Published online 16 September 2022

Nucleic Acids Research, 2022, Vol. 50, No. 21 ***e122***
https://doi.org/10.1093/nar/gkac785

Density-based detection of cell transition states to construct disparate and bifurcating trajectories

Tian Lan[1], Gyorgy Hutvagner[2], Xuan Zhang[1], Tao Liu[3], Limsoon Wong[4] and Jinyan Li[1,*]

[1]Data Science Institute and School of Computer Science, University of Technology Sydney, Ultimo, NSW 2007, Australia, [2]School of Biomedical Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia, [3]Children's Cancer Institute Australia for Medical Research, Randwick, NSW 2031, Australia and [4]School of Computing, National University of Singapore, 13 Computing Drive, 117417, Singapore

Received May 06, 2022; Revised August 22, 2022; Editorial Decision August 27, 2022; Accepted September 01, 2022

ABSTRACT

Tree- and linear-shaped cell differentiation trajectories have been widely observed in developmental biologies and can be also inferred through computational methods from single-cell RNA-sequencing datasets. However, trajectories with complicated topologies such as loops, disparate lineages and bifurcating hierarchy remain difficult to infer accurately. Here, we introduce a density-based trajectory inference method capable of constructing diverse shapes of topological patterns including the most intriguing bifurcations. The novelty of our method is a step to exploit overlapping probability distributions to identify transition states of cells for determining connectability between cell clusters, and another step to infer a stable trajectory through a base-topology guided iterative fitting. Our method precisely re-constructed various benchmark reference trajectories. As a case study to demonstrate practical usefulness, our method was tested on single-cell RNA sequencing profiles of blood cells of SARS-CoV-2-infected patients. We not only re-discovered the linear trajectory bridging the transition from IgM plasmablast cells to developing neutrophils, and also found a previously-undiscovered lineage which can be rigorously supported by differentially expressed gene analysis.

INTRODUCTION

Cell trajectory inference is referred to as a computational process to construct a cell differentiation trajectory consisting of possibly multiple lineages from single-cell RNA-sequencing data (1,2). Cell trajectory inference is essential for a variety of research areas related to molecular function annotation (3–5), cell lineage identification (6,7), intra-tumor heterogeneity analysis (8), and cancer progression understanding (9,10). Many methods have been proposed recently for accurate trajectory inference, including PAGA (11), Monocle2 (12–14), Slingshot (15), Scorpius (16), Slicer (17), TSCAN (18), STREAM (19), Tempora (20), VIA (21) and Mutrans (22). These methods typically output a cell trajectory path organized in a temporal order with each cell assigned a differentiation pseudotime. High performance of these state-of-the-art methods are mainly attributed to the high-quality transcriptomic data acquired at single-cell resolution (18,23) which contains heterogeneity patterns detectable between the cell subpopulations (15,20,24), and also owing to the sophisticated data analysis for detecting the heterogeneity patterns such as through induction of minimum spanning trees (15,18), graph partitioning (11,25) or time-series prediction when sequential data sets are involved (20,26–28).

However, novel techniques are still needed for detecting various heterogeneity patterns to construct a wider range of cell trajectories including cycles, disparate lineages or bifurcating hierarchy. Current inference by Monocle2 and Slingshot can only be able to accurately derive linear-shaped or tree-structured trajectories, without much flexibility extended for more shapes (29). Other tools such as PAGA and Slicer have capability of inferring complicated topologies like a loop or a disconnected lineage, but their accuracy may drop significantly when the data complexity rises (29). In particular, these current methods show very limited performance on the inference of the much less investigated bifurcation trajectories which are a more intriguing type of cell developmental path (30–32), involving a complex cellular hierarchy of multiple lineage specifications bridged into cycles.

Cells developed from one state to another usually undergo a smooth and gradual change of gene expression (15,33). Cells under such gradually changing states cannot be categorized into any specific, distinct type. Instead, they can only be postulated to configure a narrow transient window between two distinct cell states, and thus they are called 'vague cells' here. Expression profiles of these vague cells potentially provide key information to help tra-

*To whom correspondence should be addressed. Tel: +61 295149264; Fax: +61 295149264; Email: jinyan.li@uts.edu.au

jectory inference tools predict whether or not a cell sub-population will differentiate into other ones.

This work introduces a density-based cell trajectory inference method (DBCTI) to detect all possible clusters of vague cells for accurate inference of disparate, cyclical and bifurcating trajectories. Such vague cells serve like a function of anchors helping link cell sub-populations. By our method, a density-based clustering step is first performed to determine all sub-populations of the cells; then these populations of cells are fitted with probability distributions; subsequently the probability distributions are joined to identify clusters of transient vague cells representing cell-fate transition states.

With the density information from all these transition states, DBCTI decides whether any two states of cells should be separated, or should be branched, or should be bridged to build a loop. Such a step allows plenty of flexibility for our method to construct diverse shapes of cell trajectories fitting the data. In contrast, the rigid structure of minimum spanning tree as adopted by TSCAN, Slingshot and Waterfall, restricts the shape of trajectories not to having any disparate edges; it does not allow any cycle in the shape either (3,15,18). Another novel step in our method is an iterative fitting step guided by a base topology so that the trajectory inference can be converged into a stable trajectory. Furthermore, our method DBCTI uses the robust minimum covariance determinants (34) to mitigate the high rates of low-depth or dropout effects in single-cell RNA-sequencing data (35–37) and thus overcomes this challenging issue faced by the current trajectory inference methods in the cell-state identification step (38).

To demonstrate our method's superior performance, we show diverse trajectory patterns derived from 15 simulated and five experimental reference datasets, with comparison to those derived by three main-stream cell trajectory inference tools Monocle2 (Monocle3 unavailable under maintenance), Slingshot and TSCAN, and with comparison to those by PAGA and VIA which are the latest tools having the capabilities to infer complicated patterns such as cyclical loops and disconnected trajectories. We also compare with MuTrans, a recently published tool that utilizes a similar notion of transition states as our DBCTI for inferring cell trajectories from single-cell RNA sequencing data. In-depth case studies are then followed to understand deeper about cell trajectories of SARS-CoV-2-infected peripheral blood mononuclear cells.

MATERIALS AND METHODS

Method overview

DBCTI has four main steps in its workflow: density-based clustering of the cell population, detection of transition states of the cells, trajectory inference and pseudo-time assignment (Figure 1A–D). The input file to DBCTI is a transcript-by-cell single-cell RNA sequencing count matrix, and the method outputs a trajectory defined by the pseudo-time assigned to each cell.

Pre-processing steps in our method include cell filtering, gene selection and counts normalization to clean the count matrix. A dimension reduction step is then followed to work on the cleaned matrix to embed all of the cells into a 2D hyperspace through a principal component analysis (39) together with a t-distributed stochastic neighbour embedding tSNE (40). Then, both parametric and non-parametric density-based clustering algorithms, specifically the finite multi-variate and kernel density estimation (KDE) algorithms (41), are used to form cell clusters which are called cell states (Figure 1A).

After the cell states are detected, each of them is fitted with a probability distribution using the minimum covariance determinant (MCD) method (34). Then overlapping distributions are identified and exploited to detect 'vague cells'. Here, vague cells are defined as those cells belonging to more than one distributions (see Figure 1B, more details in Figure 1E).

Under the idea that distance-closer cells share higher similarities while the farther ones represent more heterogeneity, these vague cells (points) are collectively considered as a gene expression transient (transition) window for distance calculation, and then the distance is used to determine the connectability between the corresponding two states, thus forging a base topology of the cell trajectory (see Figure 1C, more details in Figure 1F). This step for the identification of between-states connectability is a novel and the most important step in our method. This between-states connectability identification gives rise to much flexibility for our method to potentially infer diverse shapes of trajectories.

After the identification of between-states connectability for all possible pairs of cell clusters, DBCTI goes to fit a principal curve for each pair of connectable states guided by the base topology, and refines the curve iteratively to reach a stable trajectory. Finally DBCTI assigns each cell alongside the trajectory a pseudo-time (Figure 1 D), which is proportional to the inferred distance between the pairs of cells. Detailed description of these steps are presented at the following subsections.

Steps for data pre-processing and cell embedding

DBCTI filters the data by excluding those genes and cells having gene expression less than a specified threshold. Normalization is carried out by dividing each gene's count in a cell by the total count of the genes in that cell and multiplied by a scaling factor set as 10^4 by default (similarly as many single-cell RNA-sequencing data pre-processing and analysis tools such as Seurat (42)):

$$n_{norm(g,c)} = \frac{n_{g,c}}{N_c} \times scale\ factor \quad (1)$$

where $n_{norm(g,\,c)}$ is the value after normalization, $n_{g,\,c}$ is the original count of gene g in cell c and N_c is the sum of all gene counts in cell c.

DBTCI has three options for gene selection via variation (as default), correlation score or NMF. The variation approach (similarly as taken by Seurat) calculates the variance for each gene and selects top n genes with the highest variance (default set as 2000 the same as Seurat). The option via correlation score (43) or via NMF (44,45) for gene selection requires a target gene list. The correlation approach calculates the genes' correlation to each of the targets, and for each target it chooses those genes that are most positively correlated with. The genes are then merged without dupli-

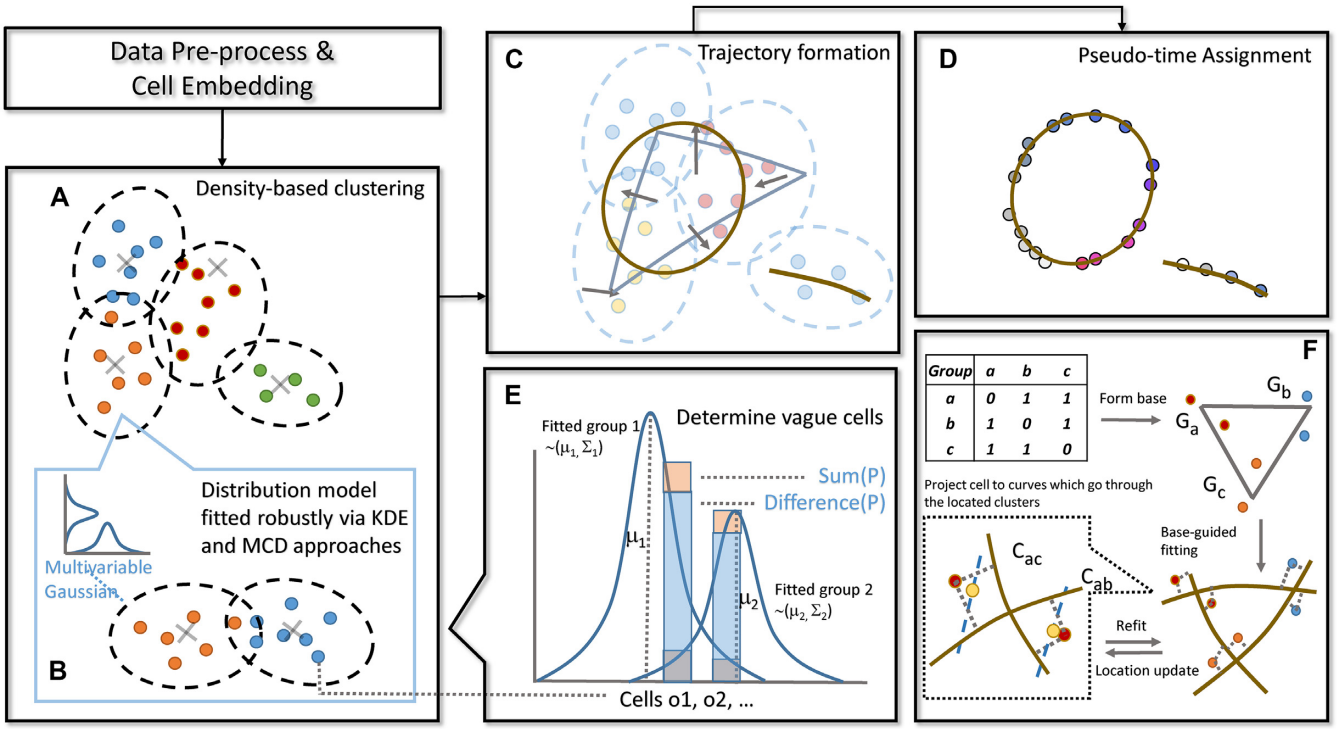


Figure 1. Main steps and workflow of DBCTI. (**A**) Cell clusters are identified via density-based clustering approaches. (**B**) Probability distributions are fitted for the cell clusters to detect vague cells. (**C**) Trajectory formation. (**D**) Pseudo-time assignment. (**E**) More details of the vague cell determination process. (**F**) More details of the trajectory formation process.

cates. The NMF approach uses the following equation for gene selection:

$$V \approx W \times H \tag{2}$$

where V is the data matrix with dimensions $g \times c$, being decomposed into W with $g \times k$ dimensions and H with $k \times c$ dimensions. K-nearest neighbour algorithm (KNN) is then used to find the top-k nearest neighbour for each target gene based on the W matrix.

After gene selection, principal component analysis is applied to extract principal components of the data, reducing the number of dimensions to n. Here, n is set as 50 (the same setting as the default parameter in Seurat pipeline), aiming to reduce dimension effectively while keeping adequate information. T-distributed stochastic neighbour embedding (tSNE) is then applied to further reduce the dimension to 2, for dealing with non-linear structures in the data. We suggest that the perplexity parameter of tSNE should be set around 10% of the total cell numbers, to preserve structures for closed cell clusters that have high similarities. The perplexity that DBCTI sets for tSNE is 30 by default, which is similarly implemented in most single-cell RNA sequencing methods (46).

Density-based clustering to determine cell sub-populations (cell states)

A density-based clustering algorithm is applied to group the cells into clusters (i.e. sub-populations or states) by fitting the data with a Gaussian finite mixture model (47):

$$f(x_i; \Psi) = \sum_{k=1}^{G} \pi_k f_k(x_i; \boldsymbol{\theta}_k) \tag{3}$$

where π_k are the weights of the k-th distribution, $f_k(\boldsymbol{x}_i; \boldsymbol{\theta}_k)$ is the density of observation x_i in the kth model f_k with model parameters $\boldsymbol{\theta}_k$, G is the total component number of distributions and Ψ is the mixture model parameters. The parameters are fitted via the expectation maximization (EM) algorithm. The number of components as well as the model type are set according to the Bayesian information criterion (BIC).

The contour plot of the clusters is built via a kernel density estimation approach (41). The final states of the cells are determined by merging some of the initial clusters with the help of the contour plot. For each of the final states, a minimum covariance determinant (MCD) estimator is used to infer a covariance matrix of the data and fit it into a two-dimensional Gaussian distribution by finding the h observations within the whole H data that has the lowest classical covariance determinant through an iteration process named C-step (34). After that, DBCTI applies an expansion factor (set as 2.5 by default) on each distribution in order to increase the area of intersection for facilitating the vague cell identification.

Identification of vague cells to annotate a transition state: our novel step

DBCTI takes a sampling approach to calculate the probability of a cell point belonging to different cell states (distributions). For each data point, an area centred at that point with a length and width both equal to $2r$ are selected (r value is set as 2 by default but tuning for this parameter won't affect much to the result). Samples are then drawn from all of the fitted distributions one-by-one. Then those points inside the selected area are counted. The probability of one cell point belonging to a state is defined as:

$$\frac{n_{c,d}}{N_d} \times 100\% \tag{4}$$

where $n_{c,d}$ is the number of sample points from distribution d dropped inside the area centred at point c, and N_d is the total number of sampled points from the distribution d.

DBCTI confirms a vague cell point c for a transition state if

$$\begin{cases} P_{c,d1} + P_{c,d2} \geq C_{sum} \\ \left| P_{c,d1} - P_{c,d2} \right| \leq C_{diff} \end{cases} \tag{5}$$

where $P_{c,d1}$ represents the probability of c belonging to the $d1$ distribution, $P_{c,d2}$ stands for the probability of c belonging to the $d2$ distribution, C_{sum} and C_{diff} are real number thresholds. In other words, if the sum of the possibility of a cell c belonging to the $d1$ distribution and that of belonging to the $d2$ distribution is greater than C_{sum}, while the absolute difference between $P_{c,d1}$ and $P_{c,d2}$ is smaller than C_{diff}, this cell is defined as a vague cell in a transition state. See Figure 1E for an illustration.

Based on our observations from numerous robustness trials and tests, we recommend (although theoretically difficult) that C_{sum} should be set around 0.8 and C_{diff} should be set around 0.5 for accurately constructing trajectories in the downstream steps. We have tested that such a criteria can capture the edge cells which are in the middle of gene expression transition between two distinct cell sub-populations.

Iterative construction towards a stable trajectory: our novel step

DBCTI decides a binary relation (link or not-link) between two cell states based on the presentation/absence of vague cells in the transition state. Two clusters are linked if the vague cell proportion in that two clusters is larger than a threshold. When two clusters are linkable (connectable), a principal curve $f(s)$ (48) is fitted for all of the points in the two clusters such that:

$$\begin{aligned} &\mathbf{f}(s) = \mathbf{E}\left[\mathbf{Y} \mid s_{\mathbf{f}}(\mathbf{y}) = s\right]; \quad \mathbf{Y} = \{\mathbf{Y_1} \cup \mathbf{Y_2}\} \\ &where \quad \mathbf{Y_1} \sim d1, \mathbf{Y_2} \sim d2 \end{aligned} \tag{6}$$

where the projection index $s_{\mathbf{f}}(\mathbf{y})$ of y is the value of λ for which $\mathbf{f}(\lambda)$ is closest to y. $f(s)$ is the expected value of the distribution for data point that has a projection index equal to s, and $\mathbf{Y}$ is a vector containing observation values from both $d1$ and $d2$ distributions, respectively ($\mathbf{Y_1}$ contains values from $d1$ and $\mathbf{Y_2}$ contains values from $d2$). The parameters of $f(s)$ are refined via the EM algorithm.

If a cluster is fitted with more than one curves linking to other different clusters, DBCTI calculates the average position of the data on the fitted principal curves projected from the original points. Suppose there are points from cluster D fitted with n principal curves where $n \geq 2$, then the average position of the data is:

$$s_{\mathbf{f}_{D1}}(\mathbf{y}) = s_{D1}, \quad s_{\mathbf{f}_{D2}}(\mathbf{y}) = s_{D2}, \ldots s_{\mathbf{f}_{Dn}}(\mathbf{y}) = s_{Dn}$$

$$f(s) = \frac{1}{n} \times \sum_{i=1}^{n} f_{Dn}(s_{Dn}) \tag{7}$$

$$with \quad s = \frac{1}{n} \times \sum_{i=1}^{n} s_{Dn} \tag{8}$$

where $s_{\mathbf{f}_{Dn}}$ is the projection index for the principal curve via cell population D_n. This process iterates till changes of the positions converge. See an illustration of the step at Figure 1(F).

We have observed that for most datasets we tested in this work, there are about 1.0% cells in two linkable clusters which have had graduate transition change of gene expression from one cluster to the other. Therefore, we set the vague cell proportion threshold as 1.0%. Users have choice to tune this parameter to infer conserved or radical cell trajectories.

Detailed step for pseudo-time assignment for each cell

DBCTI calculates a pseudo-time for each cell using the Euclidean distance between the cell position and a specified initial cell state, based on the obtained trajectory plot (Figure 1D). The interval between cell states is ignored in the calculation so that the pseudo-time can be continuous. Given a cell, DBCTI first identifies the cluster which the cell belongs to and then finds the shortest path from that cluster to the initial cluster. A pseudo-time for cell c is hence defined as:

$$T_c = \sum_{D=1}^{m-1} \sum_{n=1}^{N_D-1} d(P_{D,n}, P_{D,n+1}) + \sum_{n=1}^{C_m-1} d(P_{m,n}, P_{m,n+1}) \tag{9}$$

$$where \quad d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \tag{10}$$

where N_d is the total number of cells in the cell population d, m is the total number of cell populations on the shortest path from the population where c belongs to the initial cell state, C_m is the cell index of c in population m. A cell index represents the order of cells in its corresponding cluster. The cell labeled as the first one is the one at the end of the fitted principal curve closer to one of the previous cell populations on the obtained shortest path. To handle the situation where the trajectory diagram is disconnected, separated pseudo-time settings are used for each individual component.

Simulated and experimental datasets

We have employed total fifteen simulated datasets through the generate_dataset function in the software dyngen (49) in three rounds for performance evaluation. At each round,

Table 1. Benchmark datasets summary

Data Name	Ref. topology	Cell no.	Gene no.	Organism
Simulated	Linear	1000	1021	*In silico*
Simulated	cyclical	1000	1020	*In silico*
Simulated	Binary tree	1000	1059	*In silico*
Simulated	Disconnected	1000	1044	*In silico*
Simulated	BF. circle	1000	1014	*In silico*
Yan-dataset	Linear	90	18 573	*H. sapiens*
Nestorowa-dataset	Binary tree	472	4766	*M. musculus*
Kowalczyk-dataset	Linear	524	7748	*M. musculus*
Leng-dataset	cyclical/linear	247	17 553	*H. sapiens*
Camp-dataset	disconnected	200	16 043	*H. sapiens*

BF circle stands for bifurcating circle, *H. sapiens* for *Homo sapiens* and *M. musculus* for *Mus musculus*.

the backbones were specified as linear, cyclical, binary tree, disconnected and bifurcating cycle with other parameters set by default to generate five of the fifteen simulated single-cell RNA-sequencing datasets.

We have also used five experimental single-cell RNA-sequencing datasets, which had been considered as reference benchmark datasets in the field, to evaluate DBCTI's performance. All cells of these five experimental single-cell mRNA-seq datasets are class annotated. These datasets are abbreviated as Leng-dataset (50), Camp-dataset (51), Yan-dataset (52), Kowalczyk-dataset (53) and Nestorowa-dataset (54). The Leng-dataset (50) is sequenced from human embryonic stem cells, the Camp-dataset (51) is sequenced from human liver hepatocyte-like cells, the Yan-dataset (52) is sequenced from human early embryos and embryonic stem cells, and the Kowalczyk-dataset (53) and Nestorowa-dataset (54) are sequenced from mouse hematopoietic stem and progenitor cells respectively. To avoid batch effect, for the Camp-dataset, only those cells annotated in the same batch have been used. More details of these datasets are summarized in Table 1. Our case study and analysis involve two datasets. For the Wilk-dataset (55), the two patients are abbreviated as C1 and C2, and for the Melms-dataset (56), the patient is abbreviated as L01cov.

Comparison with six cell trajectory inference methods and performance benchmark metrics

We compare DBCTI with three main-stream cell trajectory inference tools Monocle2 (12–14), Slingshot (15) and TSCAN (18), and with PAGA (11) and VIA (21) which are the latest tools able to infer complicated patterns such as cyclical loops, and also with Mutrans (22) that uses a notion of transition states similar as DBCTI. The tools are all tested under the default parameters chosen by the dynwrap package with the infer_trajectory function except Monocle 2 is performed with its own default parameters. The cluster number k value of Mutrans is decided based on the epi value calculated by the EstClusterNum function. (We note that Monocle 3 was also considered for comparison, but it is currently under construction and re-development, not available for downloading.)

We define three metrics to quantify the performance in terms of both topological accuracy and cell cluster detection. A pattern consistency score is defined to examine whether a shape consistency exists between the inferred pattern and the reference. To get this score, all trajectories are firstly classified into one of the four basic shape categories: linear, cyclical, tree or disconnected. A linear shape is defined as a shape containing neither branching events nor cycles; A cyclical shape is defined as a closed loop; a tree structure is defined as a shape containing branching events but not containing any closed loop; a disconnected lineage is defined as a shape containing more than one sub-components regardless of the specific shapes within them; and finally a bifurcation trajectory is defined as a pattern containing both the cyclical and tree topologies. A score 1 is granted if the trajectory falls into the same category as the reference, otherwise a 0 score is granted. A pattern consistency score is a straightforward metric to examine the inference accuracy. A final score, $S_{con,D}(p, t)$, for tool t with regard to shape p is computed by averaging all the consistency scores from a collection of datasets D:

$$S_{con,D}(p,t) = \mathbf{E}[S_{con,d}(p,t)] \quad where \quad d \in D \tag{11}$$

A pattern similarity score quantifies the topological similarity between the references and the inferred trajectories. It counts the number of branching points (BP) and branches (BR) in the subcomponent i in the reference, denoted as $N_{BP,ref,i}$ and $N_{BR,ref,i}$, and those in the inferred trajectory by tools t we evaluated are denoted as $N_{BP,t,i}$ and $N_{BR,t,i}$. A *BP* is defined as a point where at least three distinct paths jointed, while a BR is defined as the maximum number of different choices of paths from one end in the graph to another end, where an end is the point that only connects to one path. We designed such metric to calculate the similarity score as both BR and BP are sensitive to the changes of topology we included so that being convenient to quantify the tool performance. The pattern similarity score of tool $tS_{sim}(t)$ is calculated as:

$$S_{sim}(t) = \begin{cases} \frac{1}{n_{com}} \sum_{i=1}^{n_{com}} \left(1 - \frac{1}{\sum_{n=1}^{N_t} E_{n,i}} \times (E_{t,i})\right) & if \quad n_{com} = n_{ref} \\ 0 & if \quad n_{com} \neq n_{ref} \end{cases} \tag{12}$$
$$where \quad E_t = \left|N_{BP,ref,i} - N_{BP,t,i}\right| + \left|N_{BR,ref,i} - N_{BR,t,i}\right|$$

where N_t is the total number of tools that have the same subcomponent number n_{com} as the reference n_{ref}, and E_t is the error score of tool t. Here, the score is normalized by the number of tools n_{tool} to help compare across different datasets.

Performance accuracy depends not only on the inferred topological features, but also on the capability of accurately cluster cells with same type together. A purity score is defined to evaluate the accuracy of the clustering in comparison with the reference. All of the tools can output a clustering index in their result. To calculate a purity score S_P, clusters are formed with the maximum number of clusters set by the reference. The numbers of points belonging to correct classes in each cluster are added up and then divided by the total number of points:

$$S_P = \frac{1}{n} \sum_{q=1}^{k} \max_{1 \le j \le l} n_q^j \tag{13}$$

where n_q^j is the number of points in cluster q but that belong to class j, k is the total number of clusters, n is the total number of points and j is the number of classes.

These scores all range between 0 and 1. The higher score a tool obtains, the better its performance is.

Other methods for dimensionality reduction and parameter robustness test

TSNE is our suggested step for DBCTI to reduce the dimensionality of the data. Other dimension reduction methods are also available, for example, PHATE and DCA, via R packages. To understand which of the three dimensionality reduction methods is more suitable for this work, we calculate the purity score for the clusters from each of the five experimental reference datasets after each of the dimensionality reduction methods is applied. For a fair comparison, the same K-means algorithm is applied to all of the dimensionality-reduced datasets to determine the cluster index, including the dimensionality-reduced datasets after our DBCTI's tSNE step. The K parameter is automatically set by the NbClust function in the NbClust package (57,58).

There are three important parameters involved in Equations (4) and (5) for the identification of vague cells. To evaluate the impact of different settings of r on the trajectory result, we use 5 values from 1 to 3 with interval step 0.5. Via the point_possibility function implemented in DBCTI, each r value is used to calculate a probability assignment matrix, which denotes the probability of each cell belonging to each fitted distribution. We quantify the robustness by calculating the distance between the matrix obtained by the above r values and by r's default value, and then average them on each cell. Suppose the probability assignment matrix contains m distributions and n cells, the distance is then calculated as:

$$D = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{m}(e_{i,j} - r_{i,j})^2 \tag{14}$$

where $e_{i,j}$ and $r_{i,j}$ are elements in the probability assignment matrices obtained from the above r values and the default r value respectively.

Similarly, we investigate on how much the parameters C_sum and C_diff affect the performance when they take different settings. Each time, the parameter C_sum is set as a value within the range from 0.65 to 0.95 with interval step 0.05, and C_diff is set within the range from 0.35 to 0.65 with interval step 0.05. Then we use all of these two-parameter combinations (namely, 7×7=49 C_sum-C_diff pairs) for the experiment. Under each of these settings and for each dataset, we compute a distance matrix of the clusters, which is a matrix deciding if any two cell populations are linkable or not. This is implemented by the connect_cluster function, where 0 stands for unlinkable and 1 as linkable. Suppose k clusters are contained in the connection matrix, we calculate the matrix distance to examine the robustness:

$$d_{i,j} = \begin{cases} 0 & if \quad e_{i,j} = r_{i,j} \\ 1 & if \quad e_{i,j} \neq r_{i,j} \end{cases} \tag{15}$$

$$D = \frac{1}{k}\sum_{j=1}^{n}\sum_{i=1}^{m} d_{i,j} \tag{16}$$

where $e_{i,j}$ and $r_{i,j}$ are elements in the connection matrices obtained from the testing C_sum and C_diff combinations and from the default value C_sum as 0.8 and C_diff as 0.5, respectively.

RESULTS

Accurate and diverse patterns of cell trajectories constructed from simulated and experimental reference datasets

Figure 2A, F, K, P and U respectively shows the linear-shape trajectory, cyclical developmental loop, binary tree, disparate lineage and the bifurcating multi-lineage hierarchy generated in our first round simulation experiment. The five trajectories of different shapes simulated in the second round are presented in Supplementary Figure S2A–E, and those in the third round are shown in Supplementary Figure S3A–E. DBCTI precisely re-constructed all of these 15 cell trajectory patterns; see Figure 2B, G, L, Q and V, Supplementary Figures S2F–J and S3F–J, where DBCTI's robustness and performance consistency can be clearly observed across the three rounds of replicate datasets.

In particular, for the re-construction of the disparate lineage patterns, DBCTI is able to not only derive the two separated lineages, but also can precisely frame the special shape for each of the two lineages, namely a closed loop and a linear pattern (see Figure 2P and Q, Supplementary Figures S2D, I and S3D, I). Of notable interests, DBCTI is able to accurately re-construct the bifurcating loop topologies (see Figure 2U and V, Supplementary Figures S2e, j and S3E, J).

There are intensively-studied and widely-recognized cell differentiation schematic graphs (59,60) for the cell populations in the five experimental datasets (i.e. Leng-dataset (50), Camp-dataset (51), Yan-dataset (52), Kowalczyk-dataset (53) and Nestorowa-dataset (54)). These graph diagrams have been considered as reference templates about these cell trajectory paths. We show them at Figure 3A, F, K, P and U for benchmarking comparison with our computationally inferred cell trajectories.

From the Camp-dataset that is related to liver cell lineages, DBCTI constructed a two-component disconnected trajectory, indicating separated populations of the mesenchymal stem cells in two branches while both developed from the same pluripotent stem cells. Our inferred cell trajectory exactly echoes with the benchmark differentiation diagram, viz. Figure 3U versus V. DBCTI predicted a linear trajectory from the Yan-dataset (Figure 3B). Alongside the trajectory, the cells are ordered in the sequence of zygote/2-cell, 4-cell, 8-cell, 16-cell and the blast-cell state according to the annotation. It is a clear cell differentiation path for early embryo cells, coinciding with the benchmark schematic graph (Figure 3A). From the Nestorowa-dataset, DBCTI derived a tree structure as shown in Figure 3G, where the MEP and GMP cells are located at two separated ends while the cells possessing higher potency are located in front of the branching point. As most of the cells at the branching point belong to the CMP population, this

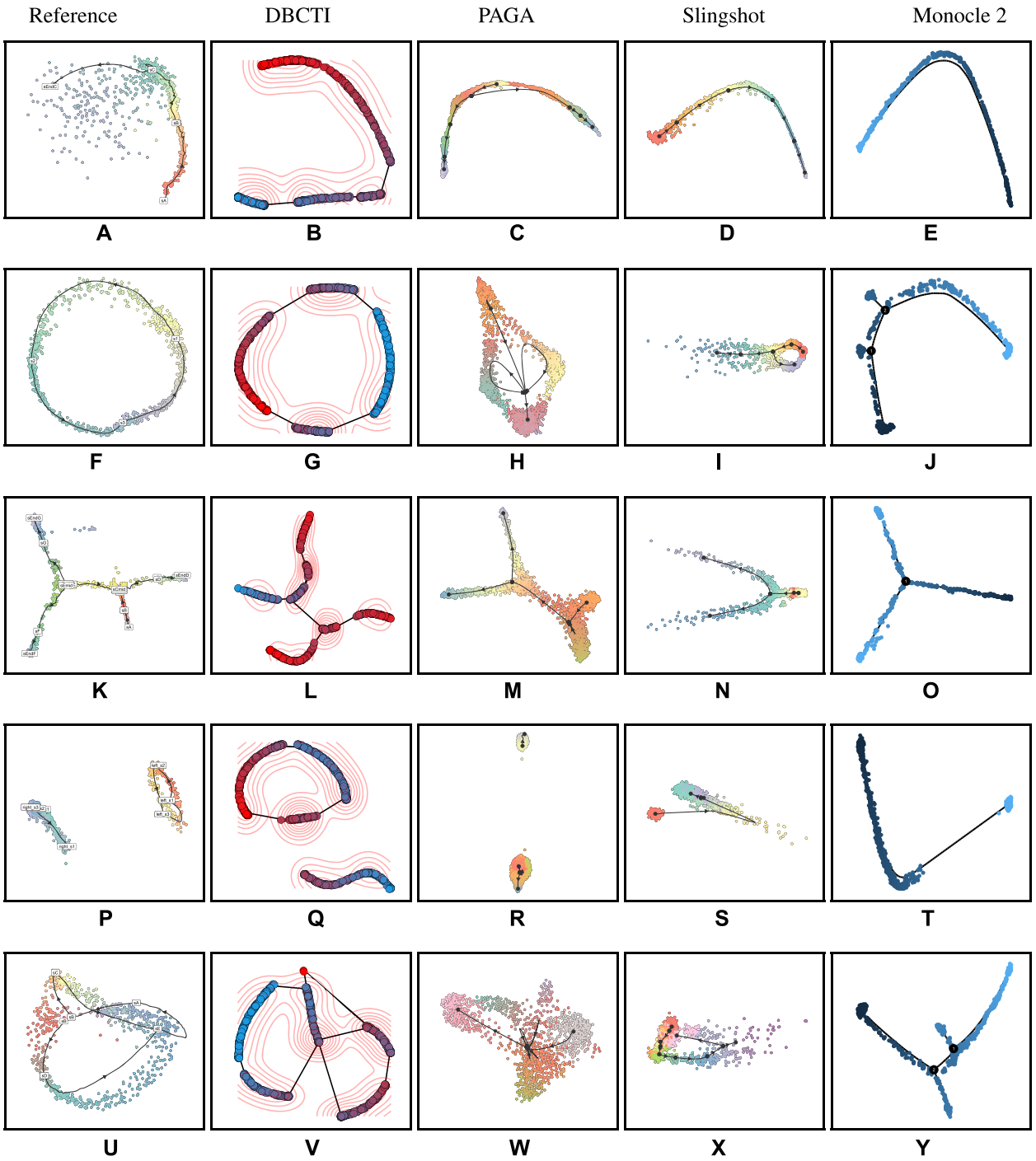


Figure 2. Reference trajectories in comparison with re-constructed trajectories from five simulated datasets by five tools. (**A**, **F**, **K**, **P**, **U**) Simulated linear lineage, cyclical shape, binary tree, disconnected trajectory and bifurcating loop. (**B**, **G**, **L**, **Q**, **V**) DBCTI re-constructed linear, cyclical, binary tree, disconnected and bifurcating trajectories (coloured by pseudo-time). (**C**, **H**, **M**, **R**, **W**) PAGA constructed trajectories from the simulated linear, cyclical, binary tree, disconnected and bifurcating trajectory datasets (coloured by pseudo-time). (**D**, **I**, **N**, **S**, **X**) Slingshot constructed trajectories from the five simulated datasets (coloured by inferred cell states). (**E**, **J**, **O**, **T**, **Y**): Monocle 2 constructed trajectories from the five simulated datasets.

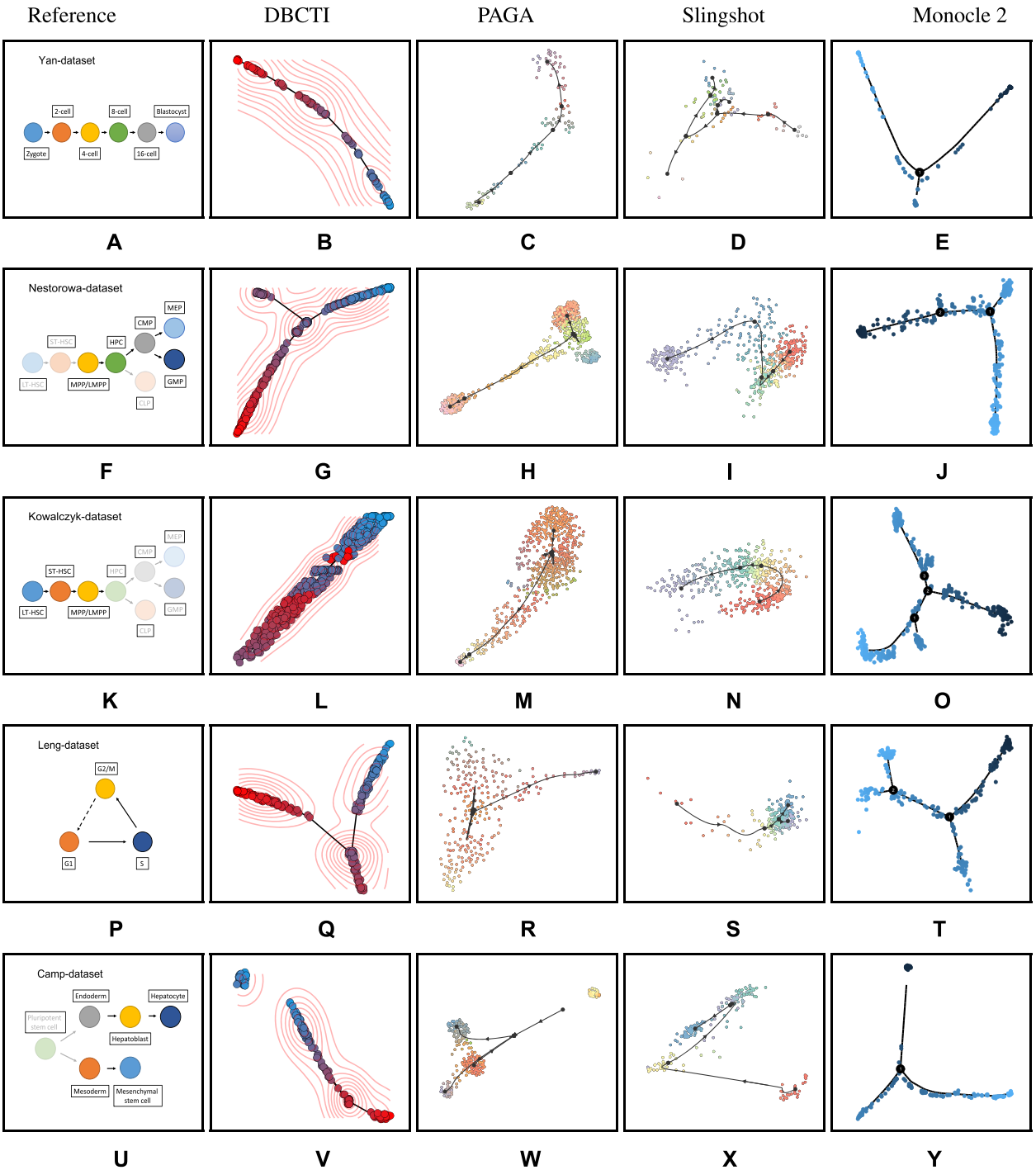


Figure 3. Widely-recognized schematic graphs of cell differentiations in comparison with computationally inferred trajectories from their experimental single-cell RNA-seq datasets. (**A**, **F**, **K**, **P**, **U**): Cell differentiation schematic graphs from the Yan-dataset, Nestorowa-dataset, Kowalczyk-dataset, Leng-dataset and Camp-dataset, where the solid colours stand for cell types annotated according to the datasets and the dashed lines represent dispensable trajectories. (**B**, **G**, **L**, **Q**, **V**) DBCTI inferred trajectories from the Yan-data, Nestorowa-data, Kowalczyk-data, Leng-data and Camp-data (coloured by pseudo-time). (**C**, **H**, **M**, **R**, **W**) PAGA inferred trajectories from the five experimental datasets (coloured by pseudo-time). (**D**, **I**, **N**, **S**, **X**) Slingshot inferred trajectories from the five experimental datasets (coloured by the cell states). (**E**, **J**, **O**, **T**, **Y**) Monocle 2 inferred trajectories from the five experimental datasets.

is highly consistent with the biological differentiation model shown in Figure 3F.

DBCTI outperforms recent tools for cell trajectory inference

We compared DBCTI with six recently proposed trajectory inference tools PAGA, Sligshot, Monocle2, VIA, TSCAN and Mutrans. We note that Monocle 3 was also considered for comparison, but it is currently under construction and re-development, not available for downloading. This performance comparison was conducted on the 15 simulated and the five experimental reference datasets (namely, we constructed total 120 cell trajectories for comparison). Results are shown in Figures 2 and 3 and Supplementary Figures S1–S10. DBCTI outperforms all of these existing tools on all of these 20 datasets. For example, on all of the three simulated datasets containing a cycle pattern, none of these tools except our DBCTI is able to infer a pure cyclical pattern. Although PAGA and VIA have inferred some trajectories containing a close loop, they are not shapes consistent with the simulated reference patterns (see Figure 2H, Supplementary Figures S7B, G, L and S9B, G, L). TSCAN or Slingshot have just constructed some opened cyclical-like patterns (Supplementary Figures S6G, L and S8G, L).

On the simulated three datasets containing a disparate lineage, DBCTI, PAGA and VIA have all built a trajectory containing two separated sub-components (Figure 2Q, R, Supplementary Figures S1I, S2I, S3I, S7D, I, N and S9D, I, N). But only DBCTI can recover the same topology for each of the two separated parts. Similarly, on the Camp-dataset where a disparate lineage exists, only DBCTI and VIA have successfully inferred a trajectory with two sub-components; DBCTI built the topology for each sub-component correctly while VIA did not.

In the re-construction of the simulated three bifurcating patterns, only our DBCTI can make accurate inference for all of them. PAGA has been just able to build a tree-like structure (see Figure 2(W) and Supplementary Figure S7(E,J,O)).

We attribute DBCTI's outstanding performance to its accurate detection of vague cells and to its base topology construction step. Specifically taking the bifurcating pattern (Figure 2U, V) as an example, it is the identification of vague cells by DBCTI that enables the closing of the loops while still allowing branched lineages between the loops.

We calculated pattern consistency scores, pattern similarity scores and purity scores to quantify the performance. Figure 4 shows the scores of the three metrics for each of the tools. DBCTI has the highest scores in all the three categories. On the simulated linear and binary tree datasets, all the tools have competitively similar scores; more than half of the tools have a score of 1, indicating a perfect inference. DBCTI maintains this excellent performance as the trajectory goes more complicated while the performance of the other tools drops. Especially on the complicated bifurcating topology, DBCTI outperforms other tools to a great extent; in fact, DBCTI is the only tool able to infer correctly these bifurcation trajectories. On the simulated cyclical datasets, apart from the topological evaluation, DBCTI also outperforms other tools in terms of purity score, with an average above 0.6 and 0.8 on the simulated and experimental datasets.

Purity scores of the cell clusters after PHATE's and DCA's dimensionality reduction

Similarly as tSNE, DCA and PHATE can be also used to carry out dimensionality reduction. We calculated the purity score for the clusters from each of the five experimental reference datasets after each of the dimensionality reduction methods is applied. For a fair comparison, the same K-means algorithm was applied to all of the dimensionality-reduced datasets to determine the cluster index, including the dimensionality-reduced datasets after our DBCTI's tSNE step. The K parameter was automatically set by the NbClust function in the NbClust package. The result shows that tSNE provides better performance on all of the five datasets, with an average purity score 0.80, much higher than PHATE's 0.70 and DCA's 0.56 (Supplementary Figure S11(P)).

Parameter tuning results for the identification of vague cells

We tested the robustness of crucial parameters involved in DBCTI's inference process for the identification of vague cells, namely parameters r, C_sum and C_diff. The investigation is to understand how much the parameters affect the performance when they take different settings. We note that the default value we set for C_sum and C_diff are 0.8 and 0.5 respectively. The investigation was carried out on all of the five experimental reference datasets (Figure 3). The parameter C_sum was set as a value within the range from 0.65 to 0.95 with interval step 0.05, and C_diff was set within the range from 0.35 to 0.65 with interval step 0.05. We used all of these two-parameter combinations (namely, $7 \times 7 = 49$ pairs) for the experiment. Under each of these settings and for each dataset, we computed a distance matrix of the clusters, which is a matrix deciding if any two cell populations are linkable or not. We found that any combination of a C_sum value ranging from 0.7 to 0.9 with a C_diff value from 0.4 to 0.6 did not make any variation on the final topology construction on all of the five datasets (Supplementary Figure S12(A-E)), showing strong robustness of the default settings to the DBCTI's inference performance.

Besides, we also found that different values set for parameter r made little impact on the inference result (Supplementary Figure S12(F)). A r value ranging from 1 to 3 with 0.5 interval step generates at most 0.008 distribution assignment changes of a cell on average in comparison with DBCTI's default r setting. We also verified that such difference did not affect the final trajectory result.

Case studies: An IgM plasmablast sub-population newly identified by DBCTI in a disconnected trajectory from SARS-CoV-2-infected peripheral blood mononuclear cells

In a recent study on peripheral immune response to severe COVID-19 (55), Wilk and colleagues identified a novel cell population (annotated as 'developing neutrophils') from severe COVID-19 patients' peripheral blood mononuclear cells, and found a linear differentiation trajectory bridging

A

Data / Tool	linear_s Linear	cyclical_s Cyclical	Binary_tree_s Tree	Disconnected_s Disconnected*	Bifurcating_cycle_s Cyclical
Monocle 2	**1.00**	0.00	**1.00**	0.00	0.00
TSCAN	**1.00**	0.00	0.00	0.00	0.00
PAGA	0.67	0.00	0.67	0.67	0.00
Slingshot	**1.00**	0.00	**1.00**	0.00	0.00
VIA	0.67	0.00	**1.00**	**1.00**	0.00
Mutrans	**1.00**	0.00	0.00	0.00	0.00
DBCTI	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Data / Tool	Yan_data Linear	Nestorowa_data Tree	Kowalczyk_data Linear	Leng_data Linear/Cyclical	Camp_data Disconnected
Monocle 2	0.00	**1.00**	0.00	0.00	0.00
TSCAN	**1.00**	0.00	**1.00**	**1.00**	0.00
PAGA	**1.00**	**1.00**	0.00	0.00	0.00
Slingshot	0.00	0.00	**1.00**	0.00	0.00
VIA	**1.00**	**1.00**	0.00	0.00	**1.00**
Mutrans	0.00	**1.00**	**1.00**	**1.00**	0.00
DBCTI	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Data / Tool	linear_s Linear	cyclical_s Cyclical	Binary_tree_s Tree	Disconnected_s Disconnected	Bifurcating_cycle_s Cyclical
Monocle 2	**1.00**	0.71	0.87	0.00	0.84
TSCAN	**1.00**	0.93	0.74	0.00	0.84
PAGA	0.64	0.93	0.89	0.67	0.88
Slingshot	**1.00**	0.80	0.87	0.83	0.84
VIA	0.36	0.76	0.91	**1.00**	0.84
Mutrans	**1.00**	0.87	0.74	0.00	0.81
DBCTI	**1.00**	**1.00**	**0.96**	**1.00**	**0.93**

Data / Tool	Yan_data Linear	Nestorowa_data Tree	Kowalczyk_data Linear	Leng_data Linear/Cyclical	Camp_data Disconnected
Monocle 2	0.75	0.87	0.65	0.71	0.00
TSCAN	**1.00**	0.87	**1.00**	**1.00**	0.00
PAGA	**1.00**	**1.00**	0.94	0.86	0.00
Slingshot	0.50	0.87	**1.00**	0.86	0.00
VIA	**1.00**	0.40	0.41	0.57	0.50
Mutrans	0.75	**1.00**	**1.00**	**1.00**	0.00
DBCTI	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

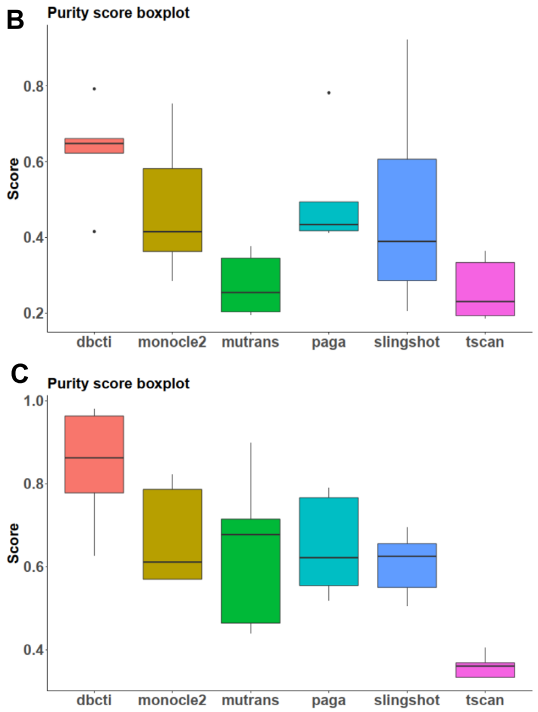


Figure 4. Quantified performance comparison between the inferred trajectories. (**A**) Pattern consistency scores (first two tables for the simulated and the experimental datasets) and pattern similarity scores (bottom two tables for the simulated datasets and the experimental datasets). All the scores for the simulated datasets were averaged with regard to the same topological pattern. (**B**) Purity scores assessed for the tools on the simulated datasets. (**C**) Purity scores on the experimental datasets.

the transition from immunoglobulin M plasmablast (IgM PB) cells to these developing neutrophils via an RNA velocity and cellular phenotypic analysis.

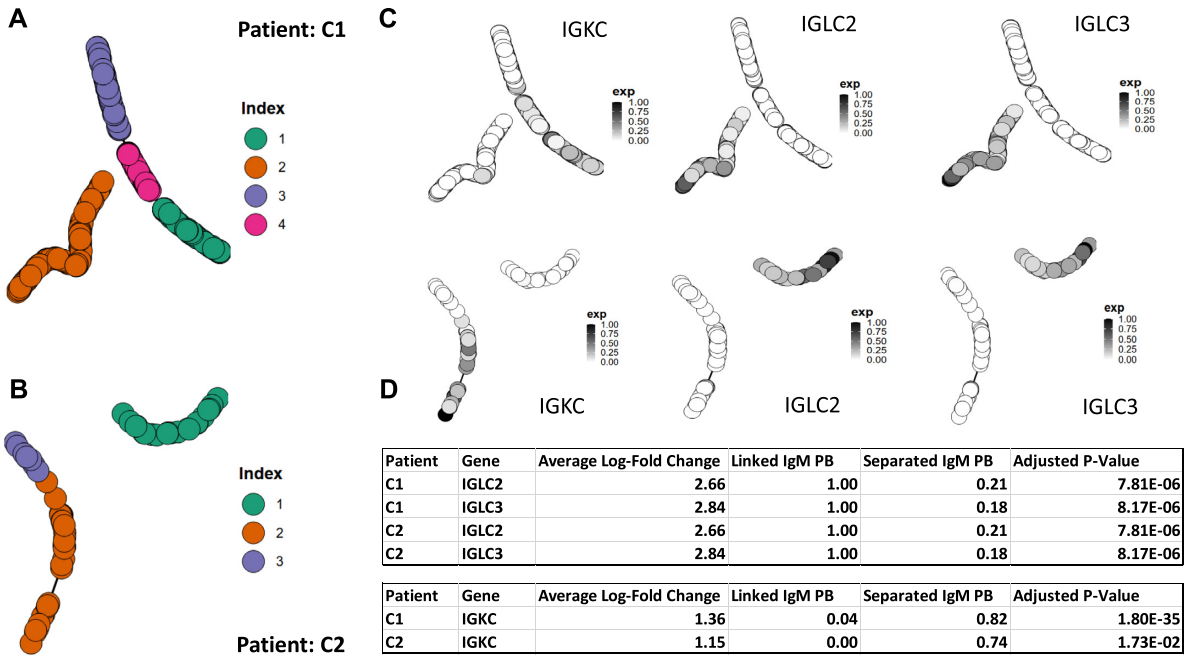
DBCTI was applied to the single-cell RNA-sequencing datasets of two patients C1 and C2 from (55) to see whether our trajectory inference could derive the same shape of the differentiation path (from the IgM PB cells to the neutrophils). Note that we applied DBCTI to the two patient samples separatively in order to minimize batch effect.

DBCTI-inferred cell trajectory from C1's dataset is a disconnected trajectory consisting of two components (Figure 5A). The bigger component is a linear differentiation path covering two clusters of IgM PB cells ordered temporally prior to a cluster of neutrophils according to their pseudo-time. This is a cell differentiation lineage exactly consistent with Wilk's discovery about the linear cell-transition from IgM PB cells to neutrophils. A similar linear differentiation path was also constructed by DBCTI from the data of patient C2 (Figure 5B). These results firmly suggest that our findings by DBCTI can re-discover significant cell trajectories reported in high-quality literature (55).

Of more interests and novelty of our findings is that C1's IgM PB cell population can be divided by DBCTI into two sub-populations: one is formed in the bigger component of the trajectory supporting the linear differentiation from IgM PBs to neutrophils, the other is all clustered in the smaller component of the trajectory (Supplementary Figure S13A, B). This suggests that not all of the IgM PB cells were developing into neutrophils, a phenomenon not revealed by Wilk's work (55).

In fact, three antigen-binding genes IGLC2, IGLC3 and IGKC were highly enriched in these two IgM PB sub-populations as shown in our differentially expressed (DE) gene analysis for the two sub-populations. Genes IGLC2 and IGLC3 were expressed significantly higher in the separated sub-population, while gene IGKC was enriched in the linear-trajectory sub-population (Figure 5C). The adjusted p-value of these genes are all far below 0.05 (Figure 5D). Again, similar expression patterns of these three genes were observed in patient C2's data. We thus hypothesize that $IGKC^{+}/IGLC2^{-}/IGLC3^{-}$ IgM PB cells have potential to be differentiated into neutrophils, while $IGKC^{-}/IGLC2^{+}/IGLC3^{+}$ IgM PB cells have no such differentiation trend.

For comparison, we constructed cell trajectories from these datasets using PAGA and Slingshot, which are two methods having better performance than the other existing tools based on the evaluation results presented in Figure 4. PAGA and Slingshot constructed cell trajectories quite different from those by DBCTI (see Figures 5 and 6). PAGA inferred loop sub-structures in the trajectories. However, these loops are unlikely to exist in biology, as it otherwise indicates that IgM PB cells could be differentiated into neutrophils and could also be differentiated back from neutrophils (61,62). Slingshot did not figure out the separate lineages due to the structure limitation of the model it uses. Overall, DBCTI has obtained more reasonable results that can be supported by the findings from the literature Wilk-paper as well as can be rigorously supported by differentially expressed gene analysis.



Patient	Gene	Average Log-Fold Change	Linked IgM PB	Separated IgM PB	Adjusted P-Value
C1	IGLC2	2.66	1.00	0.21	7.81E-06
C1	IGLC3	2.84	1.00	0.18	8.17E-06
C2	IGLC2	2.66	1.00	0.21	7.81E-06
C2	IGLC3	2.84	1.00	0.18	8.17E-06

Patient	Gene	Average Log-Fold Change	Linked IgM PB	Separated IgM PB	Adjusted P-Value
C1	IGKC	1.36	0.04	0.82	1.80E-35
C2	IGKC	1.15	0.00	0.74	1.73E-02

Figure 5. SARS-CoV-2-infected IgM PB and neutrophil cell population differentiation trajectory inferred by DBCTI from the Wilk-dataset. (**A**) Cell trajectory derived from patient C1. (**B**) Cell trajectory derived from patient C2. (**C**) Gene expression trends of selected genes (the first row is for patient C1 and the second row is for C2). (**D**) Differentially expressed gene analysis between the two IgM PB sub-populations in patients C1 and C2.

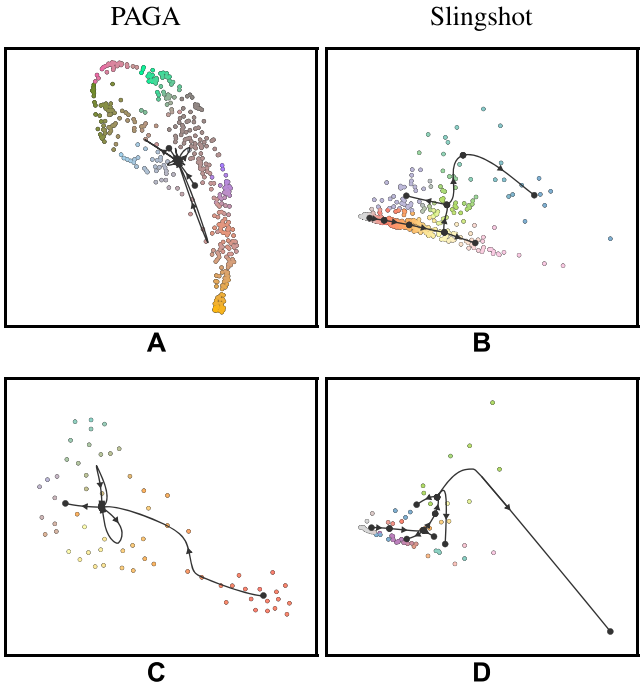


Figure 6. Inference result on the Wilk-dataset by PAGA and Slingshot. (**A**) Patient C1 sample by PAGA. (**B**) Patient C1 sample by Slingshot. (**C**) Patient C2 sample by PAGA. (**D**) Patient C2 sample by Slingshot.

DISCUSSION

We have introduced a cell trajectory inference method DBCTI which is a pipeline of steps to work on single-cell RNA-seq count matrices for generating cell trajectories with pseudo-time assigned at each cell. The basic idea of DBCTI is the assumption that a cell having a close distance with another in the embedded hyperspace should have high similarity in their expression pattern. Based on this idea, we have used density-based clustering algorithms to identify cell clusters and studied their overlapping distributions to identify vague cells and transition states. The detection of these vague cells is critically important to determine the connectivity between the clustered cell sub-populations. Different from other tools, DBCTI utilizes a 2-stage approach to build the trajectory: base topology construction as the first stage and base-guided iteratively fitting as the second. Especially, in the first stage for base topology construction with the density information from all the transition states, DBCTI can decide whether any two clusters (lineages) should be separated, or should be branched, or should be bridged to build a loop. Such a step allows plenty of flexibility for the method to construct diverse shapes of cell trajectories fitting the data. In contrast, the rigid structure of minimum spanning tree as adopted by TSCAN, Slingshot and Waterfall, restricts the shape of trajectories not to having any disparate edges; it does not allow any cycle in the shape either (3,15,18).

Cluster assignment as a crucial step, has a direct influence on the accuracy of the inferred cell trajectory. As single-cell RNA-seq data usually contains noises due to insufficient sequencing depth, DBCTI uses a model-based clustering technique to infer the initial cell cluster states. Instead of directly using the exact expression levels of cells, DBCTI fits cells into probability distributions. Besides, kernel density estimation (KDE), as a non-parametric approach, is used to jointly decide the final cluster, in case some data points do not fully satisfy the Gaussian distribution as-

sumption. Additionally, robust covariance inference technique MCD is used to fit the distribution of final determined cell states. This approach infers relatively reliable data first and, based on that, construct a covariance matrix to infer the correct shape of the distribution as well as increase the detection accuracy of transition points. All of such steps are attempted to minimize the impact of the imputations in the data. As a result, the purity scores on the cluster assignment by DBCTI is higher than other tools. In addition, the gene-selection pre-processing step is also important. A group of cells can be classified into different categories in taxonomy. As examples shown, a cell can be classified as stem cell, and a proliferating cell can be in the M cell cycle state. From this perspective, DBCTI allows to select genes based on the provided marker genes, in order to construct the corresponding trajectory, via either correlation or nonnegative matrix factorization (NMF) approach. Details of the two methods have been described in the Methods section.

DBCTI can correctly re-construct or build specific differentiation patterns ranging from simple topologies such as linear, cyclical and binary tree, to more complicated patterns like disparate lineage and bifurcating cyclical hierarchy. Its performance is better than six recently developed trajectory inference tools in terms of inferring specific pattern, reconstructing accurate topological trajectory as well as distinguishing different types of cell states.

Apart from the case study presented in Figure 5, we have actually conducted more case studies which are about glioblastoma cell lineages (Supplementary Figure S14) and SARS-CoV-2-infected lung macrophages' differentiation (Supplementary Figure S15) in Supplementary material. Detailed analysis and biological verification will be conducted in our future work.

DBCTI may lose some precision when handling more complicated trajectories. As demonstrated in Figure 4, a relatively complicated bifurcating pattern for instance, has a lower similarity score comparing to the simpler topologies. Additionally, DBCTI is unable to predict the direction of differentiation. Hence, the final estimated pseudo time is a relative but not an absolute value. As a future work, it is expected that novel and effective feature selection methods combined with dimension reduction techniques, as well as methods able to detect the differentiation ways of cells will improve DBCTI's performance, especially for the bifurcating patterns presented in Supplementary Figures S1E, S2E and S3E.

In summary, we developed a tool named DBCTI for cell trajectory inference tasks. We demonstrated that DBCTI is able to re-construct accurate and topologically flexible cell trajectories from both simulated and real single-cell RNA-sequencing datasets. Evaluation results demonstrate that DBCTI can outperform six state-of-the-art cell trajectory inference tools. Besides, we applied DBCTI to gain new hypothesis into lineages between IgM plasmablast cells and developing neutrophils in SARS-CoV-2-infected patients.

DATA AVAILABILITY

Single-cell mRNA-seq datasets can be downloaded from GEO with accession numbers as follows: Leng-dataset (GSE64016), Camp-dataset (GSE81252), Yan-dataset (GSE36552), Kowalczyk-dataset (GSE59114), Nestorowa-dataset (GSE81682), Darmanis-dataset (GSE84465), Wilk-dataset (GSE150728) and Melms-dataset (GSE171524).

A software prototype of DBCTI is available at https://github.com/tianlt/dbcti. The codes for performing all the analyses and generating figures in the paper are available at https://github.com/tianlt/dbcti-paper.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

Australian Research Council Discovery Project [DP180100 120]. Funding for open access charge: Australian Research Council Discovery Project.
Conflict of interest statement. None declared.

REFERENCES

1. Charrout,M., Reinders,M. and Mahfouz,A. (2020) Untangling biological factors influencing trajectory inference from single cell data. *NAR Genom. Bioinform.*, **2**, lqaa053.
2. Bergen,V., Lange,M., Peidli,S., Wolf,F. and Theis,F. (2020) Generalizing RNA velocity to transient cell states through dynamical modeling. *Nat. Biotechnol.*, **38**, 1408–1414.
3. Shin,J., Berg,D., Zhu,Y., Shin,J., Song,J., Bonaguidi,M., Enikolopov,G., Nauen,D., Christian,K., Ming,G. *et al.* (2015) Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell Stem Cell*, **17**, 360–372.
4. Berge,K., De Bezieux,H., Street,K., Saelens,W., Cannoodt,R., Saeys,Y., Dudoit,S. and Clement,L. (2020) Trajectory-based differential expression analysis for single-cell sequencing data. *Nat. Commun.*, **11**, 1201.
5. Pokhilko,A., Handel,A., Curion,F., Volpato,V., Whiteley,E., Bøstrand,S., Newey,S., Akerman,C., Webber,C., Clark,M. *et al.* (2021) Targeted single-cell RNA sequencing of transcription factors enhances the identification of cell types and trajectories. *Genome Res.*, **31**, 1069–1081.
6. De Micheli,A., Laurilliard,E., Heinke,C., Ravichandran,H., Fraczek,P., Soueid-Baumgarten,S., De Vlaminck,I., Elemento,O. and Cosgrove,B. (2020) Single-cell analysis of the muscle stem cell hierarchy identifies heterotypic communication signals involved in skeletal muscle regeneration. *Cell Rep.*, **30**, 3583–3595.
7. Weinreb,C., Rodriguez-Fraticelli,A., Camargo,F. and Klein,A. (2020) Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, **367**, eaaw3381.
8. Fan,J., Slowikowski,K. and Zhang,F. (2020) Single-cell transcriptomics in cancer: Computational challenges and opportunities. *Exp. Mol. Med.*, **52**, 1452–1465.
9. Bendall,S., Davis,K., Amir,E., Tadmor,M., Simonds,E., Chen,T., Shenfeld,D., Nolan,G. and Pe'er,D. (2014) Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell*, **157**, 714–725.
10. Gulati,G., Sikandar,S., Wesche,D., Manjunath,A., Bharadwaj,A., Berger,M., Ilagan,F., Kuo,A., Hsieh,R., Cai,S. *et al.* (2020) Single-cell transcriptional diversity is a hallmark of developmental potential. *Science*, **367**, 405–411.
11. Wolf,F., Hamey,F., Plass,M., Solana,J., Dahlin,J., Göttgens,B., Rajewsky,N., Simon,L. and Theis,F. (2019) PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.*, **20**, 59.
12. Trapnell,C., Cacchiarelli,D., Grimsby,J., Pokharel,P., Li,S., Morse,M., Lennon,N., Livak,K., Mikkelsen,T. and Rinn,J. (2014) The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.*, **32**, 381–386.
13. Qiu,X., Hill,A., Packer,J., Lin,D., Ma,Y. and Trapnell,C. (2017) Single-cell mRNA quantification and differential analysis with Census. *Nat. Methods*, **14**, 309–315.

14. Qiu,X., Mao,Q., Tang,Y., Wang,L., Chawla,R., Pliner,H. and Trapnell,C. (2017) Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods*, **14**, 979–982.
15. Street,K., Risso,D., Fletcher,R., Das,D., Ngai,J., Yosef,N., Purdom,E. and Dudoit,S. (2018) Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*, **19**, 477.
16. Cannoodt,R., Saelens,W., Sichien,D., Tavernier,S., Janssens,S., Guilliams,M., Lambrecht,B., De Preter,K. and Saeys,Y. (2016) SCORPIUS improves trajectory inference and identifies novel modules in dendritic cell development. bioRxiv doi: https://doi.org/10.1101/079509, 07 October 2016, preprint: not peer reviewed.
17. Welch,J., Hartemink,A. and Prins,J. (2016) SLICER: inferring branched, nonlinear cellular trajectories from single cell RNA-seq data. *Genome Biol.*, **17**, 106.
18. Ji,Z. and Ji,H. (2016) TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Res.*, **44**, e117.
19. Chen,H., Albergante,L., Hsu,J., Lareau,C., Bosco,G., Guan,J., Zhou,S., Gorban,A., Bauer,D., Aryee,M. *et al.* (2019) Single-cell trajectories reconstruction, exploration and mapping of omics data with STREAM. *Nat. Commun.*, **10**, 1903.
20. Tran,T. and Bader,G. (2020) Tempora: Cell trajectory inference using time-series single-cell RNA sequencing data. *PLoS Comput. Biol.*, **16**, e1008205.
21. Stassen,S., Yip,G., Wong,K., Ho,J. and Tsia,K. (2021) Generalized and scalable trajectory inference in single-cell omics data with VIA. *Nat. Commun.*, **12**, 5528.
22. Zhou,P., Wang,S., Li,T. and Nie,Q. (2021) Dissecting transition cells from single-cell transcriptome data through multiscale stochastic dynamics. *Nat. Commun.*, **12**, 5609.
23. Huo,L., Jiao Li,J., Chen,L., Yu,Z., Hutvagner,G. and Li,J. (2021) Single-cell multi-omics sequencing: application trends, COVID-19, data analysis issues and prospects. *Brief. Bioinform*, **22**, bbab229.
24. Wang,W., Tan,H., Sun,M., Han,Y., Chen,W., Qiu,S., Zheng,K., Wei,G. and Ni,T. (2021) Independent component analysis based gene co-expression network inference (ICAnet) to decipher functional modules for better single-cell clustering and batch integration. *Nucleic Acids Res.*, **49**, e54.
25. Grün,D., Muraro,M., Boisset,J., Wiebrands,K., Lyubimova,A., Dharmadhikari,G., Born,M., Van Es,J., Jansen,E., Clevers,H. *et al.* (2016) De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell*, **19**, 266–277.
26. Schiebinger,G., Shu,J., Tabaka,M., Cleary,B., Subramanian,V., Solomon,A., Gould,J., Liu,S., Lin,S., Berube,P. *et al.* (2019) Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, **176**, 928–943.
27. Lin,C. and Bar-Joseph,Z. (2019) Continuous-state HMMs for modeling time-series single-cell RNA-Seq data. *Bioinformatics*, **35**, 4707–4715.
28. Grønning,A., Oubounyt,M., Kanev,K., Lund,J., Kacprowski,T., Zehn,D., Røttger,R. and Baumbach,J. (2021) Enabling single-cell trajectory network enrichment. *Nat. Comp. Sci.*, **1**, 153–163.
29. Saelens,W., Cannoodt,R., Todorov,H. and Saeys,Y. (2019) A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.*, **37**, 547–554.
30. Kumar,S., Fonseca,V., Ribeiro,F., Basto,A., Doce,A., Monteiro,M., Elessa,D., Miragaia,R., Gomes,T., Piaggio,E. *et al.* (2021) Developmental bifurcation of human T follicular regulatory cells. *Sci. Immunol.*, **6**, eabd8411.
31. Balan,S., Arnold-Schrauf,C., Abbas,A., Couespel,N., Savoret,J., Imperatore,F., Villani,A., Manh,T., Bhardwaj,N. and Dalod,M. (2018) Large-scale human dendritic cell differentiation revealing notch-dependent lineage bifurcation and heterogeneity. *Cell Rep.*, **24**, 1902–1915.
32. Yao,Z., Mich,J., Ku,S., Menon,V., Krostag,A., Martinez,R., Furchtgott,L., Mulholland,H., Bort,S., Fuqua,M. *et al.* (2017) A single-cell roadmap of lineage bifurcation in human ESC models of embryonic brain development. *Cell Stem Cell*, **20**, 120–134.
33. Cao,J., Zhou,W., Steemers,F., Trapnell,C. and Shendure,J. (2020) Sci-fate characterizes the dynamics of gene expression in single cells. *Nat. Biotechnol.*, **38**, 980–988.
34. Rousseeuw,P. and Driessen,K. (1999) A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, **41**, 212–223.
35. Lan,T., Hutvagner,G., Lan,Q., Liu,T. and Li,J. (2020) Sequencing dropout-and-batch effect normalization for single-cell mRNA profiles: a survey and comparative analysis. *Brief. Bioinform.*, **22**, bbaa248.
36. Huang,M., Wang,J., Torre,E., Dueck,H., Shaffer,S., Bonasio,R., Murray,J., Raj,A., Li,M. and Zhang,N. (2018) SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods*, **15**, 539–542.
37. Lähnemann,D., Köster,J., Szczurek,E., McCarthy,D., Hicks,S., Robinson,M., Vallejos,C., Campbell,K., Beerenwinkel,N., Mahfouz,A. *et al.* (2020) Eleven grand challenges in single-cell data science. *Genome Biol.*, **21**, 31.
38. Breda,J., Zavolan,M. and Nimwegen,E. (2021) Bayesian inference of gene expression states from single-cell RNA-seq data. *Nat. Biotechnol.*, **39**, 1008–1016.
39. Jolliffe,I. and Cadima,J. (2016) Principal component analysis: a review and recent developments. *Phil. Trans. Roy. Soc. A: Math. Phys. Eng. Sci.*, **374**, 20150202.
40. Maaten,L. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.
41. Ripley,B., Venables,B., Bates,D., Hornik,K., Gebhardt,A., Firth,D. and Ripley,M. (2013) Package 'mass'. *Cran R*, **538**, 113–120.
42. Hao,Y., Hao,S., Andersen-Nissen,E., Mauck,W.III., Zheng,S., Butler,A., Lee,M., Wilk,A., Darby,C., Zager,M. *et al.* (2021) Integrated analysis of multimodal single-cell data. *Cell*, **184**, 3573–3587.
43. Michalopoulos,I., Pavlopoulos,G., Malatras,A., Karelas,A., Kostadima,M., Schneider,R. and Kossida,S. (2012) Human gene correlation analysis (HGCA): a tool for the identification of transcriptionally co-expressed genes. *BMC Res. Notes*, **5**, 265.
44. Seung,D. and Lee,L. (2001) Algorithms for non-negative matrix factorization. *Adv. Neur. Inf. Process. Syst.*, **13**, 556–562.
45. Sharma,G., Colantuoni,C., Goff,L., Fertig,E. and Stein-O'Brien,G. (2020) projectR: an R/Bioconductor package for transfer learning via PCA, NMF, correlation and clustering. *Bioinformatics*, **36**, 3592–3593.
46. Kobak,D. and Berens,P. (2019) The art of using t-SNE for single-cell transcriptomics. *Nat. Commun.*, **10**, 5416.
47. Scrucca,L., Fop,M., Murphy,T. and Raftery,A. (2016) mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *R J.*, **8**, 289.
48. Hastie,T. and Stuetzle,W. (1989) Principal curves. *J. Am. Stat. Assoc.*, **84**, 502–516.
49. Cannoodt,R., Saelens,W., Deconinck,L. and Saeys,Y. (2021) Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nat. Commun.*, **12**, 3942.
50. Leng,N., Chu,L., Barry,C., Li,Y., Choi,J., Li,X., Jiang,P., Stewart,R., Thomson,J. and Kendziorski,C. (2015) Oscope identifies oscillatory genes in unsynchronized single-cell RNA-seq experiments. *Nat. Methods*, **12**, 947–950.
51. Camp,J., Sekine,K., Gerber,T., Loeffler-Wirth,H., Binder,H., Gac,M., Kanton,S., Kageyama,J., Damm,G., Seehofer,D. *et al.* (2017) Multilineage communication regulates human liver bud development from pluripotency. *Nature*, **546**, 533–538.
52. Yan,L., Yang,M., Guo,H., Yang,L., Wu,J., Li,R., Liu,P., Lian,Y., Zheng,X., Yan,J. *et al.* (2013) Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. *Nat. Struct. Mol. Biol.*, **20**, 1131–1139.
53. Kowalczyk,M., Tirosh,I., Heckl,D., Rao,T., Dixit,A., Haas,B., Schneider,R., Wagers,A., Ebert,B. and Regev,A. (2015) Single-cell RNA-seq reveals changes in cell cycle and differentiation programs upon aging of hematopoietic stem cells. *Genome Res.*, **25**, 1860–1872.
54. Nestorowa,S., Hamey,F., Pijuan Sala,B., Diamanti,E., Shepherd,M., Laurenti,E., Wilson,N., Kent,D. and Göttgens,B. (2016) A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood J. Am. Soc. Hematol.*, **128**, e20–e31.
55. Wilk,A., Rustagi,A., Zhao,N., Roque,J., Martínez-Colón,G., McKechnie,J., Ivison,G., Ranganath,T., Vergara,R., Hollis,T. *et al.* (2020) A single-cell atlas of the peripheral immune response in patients with severe COVID-19. *Nat. Med.*, **26**, 1070–1076.

56. Melms,J., Biermann,J., Huang,H., Wang,Y., Nair,A., Tagore,S., Katsyv,I., Rendeiro,A., Amin,A., Schapiro,D. *et al.* (2021) A molecular single-cell lung atlas of lethal COVID-19. *Nature*, **595**, 114–119.
57. Krishna,K. and Murty,M. (1999) Genetic K-means algorithm. *IEEE Trans. Syst. Man Cyber. Part B (Cybernetics)*, **29**, 433–439.
58. Malika,C., Ghazzali,N., Boiteau,V. and Niknafs,A. (2014) NbClust: an R package for determining the relevant number of clusters in a data Set. *J. Stat. Softw.*, **61**, 1–36.
59. Riether,C., Schürch,C. and Ochsenbein,A. (2015) Regulation of hematopoietic and leukemic stem cells by the immune system. *Cell Death Differ.*, **22**, 187–198.
60. Miyajima,A., Tanaka,M. and Itoh,T. (2014) Stem/progenitor cells in liver development, homeostasis, regeneration, and reprogramming. *Cell Stem Cell*, **14**, 561–574.
61. Ng,L., Ostuni,R. and Hidalgo,A. (2019) Heterogeneity of neutrophils. *Nat. Rev. Immunol.*, **19**, 255–265.
62. Chu,V. and Berek,C. (2013) The establishment of the plasma cell survival niche in the bone marrow. *Immunol. Rev.*, **251**, 177–188.