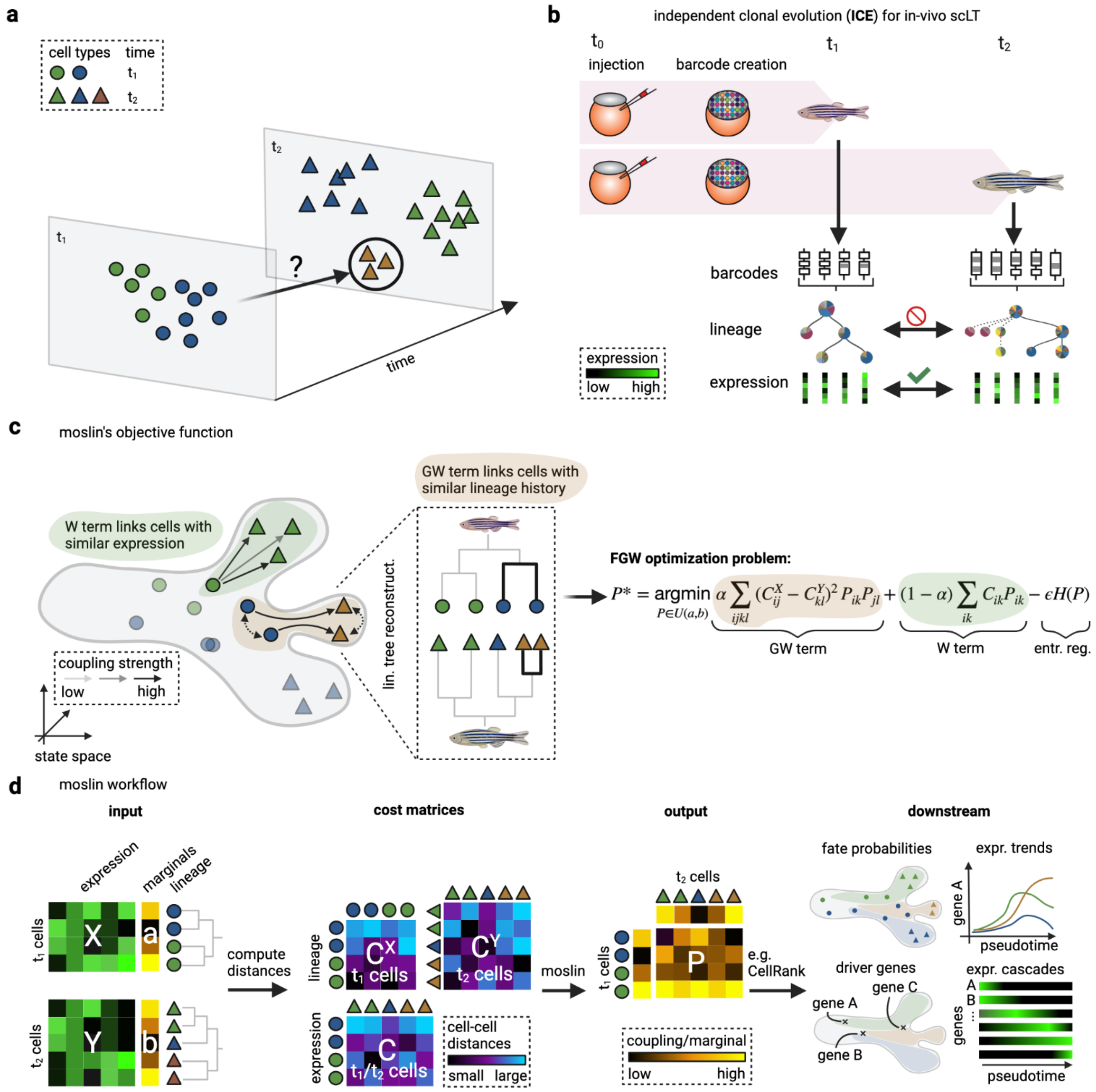

Fig. 1 | Moslin maps lineage-traced single cells across time points.

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. 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 grey 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 Gromow-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[22] to compute fate probabilities, driver genes and expression trends- or cascades. Figure created using BioRender.

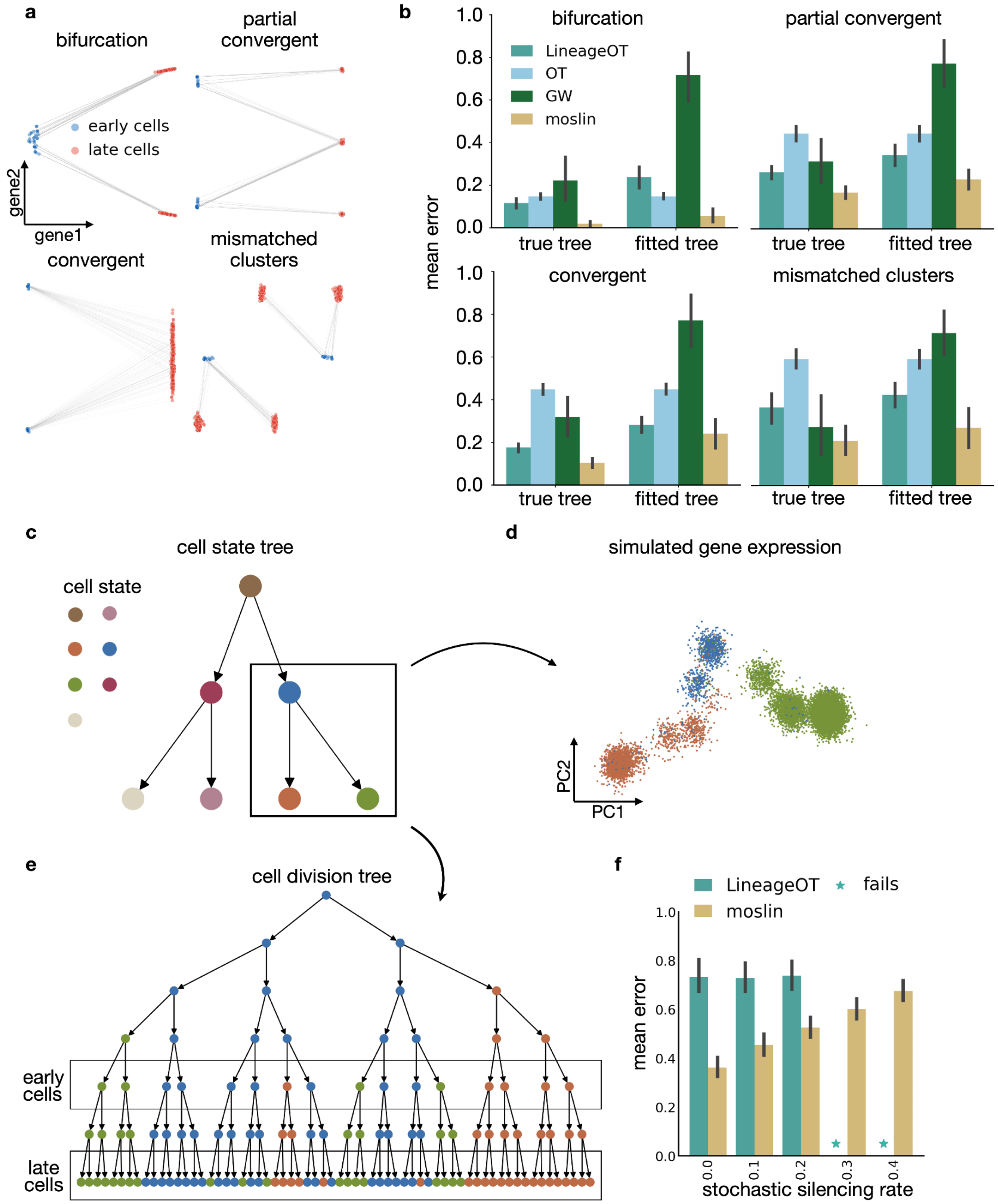

Fig. 2 | Moslin obtains accurate couplings for simple and complex trajectory topologies.

a. Visualization of the four different kinds of simulated trajectories in gene expression space. **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, OT, GW and moslin (Methods). Error bars depict the 95% confidence interval across 10 random simulations. **c-e.** Simulated tree and expression using TedSim[42]. 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 LineageOT. As a function of the *stochastic silencing rate*. Error bars depict the 95% confidence interval across 10 random simulations.

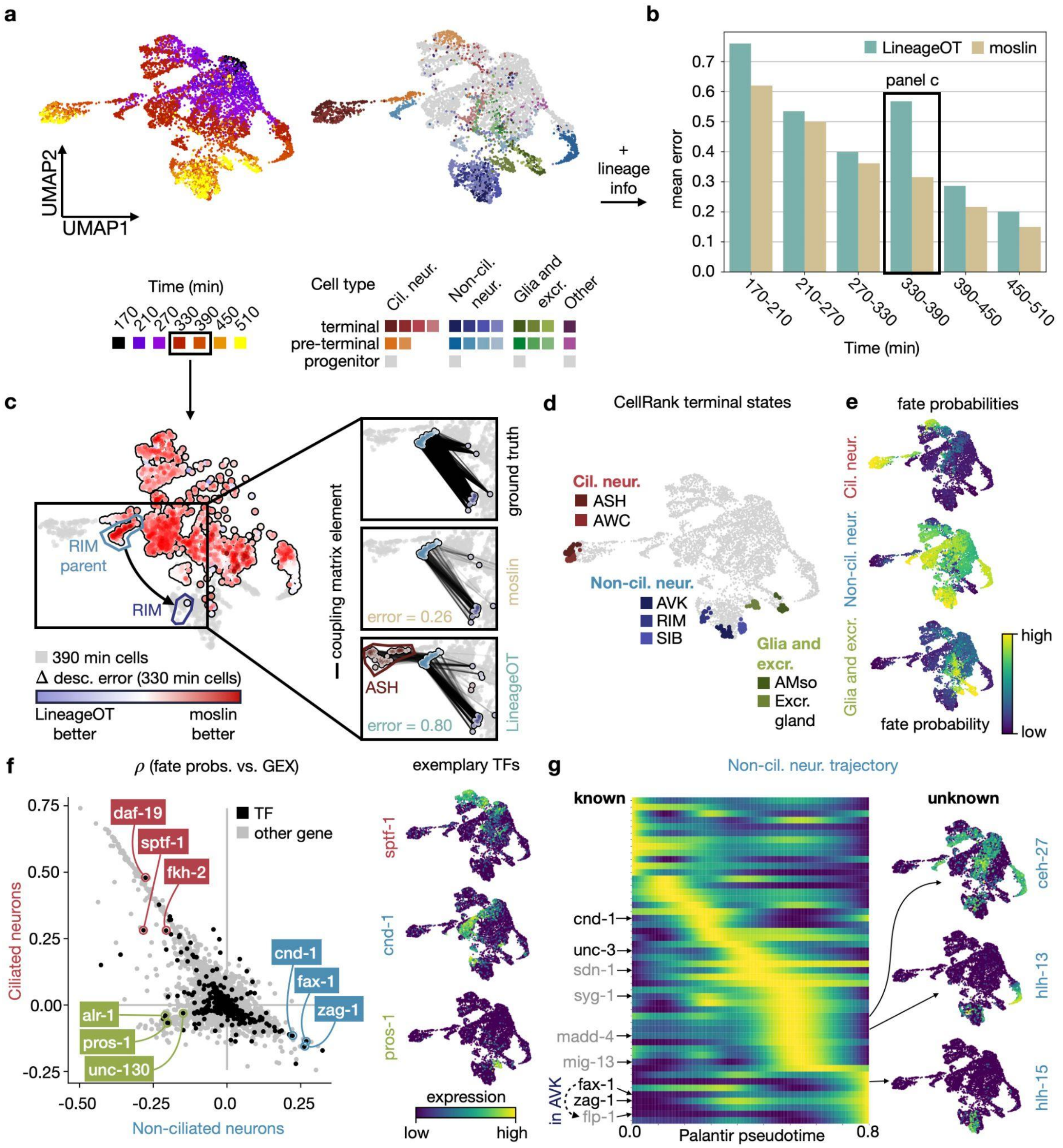

Fig. 3 | Moslin accurately captures *C. elegans* embyogenesis.

a. UMAP[82] 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 moslin and LineageOT[21] across time points (Methods). **c.** Left: UMAP of 330-390 min cells, colored in grey (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 (Methods) **d.** UMAP, showing the top 30 cells per moslin/CellRank[22] computed terminal state. **e.** UMAPs of aggregated fate probabilities towards Ciliated neurons, Non-ciliated neurons, and Glia and excretory cells (Supplementary Fig. 6 and 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 (Supplementary Table 1). 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 Supplementary Fig. 10). Each row corresponds to a gene, smoothed using fate probabilities (**e**) and the Palantir pseudotime[53] (x-axis, Supplementary Fig. 9). We annotate a few TFs, including cnd-1[48,49], fax-1[56], and zag-1[57–59] (black), and other genes, including syg-1[83–85], madd-4[86–88], and flp-1[56,89] (grey) that are known to be involved in the process (Supplementary Table 1). Right: UMAPs, showing expression of previously unknown predicted driver TFs.

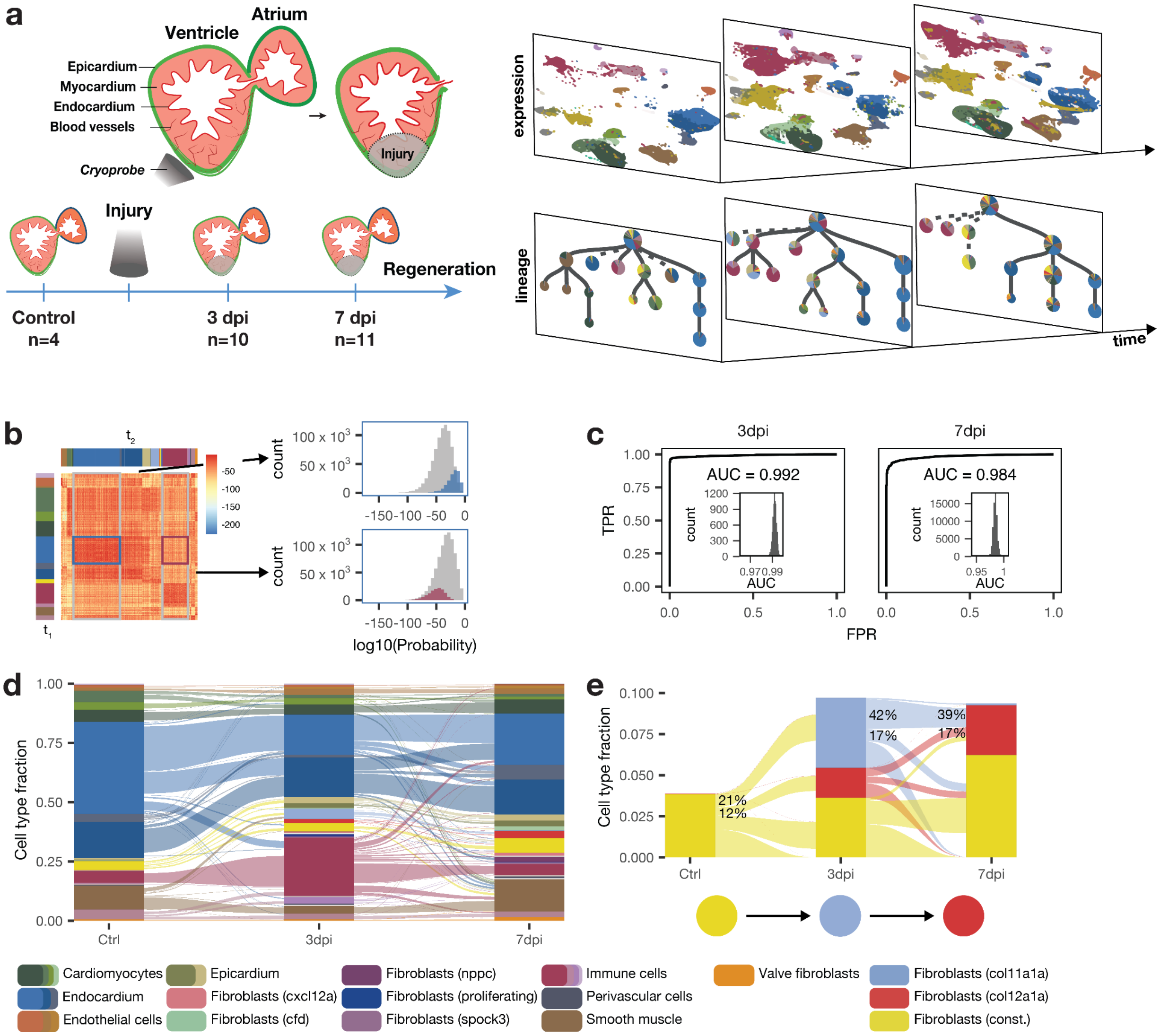

Fig. 4 | Moslin recovers lineage relations among transient fibroblast subsets.

a. Underlying data describes zebrafish heart regeneration, measured through single-cell transcriptomic and lineage profiling before injury (n=4), at 3dpi (n=9) and 7dpi (n=7)[67]. **b.** Welch's t-test for cell type persistence (Methods). **c.** ROC curves for same-cell type ancestors. **d.** Flow diagram of cell type transitions. **e.** Flow diagram of transient epicardial fibroblasts corroborates col11a1a fibroblasts as an intermediary state between constitutive and col12a1a fibroblasts.

April 14, 2023

Moslin Methods Section

Contents

1 The moslin algorithm 2
1.1 Introduction and model overview . . . 2
1.2 In vivo single-cell lineage tracing (scLT) . . . 2
1.3 Moslin's objective function for in-vivo ICE . . . 3
1.4 Downstream usage of coupling matrices . . . 6

2 Datasets 6
2.1 2-gene simulations . . . 6
2.2 TedSim simulated data . . . 7
2.3 *C. elegans* embryonic development . . . 8
2.3.1 Benchmarking moslin with LineageOT . . . 9
2.3.2 Combining moslin with CellRank for fate mapping analysis . . . 10
2.4 Zebrafish heart regeneration (LINNAEUS) . . . 11
2.4.1 Mapping datasets . . . 11
2.4.2 Test for persistence of cell states . . . 12
2.4.3 Calculating cellular flows . . . 13

1 The moslin algorithm

1.1 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[1–12] that record lineage relationships in vivo. While previous analysis approaches for this type of lineage tracing data remained limited to individual, isolated time-points[4,5,13–21], moslin embeds clonal dynamics in their temporal context.

Moslin's inputs. The input to moslin are pairs of state matrices and linege information ($X \in \mathbb{R}^{N\times G}$, ξ) and ($Y \in \mathbb{R}^{M\times G}$, ζ) 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. The lineage information arrays ξ and ζ contain the lineage tracing outcome for every cell; their exact nature depends on the lineage tracing technology (Section 1.2). Optionally, moslin takes marginal distributions $\boldsymbol{a} \in \Delta_N$ and $\boldsymbol{b} \in \Delta_M$ over cells at t_1 and t_2 for probability simplex $\Delta_N \coloneqq \{\boldsymbol{a} \in \mathbb{R}_+^N | \sum_{i=1}^N a_i = 1\}$. 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 $P \in U(\boldsymbol{a}, \boldsymbol{b})$ where $U(\boldsymbol{a}, \boldsymbol{b})$ is the set of feasible coupling matrices given by

$$U(\boldsymbol{a}, \boldsymbol{b}) \coloneqq \{P \in \mathbb{R}_+^{N\times M} | P\mathbf{1}_M = \boldsymbol{a},\ P^\top \mathbf{1}_N = \boldsymbol{b}\}\,, \tag{1}$$

for constant one vector $\mathbf{1}_N = [1, ..., 1]^\top \in \mathbb{R}^N$. 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 $U(\boldsymbol{a}, \boldsymbol{b})$ contains all matrices P which are compatible with the prescribed marginals $\boldsymbol{a}$ at t_1 and $\boldsymbol{b}$ at t_2.

With these definitions at hand, we can formalize the aim of moslin: we seek to find the coupling matrix $P \in U(\boldsymbol{a}, \boldsymbol{b})$ 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 Subsection 1.3

1.2 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[22,23], recent approaches are sequencing-based and use heritable genetic barcodes[23–27]. 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[28–30], in vivo transplantation settings[28] or in vivo regenerative systems like human PBMC and CD34+ samples[31,32] or the zebrafish fin[2]. 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.

1.3 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 $P \in U(\boldsymbol{a}, \boldsymbol{b})$ 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[33]; in particular, we use a *Fused Gromov Wasserstein*[34] (FGW) formulation which allows us to include terms for across- and within time-point similarity (Supplementary Note 1). Previous single-cell methods successfully used OT to map cells across time points without lineage information[35,36], impute gene expression in spatial data[37], predict perturbation response[38–40], learn patient manifolds[41,42], integrate data across modalities[43] and infer cell-cell communication[44]. 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, molecular similarity is conserved between t_1 and t_2; similar cell pairs at t_1 are likely to transition into similar cell pairs at t_2.
- A3: 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 challenges A2.
- Noisy or incomplete lineage readout challenges A3.

Thus, rather than enforcing A1-A3 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.

A combined approach for in vivo scLT data. In ICE, gene expression information is comparable across time points but lineage information is not (Section 1.2). 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 quantities gene expression similarity.
- A quadratic *Gromov-Wasserstein* (GW) term for incomparable features, encouraging A2 and A3. This term quantifies lineage and expression concordance.

The W term for individual comparisons. To encourage A1, we consider a W term[33] 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 $(\boldsymbol{x}_i, \boldsymbol{y}_j) \in \mathcal{X} \times \mathcal{Y}$, we construct a cost matrix, $C \in R_+^{N \times M}$ with $C_{ij} = c(\boldsymbol{x}_i, \boldsymbol{y}_j)$ 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[45]. Formally, the mapping problem is defined as

$$P^* := \underset{P \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \; \langle C, P \rangle = \underset{P \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \sum_{ij} C_{ij} P_{ij} \,, \tag{2}$$

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 and A3, we consider a GW term[33,46,47] which compares cell pairs in the source (t_1) and target (t_2) distributions in terms of their gene expression and lineage information. Given gene expression vectors and lineage information, we define two independent cost matrices, $C^{\mathcal{X}} \in R_+^{N \times N}$ and $C^{\mathcal{Y}} \in R_+^{M \times M}$ with $C^{\mathcal{X}}_{ij} = c^{\mathcal{X}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $C^{\mathcal{Y}}_{kl} = c^{\mathcal{Y}}(\boldsymbol{y}_k, \boldsymbol{y}_l)$ for cost functions $c^{\mathcal{X}}$ and $c^{\mathcal{Y}}$.

Focusing on the early time point, consider latent space samples $\boldsymbol{x}_i$ and lineage information ξ_i. Define the composite t_1-cost function

$$c^{\mathcal{X}}(\boldsymbol{x}_i, \xi_i, \boldsymbol{x}_j, \xi_j) = \beta \, c_l\big(f^{\mathcal{X}}(\xi_i), f^{\mathcal{X}}(\xi_j)\big) + (1-\beta)||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2 \,, \tag{3}$$

for parameter $\beta \in [0,1]$, controlling the weight given to lineage versus molecular state, mapping function $f^{\mathcal{X}}$, providing a representation of the lineage information at t_1, and lineage distance function c_l. Lineage information is typically noisy and incomplete; we include molecular similarity at weight $(1-\beta)$ as a regularization. Moslin supports two ways of representing lineage information:

- barcode representation: $f^{\mathcal{X}}$ is the identity and c_l quantifies hamming distance between raw barcodes.
- lineage tree representation: $f^{\mathcal{X}}$ is a lineage-tree reconstruction computed using a method like Cassiopeia[13] or LINNAEUS[1] and c_l quantifies shortest path distance along reconstructed trees.

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

$$P^* := \underset{P \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \sum_{ijkl} L\left(C^{\mathcal{X}}_{ij}, C^{\mathcal{Y}}_{kl}\right) P_{ik} P_{jl} \,, \tag{4}$$

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, A2, and A3, we combine the W with the GW term to yield moslin's objective function for in-vivo ICE data,

$$P^* = \underset{P \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}} \; \underbrace{\alpha \sum_{ijkl} L\left(C^{\mathcal{X}}_{ij}, C^{\mathcal{Y}}_{kl}\right) P_{ik} P_{jl}}_{\text{A2 and A3}} + \underbrace{(1-\alpha) \sum_{ik} C_{ik} P_{ik}}_{\text{A1}} \,, \tag{5}$$

which is known as a *Fused Gromov-Wasserstein* (FGW) problem[34] (Supplementary Note 1). 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 Equation (5) defines a quadratic programming problem; to introduce a notion of uncertainty and to speed up the optimization, we follow previous approaches[35,48] and include and entropy regularization term,

$$H(P) = -\sum_{ij} P_{ij}(\log P_{ij} - 1)\,, \tag{6}$$

and the regularized FGW objective reads

$$P^* \coloneqq \underset{P \in U(\boldsymbol{a},\boldsymbol{b})}{\operatorname{argmin}}\ \alpha \sum_{ijkl} L\left(C^{\mathcal{X}}_{ij}, C^{\mathcal{Y}}_{kl}\right) P_{ik}P_{jl} + (1-\alpha)\sum_{ik} C_{ik}P_{ik} - \epsilon H(P)\,, \tag{7}$$

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[33,48] (Supplementary Note 1). 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 endcode prior biological information. If additional information about sampled cells is available, e.g., growth- and death-rates, uncertainty, etc., we incorporate them via the marginals $\boldsymbol{a}$ and $\boldsymbol{b}$. If no additional information is available, we assign them uniformly. By default, in moslin, we choose the right marginal $\boldsymbol{b}$ 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,

$$a_i = \frac{g(\boldsymbol{x}_i)^{t_2-t_1}}{\sum_{j=1}^{N} g(\boldsymbol{x}_j)^{t_2-t_1}}\ \forall i \in \{1, ..., N\}\,, \tag{8}$$

where $g : \mathbb{R}^D \to \mathbb{R}$ is modeled as the expected value of a birth-death process with proliferation at rate $\beta(\boldsymbol{x})$ and death at rate $\delta(\boldsymbol{x})$, thus $g(\boldsymbol{x}) = e^{\beta(\boldsymbol{x})-\delta(\boldsymbol{x})}$ for $\beta(\boldsymbol{x})$ and $\delta(\boldsymbol{x})$ estimated from curated marker gene sets for proliferation and apoptosis, respectively[35].

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.

Accordingly, we allow small deviations from the exact marginals $\boldsymbol{a}$ and $\boldsymbol{b}$ in an unbalanced FGW framework[49] where we replace the hard constraint $P \in U(\boldsymbol{a}, \boldsymbol{b})$ 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 ($\boldsymbol{a}$) and right ($\boldsymbol{b}$) marginal constraints, we use two parameters $\tau_a, \tau_b \in (0, 1)$ (Supplementary Note 1). For the optimization, we employ the algorithm presented by Séjourné et al.[49] 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 **M**ulti-**O**mic **S**ingle-**C**ell **O**ptimal **T**ranport. moscot is a scalable, easy-to-use, open-source solution for OT-based analysis in single-cell genomics; it interfaces with optimal transport tools[50] (OTT) in the backend to support GPU acceleration and just-in-time compilation via JAX[51].

1.4 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 $\mathcal{P}$ 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,

$$p_{t_1}(x) := \begin{cases} \frac{1}{|\mathcal{P}|} & x \in \mathcal{P}\,, \\ 0 & \text{else}\,, \end{cases} \tag{9}$$

where x is a cell from t_1 and $|\mathcal{P}|$ corresponds to the number of cells in state $\mathcal{P}$. Following Schiebinger et al.[35], we compute t_2 descendants of cell state $\mathcal{P}$ by a push-forward operation of p_{t_1},

$$p_{t_2} = P^\top p_{t_1}\,, \tag{10}$$

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

$$q_{t_1} = P q_{t_2}\,, \tag{11}$$

where $q_{t_1}(x)$ is the probability mass that cell x contributes towards cell state $\mathcal{Q}$ 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[52], a fate mapping toolkit which 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 Schiebinger et al.[35], 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

$$P^{t_1,t_T} = P^{t_1,t_2} P^{t_2,t_3} ... P^{t_{T-1},t_T}\,. \tag{12}$$

We compute ancestors and descendants for multi-day couplings in the same way as above (Equations (10) and (11)).

2 Datasets

2.1 2-gene simulations

We use a simulation setting suggested by Forrow and Schiebinger[53] 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 Forrow and Schiebinger[53] define 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[54] as implemented in LineageOT[53]. LineageOT uses the tree (true or fitted) directly to compute the couplings. In moslin, we construct 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[55] with weights associated to edges according to "time" between two nodes. We compare the performance of moslin to LineageOT, 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 ($\alpha = 1$). We quantify method performance using the ancestor and descendant errors introduced in Forrow and Schiebinger[53]. For ground truth coupling P^* and predicted coupling P, we compare their predicted ancestors and descendants per cell using a Wasserstein-2 distance (Supplementary Note 1). To obtain the descendant error $E_D(P)$, we compute

$$E_D(P) = \sum_{i=1}^{N} a_i \, W_2^2(P^*_{i,:}, P_{i,:}) \, , \tag{13}$$

for squared Wasserstein-2 distance W_2^2 (Supplementary Note 1) and right marginal $a_i = \sum_j P_{ij}$. Similarly, to obtain the ancestor error $E_A(P)$, we compute

$$E_A(P) = \sum_{j=1}^{M} b_j W_2^2(P^*_{:,j}, P_{:,j}) \, , \tag{14}$$

for left marginal $b_j = \sum_i P_{ij}$. 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, $\hat{P} = \boldsymbol{a}\boldsymbol{b}^\top$, corresponding to an uninformative coupling with the same marginals as the predicted coupling P. Specifically, we compute $E_D(P)/E_D(\hat{P})$ and $E_A(P)/E_A(\hat{P})$, such that a value of one corresponds to an uninformative result. Our final error metric is given by the mean of the two quantities[53].

We perform a grid search to find the optimal parameters for each data set and method. For all settings, 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, $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.8, 0.9, 0.95, 0.98, 0.999\}$.

2.2 TedSim simulated data

We utilize TedSim[56] (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[57–59] 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[56], 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 are removed from the 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 Forrow and Schiebinger[53]. The scaling is defined such that: (i) the number of sites where both cells were measured is taken into account, (ii) the distance between two scars is twice the distance from scarred to unscarred sites. For LineageOT, similarly to the previous setting, the barcodes are used internally to construct a fitted tree. 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 $\epsilon \in \{1e-1, 1\}$.

2.3 *C. elegans* embryonic development

The *C. elegans* development dataset[60] contains gene expression for approx. 86k single cells, sequenced using 10x genomics. The original authors[60] mapped these cells towards the known *C. elegans* lineage tree[22] and obtained lineage information for a subset of cells. Additionally, they mapped their data towards a bulk time-series dataset[61] 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.

Preprocessing. To evaluate moslin's performance, we required ground-truth lineage information. The original study's[60] mapping inferred partial lineage information for a subset of approx. 46k cells. To obtain precise lineage information, we implemented two suggestions by Forrow and Schiebinger[53]:

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 precise 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 preprocessed the data using SCANPY[62], 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[63], 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[64]. To reduce complexity and focus on the main groups of terminal cell states, we aggregated original cluster annotations[60] 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 or 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.

2.3.1 Benchmarking moslin with LineageOT

Shared moslin/LineageOT parameters and settings. We benchmarked moslin with LineageOT on the two cell subsets (Strategy 1 and 2), using the pre-processing described above. We use the marginals $\boldsymbol{a}$ and $\boldsymbol{b}$ to capture the effects of cellular growth and death, and calculate them using the lineage tree following Forrow and Schiebinger[53]. Gene-expression distances among cells from different time-points were measured using squared Euclidean distance in the PCA space, and passed to both methods in the mean-scaled cost matrix C.

Additional moslin 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. We set $\beta = 0$, i.e. the GW term corresponds to pure lineage information. To construct the lineage cost matrices C^X and C^Y, 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[55]. The weights represent the temporal difference between a node and its parent. Additionally, we mean-scaled the C^X and C^Y cost matrices.

Additional LineageOT parameters. We run LineageOT following the original authors' reproducibility repository. LineageOT runs the Sinkhorn algorithm as implemented in python optimal transport (POT)[65] 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:

- Moslin:
  - $\alpha \in [0.01, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.98]$
  - $\epsilon \in [0.001, 0.01, 0.05, 0.1, 0.5]$
- LineageOT:
  - $\epsilon \in [0.001, 0.01, 0.05, 0.1, 0.5]$

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.

Zoom in to the 330/390 min time point pair. To visualize the transitions predicted by moslin and LineageOT for the RIM_parent population, we selected 330 min RIM_parent cells. Out of these, we 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 the moslin/LineageOT-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 moslin, LineageOT, and 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.

2.3.2 Combining moslin with CellRank for fate mapping analysis

We focused on the ABpxp lineage (Strategy 1), and run moslin with the optimal hyperparameters identified in our grid search. We filtered out cells assigned a zero value in the marginal distributions to arrive at 6,476 cells used for this analysis. In the following, we used CellRank with default parameters if not indicated otherwise.

Transition matrix construction in CellRank. CellRank[52] is a fate mapping framework that was originally designed for RNA velocity[66,67] data. In version 2, it has been extended towards other data modalities, including time-series data. We make use of this extension here to construct 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 moslin's 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' across all time points based on gene expression similarity. We combine T with T' 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 transition matrix, which we interpret as a Markov chain. We simulated 200 random walks, each containing 500 steps, to visualize the predicted cell dynamics, starting from randomly selected 170 min cells.

Identifying terminal states and computing aggregated fate probabilities. We used CellRank's GPCCA estimator[68,69] 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.

We used two-sided unequal variance Welch t-tests to asses whether fate probabilities were higher among pre-terminal cells for each terminal state group:

- for Ciliated neurons, we tested 647 pre-terminal ciliated neurons against 4,179 other pre-terminal and progenitor cells. We found $t = 40.7, P = 6.4 \cdot 10^{-182}$.
- for Non-ciliated neurons, we tested 890 pre-terminal non-ciliated neurons against 3,882 other pre-terminal and progenitor cells. We found $t = 29.3, P = 3.0 \cdot 10^{-160}$.
- for Glia and excretory cells, we tested 361 pre-terminal Glia and excretory cells against 4,227 other pre-terminal and progenitor cells. We found $t = 82.2, P = 2.1 \cdot 10^{-255}$.

Predicting driver genes. Using CellRank, we correlated each gene's expression with the computed fate probabilities across all cells and subsetted to known *C. elegans* transcription factors[70] (TFs). We focused on the top 20 most strongly correlated TFs per terminal cell group and treated these as predicted driver TFs.

Computing a Palantir pseudotime. Using Palantir[71], 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.

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[72] 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[73], through the CellRank interface.

2.4 Zebrafish heart regeneration (LINNAEUS)

The zebrafish heart regeneration dataset[74] consists of hearts from 25 organisms; four uninjured hearts (ctrl), nine at three days after injury (3dpi), and seven at seven days after injury (7dpi). We use moslin to calculate couplings P_{ik}^{ab}, with a and b denoting datasets at consecutive timepoints. For ease of reading, we will suppress indices a and b in the following unless necessary.

2.4.1 Mapping datasets

We embed the transcriptomic readout of all single cells with lineage information into a joint latent space using scVI[45], 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 timepoints. The standard parameters used are: $\alpha = 0.5$, $\epsilon = 1e-2$, $\tau_a = 0.9$, $\tau_b = 1$, and $\beta = 0.2$. To understand the influence of the hyper-parameters on the performance we re-compute the mappings, changing a single parameter at a time within the following grids: $\alpha \in \{0, 4, 0.6\}$, $\epsilon =\in \{5e-3, 1e-2, 1e-1\}$, $\tau_a \in \{0, 85, 0.95\}$, $\tau_b = 1$, and $\beta \in \{0, 1, 0.3\}$.

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[75] and the GSEA Hallmark apoptosis geneset (`https://www.gsea-msigdb.org/gsea/msigdb/human/geneset/HALLMARK_APOPTOSIS.html`). These are converted to their zebrafish orthologues using orthologues from Alliance, as previously described[74]. Next, we use these two gene sets to calculate growth rates[76]. For cells at 3dpi, 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.

2.4.2 Test for persistence of cell states

We expect that cells of the same, non-transient, type are persistent over time; cells of type A at time t_2 should, for the most part, stem from cells of type A at time t_1. This means moslin-computed couplings between those cells should be higher than those between cells of type B (with $B \neq A$) at time t_1 and cells of type A at time t_2. To test this, we first select cell types at t_1 and t_2 with more than 10 cells that exist at both time points. We then define the distribution of couplings between cells of type B at t_1 and cells of type A at t_2 as

$$\gamma(B, A) := \{P_{ik} : i \text{ type } B, k \text{ type } A\} \tag{15}$$

and perform a Welch's t-test to calculate the significance level of the hypothesis

$$\mu(\gamma(B, A)) > \mu(\gamma(\overline{B}, A)) \tag{16}$$

where $\overline{B}$ is the complement of B, i.e. all cells that are not type B (Fig. 5b), and $\mu(\gamma)$ is the mean of the population γ. Note that due to our requirement that every cell type at every time point contain at least 10 cells, all distributions here will have 100 or more datapoints, ample to assume the sample means are close to normal by the central limit theorem and therefore satisfy the normality assumption underlying a Welch's t-test. The expectation of persistent non-transient cell types means that a significant test result for $\langle\gamma(A, A)\rangle > \left\langle\gamma(\overline{A}, A)\right\rangle$ is a true positive, and a significant test result for $\langle\gamma(B, A)\rangle > \left\langle\gamma(\overline{B}, A)\right\rangle$ with $B \neq A$ is a false positive. With this formulation, we can create receiver operating characteristic (ROC) curves by iterating over the p-values from the t-tests and calculate the area under the ROC curve (AUC) value for a single combination of t_1 and t_2 datasets.

To create ROC curves for all control-3dpi and 3dpi-7dpi couplings, we perform this test between all t_1 cell types and all t_2 cell types within all combinations of datasets. We calculate AUCs to be 0.992 for control-3dpi and 0.984 for 3dpi-7dpi at hyper-parameter values $\alpha = 0.5, \beta = 0.2, \epsilon = 0.01, \tau_a = 0.9$ (Fig. 5c and Subsection 1.3). To understand the influence of individual dataset couplings, we use the same procedure to calculate AUCs for all possible subsets of dataset combinations and plot the histogram of these AUCs (Fig. 5c, inset). Finally, to understand the influence of hyper-parameters α, β, ϵ, and τ_a, we used the same procedure to calculate AUCs for couplings with different hyper-parameter values (Supp. Fig. 1). We observe no noticeable differences in AUCs.

Since the above described test follows a one-versus-one strategy, its AUC values may be inflated by a high amount of true negatives. We therefore implemented a variation of the cell type persistency test, following a one-versus-rest strategy. Here, we only test whether

$$\langle\gamma(A, A)\rangle > \left\langle\gamma(\overline{A}, A)\right\rangle. \tag{17}$$

We calculate ROC curves as above and find areas under the curve of 0.9999 and 1 (up to eight decimals). We conclude that both in one-versus-one and one-versus-rest strategies, the cell type persistency test shows a very high performance of moslin.

2.4.3 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

$$P_{AB}^{ab} = \sum_{i \in A, k \in B} P_{ik}^{ab}, \tag{18}$$

which satisfies $\sum_{AB} P_{AB} = 1$ since $\sum_{ik} P_{ik} = 1$. 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:

$$\tilde{P}_{AB} := \sum_{ab} \left(P_{AB}^{ab} \frac{\#a * \#b}{\sum_a \#a * \sum_b \#b} \right). \tag{19}$$

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

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

$$\tilde{f}_A := \sum_a \frac{\#a}{\sum_a \#a} f_A^a. \tag{20}$$

Again, $\sum_A \tilde{f}_A = 1$ 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 $\tilde{P}_{AB}$ by the total mass outgoing from A:

$$s_{AB} := \frac{\tilde{P}_{AB}}{\sum_C \tilde{P}_{AC}}, \tag{21}$$

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

$$t_{AB} = \frac{\tilde{P}_{AB}}{\sum_C \tilde{P}_{CB}}. \tag{22}$$

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 3dpi datasets and five out of seven 7dpi datasets, meaning 18 instead of 36 couplings between control and 3dpi datasets, and 30 instead of 63 couplings between 3dpi and 7dpi datasets. This method of random selection allows for a total of 7056 combinations:

$$\binom{4}{3} * \binom{9}{6} * \binom{7}{5} = 7056. \tag{23}$$

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

References

[1] Bastiaan Spanjaard, et al. Simultaneous lineage tracing and cell-type identification using crispr–cas9-induced genetic scars. *Nature biotechnology*, 36(5):469–473, 2018.

[2] Anna Alemany, et al. Whole-organism clone tracing using single-cell sequencing. *Nature*, 556(7699): 108–112, 2018.

[3] Bushra Raj, et al. Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. *Nature biotechnology*, 36(5):442–450, 2018.

[4] Michelle M Chan, et al. Molecular recording of mammalian embryogenesis. *Nature*, 570(7759):77–82, 2019.

[5] Daniel E Wagner, et al. Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo. *Science*, 360(6392):981–987, 2018.

[6] Dian Yang, et al. Lineage tracing reveals the phylodynamics, plasticity, and paths of tumor evolution. *Cell*, 185(11):1905–1923, 2022.

[7] Anna Minkina, et al. Tethering distinct molecular profiles of single cells by their lineage histories to investigate sources of cell state heterogeneity. *bioRxiv*, 2022.

[8] Zhisong He, et al. Lineage recording in human cerebral organoids. *Nature methods*, 19(1):90–99, 2022.

[9] Kamen P Simeonov, et al. Single-cell lineage tracing of metastatic cancer reveals selection of hybrid emt states. *Cancer cell*, 39(8):1150–1162, 2021.

[10] Bushra Raj, et al. Emergence of neuronal diversity during vertebrate brain development. *Neuron*, 108 (6):1058–1074, 2020.

[11] Sarah Bowling, et al. An engineered crispr-cas9 mouse line for simultaneous readout of lineage histories and gene expression profiles in single cells. *Cell*, 181(6):1410–1422, 2020.

[12] Jeffrey J Quinn, et al. Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts. *Science*, 371(6532), 2021.

[13] Matthew G Jones, et al. Inference of single-cell phylogenies from lineage tracing data using cassiopeia. *Genome biology*, 21(1):1–27, 2020.

[14] Wuming Gong, et al. Benchmarked approaches for reconstruction of in vitro cell lineages and in silico models of c. elegans and m. musculus developmental trees. *Cell Systems*, 2021.

[15] Sophie Seidel and Tanja Stadler. Tidetree: A bayesian phylogenetic framework to estimate single-cell trees and population dynamic parameters from genetic lineage tracing data. *bioRxiv*, 2022.

[16] Naoki Konno, et al. Deep distributed computing to reconstruct extremely large lineage trees. *Nature Biotechnology*, 40(4):566–575, 2022.

[17] Robert Wang, et al. Theoretical guarantees for phylogeny inference from single-cell lineage tracing. *bioRxiv*, 2021.

[18] Matthew G Jones, et al. Phylovision: Interactive software for integrated analysis of single-cell transcriptomic and phylogenetic data. *bioRxiv*, 2021.

[19] Hamim Zafar, et al. Single-cell lineage tracing by integrating crispr-cas9 mutations with transcriptomic data. *Nature communications*, 11(1):1–14, 2020.

[20] Khalil Ouardini, et al. Reconstructing unobserved cellular states from paired single-cell lineage tracing and transcriptomics data. *bioRxiv*, 2021.

[21] Caleb Weinreb and Allon M Klein. Lineage reconstruction from clonal correlations. *Proceedings of the National Academy of Sciences*, 117(29):17041–17048, 2020.

[22] John E Sulston, et al. The embryonic cell lineage of the nematode caenorhabditis elegans. *Developmental biology*, 100(1):64–119, 1983.

[23] Sadie VanHorn and Samantha A Morris. Next-generation lineage tracing and fate mapping to interrogate development. *Developmental cell*, 56(1):7–21, 2021.

[24] Chloé S Baron and Alexander van Oudenaarden. Unravelling cellular relationships during development and regeneration using genetic lineage tracing. *Nature reviews molecular cell biology*, 20(12):753–765, 2019.

[25] Daniel E Wagner and Allon M Klein. Lineage tracing meets single-cell omics: opportunities and challenges. *Nature Reviews Genetics*, 21(7):410–427, 2020.

[26] Roberto Moreno-Ayala and Jan Philipp Junker. Single-cell genomics to study developmental cell fate decisions in zebrafish. *Briefings in Functional Genomics*, 20(6):420–426, 2021.

[27] Pedro Olivares-Chauvet and Jan Philipp Junker. Inclusion of temporal information in single cell transcriptomics. *The International Journal of Biochemistry & Cell Biology*, 122:105745, 2020.

[28] Caleb Weinreb, et al. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367(6479), 2020.

[29] Brent A Biddy, et al. Single-cell mapping of lineage and identity in direct reprogramming. *Nature*, 564 (7735):219–224, 2018.

[30] Killian Hurley, et al. Reconstructed single-cell fate trajectories define lineage plasticity windows during differentiation of human psc-derived distal lung progenitors. *Cell Stem Cell*, 26(4):593–608, 2020.

[31] Livius Penter, et al. Longitudinal single-cell dynamics of chromatin accessibility and mitochondrial mutations in chronic lymphocytic leukemia mirror disease history. *Cancer Discovery*, 2021.

[32] Caleb A Lareau, et al. Massively parallel single-cell mitochondrial dna genotyping and chromatin profiling. *Nature biotechnology*, 39(4):451–461, 2021.

[33] Gabriel Peyré, et al. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[34] Titouan Vayer, et al. Fused gromov-wasserstein distance for structured objects. *Algorithms*, 13(9):212, 2020.

[35] Geoffrey Schiebinger, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

[36] Alexander Tong, et al. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *International conference on machine learning*, pages 9526–9536. PMLR, 2020.

[37] Mor Nitzan, et al. Gene expression cartography. *Nature*, 576(7785):132–137, 2019.

[38] Charlotte Bunne, et al. Learning single-cell perturbation responses using neural optimal transport. *bioRxiv*, 2021.

[39] Charlotte Bunne, et al. Proximal optimal transport modeling of population dynamics. In *International Conference on Artificial Intelligence and Statistics*, pages 6511–6528. PMLR, 2022.

[40] Charlotte Bunne, et al. Supervised training of conditional monge maps. *arXiv preprint arXiv:2206.14262*, 2022.

[41] William S Chen, et al. Uncovering axes of variation among single-cell cancer specimens. *Nature methods*, 17(3):302–310, 2020.

[42] Alexander Y Tong, et al. Diffusion earth mover's distance and distribution embeddings. In *International Conference on Machine Learning*, pages 10336–10346. PMLR, 2021.

[43] Pinar Demetci, et al. Scot: Single-cell multi-omics alignment with optimal transport. *Journal of Computational Biology*, 29(1):3–18, 2022.

[44] Zixuan Cang and Qing Nie. Inferring spatial and signaling relationships between cells from single cell transcriptomic data. *Nature communications*, 11(1):1–13, 2020.

[45] Adam Gayoso, et al. A python library for probabilistic analysis of single-cell omics data. *Nature biotechnology*, 40(2):163–166, 2022.

[46] Gabriel Peyré, et al. Gromov-wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pages 2664–2672. PMLR, 2016.

[47] Facundo Mémoli. Gromov–Wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011.

[48] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[49] Thibault Séjourné, et al. The unbalanced gromov wasserstein distance: Conic formulation and relaxation. *Advances in Neural Information Processing Systems*, 34:8766–8779, 2021.

[50] Marco Cuturi, et al. Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*, 2022.

[51] Roy Frostig, et al. Compiling machine learning programs via high-level tracing. *Systems for Machine Learning*, 2018.

[52] Marius Lange, et al. Cellrank for directed single-cell fate mapping. *Nature methods*, 19(2):159–170, 2022.

[53] Aden Forrow and Geoffrey Schiebinger. Lineageot is a unified framework for lineage tracing and trajectory inference. *Nature communications*, 12(1):1–10, 2021.

[54] Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425, 1987.

[55] Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, pages 287–290. 2022.

[56] Xinhai Pan, et al. Tedsim: temporal dynamics simulation of single-cell rna sequencing data and cell division history. *Nucleic acids research*, 50(8):4272–4288, 2022.

[57] Haifan Lin and Trista Schagat. Neuroblasts: a model for the asymmetric division of stem cells. *Trends in Genetics*, 13(1):33–39, 1997.

[58] Sean J Morrison and Judith Kimble. Asymmetric and symmetric stem-cell divisions in development and cancer. *nature*, 441(7097):1068–1074, 2006.

[59] Juergen A Knoblich. Mechanisms of asymmetric stem cell division. *Cell*, 132(4):583–597, 2008.

[60] Jonathan S Packer, et al. A lineage-resolved molecular atlas of c. elegans embryogenesis at single-cell resolution. *Science*, 365(6459), 2019.

[61] Tamar Hashimshony, et al. Spatiotemporal transcriptomics reveals the evolutionary history of the endoderm germ layer. *Nature*, 519(7542):219–222, 2015.

[62] F Alexander Wolf, et al. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

[63] Rahul Satija, et al. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33 (5):495–502, 2015.

[64] Leland McInnes, et al. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[65] Rémi Flamary, et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021.

[66] Gioele La Manno, et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018.

[67] Volker Bergen, et al. Generalizing rna velocity to transient cell states through dynamical modeling. *Nature biotechnology*, 38(12):1408–1414, 2020.

[68] Bernhard Reuter, et al. Generalized markov modeling of nonreversible molecular kinetics. *The Journal of chemical physics*, 150(17):174103, 2019.

[69] Bernhard Reuter, et al. Generalized markov state modeling method for nonequilibrium biomolecular dynamics: exemplified on amyloid β conformational dynamics driven by an oscillating electric field. *Journal of Chemical Theory and Computation*, 14(7):3579–3594, 2018.

[70] Wen-Kang Shen, et al. Animaltfdb 4.0: a comprehensive animal transcription factor database updated with variation and expression annotations. *Nucleic Acids Research*, 51(D1):D39–D45, 2023.

[71] Manu Setty, et al. Characterization of cell fate probabilities in single-cell data with palantir. *Nature biotechnology*, 37(4):451–460, 2019.

[72] David Van Dijk, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174 (3):716–729, 2018.

[73] Simon N Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semi-parametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1):3–36, 2011.

[74] Bo Hu, et al. Origin and function of activated fibroblast states during zebrafish heart regeneration. *Nature genetics*, 54(8):1227–1237, 2022.

[75] Rahul Satija, et al. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33 (5):495–502, 2015.

[76] Geoffrey Schiebinger, et al. Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming. *Cell*, 176(4):928–943, 2019.