nature biotechnology

Article

https://doi.org/10.1038/s41587-024-02186-3

Gene trajectory inference for single-cell data by optimal transport metrics

Received: 19 December 2022

Accepted: 26 February 2024

Published online: 05 April 2024

Check for updates

Rihao Qu[1,2,3,11], Xiuyuan Cheng[4,11], Esen Sefik[3], Jay S. Stanley III[5], Boris Landa[5], Francesco Strino[6], Sarah Platt[2,7], James Garritano[5], Ian D. Odell[3,7], Ronald Coifman[5,8,9], Richard A. Flavell[3,10,12], Peggy Myung[2,7,12] & Yuval Kluger[1,2,5,12]

Single-cell RNA sequencing has been widely used to investigate cell state transitions and gene dynamics of biological processes. Current strategies to infer the sequential dynamics of genes in a process typically rely on constructing cell pseudotime through cell trajectory inference. However, the presence of concurrent gene processes in the same group of cells and technical noise can obscure the true progression of the processes studied. To address this challenge, we present GeneTrajectory, an approach that identifies trajectories of genes rather than trajectories of cells. Specifically, optimal transport distances are calculated between gene distributions across the cell–cell graph to extract gene programs and define their gene pseudotemporal order. Here we demonstrate that GeneTrajectory accurately extracts progressive gene dynamics in myeloid lineage maturation. Moreover, we show that GeneTrajectory deconvolves key gene programs underlying mouse skin hair follicle dermal condensate differentiation that could not be resolved by cell trajectory approaches. GeneTrajectory facilitates the discovery of gene programs that control the changes and activities of biological processes.

Dynamic gene expression changes often specify mechanisms through which cells determine state and function. Indeed, tightly regulated gene cascades underlie a myriad of fundamental processes, such as cell cycle (CC)/mitosis[1–4] and tissue/organ differentiation[5–8]. With the emergence of single-cell RNA-sequencing (scRNA-seq) platforms, cell trajectory inference techniques[9–19] are widely applied to study the cellular dynamics of biological processes. These techniques use single-cell whole-transcriptome data to organize cells into lineages and infer a unidimensional latent variable (that is, pseudotime[20]) that describes a cell's position along a lineage process. After pseudotime construction, gene dynamics underlying a biological process can be inferred by tracking the changing patterns of their expression levels along the cell pseudotime[12,15,21].

However, when cells undergo multiple processes in parallel (for example, CC coupled with cell differentiation[22] or circadian clock[23]) and each process is governed by a different set of genes, cell pseudotime learned by organizing cells using the collective genes becomes less informative, as it mixes the effects of multiple processes. Mathematically, when multiple processes that are not strongly correlated with each other co-occur in the same group of cells, cell geometry

[1]Computational Biology & Bioinformatics Program, Yale University, New Haven, CT, USA. [2]Department of Pathology, Yale University School of Medicine, New Haven, CT, USA. [3]Department of Immunobiology, Yale University School of Medicine, New Haven, CT, USA. [4]Department of Mathematics, Duke University, Durham, NC, USA. [5]Program in Applied Mathematics, Yale University, New Haven, CT, USA. [6]PCMGF Limited, Watford, UK. [7]Department of Dermatology, Yale University School of Medicine, New Haven, CT, USA. [8]Department of Mathematics, Yale University, New Haven, CT, USA. [9]Department of Electrical Engineering, Yale University, New Haven, CT, USA. [10]Howard Hughes Medical Institute, Yale University School of Medicine, New Haven, CT, USA. [11]These authors contributed equally: Rihao Qu, Xiuyuan Cheng. [12]These authors jointly supervised this work: Richard A. Flavell, Peggy Myung, Yuval Kluger. e-mail: yuval.kluger@yale.edu

(determined by these processes) cannot be effectively parametrized by a common single latent variable. Hence, organizing cells into unidimensional lineages is no longer appropriate.

To address this challenge, we propose GeneTrajectory, an approach to studying dynamic processes that does not rely on unidimensional parameterization of the cell manifold. GeneTrajectory allows us to deconvolve multiple, independent processes with sequential gene dynamics. In contrast to cell trajectory approaches, GeneTrajectory constructs trajectories of genes rather than trajectories of cells. Our algorithm dissects out gene programs from the whole transcriptome, eliminating the need for initial cell trajectory construction or the specification of the initial and terminal cell states for each process. Using this method, genes that sequentially contribute to a given biological process can be extracted and organized into a respective gene trajectory that reveals the successive order of gene activity.

In this work, we begin by showing GeneTrajectory's efficacy for unraveling gene dynamics through simulation experiments and application to a human myeloid lineage dataset. Subsequently, we use our approach on a mouse embryonic skin dataset to demonstrate that GeneTrajectory can resolve critical cell state transitions during the early-stage development of hair follicles[5,24]. Our results indicate that GeneTrajectory extracts gene geometry without the need for constructing cell pseudotime, revealing independent trajectories of concurrent processes that are otherwise obscured by cell pseudotime approaches.

Results

Computing optimal transport between genes over the cell graph

A progressive dynamic biological process is usually governed by a finely regulated gene cascade[25–27], in which genes are activated and deactivated in a temporal order along the process, dictating the transcriptomic changes of underlying cell states. Moreover, cells can participate in multiple processes simultaneously, either in a dependent or independent manner. For instance, we illustrate two contrasting scenarios by considering the concurrence of a linear process (for example, differentiation) and a cyclic process (for example, CC; Fig. 1a). When these two processes are strictly dependent on each other, they can be parameterized by a common latent variable and result in a one-dimensional cell curve. In this scenario, it is straightforward to assign a meaningful pseudotime for the cells by ordering them along the curve. However, deconvolving genes into two processes and retrieving their pseudotemporal order in each process is not immediately apparent, which requires additional postprocessing (for example, clustering gene dynamics along the cell pseudotime[12]). In contrast, when these two processes are independent, cells fall into a manifold (as a Cartesian product of these two processes) with an intrinsic dimension >1. These processes do not share a common latent variable, thus gene dynamics inference based on unidimensional interpolation along the cell–cell manifold is no longer appropriate. In practice, the weak and stochastic nature of the dependency between concurrent biological processes can complicate the extraction of the cell path and the construction of cell pseudotime.

Here we present GeneTrajectory, an approach to inferring gene processes through learning the gene–gene geometry without one-dimensional parameterization of the cell manifold (Fig. 1b). Specifically, GeneTrajectory quantifies the distance of genes based on their expression distributions over a cell graph using optimal transport (OT) metrics (Fig. 1d). Previously, OT metrics (for example, Wasserstein distance) have been applied in a wide range of scenarios in single-cell analysis, including (1) defining a distance measure between cells[28,29] or cell populations[30], (2) constructing cell trajectories[31,32], (3) spatial reconstruction of single-cell transcriptome profiles[33,34] and (4) multi-omics data integration[35]. In these works, the dissimilarity was quantified either between a pair of cells or between a pair of cell populations. In our work, we distinctively define the graph-based Wasserstein distance between pairs of genes to study their underlying pseudotemporal dynamics. Specifically, we normalize the expression of a gene into a probabilistic distribution over cells and then compute the Wasserstein distances between gene distributions in the cell graph (Fig. 1d). Here the cell graph is constructed in a way that provides a representation of cells, which preserves the cell manifold structure in the high-dimensional space (Fig. 1c). In this construction, the graph-based Wasserstein distance between pairwise gene distributions has the following characteristics: (1) it takes into account the geometry of cells; that is, it assigns a higher cost to transport a point mass from one cell to a distant cell as compared to its adjacent neighbors. (2) It prevents the transport across the ambient cell space, which is often a problematic issue when using spatial distance measures (for example, the Euclidean distance in the cell space).

In our approach, the computation of gene–gene Wasserstein distances is based on the following two steps (Table 1):

- Construct a cell graph. As an initial step, we learn a reduced-dimensional cell embedding that can capture and represent the cell manifold structure in the original high-dimensional space. Next, we construct a k-nearest neighbor (kNN) graph of cells based on their relative distances in the cell embedding (Fig. 1c). This establishes a cell–cell connectivity map that serves as the 'roadmap' for transporting gene distributions in the next step. Here, for a given pair of cells u and v, we search for the shortest path connecting them in the kNN cell graph and denote its length as the graph distance $d_G(u, v)$ between cells u and v. This graph distance will be used to define the cost of transporting a point mass between cells u and v in the next step.
- Compute gene–gene Wasserstein distances over the cell graph. We model the expression level of genes as discrete distributions on the cell graph. Specifically, we divide the original expression level of a given gene in each cell by the sum of its expression level in all cells. We then define the distance between two gene distributions by the graph-based Wasserstein-p distance (W_p distance, $1 \le p < \infty$; Fig. 1c,d). Accordingly, the transport cost between cells u and v is defined as $C_{u,v} = d_G(u, v)^p$. Here p is a user-defined parameter, and $p = 1$ refers to the well-known Earth Mover's distance. Algorithmic details are described in 'Step 2. Compute graph-based Wasserstein distances between genes'.

In practice, computing the Wasserstein distance between all pairwise gene distributions can be computationally expensive. When the cell graph is large, the time cost for finding the OT solution increases exponentially. In our framework, we have designed two strategies to accelerate the computation based on (1) cell graph coarse-graining, and (2) gene graph sparsification (details in 'Step 2. Compute graph-based Wasserstein distances between genes').

Gene trajectory construction

The gene–gene Wasserstein distance captures the pseudotemporal relations of genes in the sense that if two genes are activated consecutively along a biological process, their distributions are expected to have a substantial overlap in the cell graph and thus have a small Wasserstein distance between each other (Fig. 1e). To visualize the geometry of all genes, we convert pairwise gene–gene Wasserstein distances into gene–gene affinities and use diffusion map to get a low-dimensional representation of genes. If dynamical cascades of gene activation and deactivation exist in the data, viewing the gene embedding by a combination of leading diffusion map eigenvectors delineates trajectories of genes (Fig. 1f). Each trajectory is linked with a specific gene program that dictates the underlying biological process.

In our approach, the extraction of gene trajectories is performed in a sequential manner (Fig. 1g). To identify the first trajectory, we search

https://doi.org/10.1038/s41587-024-02186-3

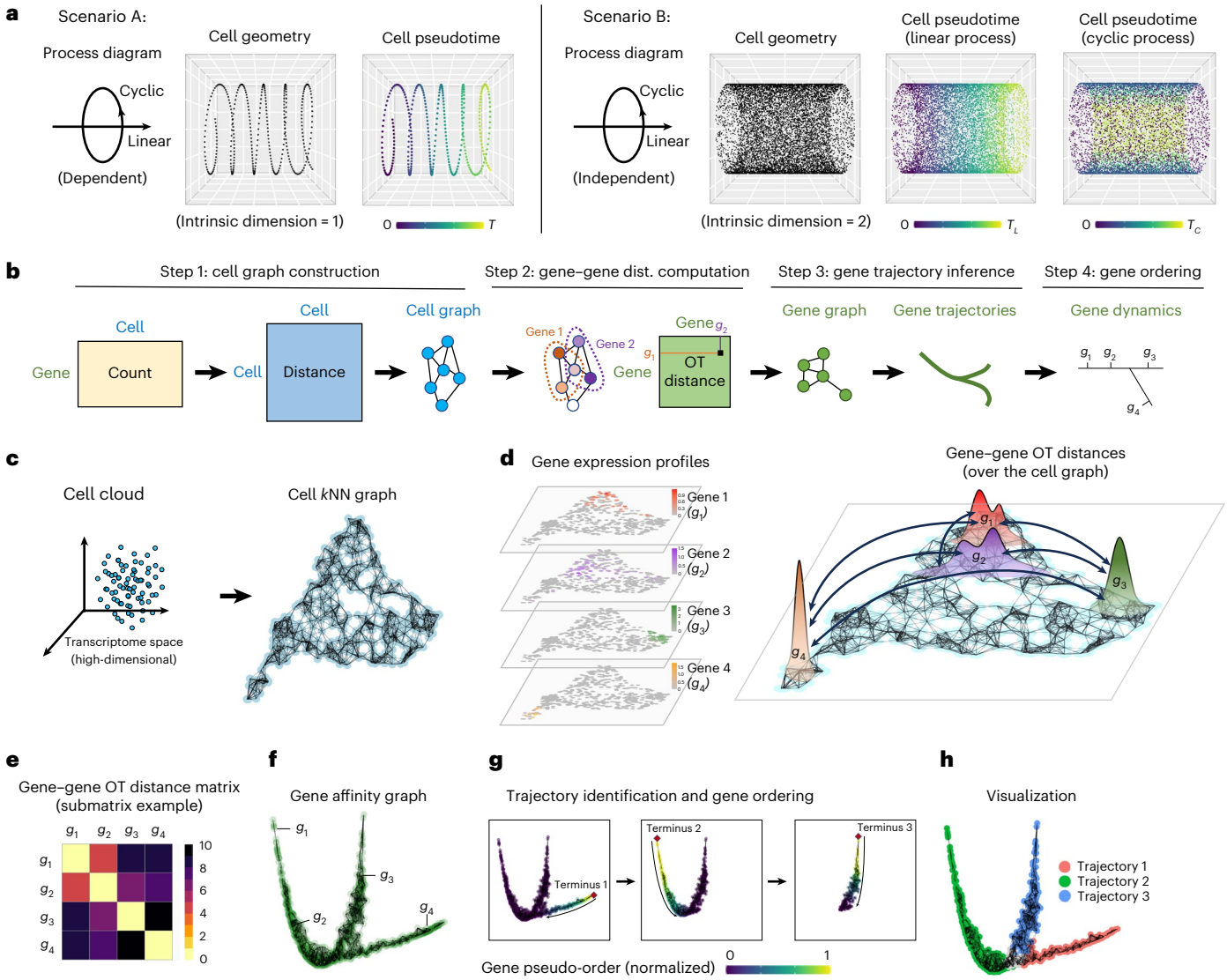


Fig. 1 | Overview of GeneTrajectory. a, Illustration of two scenarios when a linear process and a cyclic process are dependent or independent of each other, resulting in cell manifolds with different intrinsic dimensions and requiring distinct pseudotime parametrizations. **b**, Schematic representation of the major workflow of GeneTrajectory. **c**, Construction of cell *k*NN graph. **d**, Computation of graph-based OT (Wasserstein) distances between paired gene distributions (four representative genes are shown) over the cell graph. Gene distributions are defined by their normalized expression levels over cells. **e**, Heatmap of OT (Wasserstein) distances for genes g_1–g_4 in **d**. **f**, Construction of gene graph based on gene–gene affinities (transformed from gene–gene Wasserstein distances). **g**, Sequential identification of gene trajectories using a diffusion-based strategy. The initial node (terminus 1) is defined by the gene with the largest distance from the origin in the diffusion map embedding. A random-walk procedure is then used on the gene graph to select the other genes that belong to the trajectory terminated at terminus 1. After retrieving genes for the first trajectory, we identify the terminus of the subsequent gene trajectory among the remaining genes and repeat the steps above. This is done iteratively until all detectable trajectories are extracted. **h**, Diffusion map visualization of gene trajectories.

for the gene that has the largest distance from the origin of diffusion map embedding, which serves as the terminus of the first gene trajectory. To retrieve the other genes along the first trajectory, we take that terminus gene as the starting point of a diffusion process. Specifically, we assign a unit point mass to that gene and then diffuse the mass to the other genes. As the probability mass propagates along the gene trajectory from its terminus, the trajectory can be retrieved by a heuristic thresholding procedure ('Step 3. Construct gene trajectories'). After retrieving genes for the first trajectory, we identify the terminus of the subsequent gene trajectory among the remaining genes and iterate the same procedure, until all detectable gene trajectories are extracted (Fig. 1g,h).

To order the genes along a given trajectory, we retain only these genes to recompute a diffusion map embedding based on their pairwise gene–gene Wasserstein distances. The obtained first nontrivial eigenvector of the diffusion map embedding provides an intrinsic ordering of the genes along that trajectory[36,37].

To examine how the gene order along a given gene trajectory is reflected over the cell graph, we can track how these genes are expressed across different regions in the cell embedding. Specifically, we first group genes along each gene trajectory into successive bins and generate a cell embedding 'snapshot' for each bin. In each snapshot, we color the cells according to the fraction of genes (from that bin) that they express. By plotting the expression level of each gene bin on the cell embedding, we can visualize how the underlying biological process progresses across cell populations.

Assessing GeneTrajectory's performance using simulation

Assuming that a progressive biological process is temporally dictated by a sequence of genes, we simulated several artificial scRNA-seq

https://doi.org/10.1038/s41587-024-02186-3

Table 1 | List of core notations in Methods

u, v	Index of cells
i, j	Index of genes
m	Original number of cells
n	Original number of genes
m'	Reduced number of cells after coarse-graining
$\delta^{(p)}(\rho, \rho')$	Wasserstein-P distance between distributions ρ and ρ'
$d_E(u, v)$	Euclidean distance between cell u and v
$d_G(u, v)$	Graph distance between cells u and v
C	Transport cost matrix on the cell graph. $C_{u,v}$ represents the cost of transport between cell u and v
C'	Transport cost matrix on the coarse-grained cell graph
M	kNN membership matrix for the cell graph. $M(u, a)=1/\lvert a\rvert$ if and only if the cell u belongs to the ath subset, where $\lvert a\rvert$ represents the number of cells in that subset; otherwise $M(u, a)=0$
A	Gene–gene affinity matrix
P	Row-normalized gene–gene affinity matrix (as the random-walk matrix)
S	Diffusion map (spectral) embedding of genes

datasets with a variety of gene dynamics by modeling the change of gene expression over time (Extended Data Fig. 1a,b; 'Workflow of gene dynamics simulation'). Specifically, for a gene involved in a given biological process, we simulate its expected expression level $\lambda(t)$ as a function of time t. For clarity, we note that t represents the pseudotime of a biological process, linked with the cell state (for example, differentiation status) rather than the actual time (for example, specific day of a developmental process). Here we use multiple parameters to account for the heterogeneity of gene expression profiles in single-cell data, including the variation of duration time and expression intensities (details in 'Workflow of gene dynamics simulation'). For each cell state at t along a biological process, we apply a Poisson sampling to generate the observed expression level of each gene by taking $\lambda(t)$ as the mean of Poisson distribution. In these simulation experiments, we know the ground truth of both the pseudotime of each cell in the corresponding biological process and the temporal order of genes that dictate each process. Finally, we incorporate an optional step to account for sequencing depth. This is achieved by sampling a specified number of nonzero entries from the original count matrix. This procedure enables us to generate an artificial dataset with varying levels of missing data.

We first simulated (1) a cycling process in which the change of gene expression shows a periodical pattern over time (Fig. 2a and Extended Data Fig. 1c), and (2) a process with a branching point where it diverges into two different lineages (Fig. 2b and Extended Data Fig. 1d). Inspection of the gene trajectories in these two simulation examples reveals similar layouts with their cell embeddings (Fig. 2a,b). The ordering of genes along each gene trajectory shows a high concordance with the ground truth (Supplementary Table 1).

We next, created two scenarios that simulate a mixture of two concurrent processes (Fig. 2c,d and Extended Data Fig. 1e,f). Specifically, one process mimics cell differentiation (linear or branched in a multilayered fashion), and the other mimics the CC. In these two scenarios, each cell state is determined by two independent hidden variables—a pseudotime along the differentiation process and a pseudotime in the CC. For each process, we simulated an exclusive set of genes with distinct dynamic characteristics (Extended Data Fig. 1e,f; 'Workflow of gene dynamics simulation'), generating a cell manifold with a cylinder-shaped or a coral-shaped structure (Fig. 2c,d). In both scenarios, our approach deconvolves the original mixture of two processes into two gene trajectories representing a (linear or tree-like) differentiation process and a (circular) CC process. Along each trajectory, genes are ordered in high concordance with the ground truth (Supplementary Table 1), indicating that GeneTrajectory allows deconvolving a mixture of biological processes that take place simultaneously in the same group of cells.

GeneTrajectory resolves myeloid gene dynamics

We demonstrate GeneTrajectory's application using myeloid lineage differentiation, a classical biological system with a well-defined bifurcation of two major lineages[38,39]. We extracted human myeloid cells from a public 10× Genomics peripheral blood mononuclear cell (PBMC) dataset and identified four cell types based on canonical markers (Fig. 3a and Extended Data Fig. 2b,c). These included $CD14^+$ monocytes, intermediate monocytes with high expression of HLA-DR (Human Leukocyte Antigen – DR isotype), $CD16^+$ monocytes and myeloid type-2 dendritic cells. The UMAP visualization of the cell embedding shows a continuum of cell states underlying myeloid lineage genesis, comprising monocyte maturation and dendritic cell differentiation. Human monocyte maturation involves the upregulation of CD16 on a subset of $CD14^+$ classical monocytes[40]. Specifically, $CD14^+$ monocytes first transition into an intermediate subset of monocytes and then differentiate into $CD16^+$ nonconventional monocytes with distinct effector functions.

We used GeneTrajectory to identify three gene trajectories, each representing a specific aspect of the myeloid lineage differentiation process (Fig. 3b). Viewing the gene bin plots of Trajectory 1 illustrates that a subset of $CD14^+$ monocytes start a differentiation cascade and gradually shift toward $CD16^+$ monocytes, which suggests Trajectory 1 captures the gene dynamics underlying the early stage of monocyte maturation (Fig. 3c). Notably, *CLEC5A*, *RETN*, *CCR2* and *SELL* (CD62L) are known to be associated with the initial $CD14^+$ monocyte cellular state[40] and are highlighted as part of Trajectory 1 (Fig. 3b). Subsequently, the ordering of genes that define Trajectory 2 provides a pseudotemporal view on the later stage of $CD16^+$ monocyte differentiation (Fig. 3d). This process is primarily driven in response to cytokine colony-stimulating factor 1 (CSF1) and requires *CSF1R*[41]. While ordered after *CSF1R*, *ICAM2* is known to be constitutively expressed in $CD16^+$ monocytes and is necessary for their patrolling ability across the endothelium of blood vessels[41,42]. Coming toward the end, *C1QA*, *C1QB*[43] and *FCGR3A* markers broadly expressed by fully differentiated $CD16^+$ monocytes are identified. In addition, we retrieved a third gene trajectory (Trajectory 3) that marks the differentiation of type-2 dendritic cells as a distinct myeloid lineage (Fig. 3e). Myeloid type-2 dendritic cells have the following two subsets: $CD14^+$ and $CD14^-$. Specifically, the $CD14^+$ subset shares overlapping features with $CD14^+$ monocytes, whereas the $CD14^-$ subset is delineated here as corresponding with a separate gene trajectory[44]. In contrast to $CD16^+$ monocytes, these $CD14^-$ dendritic cells differentiate in response to *GMCSF* and *IL4*, in line with expression of *CCR5*, *CD2*, *CLEC10A*, *CD72*, *CD1C* and *PKIB*[45] (Fig. 3b and Extended Data Fig. 2a). Notably, GeneTrajectory does not necessitate specification of the initial and terminal cell states for each process, while those states can be automatically revealed by inspecting the cell population that express the endpoint genes of each gene trajectory.

Deconvolving gene processes in dermal condensate genesis

Hair follicle dermal condensates (DCs) emerge in the skin dermis around embryonic day 14.5 (E14.5) and have an essential role in hair follicle formation. Morphogenetic signals, including Wnt/β-catenin signaling, are critical for the differentiation of DC cells[5,24,46]. We collected skin from E14.5 wild-type (WT) and paired K14Cre; $Wntless^{fl/fl}$ (Wls) mutant embryos for scRNA-seq (Fig. 4a). The genetic defect in the mutant results in attenuated dermal Wnt signaling and a lack of DCs and hair follicles[47,48] (Fig. 4b–c and Extended Data Fig. 3a).

https://doi.org/10.1038/s41587-024-02186-3

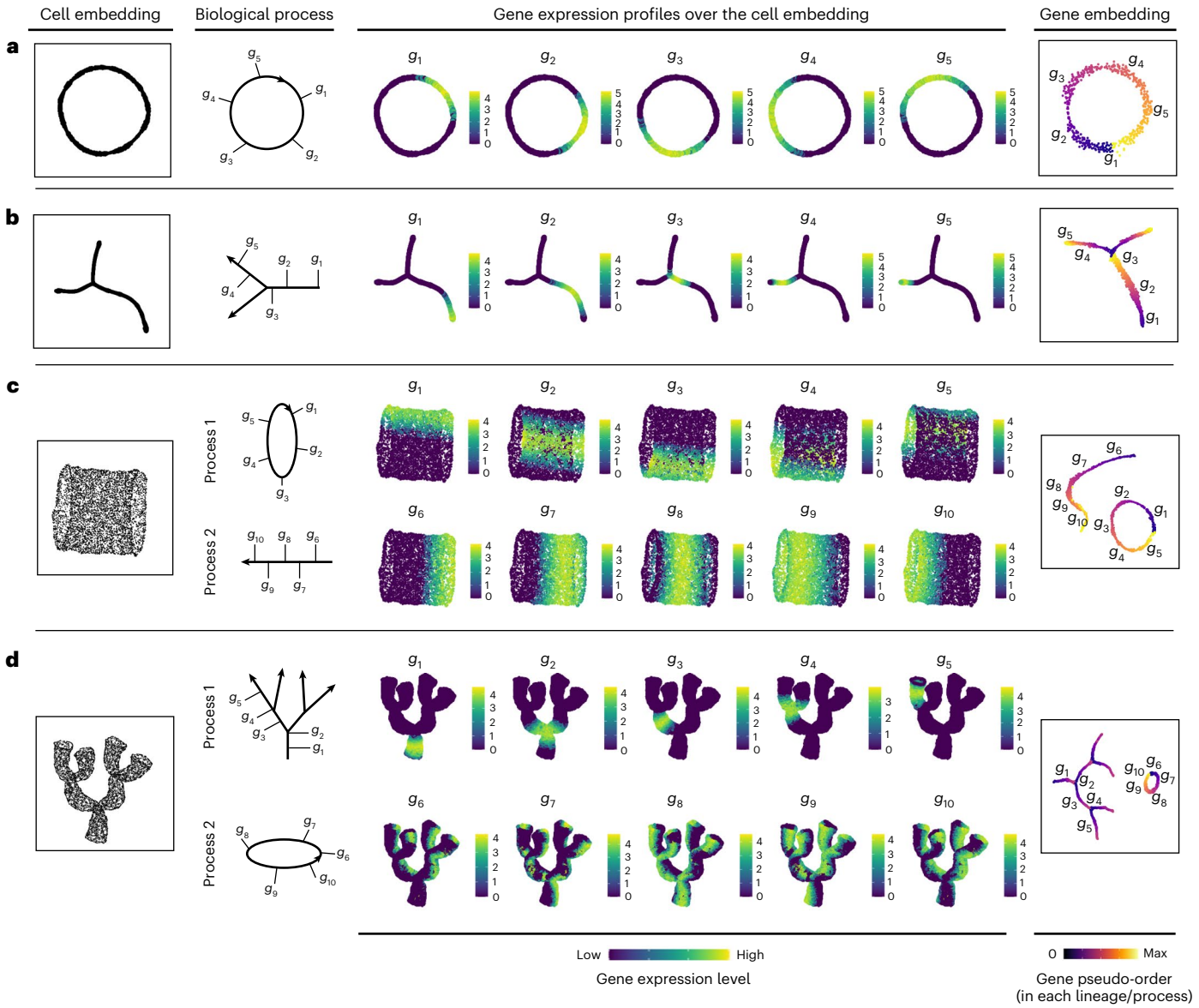


Fig. 2 | GeneTrajectory performance assessment based on simulation experiments. a, Simulation of a cycling process (CC). The cell embedding and gene embedding showcase the same topology that has a ring-shaped structure. **b**, Simulation of a differentiation process with two lineages. The cell embedding and gene embedding showcase the same topology that has a bifurcating tree structure. **c**, Simulation of a linear differentiation process coupled with CC. The cell embedding and gene embedding showcase distinct topologies. Cells are organized along a cylinder-shaped manifold that has an intrinsic dimension of two. Genes that contribute to the two processes are deconvolved and organized along a ring-shaped trajectory and a linear trajectory. **d**, Simulation of a multilevel lineage differentiation process coupled with CC. The cell embedding and gene embedding showcase distinct topologies. Cells are organized along a coral-shaped manifold that has an intrinsic dimension of two. Genes that contribute to the two processes are deconvolved and organized along a ring-shaped trajectory and a multilayered-tree-structured trajectory. (**a** and **b** are visualized by *t*-SNE (*t*-distributed stochastic neighbor embedding); **c** and **d** are visualized by UMAP (uniform manifold approximation and projection). The first column shows the cell embedding; the second column delineates the progressive dynamics of the simulated process with five genes selected along each process; the third to seventh columns show the expression of selected genes in the cell embedding following their pseudotemporal order; the eighth column displays the embedding of genes, colored by the ground truth of gene pseudo-order).

Visualizing cells on UMAP reveals a continuum of cell states composed of lower dermal cells (*Dkk2*$^+$) and Wnt-activated upper dermal cells (*Dkk1*$^+$ or *Lef1*$^+$), which include DC cells (*Sox2*$^+$; Fig. 4c and Extended Data Fig. 3a). We applied GeneTrajectory to the combined dermal cell populations and extracted three prominent gene trajectories that correspond to lower dermis (LD) differentiation, DC differentiation and CC (Fig. 4d). Specifically, we examined the CC gene ordering by checking the distribution of genes associated with different CC phases along the gene trajectory (Extended Data Fig. 3b). Wnt signaling pathway genes (for example, *Lef1* and *Dkk1*) and SHH (Sonic Hedgehog) signaling pathway genes (for example, *Ptch1* and *Gli1*), two morphogenetic signals shown to be necessary and sufficient for DC differentiation[5], are present in the DC gene trajectory. Notably, the upper dermal cell embedding integrates a mixture of biological processes (CC and DC differentiation) that co-occur within the same cell population. By using GeneTrajectory, each biological process can be deconvolved from the other and independently examined. Viewing the gene bin plots for the CC and DC gene trajectories together reveals that DC progenitors actively proliferate throughout all stages and then exit the CC at the terminus of DC differentiation (Fig. 4e). These data imply that DC cells are the immediate progeny of proliferative progenitors in the upper dermis.

https://doi.org/10.1038/s41587-024-02186-3

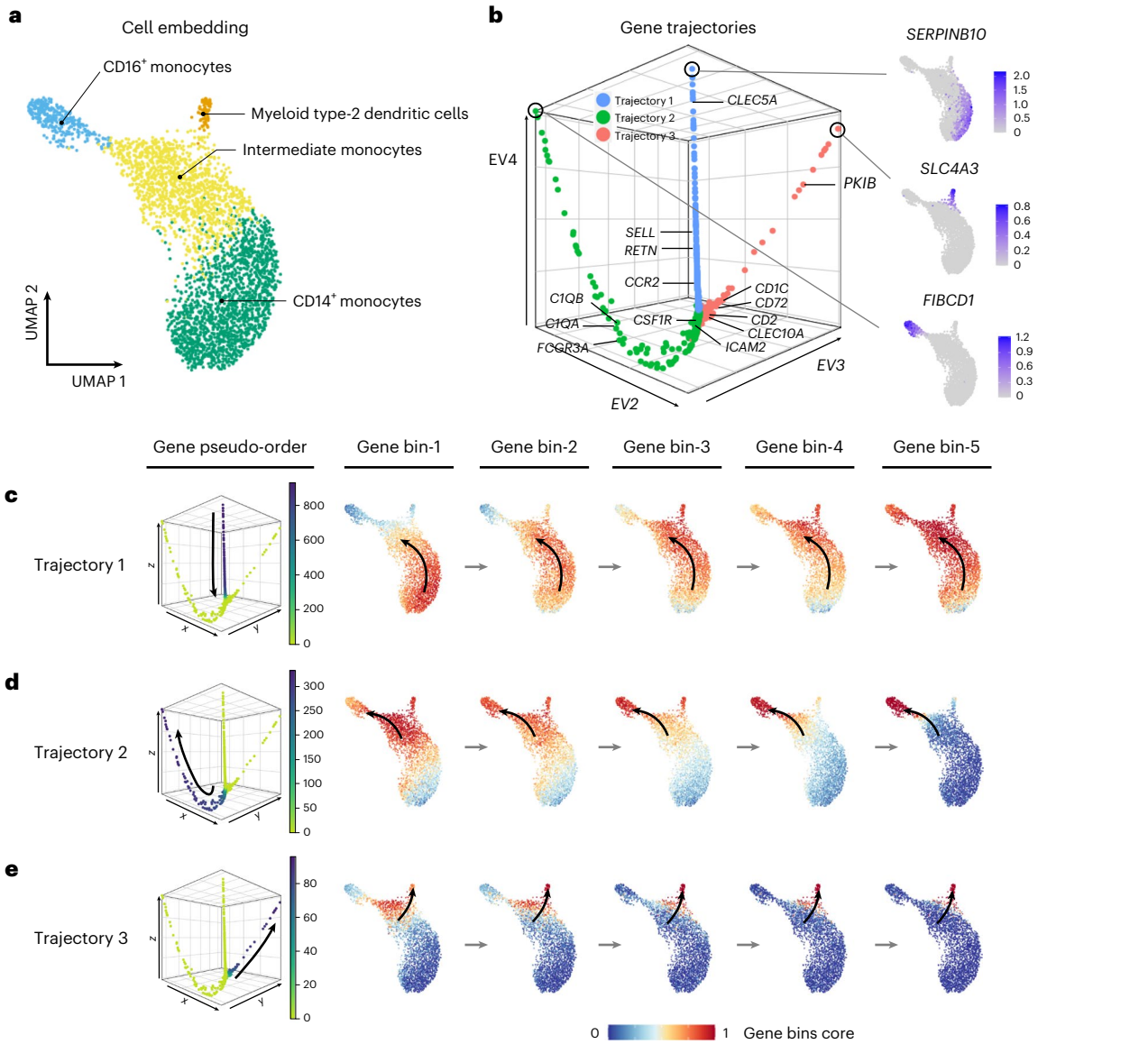


Fig. 3 | Gene trajectory inference on a myeloid scRNA-seq dataset. a, UMAP of myeloid cell population colored by cell types. **b**, DM (Diffusion Map) embedding of the gene graph based on gene–gene Wasserstein distances, visualized using the three leading nontrivial eigenvectors. Three prominent gene trajectories are identified. Expression profiles of the genes at the terminus of these trajectories are shown, each indicating a distinct myeloid cell state. **c**–**e**, Gene bin plots showing the gene expression activities along each gene trajectory (Trajectory 1 (**c**), Trajectory 2 (**d**) and Trajectory 3 (**e**)) over the cell embedding. Genes along each trajectory are ordered and then split into five equal-sized bins. Gene bin score is defined by the proportion of genes (from each bin) expressed in each cell. Arrows indicate the path of gene distribution progression over the cells.

GeneTrajectory identifies biological defects in Wls mutant

We next use GeneTrajectory to examine how attenuated Wnt signaling affects the DC differentiation gene program. By tracking the expression status of genes along each gene trajectory in the WT and mutant cells (Fig. 5a), we did not detect a difference between the mutant and control with respect to the CC and LD gene trajectories (Extended Data Fig. 4a,b). However, along the DC gene trajectory, Wls mutant cells fail to express later-stage DC markers, indicating the defect is specific to DC differentiation. Visualizing gene bin plots for the DC gene trajectory shows that mutant cells fail to progress in the DC differentiation process (Fig. 5e,f).

Moreover, gene trajectory inference allows us to define a specific stage of cell state transition by specifying a gene window along the gene trajectory. To understand how genetic mutation affects DC differentiation, we use GeneTrajectory to stratify the pool of progenitors by different stages of DC differentiation. Considering genes in each bin as markers indicative of a specific DC differentiation stage, we first identified cells that express more than half of the genes in the last bin as cells in the final stage of differentiation (stage 7). Among the remaining cells, we identified the cells that express more than half of the genes in the sixth bin as progenitors in stage 6. We repeated this procedure iteratively until all seven gene bins were associated with their matched cell populations (Fig. 5e,f and Extended Data Fig. 4c).

By comparing the composition of progenitors in different stages between the WT and Wls mutant, we found that mutant cells fail to express most of the markers after stage 4, when key markers in Wnt (for example, *Lef1*) and SHH (for example, *Gli1* and *Ptch1*) signaling pathways are upregulated in the WT condition (Fig. 5e,f and Supplementary Table 2). The average expression level of Wnt target genes is uniformly lower in the mutant than in the WT condition (Fig. 5c and Extended Data Fig. 4d), while the proportion of cells in the G1 phase of the CC is higher in the mutant across all stages (Fig. 5b). Consistent with this,

https://doi.org/10.1038/s41587-024-02186-3

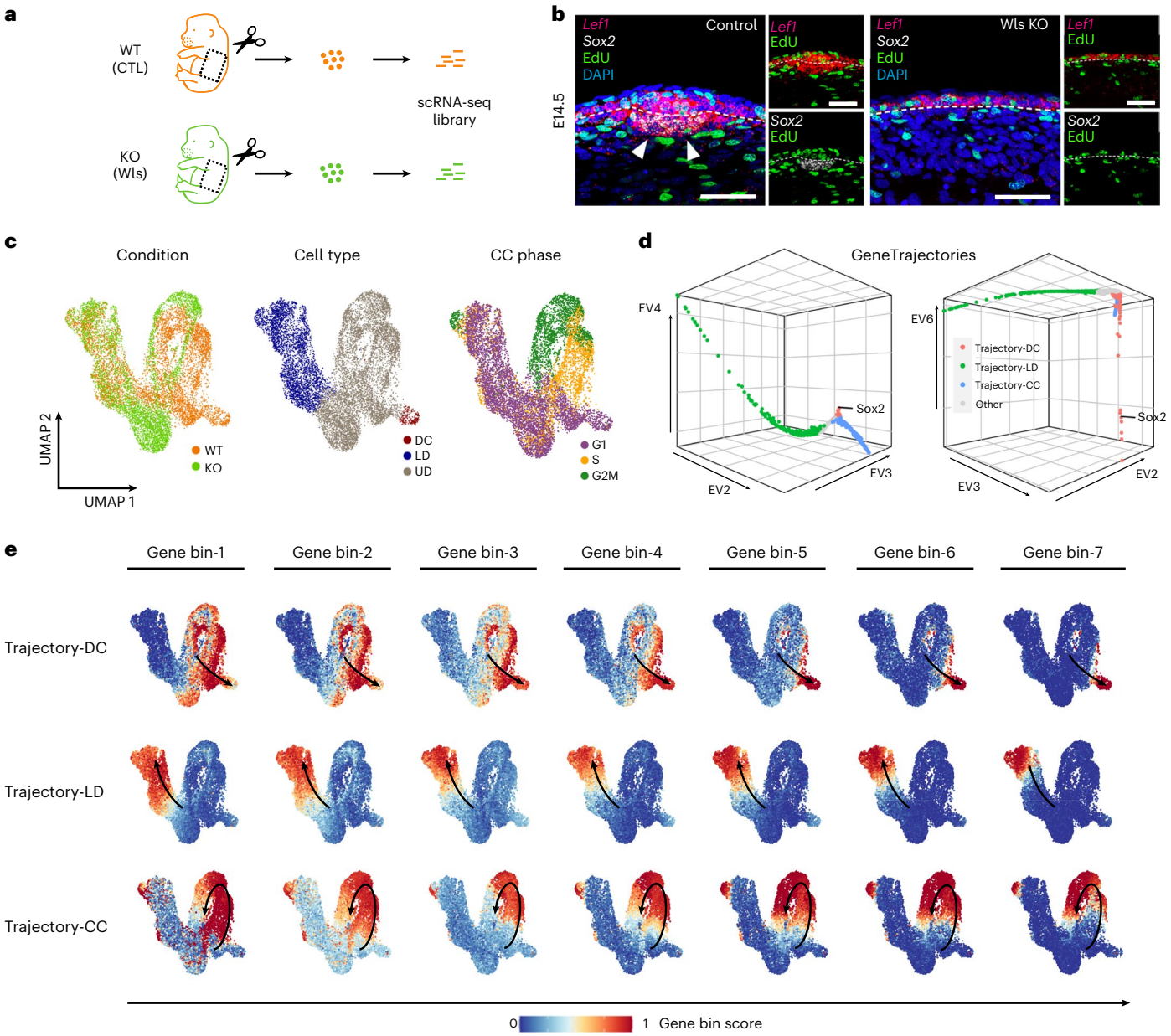


Fig. 4 | GeneTrajectory deconvolves two mixed processes during DC genesis. **a**, Experimental design of extracting skin tissue from a pair of WT and Wls KO embryos at day E14.5 for scRNA-seq. **b**, FISH images (scale bar = 50 µm) showing the spatial distribution of *Lef1*, *Sox2*, EdU nucleotide and DAPI in the upper dermis of WT and Wls KO. EdU is a nucleotide that is incorporated by cells in the S phase of the CC. $n = 8$ (WT) and $n = 9$ (KO) embryos examined over four biologically independent experiments with similar results. **c**, UMAP of cells color coded by cell types, conditions and CC phases. **d**, DM (Diffusion Map) embedding of the gene graph to visualize three identified gene trajectories (two different combinations of leading nontrivial eigenvectors are displayed). **e**, Gene bin plots delineating the dynamics of each process (including DC differentiation, LD differentiation and CC), in which genes along each trajectory are split into seven equal-sized bins. Gene bin score is defined by the proportion of genes (from each bin) expressed in each cell. Arrows indicate the path of gene distribution progression over the cells. Upper dermis, UD; lower dermis, LD; dermal condensate, DC; cell cycle, CC; wildtype, WT; control, CTL; knockout, KO.

the rate of EdU nucleotide incorporation (S phase) is lower in the Wls mutant (Figs. 4b and 5d). These data suggest that higher levels of Wnt signaling are necessary to maintain a normal rate of cell proliferation across the DC differentiation process until DC progenitors exit the CC at stage 7. These results also raise the notion that dermal proliferation itself may directly regulate dermal cell state progression during the DC differentiation process.

Comparison of GeneTrajectory to cell trajectory methods

We compared GeneTrajectory with five cell trajectory methods as follows: Monocle 2 (ref. 16), Monocle 3 (ref. 10), Slingshot[9], PAGA[11] and CellRank[15]. In the simulations of two co-occurring processes, we assessed performance by calculating the Spearman correlation between the gene ordering inferred from each approach and the ground truth. To order genes based on these cell trajectory inference methods, we first constructed the cell pseudotime using their default pipelines ('Comparing GeneTrajectory with cell trajectory methods in terms of gene ordering inference'). Subsequently, we fitted generalized additive models (GAM)[49,50] to find the peak location of each gene expression along the cell pseudotime. The genes were then ordered based on these peak locations. GeneTrajectory achieved the best performance in recovering gene order for both cyclic and linear processes (Fig. 6a,b) in simulation experiments, showing remarkable robustness to variations in cell numbers and sparsity levels of the count matrix.

https://doi.org/10.1038/s41587-024-02186-3

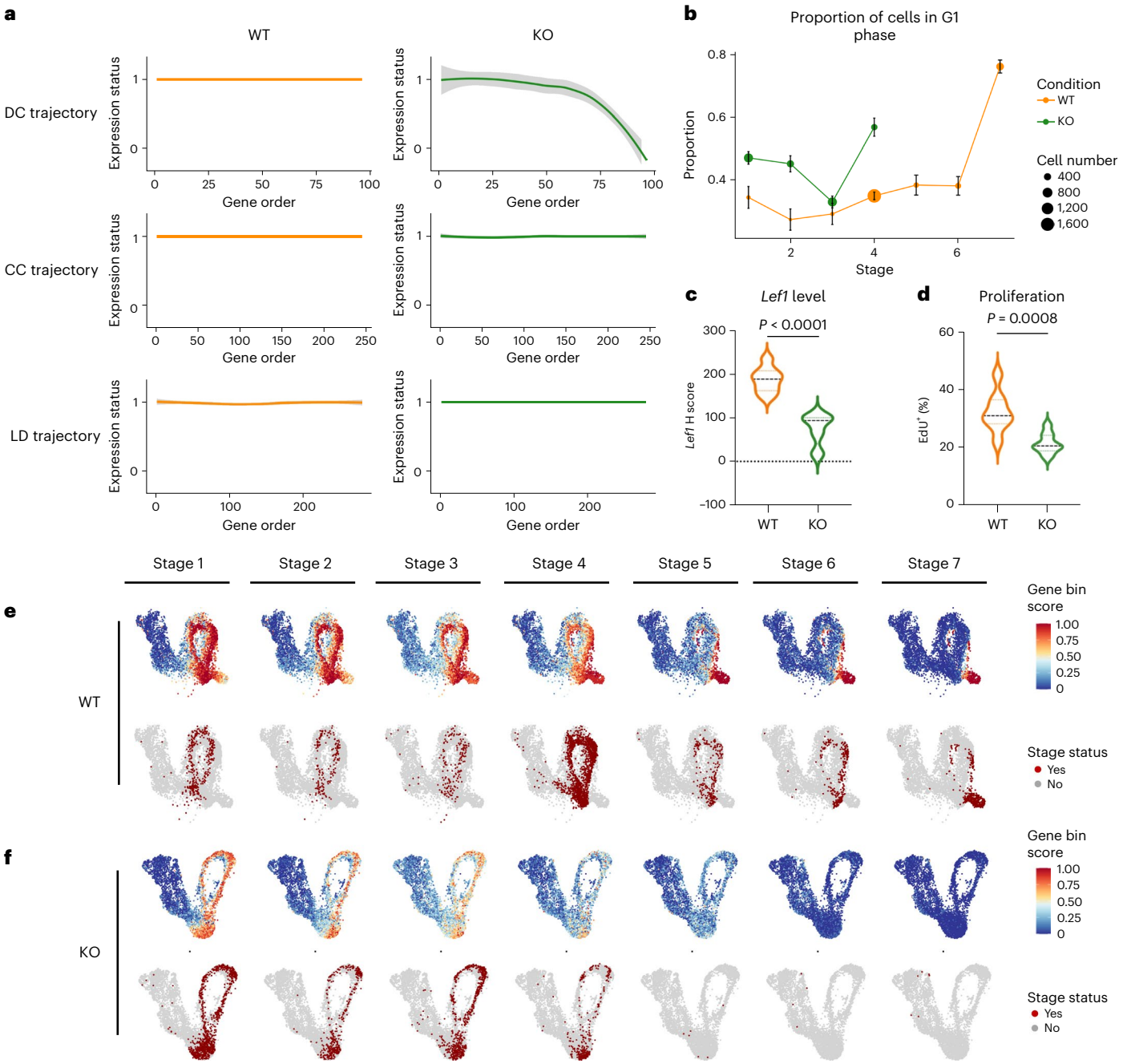


Fig. 5 | Gene dynamics comparative analysis. a, Gene expression status (smoothed) along each gene trajectory in two conditions (0, expressed in fewer than 1% of cells; 1, otherwise). **b**, Change in G1 proportion across seven stages of DC differentiation. Error bar: mean ± s.e., *n* = the number of cells in each stage of the corresponding condition. Dots in stages 5–7 of the KO are omitted when the number of cells is ≤10. **c**, *Lef1* transcript levels (*H* score) quantified by FISH in UD in two conditions. **d**, Percentage of Edu in UD in two conditions. EdU is a nucleotide incorporated by cells in the S phase of CC. **e**, Gene bin plots of the DC gene trajectory over the WT cell embedding. Cells involved in the DC differentiation process are stratified into seven different stages. **f**, Gene bin plots of the DC gene trajectory over the Wls KO cell embedding. Cells involved in the DC differentiation process are stratified into seven different stages. Violin plots: *n* = 8 (WT) and *n* = 9 (KO) embryos examined over four biologically independent experiments. Statistical analysis was performed using two-sided Student's *t* test. Lines indicate 75th, 50th and 25th percentiles.

In our real-world example of DC development, we examined the order of known markers during DC differentiation (Fig. 6c,d). GeneTrajectory recovered the correct ordering—Wnt target genes *Dkk1*/*Grem1*/*Lef1* and *Bmp4* emerge first along this process. Dermal Wnt signaling is known to be required for SHH activation[47,48]. Accordingly, the emergence of Wnt target genes is succeeded by the expression of SHH target genes (*Gli1*/*Ptch1*), which precedes the upregulation of the CC inhibitor, *Cdkn1a*, and terminates with the expression of mature DC markers (*Sox2*/*Sox18*/*Foxd1*). In contrast, SlingShot, Monocle 2 and Monocle 3 were unsuccessful in generating a reasonable sequence for these genes. PAGA failed to generate a distinguishable ordering of later-stage markers. CellRank incorrectly placed the DC marker (*Sox2*) before *Gli1* and failed to define the ordering for *Bmp4*/*Lef1* and *Cdkn1a*.

Moreover, manually regressing out known coexisting biological effects (for example, CC) does not guarantee an accurate recovery of gene dynamics when using cell trajectory inference methods. For instance, in our dermal example, regressing out CC effects resulted in persistent incorrect gene orderings for SlingShot, Monocle 2,

https://doi.org/10.1038/s41587-024-02186-3

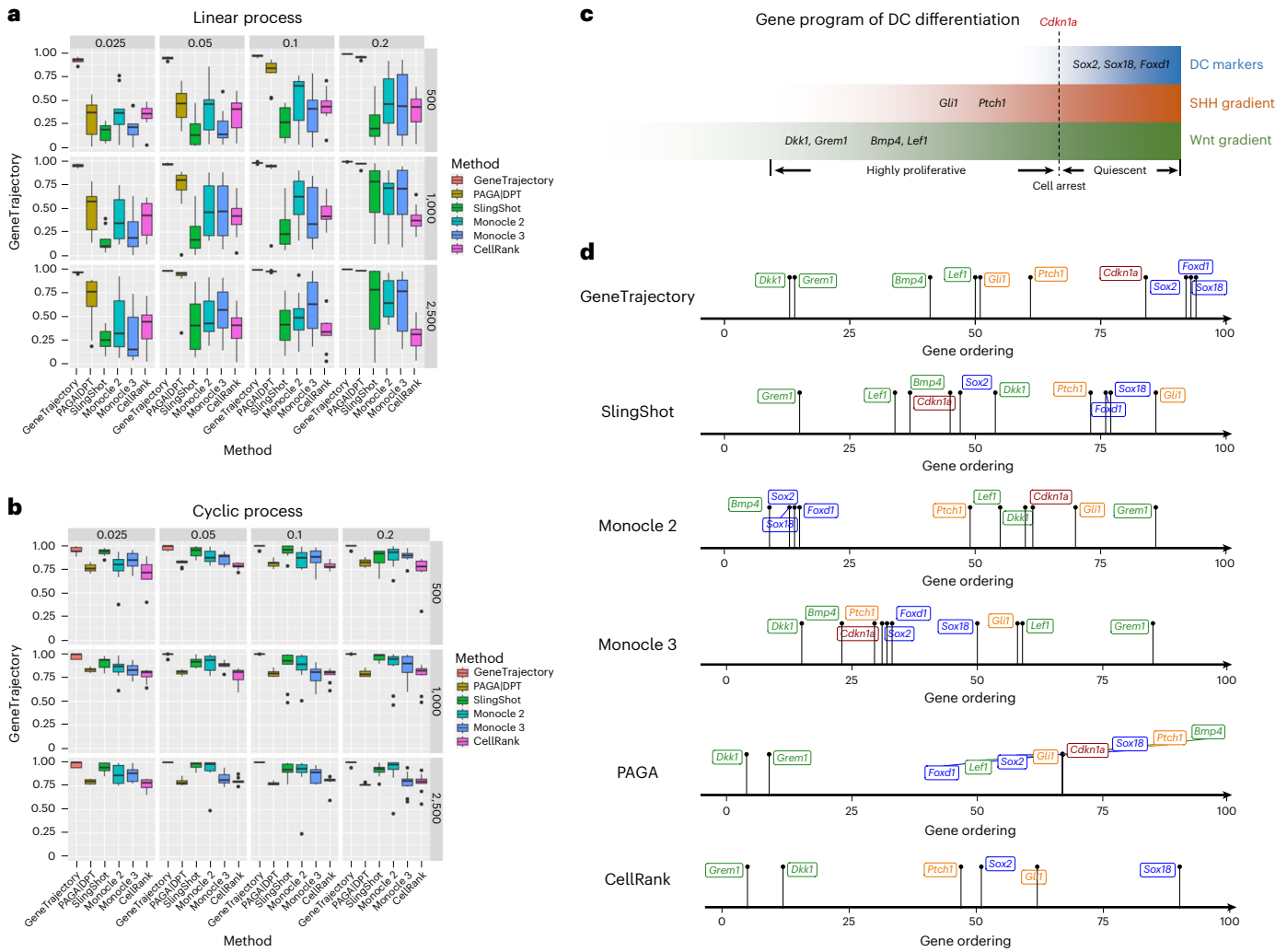


Fig. 6 | GeneTrajectory outperforms other methods in inferring gene ordering along concurrent processes. a,**b**, Comparison of GeneTrajectory with other approaches on simulated data (corresponding to the third simulation example in Fig. 2) of simultaneous linear process (**a**) and cyclic process (**b**), with varying sample size and sparsity level of the count matrix (the numbers in the vertical gray boxes correspond to sample size, and those in the horizontal gray boxes correspond to the percentage of nonzero entries in each count matrix). **c**, Schematic representation of the key genes activated during the DC differentiation process. **d**, Gene ordering results obtained by different methods on the DC genesis data. Box plots: the box represents the IQR, with the line inside the box indicating the median. Whiskers extend to a maximum of 1.5× IQR beyond the box, with outliers represented as individual points. IQR, interquartile range.

Monocle 3, PAGA and CellRank (Extended Data Fig. 5), suggesting that CC regression is not sufficient to deconvolve the intertwined gene dynamics. This underscores the advantage of GeneTrajectory that it is capable of detecting and disentangling multiple gene programs when they are present.

Discussion

We developed GeneTrajectory, an approach for constructing gene trajectories where each trajectory comprises genes organized in a pseudotemporal order that characterizes the transcriptional dynamics of a specific biological process. GeneTrajectory uses optimal-transport-based gene–gene dissimilarity metrics. These metrics naturally leverage the underlying geometry of the cell–cell graph to reveal a coherent relation among genes that are involved in progressive processes. Importantly, GeneTrajectory bypasses the need for constructing cell pseudotime, which is a common requirement in existing methods. This renders it broadly applicable in scenarios where cells do not form into clear lineages.

It is worthwhile to note that cell trajectory inference and gene trajectory inference can complement each other to address different types of questions. Cell trajectory inference aims to define biological processes by lineages of cells, while gene trajectory inference associates each process with a sequence of genes. As demonstrated above, when cells participate in concurrent processes, cell trajectory inference may fail to deconvolve them. Similarly, when one gene participates in multiple biological processes, theoretically, it should be placed at the joint of gene trajectories. However, if that gene is expressed across many cells, it may have a small Wasserstein distance to genes that are homogeneously expressed (uninformative genes). As a result, it will be colocalized with uninformative genes in the gene embedding, causing difficulty for GeneTrajectory to distinguish them. Moreover, there are multiple aspects of our proposed algorithm that could be further refined. For instance, the branch identification procedure requires interactive optimization and might exhibit instability if the branches differ substantially in length and size. In addition, GeneTrajectory cannot automatically infer the directionality of progression along each trajectory. The directionality can be determined by checking whether the endpoint genes in each trajectory are initial stage markers or terminal stage markers of the corresponding process. Another important aspect is that the idea of using the OT distance between genes over

cell–cell graphs could have other potential applications beyond the inference of gene programs and their dynamics. Intuitively, after we compute the gene–gene affinity matrix, we can iteratively improve the organization of cells by an OT distance between the cells over the gene–gene graph. This approach warrants further investigation from theoretical and practical perspectives.

In this work, we demonstrated the utility of GeneTrajectory to unravel gene dynamics using scRNA-seq data. However, our method can be generalized to other single-cell modalities, including but not limited to scATAC–seq[51] and spatial transcriptomics[52]. Specifically, we anticipate that GeneTrajectory can be applied to resolve biological processes using dual modalities[53] at the same time. For instance, we can quantify the pairwise distances between the distributions of gene expression and chromatin accessibility, which facilitates understanding the interplay between epigenetic dynamics and transcriptomic dynamics that underlie biological processes.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41587-024-02186-3.

References

1. Mahdessian, D. et al. Spatiotemporal dissection of the cell cycle with single-cell proteogenomics. *Nature* **590**, 649–654 (2021).
2. Scialdone, A. et al. Computational assignment of cell-cycle stage from single-cell transcriptome data. *Methods* **85**, 54–61 (2015).
3. Skinner, S. O. et al. Single-cell analysis of transcription kinetics across the cell cycle. *eLife* **5**, e12175 (2016).
4. Cao, J., Zhou, W., Steemers, F., Trapnell, C. & Shendure, J. Sci-fate characterizes the dynamics of gene expression in single cells. *Nat. Biotechnol.* **38**, 980–988 (2020).
5. Qu, R. et al. Decomposing a deterministic path to mesenchymal niche formation by two intersecting morphogen gradients. *Dev. Cell* **57**, 1053–1067 (2022).
6. Macaulay, I. C. et al. Single-cell RNA-sequencing reveals a continuous spectrum of differentiation in hematopoietic cells. *Cell Rep.* **14**, 966–977 (2016).
7. Chu, L.-F. et al. Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biol.* **17**, 173 (2016).
8. Chen, R., Wu, X., Jiang, L. & Zhang, Y. Single-cell RNA-seq reveals hypothalamic cell diversity. *Cell Rep.* **18**, 3227–3241 (2017).
9. Street, K. et al. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* **19**, 477 (2018).
10. Cao, J. et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature* **566**, 496–502 (2019).
11. 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).
12. Van den Berge, K. et al. Trajectory-based differential expression analysis for single-cell sequencing data. *Nat. Commun.* **11**, 1201 (2020).
13. Deconinck, L., Cannoodt, R., Saelens, W., Deplancke, B. & Saeys, Y. Recent advances in trajectory inference from single-cell omics data. *Curr. Opin. Syst. Biol.* **27**, 100344 (2021).
14. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.* **37**, 547–554 (2019).
15. Lange, M. et al. CellRank for directed single-cell fate mapping. *Nat. Methods* **19**, 159–170 (2022).
16. Qiu, X. et al. Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods* **14**, 979–982 (2017).
17. Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F. & Theis, F. J. Diffusion pseudotime robustly reconstructs lineage branching. *Nat. Methods* **13**, 845–848 (2016).
18. Setty, M. et al. Characterization of cell fate probabilities in single-cell data with Palantir. *Nat. Biotechnol.* **37**, 451–460 (2019).
19. Lönnberg, T. et al. Single-cell RNA-seq and computational analysis using temporal mixture modeling resolves Th1/Tfh fate bifurcation in malaria. *Sci. Immunol.* **2**, eaal2192 (2017).
20. Tritschler, S. et al. Concepts and limitations for learning developmental trajectories from single cell genomics. *Development* **146**, dev170506 (2019).
21. 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).
22. Ruijtenberg, S. & van den Heuvel, S. Coordinating cell proliferation and differentiation: antagonism between cell cycle regulators and cell type-specific gene expression. *Cell Cycle* **15**, 196–212 (2016).
23. Rougny, A., Paulevé, L., Teboul, M. & Delaunay, F. A detailed map of coupled circadian clock and cell cycle with qualitative dynamics validation. *BMC Bioinformatics* **22**, 240 (2021).
24. Gupta, K. et al. Single-cell analysis reveals a hair follicle dermal niche molecular differentiation trajectory that begins prior to morphogenesis. *Dev. Cell* **48**, 17–31 (2019).
25. Sood, P. et al. Modular, cascade-like transcriptional program of regeneration in stentor. *eLife* **11**, e80778 (2022).
26. Zhu, H., Zhao, S. D., Ray, A., Zhang, Y. & Li, X. A comprehensive temporal patterning gene network in *Drosophila* medulla neuroblasts revealed by single-cell RNA sequencing. *Nat. Commun.* **13**, 1247 (2022).
27. Li, J. et al. Systematic reconstruction of molecular cascades regulating GP development using single-cell RNA-seq. *Cell Rep.* **15**, 1467–1480 (2016).
28. Huizing, G.-J., Peyré, G. & Cantini, L. Optimal transport improves cell–cell similarity inference in single-cell omics data. *Bioinformatics* **38**, 2169–2177 (2022).
29. Bellazzi, R., Codegoni, A., Gualandi, S., Nicora, G. & Vercesi, E. The gene mover's distance: single-cell similarity via optimal transport. Preprint at *arXiv* 10.48550/arXiv.2102.01218 (2021).
30. Orlova, D. Y. et al. Earth mover's distance (EMD): a true metric for comparing biomarker expression levels in cell populations. *PLoS ONE* **11**, e0151859 (2016).
31. Schiebinger, G. et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell* **176**, 928–943 (2019).
32. Zhang, S., Afanassiev, A., Greenstreet, L., Matsumoto, T. & Schiebinger, G. Optimal transport analysis reveals trajectories in steady-state systems. *PLoS Comput. Biol.* **17**, e1009466 (2021).
33. Cang, Z. & Nie, Q. Inferring spatial and signaling relationships between cells from single cell transcriptomic data. *Nat. Commun.* **11**, 2084 (2020).
34. Moriel, N. et al. NovoSpaRc: flexible spatial reconstruction of single-cell gene expression with optimal transport. *Nat. Protoc.* **16**, 4177–4200 (2021).
35. Demetci, P., Santorella, R., Sandstede, B., Noble, W. S. & Singh, R. SCOT: single-cell multi-omics alignment with optimal transport. *J. Comput. Biol.* **29**, 3–18 (2022).
36. Coifman, R. R. & Lafon, S. Diffusion maps. *Appl. Comput. Harmon. Anal.* **21**, 5–30 (2006).
37. Singer, A. From graph to manifold Laplacian: the convergence rate. *App. Comput. Harmon. Anal.* **21**, 128–134 (2006).
38. Tacke, F. & Randolph, G. J. Migratory fate and differentiation of blood monocyte subsets. *Immunobiology* **211**, 609–618 (2006).
39. Van de Veerdonk, F. L. & Netea, M. G. Diversity: a hallmark of monocyte society. *Immunity* **33**, 289–291 (2010).

40. Patel, A. A. et al. The fate and lifespan of human monocyte subsets in steady state and systemic inflammation. *J. Exp. Med.* **214**, 1913–1923 (2017).
41. Chitu, V. & Stanley, E. R. Colony-stimulating factor-1 in immunity and inflammation. *Curr. Opin. Immunol.* **18**, 39–48 (2006).
42. Imhof, B. A. & Dunon, D. Leukocyte migration and adhesion. *Adv. Immunol.* **58**, 345–416 (1995).
43. Ghebrehiwet, B., Hosszu, K. K., Valentino, A., Ji, Y. & Peerschke, E. I. Monocyte expressed macromolecular C1 and C1q receptors as molecular sensors of danger: implications in SLE. *Front. Immunol.* **5**, 278 (2014).
44. Heger, L. et al. Subsets of $CD1c^+$ DCs: dendritic cell versus monocyte lineage. *Front. Immunol.* **11**, 559166 (2020).
45. Higashi, N. et al. The macrophage C-type lectin specific for galactose/N-acetylgalactosamine is an endocytic receptor expressed on monocyte-derived immature dendritic cells. *J. Biol. Chem.* **277**, 20686–20693 (2002).
46. Myung, P., Andl, T. & Atit, R. The origins of skin diversity: lessons from dermal fibroblasts. *Development* **149**, dev200298 (2022).
47. Chen, D., Jarrell, A., Guo, C., Lang, R. & Atit, R. Dermal β-catenin activity in response to epidermal Wnt ligands is required for fibroblast proliferation and hair follicle initiation. *Development* **139**, 1522–1533 (2012).
48. Fu, J. & Hsu, W. Epidermal Wnt controls hair follicle induction by orchestrating dynamic signaling crosstalk between the epidermis and dermis. *J. Invest. Dermatol.* **133**, 890–898 (2013).
49. Hastie, T. J. *Generalized Additive Models*, pp. 249–307 (Routledge, 2017).
50. Wood, S. *mgcv: Mixed GAM Computation Vehicle with GCV/AIC/REML Smoothness Estimation* (University of Bath, 2012).
51. Pott, S. & Lieb, J. D. Single-cell ATAC–seq: strength in numbers. *Genome Biol.* **16**, 172 (2015).
52. Ståhl, P. L. et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* **353**, 78–82 (2016).
53. Macaulay, I. C., Ponting, C. P. & Voet, T. Single-cell multiomics: multiple measurements from single cells. *Trends Genet.* **33**, 155–168 (2017).

Methods

Workflow

The major workflow of GeneTrajectory comprises the following four main steps. Core notations are listed in Table 1.

- Step 1—build a cell–cell *k*NN graph in which each cell is connected to its *k*NNs. Find the shortest path connecting each pair of cells in the graph and denote its length as the graph distance between cells.
- Step 2—compute pairwise graph-based Wasserstein distance between gene distributions, which quantifies the minimum cost of transporting the distribution of a given gene into the distribution of another gene in the cell graph.
- Step 3—generate a low-dimensional representation of genes (using diffusion map by default) based on the gene–gene Wasserstein distance matrix. Identify gene trajectories in a sequential manner.
- Step 4—determine the order of genes along each gene trajectory.

Step 1. Construct a cell–cell graph and define graph distances. *Data preprocessing*. The data preprocessing contains the following steps:

(1) standard preprocessing of the count matrix (m cells and n genes).
(2) dimension reduction.

Standard preprocessing—the original count matrix (cell-by-gene) is first preprocessed by using the standard pipeline in single-cell analysis, including library normalization, top variable gene selection and scaling.

Dimension reduction—due to the low-rank nature of single-cell data, we run dimensionality reduction on the original count matrix to generate a low-dimensional representation of the cell geometry (cell embedding). Commonly used methods include PCA, *t*-SNE, UMAP and diffusion maps. By default, we apply PCA for the initial step of dimensionality reduction and retain the leading n (typically around 30–100) principal components (PCs). Then we use diffusion map to generate a manifold-preserving low-dimensional representation of cells. Specifically, for a given pair of cells u and v, we calculate the Euclidean distance $d_E(u,v)$ between their coordinates of the leading n PCs. We then convert it into an affinity measure $a(u,v)$ using the following Gaussian kernel with a local-adaptive bandwidth:

$$a(u,v)=\frac{1}{2}\left(\exp\left\{-\frac{d_E^2(u,v)}{\sigma(u)^2}\right\}+\exp\left\{-\frac{d_E^2(u,v)}{\sigma(v)^2}\right\}\right),\quad u,v=1,\cdots,m,$$

where $\sigma(u)$ represents the Euclidean distance between cell u and its *k*NNs in the PC space. Using a local-adaptive bandwidth allows us to automatically adjust the kernel size based on the local cell density in the original cell space. After we get the affinities between all pairs of cells, we apply the diffusion map algorithm and retain its leading n' eigenvectors as a low-dimensional representation of cells for the subsequent cell graph construction, which preserves the geometric information of the cell manifold.

Cell–cell graph distance computation. When cell geometry presents a low-dimensional manifold structure, the OT should be always done across the cell manifold instead of taking a shortcut through empty regions in the ambient space where there are no cells. Here we build a cell *k*NN graph in which we connect each cell to its *k*NNs in the dimensionality-reduced cell space. For a given pair of cells u and v, we search for the shortest path connecting them in the *k*NN cell graph and denote its length as the graph distance $d_G(u,v)$ between cells u and v. Theoretically, in the limit of a large number of cells, the graph distances constructed in this way reveal manifold geodesic distances, which are the intrinsic cell–cell distances[54,55].

Step 2. Compute graph-based Wasserstein distances between genes. We model the expression level of a gene as a discrete distribution on the cell graph. Specifically, let $g_i(u)$ represents the expression level of gene i in cell u, we then define the distribution of gene i by:

$$\rho_i(u)=g_i(u)\Big/\sum_{v=1}^{m}g_i(v). \tag{1}$$

It has the following properties: (1) $\rho_i\in\mathbb{R}_+^m$; (2) $\sum_v\rho_i(v)=1$. We then define the distance between two genes by the W_p distance ($1\le p<\infty$) between their distributions on the cell graph. Namely, the W_p distance $\delta^{(p)}(\rho_i,\rho_j)$ between gene i and gene j quantifies their dissimilarity. Technically, the W_p distance can be computed by solving a discrete OT mapping over the cell graph. Details are described below.

W_p distance formulation and computation. Here we set up some mathematical notations as follows: for a graph consisting of m nodes $V=\{1,\ldots,m\}$, a graph distribution is a non-negative vector $\rho\in\mathbb{R}_+^m$ such that the sum of its elements is equal to one and the distribution assigns measure $\rho(u)$ to node u. We assume the graph is equipped with a graph ground distance $d_G(u,v)$ for $u,v\in V$. Specifically, the graph distance d_G is used to specify the cost of the OT, that is, the cost matrix C is defined as $C_{u,v}=d_G(u,v)^p$. As mentioned in Step 1. Construct a cell–cell graph and define graph distances, we denote the shortest path distance on a *k*NN graph as d_G, while the computational method also allows other options of d_G or even letting the cost matrix take a more general form. For two graph distributions ρ and ρ', the W_p distance is defined as:

$$\delta^{(p)}(\rho,\rho')=\min_{F\in\Pi_{\rho,\rho'}}\langle F,C\rangle^{1/p}, \tag{2}$$

where $\Pi_{\rho,\rho'}=\{F,F_{u,v}\ge 0,\sum_v F_{u,v}=\rho(u)\text{ for all }u,\sum_u F_{u,v}=\rho'(v)\text{ for all }v\}$ denotes the set of transport plan F that pushes from the source distribution ρ to the target distribution ρ'.

Improve computational efficiency. In practice, the minimization in equation (2) can be solved by linear programming, which is computationally prohibitive on large cell graph and between all the pairs of genes. To reduce the cost of computing gene–gene W_p distances, we have designed two strategies to accelerate the computation based on (1) cell graph coarse-graining and (2) gene graph sparsification. Briefly, cell graph coarse-graining aims to reduce the cell number by aggregating the nearest cells into 'meta-cells'. Gene graph sparsification aims to skip the computation for two gene distributions if they are very far away from each other at a coarse-grained level, as they are unlikely to participate in the same biological process. We note that while coarse-graining the cell graph to a crude scale can make it fast for computation, it may lose accuracy and compromise the resolution. Hence, users should judiciously choose the level of coarse-graining based on the capacity of their computing resources.

(1) Cell graph coarse-graining. We coarse-grain the cell graph by aggregating m cells into m' 'meta-cells' using *the k*-means clustering algorithm. Specifically, let M be the m-by-m' membership matrix where $M(u,a)=1/|a|$ if and only if the cell u belongs to the ath subset where $|a|$ represents the number of cells in that subset, otherwise $M(u,a)=0$, then we define an updated transport cost matrix C' on the coarse-grained cell graph by M^TCM. Accordingly, the expression level of a given gene in each 'meta-cell' is defined by the sum of its expression level in all the cells in that subset. Intuitively, this procedure can be viewed as providing an approximation of a cell graph with fewer cell nodes.
(2) Gene affinity graph sparsification. We sparsify the gene affinity graph by zeroing out the entries where their pairwise

Wasserstein distances are greater than a threshold. The threshold is selected such that affinities associated with distances greater than it will be exponentially small and thus contribute negligibly to the gene affinity graph. The threshold is adaptively estimated for each cell using the approximate Wasserstein distance on a coarse-grained cell graph (strategy 1) which allows fast computation.

Specifically, this is formulated in the following way: if we want to construct the gene–gene Wasserstein distance matrix on a cell graph of an original size m, we first coarse-grain m cells into m' 'meta-cells' using the procedure in strategy 1, where m' is a size that can be quickly handled. Based on the gene-by-gene Wasserstein distance matrix constructed on m' 'meta-cells', we identify the αk nearest neighbors for each gene (where α is the predefined parameter and k is the neighborhood size to construct the local-adaptive kernel for computing the diffusion map). Going back to the computation on the original cell graph, we then only compute the Wasserstein distance between a pair of genes if one of them is included in the other's αk nearest neighbors. Practically, this can reduce the running time to $2\alpha k/m$ of the original, which computes Wasserstein distances for all pairs of genes.

Step 3. Construct gene trajectories. After we obtain the gene–gene Wasserstein distance matrix, we convert it into an affinity matrix A using a local-adaptive Gaussian kernel. Specifically, the kernel bandwidth for each gene is defined by the distance to its kNN (similar to 'Step 1. Construct a cell–cell graph and define graph distances'). The affinity between gene i and gene j is defined by:

$$A_{i,j} = \frac{1}{2}\left(\exp\left\{-\frac{\delta^{(p)}(\rho_i,\rho_j)^2}{(\sigma_i)^2}\right\} + \exp\left\{-\frac{\delta^{(p)}(\rho_i,\rho_j)^2}{(\sigma_j)^2}\right\}\right). \quad (3)$$

Here ρ_i represents the distribution of gene i and σ_i represents the kth smallest Wasserstein distance between gene i and other genes. K is an integer parameter to be specified by the user, which controls the size of the local neighborhood on the graph (in the sense that A_{ij} is only large on a subject of genes j that are sufficiently close to i). The affinity matrix A in equation (3) is used to construct a random walk on the gene–gene graph (see below in the bullet point—diffusion of probability mass on the gene graph). The random walk constructed from affinity A allows us to apply Diffusion Map to obtain a low-dimensional embedding of the genes.

Next, extracting gene trajectories is processed in a sequential manner when the gene graph exhibits a tree structure. Briefly, we first identify an 'extremum' gene as the terminus for the first gene trajectory and then use a diffusion strategy to retrieve genes belonging to that trajectory where the terminus gene serves as the initial node of the diffusion process.

The details are summarized below:

- Selection of the initial node. We retain the top d nontrivial diffusion map eigenvectors as the low-dimensional spectral embedding of genes, denoted by S. Let S_i represents the spectral coordinates of gene i, we choose the gene with the largest L_2 embedding norm $\max_i ||S_i||_2$ as the starting point of diffusion on the gene graph. The assumption here is that the gene with the largest distance from the origin of spectral embedding corresponds to the terminus of a specific gene trajectory.
- Diffusion of probability mass on the gene graph. The diffusion is performed by propagating a point mass from the initial node in the gene graph. Here the initial probability mass $\mathbf{p}_0$ can be formulated as the following unit vector:

$$\mathbf{p}_0 = (0, \ldots, 0, 1, 0, \ldots, 0).$$

Suppose gene j is selected as the initial node; then only the jth entry of p_0 is equal to 1, while all other entries are zeros. We then construct a random-walk matrix P by row-wise normalizing the gene–gene affinity matrix A. Specifically, P is defined by:

$$P = D^{-1}A,$$

where D is the degree matrix of A (that is, D is a diagonal matrix where $D_{ii} = \sum_j A_{ij}$). Calculating $\mathbf{p}_1 = P\mathbf{p}_0$ gives the updated probability mass (over genes) after the first time of diffusion. We run the diffusion up to t times (the integer t is a tunable parameter) on the gene graph to get the t-step probability mass $\mathbf{p}_t = P^t\mathbf{p}_0$. We then select the genes $\{j, s.t., \mathbf{p}_t(j) > \tau_0 \max_{j'} \mathbf{p}_t(j')\}$ as members of the first gene trajectory. Here τ_0 is a thresholding parameter, which in practice can be set to be in the range of 0.02–0.05. Throughout the experiments in this paper, we choose $\tau_0 = 0.02$.

After the genes that belong to the first gene trajectory are extracted, we repeat the abovementioned procedure on the remaining genes to get the second gene trajectory, and then the third, etc. This algorithm allows retrieving a series of gene trajectories successively until all detectable trajectories are identified.

Step 4. Order genes along each trajectory. To determine the gene ordering along a given gene trajectory, we first extract the corresponding submatrix of gene-by-gene Wasserstein distances as computed in 'Step 2. Compute graph-based Wasserstein distances between genes'. That is, we only focus on the genes that are the members of that trajectory. We then recompute the diffusion map on the Wasserstein distance submatrix to obtain a new spectral embedding of genes in that trajectory. The first nontrivial eigenvector (EV2) of the new diffusion map embedding provides an ordering of the genes along that trajectory, according to the spectral convergence theory of diffusion map[36,37]. Specifically, genes are ordered based on ranking their coordinates along EV2.

Experiments and analyses

Here we present the details for the following: (i) simulation experiments ('Workflow of gene dynamics simulation' and 'Generalizing count model using negative binomial distribution to account for overdispersion'), (ii) the biological experiments of mouse embryo skin sample preparation ('Experimental details of mouse embryo skin sample preparation'), (iii) the analyses on real-world biological datasets ('Analytical details of real-world examples'), (iv) comparing Wasserstein distance with other canonical metrics for learning gene geometry ('Comparing the Wasserstein metric to other canonical metrics for learning gene geometry'), (v) comparing GeneTrajectory with cell trajectory methods in terms of gene ordering inference ('Comparing GeneTrajectory with cell trajectory methods in terms of gene ordering inference') and (vi) the robustness evaluation experiments and guidelines on parameter selection ('Hyperparameter selection guidelines and robustness evaluation').

Workflow of gene dynamics simulation. We present the details of our simulation framework for the four examples in Fig. 2, including

(1) a cyclic process,
(2) a differential process with two lineages,
(3) a linear differentiation process coupled with CC,
(4) a multilevel lineage differentiation process coupled with CC.

For illustrative purposes, we first introduce the simulation procedure on a simple linear process. The corresponding plots are shown in Extended Data Fig. 1a,b.

Illustrative example: a linear process. To demonstrate the simplest scenario (Extended Data Fig. 1a,b), we simulate a linearly progressive biological process in $[0, T]$, where $t = 0$ corresponds to the initial cell

state and $t = T$ corresponds to the terminal cell state. We simulate a set of genes $\{g_i, i = 1, \cdots, n\}$, where each g_i is a non-negative vector in $\mathbb{R}^m$ and $g_i(u)$ represents the gene expression at cell $u, u = 1, \cdots, m$.

In this example, we let each cell u be uniquely associated with a pseudotime t_u, which is i.i.d. uniformly distributed on $[0, T]$. Our procedure is to first construct for each gene i a continuous function $\lambda_i(t)$ on $t \in [0, T]$ and then obtain the gene expression vectors g_i from $\lambda_i(t)$ based on Poisson sampling. Specifically, the simulation procedure consists of the following two steps:

- Simulate expected gene expression levels along the process.
  For each i, we define a function $\lambda_i(t)$, where $\lambda_i(t_u)$ represents the expected gene expression level of gene i at cell u. The function $\lambda_i(t)$ is associated with a 'peak time' t_i^*, which represents the time point when gene i reaches the peak of its expected expression level. The time t_i^* is uniformly sampled from $[0, T]$. The function $\lambda_i(t)$ then takes a parametric expression as

$$\lambda_i(t) = \gamma_1 \alpha_i e^{-\frac{|t - t_i^*|^2}{\gamma_2 d_i^2}}, \tag{4}$$

  where parameters γ_1 and γ_2 are predefined positive scalars, and α_i and d_i are positive random variables to account for the variation in duration length and expression intensity of different genes. Specifically, we draw d_i and α_i from log-normal prior distributions as below:

$$d_i \sim \mathrm{LN}\left(\mu_d, \sigma_d^2\right); \qquad \alpha_i \sim \mathrm{LN}\left(\mu_\alpha, \sigma_\alpha^2\right).$$

- Sample gene reads from a Poisson distribution.
  In reality, the sequencing process is based on capturing molecules (for example, DNA or RNA fragments) in a random manner. To mimic the randomness in the sequencing process, we simulate $g_i(u)$ as from a Poisson distribution with a rate $\lambda_i(t_u)$, namely,

$$g_i(u) \sim \mathrm{Poi}(\lambda_i(t_u)), \quad u = 1, \cdots, m, \tag{5}$$

  independently across all u and i. This gives $g_i \in \mathbb{R}_+^m$ as desired.
- (Optional) sparsify the count matrix by sampling nonzero entries.
  Finally, we incorporate an optional step to account for sequencing depth. This is achieved by randomly selecting a specified number of entries from the original count matrix without replacement and subsequently zeroing out the remaining entries. The probability that an entry is selected is proportional to its original expression value. This procedure enables us to generate an artificial dataset with varying levels of missing data.

Example A: a cyclic process. To simulate a biological process with cyclic dynamics (for example, CC) (Fig. 2a), based on the former setup of the linear process simulation, we only need to modify equation (4) as

$$\lambda_i(t) = \gamma_1 \alpha_i e^{-\frac{\min\left(|t - t_i^*|, |t + T - t_i^*|, |t - T - t_i^*|\right)^2}{\gamma_2 d_i^2}}. \tag{6}$$

All other details are the same as in 'Workflow of gene dynamics simulation'.

Example B: a differentiation process with two lineages. To simulate a biological process with a bifurcating structure (for example, myeloid lineage differentiation) (Fig. 2b), we represent the underlying cell state by a generalized pseudotime vector $\boldsymbol{\tau}$ comprising three pseudotime variables

$$\boldsymbol{\tau}_u = \left(t_u^{(0)}, t_u^{(1)}, t_u^{(2)}\right). \tag{7}$$

Here $t_u^{(0)} \in [0, T^{(0)}]$ represents the pseudotime along the initial process before lineage differentiation, $t_u^{(1)} \in [0, T^{(1)}], t_u^{(2)} \in [0, T^{(2)}]$ each represents the pseudotime along the process of lineage 1 and lineage 2 differentiation.

Specifically, if a cell u is along the initial process, then $t_u^{(0)} \geq 0, t_u^{(1)} = t_u^{(2)} = 0$. If a cell u is along lineage 1, then $t_u^{(0)} = T^{(0)}, t_u^{(1)} \geq 0, t_u^{(2)} = 0$. If a cell u is along lineage 2, then $t_u^{(0)} = T^{(0)}, t_u^{(1)} = 0, t_u^{(2)} \geq 0$.

Similarly, we generate $\tau_i^* = (t_i^{(0)*}, t_i^{(1)*}, t_i^{(2)*})$ based on the same procedure as described above to represent the 'time point' that gene i reaches the peak of its expected expression level. Here the expectation of the expression level of gene i at the time point τ is given by:

$$\lambda_i(\tau) = \gamma_1 \alpha_i e^{-\frac{||\tau - \tau_i^*||_1^2}{\gamma_2 d_i^2}}. \tag{8}$$

Parameters $\gamma_1, \gamma_2, \alpha_i$ and d_i are defined based on the same procedure in 'Workflow of gene dynamics simulation'. We then simulate

$$g_i(u) \sim \mathrm{Poi}(\lambda_i(\tau_u)), \quad u = 1, \cdots, m, \tag{9}$$

independently across u and i, similarly as in equation (5).

Example C: a linear differentiation process coupled with CC. In this example (Fig. 2c), we simulate genes for a linear process and a cyclic process independently and then put them together. Specifically, we associate each cell u with a generalized pseudotime vector $\boldsymbol{\tau}$ comprising two pseudotime variables $\boldsymbol{\tau}_u = (t_u^{(1)}, t_u^{(2)})$. Here $t_u^{(1)} \in [0, T^{(1)}]$ represents the pseudotime along the linear process, while $t_u^{(2)} \in [0, T^{(2)}]$represents the pseudotime along the cyclic process. The sampling processes to generate $\{t_u^{(1)}\}$ and $\{t_u^{(2)}\}$ are independent.

Next, we simulate two sets of genes using the procedure same as in 'Workflow of gene dynamics simulation' but with a different definition of the Poisson rate function $\lambda_i(\boldsymbol{\tau})$. Specifically, the first list of genes $\{g_i\}$, $1 \leq i \leq n_1$, is simulated with

$$\lambda_i(\boldsymbol{\tau}) = \gamma_1 \alpha_i e^{-\frac{|t^{(1)} - t_i^{(1)*}|^2}{\gamma_2 d_i^2}}. \tag{10}$$

The second list of genes $\{g_j\}, n_1 < j \leq n$, is defined with

$$\lambda_j(\boldsymbol{\tau}) = \gamma_1 \alpha_j e^{-\frac{\min\left(|t^{(2)} - t_j^{(2)*}|, |t^{(2)} + T^{(2)} - t_j^{(2)*}|, |t^{(2)} - T^{(2)} - t_j^{(2)*}|\right)^2}{\gamma_2 d_j^2}}. \tag{11}$$

Notably, the first list of genes contributes to the linear process, while the second list of genes contributes to the cyclic process. This simulation results in a cylinder-like cell manifold in the high-dimensional space.

Example D: a multilevel lineage differentiation process coupled with CC. In this example (Fig. 2d), we simulate genes for a two-level tree-structured process and a cyclic process independently and then put them together. In the general case, let us consider simulating a n-level bifurcating process, in which the initial process P_0 first splits into two lineages (P_1 and P_2), then each lineage proceeds independently and further splits into another two sublineages ($P_{1.1}$ and $P_{1.2}$, $P_{2.1}$ and $P_{2.2}$), and each sublineage divides again in an iterative manner until a n-level tree structure is generated. At the same time, all the cells are involved in a cyclic process. Here each cell u can be associated with a generalized pseudotime vector $\boldsymbol{\tau}$ comprising 2^n pseudotime variables $\boldsymbol{\tau}_u = (t_u^{(0)}, t_u^{(1)}, \ldots, t_u^{(2^n)})$, each of the first $2^n - 1$ pseudotime variables corresponds to a pseudotime location along $P_0, P_1, P_2, P_{1.1}, P_{1.2}, P_{2.1}, P_{2.2}, \ldots, P_{\underbrace{1.1\ldots1}_{n}}, \ldots, P_{\underbrace{2.2\ldots2}_{n}}$, respectively. For generating the instances of these $2^n - 1$ pseudotime variables, we adopt the similar framework as applied in 'Workflow of gene

dynamics simulation' by requiring that when a cell u is along a daughter lineage, its pseudotime variables corresponding to all the parent processes are set to the largest possible values, and its pseudotime variables corresponding to other processes (excluding the daughter lineage itself) are all set to 0. Besides, $t_u^{(2^n)}$ represents the pseudotime of cell u in the cyclic process, which is independent from the other $2^n - 1$ pseudotime variables.

Next, we simulate two sets of genes using the procedure same as in 'Workflow of gene dynamics simulation' but with a different definition of the Poisson rate function $\lambda_i(\boldsymbol{\tau})$. Specifically, the first list of genes $\{g_i\}$, $1 \le i \le n_1$, is simulated with

$$\lambda_i(\boldsymbol{\tau}) = \gamma_1 \alpha_i e^{-\frac{\left(||\boldsymbol{\tau} - \boldsymbol{\tau}_i^*||_1 - |t^{(2^n)} - t_i^{(2^n)*}|\right)^2}{\gamma_2 d_i^2}}. \quad (12)$$

The second list of genes $\{g_j\}$, $n_1 < j \le n$, is defined with

$$\lambda_j(\boldsymbol{\tau}) = \gamma_1 \alpha_j e^{-\frac{\min\left(|t^{(2^n)} - t_j^{(2^n)*}|, |t^{(2^n)} + T^{(2^n)} - t_j^{(2^n)*}|, |t^{(2^n)} - T^{(2^n)} - t_j^{(2^n)*}|\right)^2}{\gamma_2 d_j^2}}. \quad (13)$$

Notably, the first list of genes contributes to the tree-structured differentiation process, while the second list of genes contributes to the cyclic process. This simulation results in a coral-like cell manifold in the high-dimensional space.

Details of the simulation examples. For the examples shown in Fig. 2, the evaluation outputs can be found in Supplementary Table 1. Each example was tested through ten replicates. Specifically, in the first example, we simulated 1,000 cells, 500 genes underlying the cyclic process. In the second example, we simulated 1,000 cells, 500 genes for the initial process and 250 genes for each daughter lineage process. In the third example, we simulated 5,000 cells, 500 genes for the linear process and 500 genes for the cyclic process. In the fourth example, we simulated 10,500 cells, 400 genes for the cyclic process and 200 genes for each sublineage process. For all these samples, we adopted the following model parameters: $\gamma_1 = 25, \mu_d = 0, \mu_\alpha = 0, \sigma_d = 0.25, \sigma_\alpha = 0.25$. We chose $T = 10$ in the fourth example, while $T = 15$ in the other examples. We chose $\gamma_2 = 2$ for simulating the cyclic process, while $\gamma_2 = 8$ for simulating the other processes. In simulation experiments, genes along a circular trajectory are ordered by their angular coordinates of the first two nontrivial diffusion map eigenvectors.

Generalizing count model using negative binomial distribution to account for overdispersion. To investigate the impact of dispersion on the performance of GeneTrajectory, specifically in terms of gene ordering, we performed a negative binomial variant of our second and third simulation experiments in Fig. 2. For each dataset, we simulated three distinct sparsity levels (5%, 10% and 20%). For each sparsity level, we tested four different dispersion levels (parameterized by θ), each comprising ten technical replicates. A lower θ value indicates higher dispersion. We evaluated the consistency between the inferred gene ordering and the ground truth by calculating their Spearman correlation (Supplementary Fig. 1). It shows that GeneTrajectory exhibits remarkable stability across all sparsity and dispersion levels.

Experimental details of mouse embryo skin sample preparation. *Mice*. K14Cre (ref. 56) mice were bred to Wntless$^{fl/fl}$ (ref. 57) mice. A random population of both male and female embryos was used for all experiments. All procedures involving animal subjects were performed under the approval of the Institutional Animal Care and Use Committee of the Yale School of Medicine.

EdU administration. To assess proliferation, EdU was administered to pregnant mice intraperitoneally (25 μg gm^{-1}) and embryos were collected after 1.5 h.

In situ hybridization. In total, 10% of formalin-fixed paraffin-embedded (FFPE) whole embryos were used for histological analysis. FFPE specimens were subsectioned at 10 μm thickness. The RNAscope Multiplex Fluorescent Detection Kit v2 (ACDBio, 323110) was used for single-molecule fluorescence in situ hybridization (FISH) according to the manufacturer's protocol. Briefly, subsections were deparaffinized and permeabilized with hydrogen peroxide followed by antigen retrieval and protease treatment before probe hybridization. After hybridization, amplification and probe detection were done using the Amp 1–3 reagents. Probe channels were targeted using the provided HRP-C1-3 reagents and TSA (tyramide signal amplification) fluorophores—Cy3 (Akoya Biosciences, NEL744001KT), Cy5 (Akoya Biosciences, NEL745001KT) and fluorescein (Akoya Biosciences, NEL741001KT). EdU staining was done using the Click-it EdU Imaging Kit Alexa 488 (Life Technologies, c10338) according to the manufacturer's instructions. Nuclear counter-stain was done using Hoechst 33342 (Invitrogen, H3570) before mounting with SlowFade Mountant. RNA scope probes used (ACDBio)—Mm-Lef1 (441861) and Mm-Sox2 (401041).

Microscopy. FISH paraffin-embedded images were acquired using the Leica TCS SP8 Gated STED 3X super-resolution confocal microscope with a ×40 oil immersion (Numerical Aperture 1.3) objective lens, scanned at 5 μm thickness, 1,024 × 1,024 pixel width, 400 Hz.

Single-cell dissociation. Embryonic dorsolateral/flank skin was microdissected from E14.5 littermate control and mutant embryos and dissociated into a single-cell suspension using 0.25% trypsin (Gibco, Life Technologies) for 20 min at 37 °C. After genotyping, two to three embryos were pooled by condition. Single-cell suspensions were then stained with DAPI (Thermo Fisher Scientific, NBP2-31156) just before fluorescence-activated cell sorting.

Fluorescence-activated cell sorting. DAPI-excluded live skin cells were sorted on a BD FACS Aria II (BD Biosciences) sorter with a 100 μm nozzle. Cells were sorted in bulk and submitted for 10X Genomics library preparation at 0.75–1.0 × 10^6 ml^{-1} concentration in 4% fetal calf serum/phosphate buffered saline (FCS/PBS) solution.

H-score quantification. For quantification based on FISH, cells with 4–5 dots were considered positive (according to the RNAScope manufacturer's instructions) and subsections from a total of $n = 4$ different embryos were examined. To measure RNA expression levels, *H* scores were calculated according to ACDBio manufacturer's instructions—a cell with 0 dot is scored 0, 1–3 dots is scored 1, 4–9 dots is scored 2, 10–15 dots and/or less than 10% clustered dots is scored 3 and more than 15 dots and/or more than 10% clustered dots is scored 4; then the final *H* score of a given cell type A is calculated by summing the (% cells scored *B* within all cells in *A*) × *B* for score *B* in 0–4.

scRNA-seq and library preparation. Chromium Single Cell 3′ GEM Library and Gel Bead Kit v3.1 (PN-1000121) were used according to the manufacturer's instructions in the Chromium Single Cell 3′ Reagents Kits V3.1 User Guide. After cDNA libraries were created, they were subjected to Novaseq 6000 (Illumina) sequencing. For each scRNA-seq experiment, control and littermate mutant samples were prepared in parallel at the same time, pooled and sequenced on the same lane.

Analytical details of real-world examples. *Human myeloid dataset analysis*. Myeloid cells were extracted from a publicly available 10× scRNA-seq dataset (https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_v3). QC (quality

control) was performed using the same workflow in https://github.com/satijalab/Integration2019/blob/master/preprocessing_scripts/pbmc_10k_v3.R. After standard normalization, highly variable gene selection and scaling using the Seurat R package[58], we applied PCA and retained the top 30 PCs. Four subclusters of myeloid cells were identified based on Louvain clustering with a resolution of 0.3. Wilcoxon rank-sum test was used to find cluster-specific gene markers for cell type annotation.

For gene trajectory inference, we first applied diffusion map on the cell PC embedding (using a local-adaptive kernel, each bandwidth is determined by the distance to its kNN, $k = 10$) to generate a spectral embedding of cells. We constructed a cell–cell kNN ($k = 10$) graph based on the coordinates of the top five nontrivial diffusion map eigenvectors. Among the top 2,000 variable genes, genes expressed by 0.5–75% of cells were retained for pairwise gene–gene Wasserstein distance computation. The original cell graph was coarse-grained into a graph of size 1,000. We then built a gene–gene graph where the affinity between genes is transformed from the Wasserstein distance using a Gaussian kernel (local-adaptive, $k = 5$). Diffusion map was used to visualize the embedding of the gene graph. For trajectory identification, we used a series of time steps (11, 21 and 8) to extract three gene trajectories. Gene ordering was done based on the algorithm described in 'Step 4. Order genes along each trajectory'.

Mouse embryo skin data analysis. We separated dermal cell populations from the newly collected mouse embryo skin samples ('Experimental details of mouse embryo skin sample preparation'; aligned to the mouse genome mm10 by CellRanger v6.1.2). Cells from the WT and the Wls mutant were pooled for analyses. After standard normalization, highly variable gene selection and scaling using Seurat, we applied PCA and retained the top 30 PCs. Three dermal cell types were stratified based on the expression of canonical dermal markers, including *Sox2*, *Dkk1* and *Dkk2*. For gene trajectory inference, we first applied diffusion map on the cell PC embedding (using a local-adaptive kernel bandwidth, $k = 10$) to generate a spectral embedding of cells. We constructed a cell–cell kNN ($k = 10$) graph based on the coordinates of the top ten nontrivial diffusion map eigenvectors. Among the top 2,000 variable genes, genes expressed by 1–50% of cells were retained for pairwise gene–gene Wasserstein distance computation. The original cell graph was coarse-grained into a graph of size 1,000. We then built a gene–gene graph where the affinity between genes is transformed from the Wasserstein distance using a Gaussian kernel (local-adaptive, $k = 5$). Diffusion map was used to visualize the embedding of the gene graph. For trajectory identification, we used a series of time steps (9, 16 and 5) to sequentially extract three gene trajectories. To compare the differences between the WT and the Wls mutant, we stratified Wnt-active upper dermal cells into seven stages according to their expression profiles of the genes binned along the DC gene trajectory.

CC gene trajectory validation. We extracted the Cyclebase[59] gene list from Supplementary Table 5 in ref. 60, in which genes are categorized into groups of G1/S, S, G2, G2/M and M phase markers. We also incorporated histone genes into the S phase gene list as they are upregulated during the S phase for the active synthesis of histone proteins[61]. We plotted the distribution of genes from different phases along the gene trajectory associated with the CC process in the dermal example (Extended Data Fig. 3b). We observed that genes corresponding to the G1/S phase were located around the start of the gene trajectory, followed by a group of genes highly expressed during the S phase. G2M-related genes were located along the second half of the gene trajectory. Specifically, G2 genes appeared in the middle of the trajectory, followed by a group of genes regulating the switch from G2 to M. Genes associated with the M phase were found around the end of the trajectory. This indicates that GeneTrajectory can effectively capture gene dynamics associated with different phases of the CC.

Different visualizations of gene embedding. Gene embedding visualization is agnostic to gene–gene distance computation and trajectory identification. Different ways of gene embedding visualization for the two real-world examples included in the manuscript are shown and compared in Supplementary Fig. 2. We would advise users to apply diffusion-based visualization techniques, for example, diffusion map or PHATE[62], to display the trajectories, as they were designed to capture and reveal the connectivity of graphs.

Assessing the stability of capturing gene processes in the dermal example. After identifying three prominent gene trajectories by running GeneTrajectory on the original cell graph (with the maximum of iteration = 5,000 when calculating gene–gene distances), we constructed a new cell graph using only the genes extracted from each gene trajectory. We then reran the gene trajectory inference on each new cell graph for (1) all the genes and (2) the same set of genes that were used to construct the new cell graph (Supplementary Fig. 3). We found that the ordering of the genes used to define the new cell graph stayed in a high degree of consistency with their original ordering inferred by our method (when we constructed the cell graph using all genes). This consistency highlights the stability of GeneTrajectory in inferring gene dynamics underlying each process, unaffected by the presence of coexisting gene programs and biological effects.

Meanwhile, we observed potential caveats of iteratively running GeneTrajectory on the cell graphs constructed using the genes along a previously identified gene trajectory. This is because, in each iteration, the cell graph is only determined by the subset of genes corresponding to a specific process. There is no theoretical guarantee that the cell graph still encodes the geometric information necessary for identifying a gene trajectory associated with a different process. In other words, the new cell graph may distort the cell geometry for the other processes.

Comparing the Wasserstein metric to other canonical metrics for learning gene geometry. We conducted an extensive benchmark on using different metrics (including the Earth Mover's distance, Euclidean distance, Pearson correlation distance, Spearman correlation distance, Cosine similarity, total variation distance (equivalent to L1 distance or Manhattan distance in its discrete form), Jensen–Shannon distance and Hellinger distance) to learn gene geometry in simulation datasets (Supplementary Fig. 4). Datasets for evaluation were generated based on simulations (corresponding to the second and third simulation examples in Fig. 2). Specifically, we simulated datasets with three different sequencing depths (that is, the percentage of nonzero entries in the gene-by-cell count matrix = 2.5%, 5% and 10%), each having ten replicates. To evaluate the performance, we calculated the Spearman correlation between each inferred gene ordering and the ground truth. The Wasserstein distance recovers gene ordering more accurately and robustly than other metrics.

Comparing GeneTrajectory with cell trajectory methods in terms of gene ordering inference. We performed a benchmark to compare GeneTrajectory with five representative cell trajectory inference methods, Monocle 2 (ref. 16), Monocle 3 (ref. 10), Slingshot[9], PAGA[11] and CellRank[15]. We assessed their performances on the following two types of datasets:

- simulation datasets (corresponding to the third simulation example in Fig. 2) with varying sparsity levels of the count matrix (that is, the percentage of nonzero entries in the gene-by-cell count matrix = 2.5%, 5%, 10% and 20%) and different numbers of cells (500, 1,000 and 2,500).
- the real-world dermal dataset depicted in Figs. 4 and 5 with or without cell cycle effects regression.

For these cell trajectory inference approaches, after cell pseudotime inference, we leveraged GAM using the mgcv[63] (Mixed GAM

Computation Vehicle with Automatic Smoothness Estimation) R package to smooth the gene expression along the cell pseudotime, followed by ordering the genes based on their peak locations. For the simulation datasets, we calculated the Spearman correlation between the true gene order and the inferred gene order by each method. For the dermal dataset, the assessment is done by examining the ordering of experimentally verified markers during DC differentiation.

In summary, Monocle 2 and Monocle 3 require the specification of a starting (root) cell state to generate cell pseudo-order. In simulation experiments, we chose the cell with the ground truth pseudotime $t = 0$ as the starting cell state. In the dermal dataset, we first looked at the diffusion map embedding of cells to define the tip cell (expressing *Sox2*) as the terminal cell state of DC differentiation process. We then chose the upper dermal cell that has the largest distance (in the transcriptome space) to the terminal cell state as the starting cell state. PAGA and SlingShot require the specification of a starting cell cluster to create cell pseudotime. Based on the same strategy as described above, we chose the cluster containing the starting cell state as the starting cell cluster. The core steps in each analysis workflow for cell trajectory inference methods are summarized below.

- SlingShot—we used the getLineages function to construct the minimum spanning tree(s) on cell clusters. We then fitted principal curves using the getCurves function, which served as the basis for cell pseudotime construction.
- PAGA—cells were reclustered using the Leiden method implemented in the Scanpy toolkit. We constructed the PAGA graph of these cell clusters and inferred the progression of cells through geodesic distance along the graph using scanpy.tl.dpt.
- Monocle 2—we used the built-in DDRTree method for cell dimension reduction. We used the orderCells function to generate the cell ordering while the root state was defined by the starting cell state as noted above.
- Monocle 3—cells were partitioned using the built-in Louvain method. We learned the principal graph across all partitions and then ordered the cells using the order_cells function.
- CellRank—for the simulation experiments, because we don't have the information about the spliced and unspliced read counts, we used CellRank's CytoTRACEKernel to infer the transition dynamics and cell pseudotime. For the dermal example, we applied CellRank based on RNA velocity inference. Specifically, the spliced/unspliced counts were quantified by the velocyto toolkit. We used scVelo's dynamical model[64] to infer RNA velocities. CellRank was then applied to infer the initial states and terminal states of transition and construct cell lineages. We selected the cell lineage that terminates its transition at the DC cell population and fitted GAM models (built-in CellRank) to order the genes along the cell pseudotime of the selected lineage.

Hyperparameter selection guidelines and robustness evaluation. We would advise users to choose and determine the parameters according to the following standards:

- If users choose to use diffusion maps (or PCA) to generate a cell embedding. The number of eigenvectors (or PCs) for cell graph dimensionality reduction can be ascertained by examining the eigenvalues in descending order to identify an eigengap or the point where the spectrum starts to flatten out.
- The k in cell kNN graph construction is a user-defined hyperparameter. The chosen value for k should ensure the cell graph is fully connected.
- The number of gene programs is determined by the number of branches (gene trajectories) identifiable from the gene graph. This determination is made interactively during the process of branch identification. Specifically, when a new branch is being extracted, we exclude the genes that have already been assigned to existing branches. Subsequently, we identify one of the remaining genes that is most distant from the origin of diffusion embedding as the tip of the next branch. If the remaining genes visually form an indistinct cloud that does not exhibit a trajectory structure, we cease the process of branch identification.
- The time step t for random walks in each iteration of branch identification is interactively determined by inspecting the gene embedding. Specifically, when t increases, a greater number of genes are incorporated as the members of the branch to be extracted. The optimal t for extracting each branch should yield the longest trajectory without incorporating the genes in the indistinct cloud.
- The number of gene bins for visualization is determined by the resolution users wish to inspect for shifting patterns in gene distributions over cell embedding. An ideal number would be between 5 and 10. The choice of bin number does not affect gene trajectory inference.

We conducted an extensive evaluation to assess the robustness of GeneTrajectory with varying combinations of parameters. These parameters included k for constructing cell–cell kNN graphs, n_{dim} for dimensionality reduction and k_a for determining local-adaptive kernel bandwidths in diffusion map construction. To assess GeneTrajectory's performance on simulated datasets, we computed the Spearman correlation between the inferred gene ordering and the ground truth ordering. For the real-world examples, we performed a cross-validation by examining the Spearman correlation between all pairs of inferred gene orderings. The results of this evaluation are depicted in Supplementary Figs. 5–7. Specifically, these experiments include:

- We simulated bifurcation datasets and cylindrical datasets (corresponding to the second and third simulation examples in Fig. 2) with varying sparsity levels (that is, the percentage of nonzero entries in the gene-by-cell count matrix = 2.5%, 5%, 10% and 20%, each has ten replicates; each replicate includes 1,000 cells). We tested GeneTrajectory using a combination of $k = 5, 10, 15, 20, 25$ and $n_{\text{dim}} = 5, 10, 15, 20, 25$. The evaluation outputs are shown in Supplementary Fig. 5.
- Using the same simulation datasets mentioned above, we tested GeneTrajectory using $k_a = 5, 10, 15, 20, 25$ for constructing the diffusion embedding of cells. The evaluation outputs are shown in Supplementary Fig. 6.
- In two real-world examples included in this manuscript, we tested GeneTrajectory on cell graphs constructed using different numbers of eigenvectors ($n_{\text{dim}} = 5, 10, 15, 20, 25$). The evaluation outputs are shown in Supplementary Fig. 7.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The human PBMC scRNA-seq dataset is available at https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_v3. The mouse embryonic skin dataset generated and analyzed in this study is available from the Gene Expression Omnibus with the accession GSE255534. The processed Seurat data objects for these two datasets are available at Figshare (https://doi.org/10.6084/m9.figshare.25243225). The Cyclebase gene list was extracted from Supplementary Table 5 in ref. 60.

Code availability

The R package of GeneTrajectory and the code used for data analysis are available on GitHub (https://github.com/KlugerLab/GeneTrajectory).

References

54. Balasubramanian, M. & Schwartz, E. L. The isomap algorithm and topological stability. *Science* **295**, 7 (2002).
55. Bernstein, M., De Silva, V., Langford, J. C. & Tenenbaum, J. B. *Graph Approximations to Geodesics on Embedded Manifolds* Technical Report (Department of Psychology, Stanford University, 2000).
56. Dassule, H. R., Lewis, P., Bei, M., Maas, R. & McMahon, A. P. Sonic hedgehog regulates growth and morphogenesis of the tooth. *Development* **127**, 4775–4785 (2000).
57. Carpenter, A. C., Rao, S., Wells, J. M., Campbell, K. & Lang, R. A. Generation of mice with a conditional null allele for Wntless. *Genesis* **48**, 554–558 (2010).
58. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* **33**, 495–502 (2015).
59. Santos, A., Wernersson, R. & Jensen, L. J. Cyclebase 3.0: a multi-organism database on cell-cycle regulation and phenotypes. *Nucleic Acids Res.* **43**, D1140–D1144 (2015).
60. Liu, Z. et al. Reconstructing cell cycle pseudo time-series via single-cell transcriptome data. *Nat. Commun.* **8**, 22 (2017).
61. Günesdogan, U., Jäckle, H. & Herzig, A. Histone supply regulates s phase timing and cell cycle progression. *eLife* **3**, e02443 (2014).
62. Moon, K. R. et al. Visualizing structure and transitions in high-dimensional biological data. *Nat. Biotechnol.* **37**, 1482–1492 (2019).
63. Wood, S. & Wood, M. S. Package 'mgcv'. scholar.google.com/citations?view_op=view_citation&hl=it&user=EskiIyEAAAAJ&citation_for_view=EskiIyEAAAAJ:kh2fBNsKQNwC (2015).
64. Bergen, V., Lange, M., Peidli, S., Wolf, F. A. & Theis, F. J. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nat. Biotechnol.* **38**, 1408–1414 (2020).

Acknowledgements

The authors thank J. Yang and M. Roulis for fruitful discussions. This study was supported by the National Institutes of Health (NIH) under grants R01GM131642 (to Y.K. and X.C.), UM1DA051410, U54AG076043, U54AG079759, P50CA121974 and U01DA053628 (to Y.K.). X.C. is also partially supported by the National Science Foundation (NSF) grant DMS-2237842. P.M. is supported by the National Institute of Arthritis and Musculoskeletal and Skin Diseases (NIAMS) grant R01AR076420.

Author contributions

R.Q., X.C. and Y.K. conceived the project and designed the framework. R.Q. and X.C. developed the method. R.Q. performed data analysis and wrote the manuscript. X.C. developed the computation methodology based on mathematical theories and contributed to writing. P.M. performed the experiments and interpreted the findings. Y.K., P.M., J.S.S. and E.S. contributed to the writing and offered vital insights into improving the work. E.S., P.M., R.A.F. and I.D.O. contributed to the overall biological interpretation. B.L. and R.C. offered conceptual insights related to the theoretical framework. F.S. assisted in software implementation. S.P. assisted in experimental data analysis. J.G. assisted in writing.

Competing interests

R.A.F. is an advisor to GlaxoSmithKline, Zai Lab and Ventus Therapeutics. F.S. is employed as a director by PCMGF Limited. I.D.O. is the founder and president of Plythera and receives research funding from Ventus Therapeutics and SenTry.

Additional information

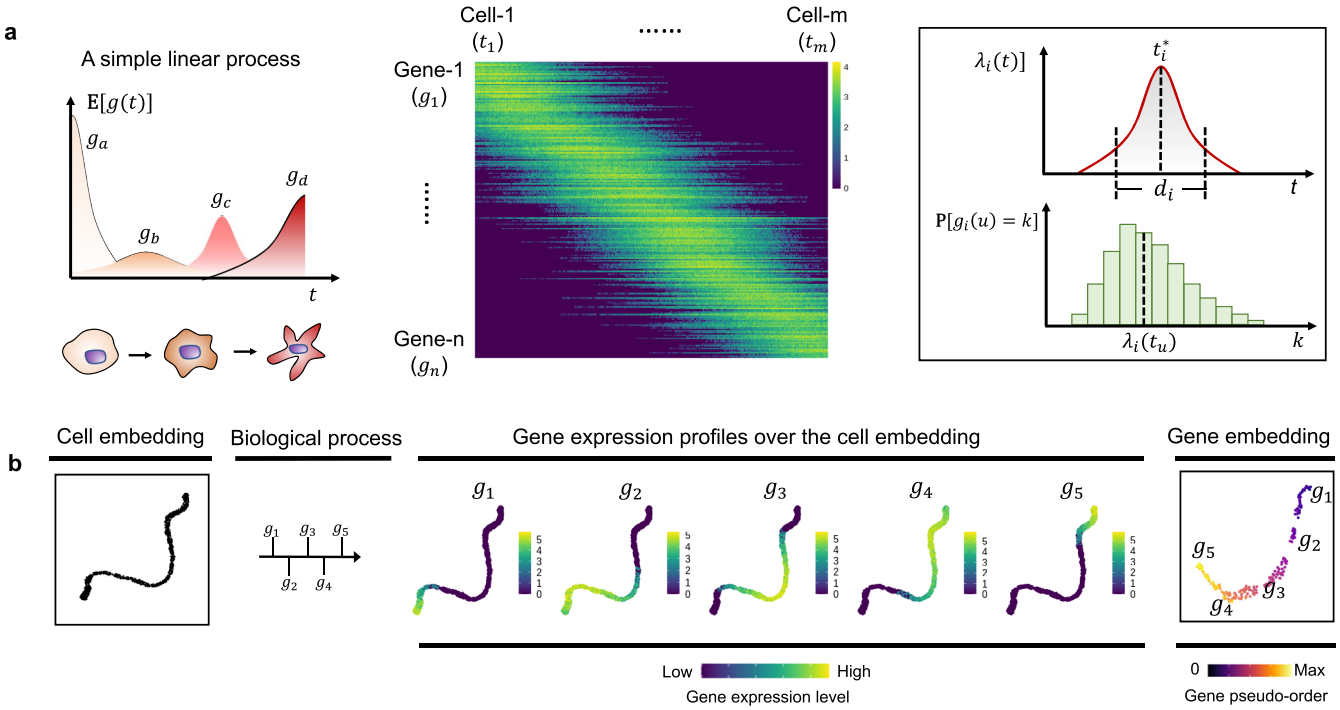
Extended data is available for this paper at https://doi.org/10.1038/s41587-024-02186-3.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41587-024-02186-3.

Correspondence and requests for materials should be addressed to Yuval Kluger.

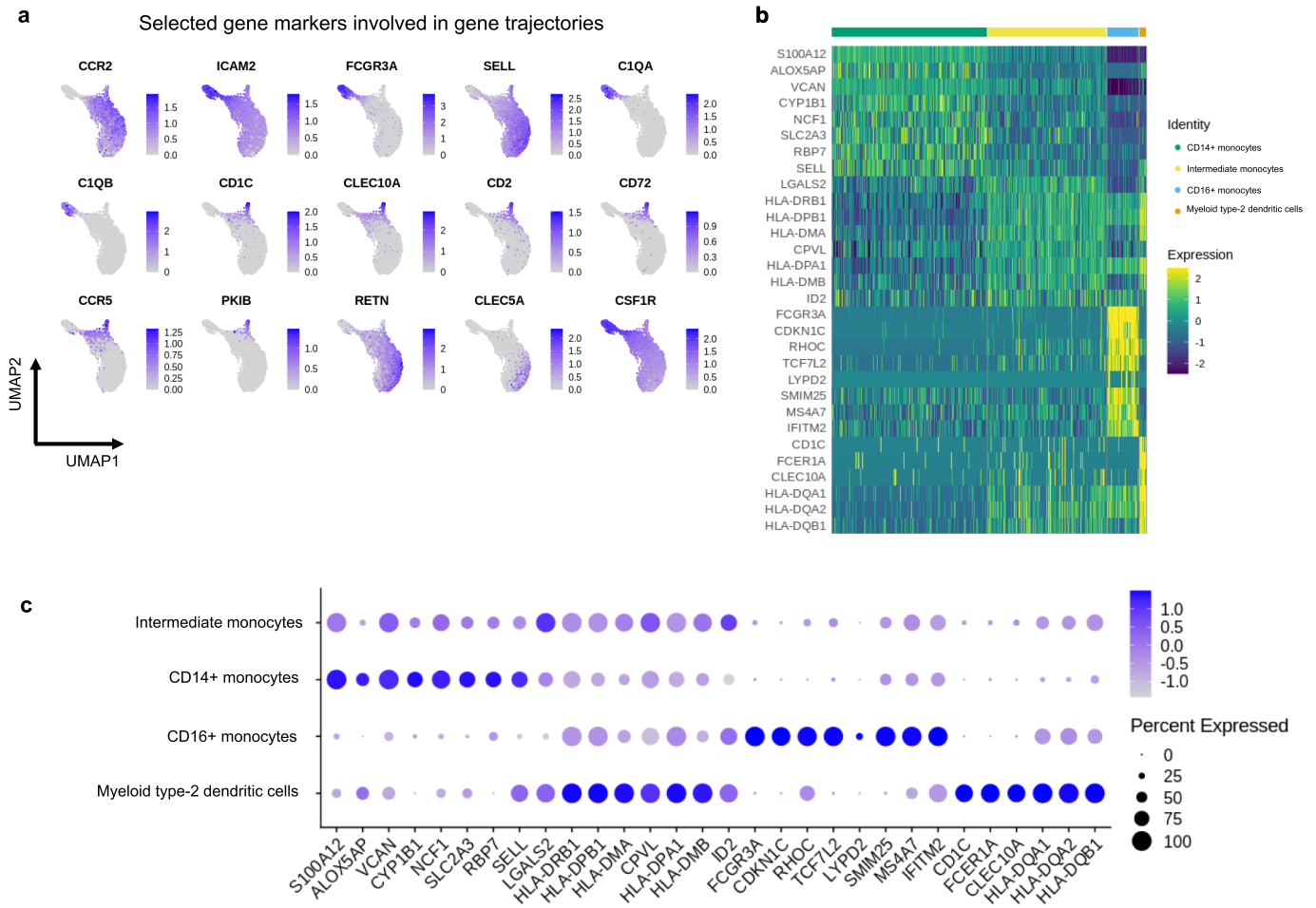
Peer review information *Nature Biotechnology* thanks Xiaojie Qiu and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

https://doi.org/10.1038/s41587-024-02186-3



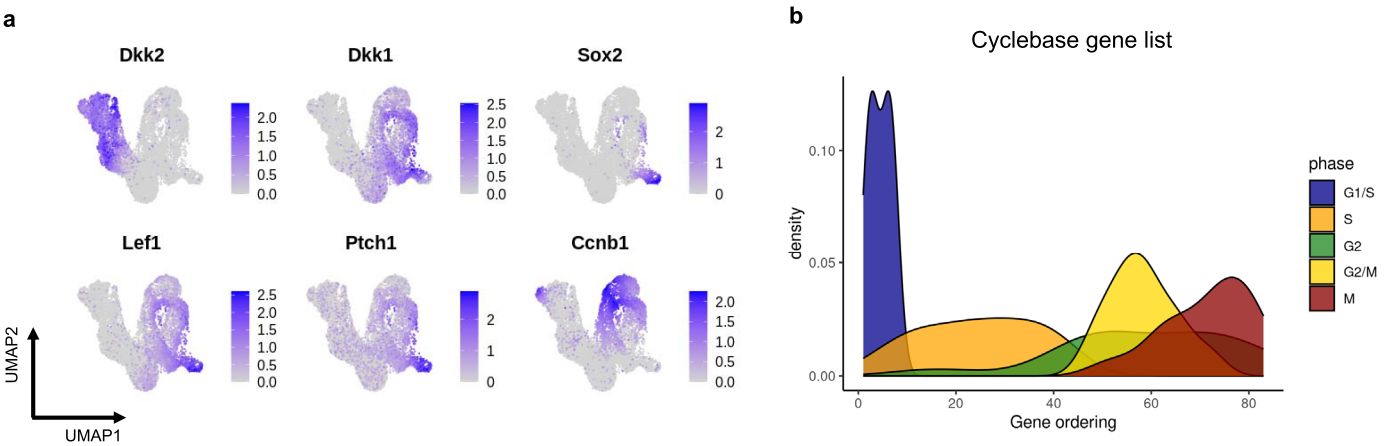
Extended Data Fig. 1 | Simulation framework and dataset visualization. **a**. Illustration of GeneTrajectory simulation framework. A simple linear differentiation process simulation is shown. Each cell is associated with a pseudotime *t* along the process. For each gene, its expected expression level is modeled as a bell-shaped function of *t*, its real expression level in a given cell is drawn from a Poisson distribution (see details in Methods). **b**. GeneTrajectory analysis on the simulated data in **a**. The first panel shows the UMAP embedding of cells; the second panel delineates the progressive dynamics of the simulated biological process with five genes selected along each process; the 3rd–7th panels show the expression of selected genes in the cell embedding following their pseudotemporal order; The 8th panel displays the UMAP embedding of genes, colored by the ground truth of gene pseudo-order. **c**–**f**. Gene-by-cell count matrices visualized by heatmaps (in log scale). Each row corresponds to a gene, each column corresponds to a cell. Each heatmap corresponds to a simulation example in Fig. 2.

https://doi.org/10.1038/s41587-024-02186-3



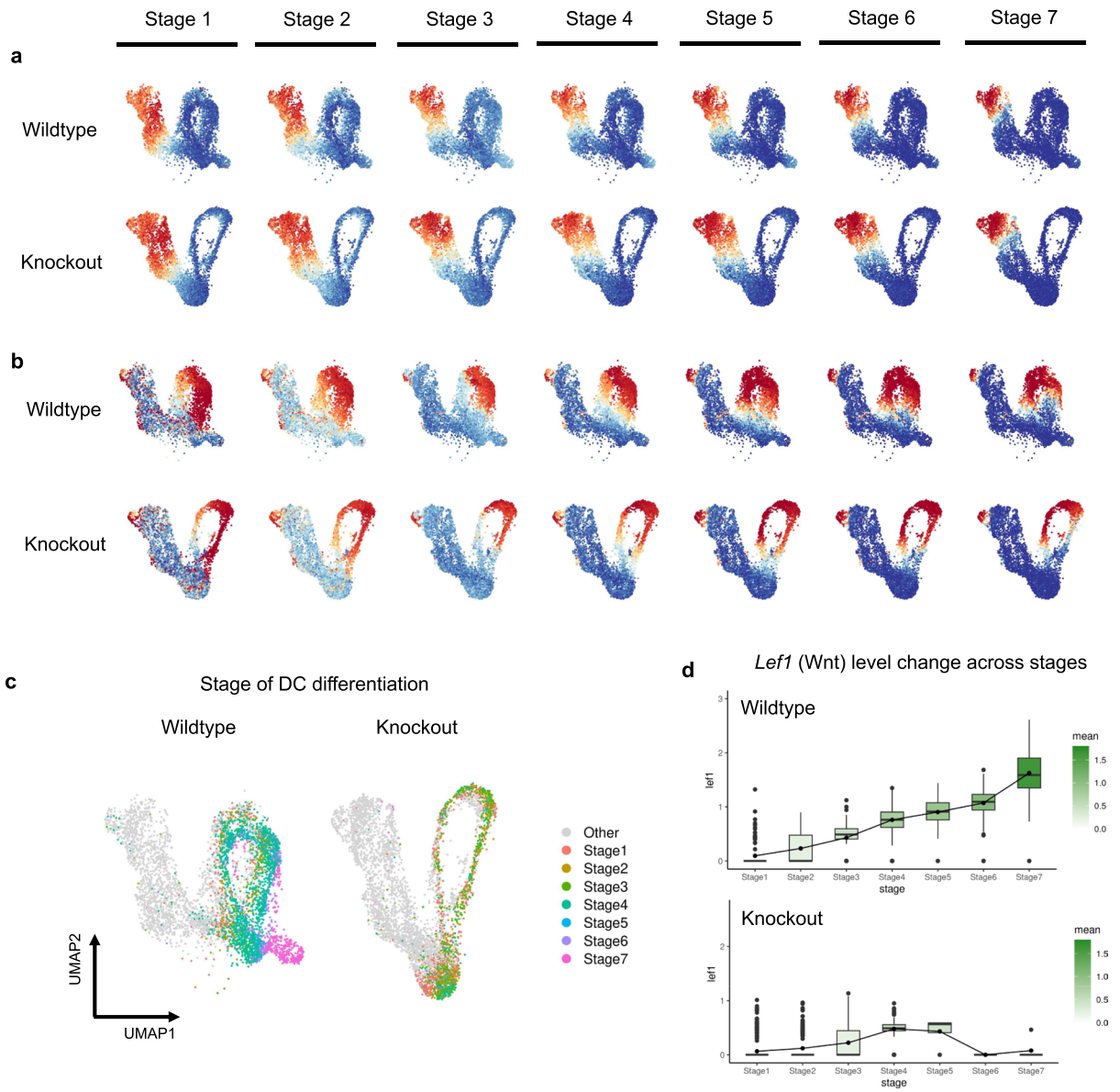
Extended Data Fig. 2 | Myeloid cell type stratification. a. UMAPs of selected well-studied myeloid gene markers identified along gene trajectories. **b**. Heatmap of cell-type specific gene markers (showing for each cell type the genes with the highest fold change in the average expression between that cell type and the remaining ones). **c**. Dot plot of cell-type specific gene markers in **b**. The color here indicates the average expression level of each gene in the corresponding cell type (after scaling).

https://doi.org/10.1038/s41587-024-02186-3



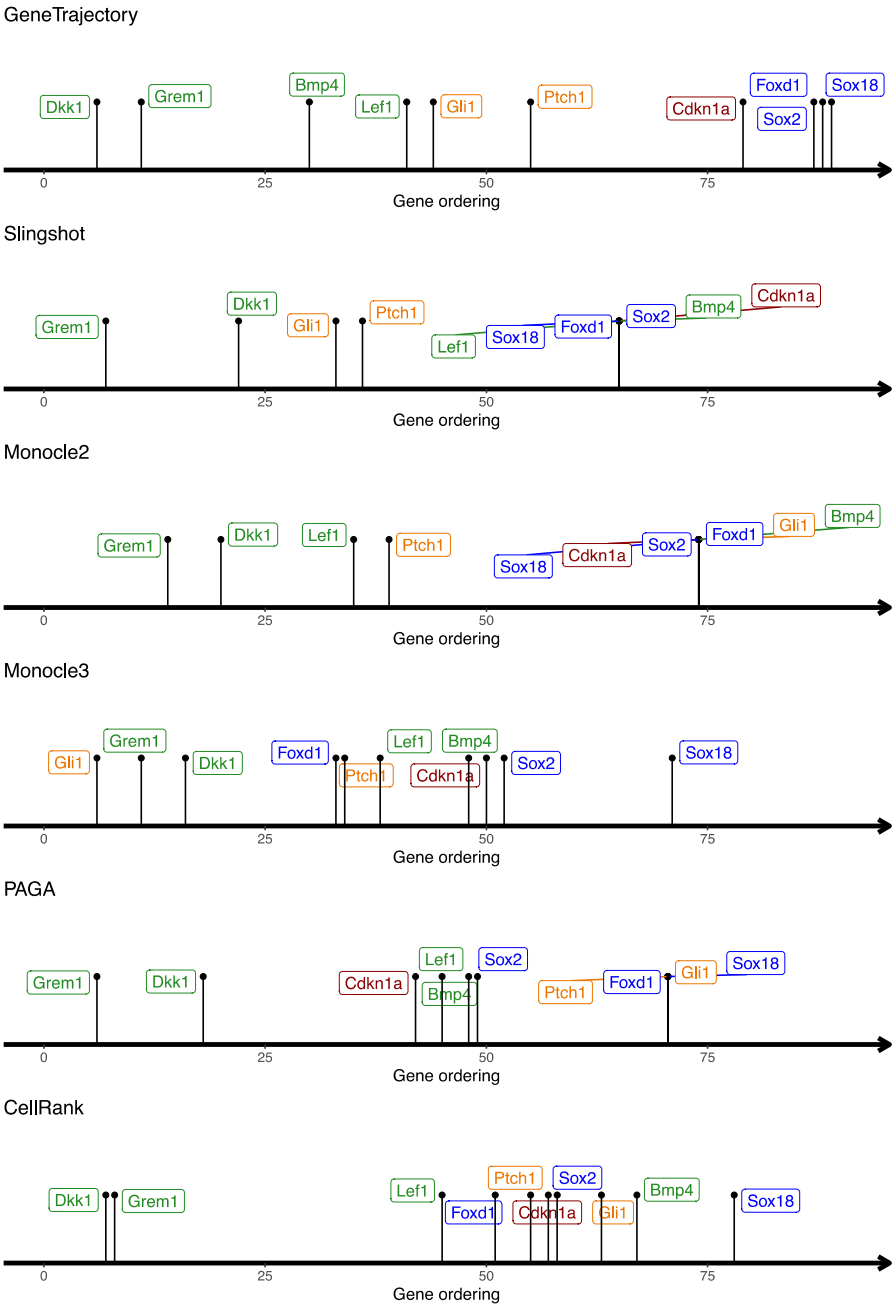
Extended Data Fig. 3 | Dermal cell type stratification. a. UMAPs of gene expression profiles. **b.** Distribution of genes associated with different cell cycle phases along the CC gene trajectory.

https://doi.org/10.1038/s41587-024-02186-3



Extended Data Fig. 4 | Gene dynamics comparison between the wild type and Wls mutant. a. Gene bin plots of the LD gene trajectory, split by condition. **b**. Gene bin plots of the CC gene trajectory, split by condition. **c**. Cell UMAPs are colored by the cell states which are categorized into multiple stages, split by two conditions. **d**. Change of *Lef1* (Wnt) level across all stages, split by condition. *Lef1* level is uniformly lower in the Wls KO than in the wild type. The box represents the interquartile range (IQR), with the line inside the box indicating the median. Whiskers extend to a maximum of 1.5× IQR beyond the box, with outliers represented as individual points.

https://doi.org/10.1038/s41587-024-02186-3



Extended Data Fig. 5 | Gene ordering results obtained by different methods on the dermal condensate genesis data. The orderings of key genes activated during the dermal condensate differentiation process are delineated. Cell cycle effects were regressed out when constructing the cell graph.

nature portfolio

Corresponding author(s): Yuval Kluger, Peggy Myung, Richard A. Flavell

Last updated by author(s): 02/18/2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	A mouse embryonic skin dataset (including scRNA-seq data generated from a pair of wildtype and Wntless knockout mice) was collected for data analysis in this study.
Data analysis	GeneTrajectory 0.1.0, Seurat 4.3.0, mgcv 1.9-0, slingshot 2.8.0, monocle 2.22.0, monocle3 1.3.4, cellrank 1.5.1, scvelo 0.2.5, velocyto 0.17.17, Cellranger 6.1.2, scanpy 1.9.1

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The human peripheral blood mononuclear cell (PBMC) scRNA-seq dataset is available at https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/ pbmc 10k v3. The mouse embryonic skin dataset generated and analyzed in this study is available from the Gene Expression Omnibus (GEO) with

the accession number GSE255534. The processed Seurat data objects for these two datasets are available at Figshare (dx.doi.org/10.6084/m9.figshare.25243225). The Cyclebase gene list was extracted from the Supplementary Table 5 in https://doi.org/10.1038/s41467-017-00039-z..

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	n/a
Population characteristics	n/a
Recruitment	n/a
Ethics oversight	n/a

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	GeneTrajectory was applied to multiple simulation datasets (each with 10 replicates) and two real-world biological datasets. The number of datasets involved in the study was determined based on the availability of data and the complexity of the research question. For the mouse skin experiments, n = 8 (WT) and n = 9 (KO) embryos examined over 4 biologically independent experiments with similar results.
Data exclusions	PBMC myeloid example: Data was downloaded from https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_v3. QC was performed using the same workflow in https://github.com/satijalab/Integration2019/blob/master/preprocessing_scripts/pbmc_10k_v3.R. Myeloid cells were extracted and retained in the analyses using markers CD14, FCGR3A, CD1C. Dermal condensate example: Cells with nGene <= 1000 or nGene >= 6000 or mitochondrial ratio > 10% were removed for QC. Dermal cells were extracted and retained in the analyses using markers Dkk1, Dkk2, Lef1, Sox2.
Replication	For Wls KO and control samples, scRNA-seq inferences and gene expression were validated using quantitative FISH as well as other data not shown (whole mount volumetric immunofluorescent staining of dermis and epidermis with Sox2, Sox9, EdU). The phenotype and finding were repeated n=4 for biological experiments and 2 embryos per condition were pooled for scRNA-seq experiments. The RNA findings are consistent with FISH results (where both females and males were used) and another unpublished scRNA-scATAC-seq (multiome) experiment (data not shown), which will be provided to reviewers if requested.
Randomization	Embryos were randomly prepared simultaneous for scRNA-seq preparation and pooled only after genotyping for mutant alleles. Biological experiments were not randomized, as it would be impractical to randomly select samples to perform FISH (only 25% mutant per litter).
Blinding	Both control and KO embryos were from the same litter and skin cells from each embryo were prepared in a blinded fashion until just before submitting for scRNA library preparation. The genotypes were determined after preparation in order to pool and submit samples by condition. The library preparation and sequencing were done blinded. For FISH analysis and proliferation measurements of control and KO, it was not possible to truly blind the investigators, as the KO has no hair follicles. Nevertheless, the FISH staining procedure and EdU administration was done in a blinded fashion. All RNA FISH quantifications were done by standard scoring of dots per cell within the upper dermis.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☐	☒ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Animals and other research organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research, and Sex and Gender in Research

Laboratory animals	Species: mus musculus, strain K14Cre;Wlsflox/flox (Dassule et al., 2000 and Carpenter et al., 2010); age: E14.5. All mice were housed according to an approved IACUC protocol within a limited-access animal facility that maintains a strict light cycle from 7 AM - 7 PM. Mouse room temperature and humidity are monitored continuosly and maintained at 22 (range 21-23) degrees celsius and relative humidity of 30-70%. All pregnant females are housed in a separate clean cage (maximum 5 mice per cage) with an automated water dispenser and basic mouse food and clean bedding that is changed weekly. Trained YARC veterinarians assess mice daily for signs of illness or poor feeding and alert investigators for signs of poor health.
Wild animals	No wild animals were used in the study.
Reporting on sex	For scRNA-seq samples, male embryos for each condition were used (using XY PCR to determine sex); 2 embryos per condition were pooled. Both male and female embryos were used to assess dermal condensate/hair follicle formation and transcript levels by FISH with no significant differences detected based on sex. Further, based on our other scRNA-seq and biological experiments in which equal number of both male and female sexes were pooled or only male or female embryos were used , there were no significant differences between sexes at this gestational age with respect to dermal differentiation genes or clusters.
Field-collected samples	No field collected samples were used in the study.
Ethics oversight	All procedures involving animal subjects were performed under the approval of the Institutional Animal Care and Use Committee of the Yale School of Medicine. All other regulatory standards were met in accordance with Yale's Environmental Health and Safety standards.

Note that full information on the approval of the study protocol must also be provided in the manuscript.