Cell-connectivity-guided trajectory inference from single-cell data

Smolander, Johannes; Junttila, Sini; Elo, Laura L.

Published in:
Bioinformatics

DOI:
10.1093/bioinformatics/btad515

Published: 01/09/2023

Document Version
Final published version

Document License
CC BY

Link to publication

Please cite the original version:
Smolander, J., Junttila, S., & Elo, L. L. (2023). Cell-connectivity-guided trajectory inference from single-cell data. *Bioinformatics*, *39*(9), Article btad515. https://doi.org/10.1093/bioinformatics/btad515

Bioinformatics, 2023, **39(9)**, btad515
https://doi.org/10.1093/bioinformatics/btad515
Advance Access Publication Date: 25 August 2023
Original Paper

OXFORD

Gene expression

Cell-connectivity-guided trajectory inference from single-cell data

Johannes Smolander [1], **Sini Junttila** [1], **Laura L. Elo**[1,2,*]

[1]Turku Bioscience Centre, University of Turku and Åbo Akademi University, 20520 Turku, Finland
[2]Institute of Biomedicine, University of Turku, 20520 Turku, Finland

*Corresponding author. Turku Bioscience Centre, University of Turku and Åbo Akademi University, Tykistökatu 6, 20520 Turku, Finland. E-mail: laura.elo@utu.fi

Associate Editor: Christina Kendziorski

Abstract

Motivation: Single-cell RNA-sequencing enables cell-level investigation of cell differentiation, which can be modelled using trajectory inference methods. While tremendous effort has been put into designing these methods, inferring accurate trajectories automatically remains difficult. Therefore, the standard approach involves testing different trajectory inference methods and picking the trajectory giving the most biologically sensible model. As the default parameters are often suboptimal, their tuning requires methodological expertise.

Results: We introduce Totem, an open-source, easy-to-use R package designed to facilitate inference of tree-shaped trajectories from single-cell data. Totem generates a large number of clustering results, estimates their topologies as minimum spanning trees, and uses them to measure the connectivity of the cells. Besides automatic selection of an appropriate trajectory, cell connectivity enables to visually pinpoint branching points and milestones relevant to the trajectory. Furthermore, testing different trajectories with Totem is fast, easy, and does not require in-depth methodological knowledge.

Availability and implementation: Totem is available as an R package at https://github.com/elolab/Totem.

1 Introduction

Single-cell RNA-sequencing (scRNA-seq) enables researchers to quantify the transcriptome of thousands or even millions of cells simultaneously at the single-cell level. The cells in a tissue undergo transcriptomic changes that are part of biological processes. If the changes happen gradually, the tissue contains cells derived from different stages of the process. If a correct trajectory that accurately models these different stages can be generated from the scRNA-seq data, this enables researchers to unravel the transcriptomic mechanisms of the processes. The modelling of trajectories from scRNA-seq data has formed a new branch in computational biology, known as trajectory inference.

A vast number of trajectory inference methods has been developed (Saelens *et al.* 2019, Deconinck *et al.* 2021). While these methods have proven to be valuable in many situations, their use is still difficult for several reasons. The main reason is that a single method used with default settings often fails to generate a trajectory that can capture all relevant parts (milestones) of the trajectory and also accurately model the correct milestone network. Therefore, a popular approach is to test different methods and tune their parameters until a biologically sensible or otherwise satisfactory trajectory is obtained.

Slingshot (Street *et al.* 2018) has become one of the most popular trajectory inference methods after a comparison study (Saelens *et al.* 2019) showed it was one of the best-performing methods for inferring tree-shaped trajectories. A tree can be any trajectory in which the parts are not disconnected and the trajectory has no cycles. Slingshot requires as input a clustering of cells that is used to construct a Minimum Spanning Tree (MST), which is subsequently smoothed using the simultaneous principal curves algorithm to obtain a directed trajectory, which also includes pseudotime, a measure of the differentiation stage of the cells. While methods such as the Average Silhouette Width (ASW) (Rousseeuw 1987) and the Variance Ratio Criterion (VRC) (Caliński and Harabasz 1974) can be used to select the clustering automatically, their weakness is in their tendency to select a clustering with an overly small number of clusters. The resulting trajectories are often over-simplistic and lack especially small milestones. Therefore, a more reliable approach can be to manually generate a clustering that includes all the relevant milestones, but this requires careful parameter testing and validation using markers to ensure that the clustering is optimal. However, even then the resulting MST may not correlate well with the true milestone network because the MST is generated using a distance matrix that is sensitive to the preprocessing steps (e.g. dimensionality reduction), the clustering structure, and the choice of the distance metric.

To provide a system with better user experience for researchers who perform trajectory inference, we developed a new tool, Totem, for the inference of tree-shaped trajectories from single-cell data (Fig. 1). Totem generates a large number of dissimilar clustering results for the cells (by default, 10 000) using a *k*-medoids algorithm. The number of clusters (*k*) and the structure of the clustering results vary, generating vastly different trajectories when used as the basis for constructing the MST.

Received: 10 November 2022; **Revised:** 24 July 2023; **Editorial Decision:** 13 August 2023; **Accepted:** 23 August 2023

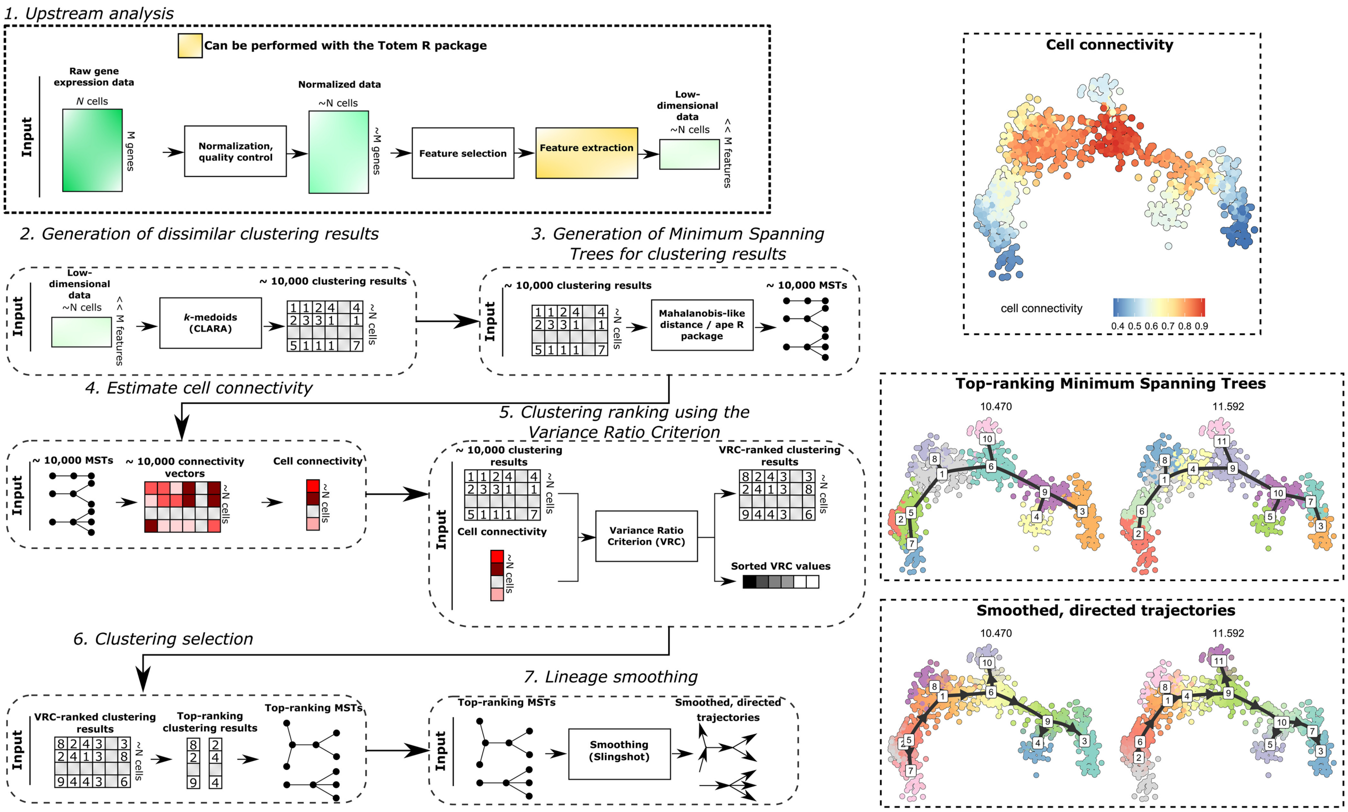
1. Upstream analysis
Can be performed with the Totem R package
Input
Raw gene expression data
N cells
M genes
Normalization, quality control
Normalized data
~N cells
~M genes
Feature selection
Feature extraction
Low-dimensional data
~N cells
<< M features
2. Generation of dissimilar clustering results
Input
Low-dimensional data
~N cells
<< M features
k-medoids (CLARA)
~ 10,000 clustering results
~N cells
3. Generation of Minimum Spanning Trees for clustering results
~ 10,000 clustering results
Mahalanobis-like distance / ape R package
~ 10,000 MSTs
4. Estimate cell connectivity
~ 10,000 MSTs
~ 10,000 connectivity vectors
Cell connectivity
5. Clustering ranking using the Variance Ratio Criterion
~ 10,000 clustering results
Cell connectivity
Variance Ratio Criterion (VRC)
VRC-ranked clustering results
Sorted VRC values
6. Clustering selection
VRC-ranked clustering results
Top-ranking clustering results
Top-ranking MSTs
7. Lineage smoothing
Top-ranking MSTs
Smoothing (Slingshot)
Smoothed, directed trajectories
Cell connectivity
cell connectivity
0.4 0.5 0.6 0.7 0.8 0.9
Top-ranking Minimum Spanning Trees
10.470
11.592
Smoothed, directed trajectories
10.470
11.592

Figure 1. Schematic illustration of the Totem workflow. (1) A gene expression matrix is used as input, which needs to be preprocessed upstream of Totem analysis, including normalization, quality control, and possible batch correction and feature selection. Further dimensionality reduction (feature extraction) can also be performed using Totem. For the low-dimensional data matrix, Totem generates (2) a large set of clustering results using a *k*-medoids clustering algorithm (CLARA), (3) a Minimum Spanning Tree (MST) for each clustering, which models the cluster (milestone) network, and (4) estimates the connectivity of the clusters in each MST (ratio of the number of edges a cluster has in MST and the number of clusters). The connectivity vectors are averaged to generate the cell connectivity measure (top right), which helps to locate branching points and milestones that are relevant to model. (5) The clustering results are ranked based on the Variance Ratio Criterion (VRC) of the cell connectivity measure, and (6) top-ranking clustering results and their corresponding MSTs are selected for further analysis (middle-right). (7) The selected MSTs are smoothed using the Slingshot algorithm to obtain directed trajectories (lower right). The trajectory can be exported as objects that can be used downstream of Totem analysis, such as differential expression analysis.

To select a clustering from the large set of clustering results, Totem uses a new measure called cell connectivity, which is based on counting the number of edges (connections) between the clusters in an MST. The ratio of the number of edges and clusters (connectivity) is calculated for each cell and clustering, and the connectivity vectors are averaged to obtain the cell connectivity of the cells. The cell connectivity acts as a useful baseline for selecting an appropriate trajectory and deciding which milestones to include in the trajectory, while also helping to give a visual overview of the milestone network, i.e. how the milestones are connected and where the branching points and leaf nodes are located. Importantly, a key feature of Totem for its usability is that it allows to quickly and easily browse different MSTs, from which the user can select one or several MSTs for further analysis.

To benchmark Totem, we performed a comprehensive comparison using the dynverse benchmarking framework (Saelens *et al.* 2019), which comprises over 200 tree-shaped trajectories. We compared Totem with the popular Slingshot tool (Street *et al.* 2018, Saelens *et al.* 2019) and the more recently introduced TinGa (Todorov *et al.* 2020), which is based on a growing neural gas (GNG) model (Fritzke 1994) and can also model trajectories that are more complex than trees. In particular, to evaluate the performance of the connectivity-based criterion of Totem for clustering selection in trajectory inference, we compared it with the popular ASW and VRC clustering selection methods. Because a single clustering is rarely optimal, and thus the users are often forced to try multiple clustering results to optimize the trajectories, we investigated the performance of the trajectories generated based on multiple top-ranked clustering results, giving a deeper overview of the performance. Finally, we demonstrate the usefulness of the cell connectivity measure with a few examples that show how the measure can aid trajectory inference by helping to pinpoint relevant branching points and milestones.

2 Materials and methods

2.1 Totem

In the following subsections, we go through the main steps of Totem trajectory inference workflow. Figure 1 illustrates the basic workflow of Totem.

2.1.1 Upstream analysis

Upstream of trajectory inference, Totem assumes that the input gene expression matrix has been normalized and quality control has been performed to remove bad-quality cells (Ilicic *et al.* 2016). scRNA-seq data analysis toolkits like Seurat

(Butler *et al.* 2018) and Scanpy (Wolf *et al.* 2018) can perform the normalization and quality control.

Trajectory inference methods require a low-dimensional (from 2 to 50) embedding as input. The dimensionality reduction steps can be customized as the user sees best prior to Totem analysis, but Totem can also be used to transform gene counts into new features (feature extraction) with functions that utilize the dyndimred R package. In scRNA-seq data analysis, the most common way to perform feature extraction is the Principal Component Analysis (PCA) in which the number of principal components typically varies from 10 to 30. Alternatively, methods like Multi-Dimensional Scaling (MDS) or the more scalable landscape MDS (LMDS) can be used to generate an embedding with a smaller number of dimensions (e.g. from 2 to 5), which is usually not a sufficient number in PCA of scRNA-seq data. *t*-distributed stochastic neighbour embedding (*t*-SNE) (Maaten and van der Hinton 2008) and Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) (McInnes *et al.* 2018) are commonly used only for visualization, and they should be used with caution if used as input in trajectory inference. As the default dimensionality reduction method in Totem, we use the 5-dimensional LMDS. Moreover, it is generally a good idea to reduce the number of features prior to feature extraction by selecting highly variable genes (HVGs), which can be performed using methods like Seurat (Butler *et al.* 2018) and Scanpy (Wolf *et al.* 2018). If the dataset has batch effects, and the user does not want to model these differences in trajectory inference, data integration methods (Butler *et al.* 2018, Luecken *et al.* 2022) can be used.

2.1.2 Generation of a large set of clustering results

To generate a large set of dissimilar clustering results that can be used as the basis for constructing the MST, we use the CLARA *k*-medoids clustering algorithm (Peter and Rousseeuw 1990) from the cluster R package, which performs fast *k*-medoids clustering. We run the algorithm L times (by default $L = 10\,000$) and filter out clustering results that have clusters with fewer than five cells to prevent overly small clusters, which are usually not interesting. The number of remaining clustering results (L') can be close to L or deviate from it considerably depending on the dataset. The number of clusters (k) varies by default from 3 to 20, but can be adjusted by the user if more complex trajectories with more milestones are expected. The default upper limit (20) is well above the average number of clusters (milestones) in the trajectories of the dynverse benchmarking framework. We also activate the R random number generation (RNG) in the clara R function so that the program returns different results with different RNG seeds.

2.1.3 Generation of minimum spanning trees for clustering results

For each clustering, we generate an MST by representing the clusters of cells as vertices in a graph and using the covariance-based approach of Slingshot (Street *et al.* 2018) to measure distances between the clusters. The Mahalanobis-like distances are calculated from the covariance matrix for cells in each cluster, while also considering the shape and spread of the clusters. We use the ape R package to perform the MST estimation (Paradis and Schliep 2019).

2.1.4 Estimate cell connectivity

For each MSTj ($j = 1, \ldots, L'$), we measure the connectivity $c_{ij} = d_{ij}/k_j$ of the ith vertex (cluster) in the graph by dividing the degree of the vertex d_{ij}, i.e. the number of edges that are connected to the vertex, by the number of vertices k_j in the graph. We then create a connectivity vector $\boldsymbol{c}_j$ that contains the connectivity values for all N cells corresponding to MST j, scale the connectivity values so that the maximum connectivity of each graph is always one, and calculate the cell connectivity of all the cells ($\bar{\boldsymbol{c}}$) by averaging over the L' connectivity vectors using the arithmetic mean:

$$\bar{\boldsymbol{c}} = \frac{1}{L'} \sum_{j=1}^{L'} \frac{\boldsymbol{c}_j}{\max(\boldsymbol{c}_j)} \tag{1}$$

The cell connectivity is higher for cell populations that are farther from the leaf parts of the trajectory (Fig. 1). We used the arithmetic mean because it ensures that the connectivity gradient is continuous, as opposed to the median approach (Supplementary Fig. S1).

2.1.5 Clustering selection using the variance ratio criterion

To find a clustering that captures the tree structure of the data and gives clusters that are well defined, we use VRC, also known as the Calinski–Harabasz score (Caliński and Harabasz 1974), which measures the ratio of the between-cluster dispersion and the within-cluster dispersion. Instead of using the whole dimensionally reduced data as input, we use the one-dimensional cell connectivity vector ($\bar{\boldsymbol{c}}$) to find clusters that have cells with similar connectivity within the clusters but different compared to the other clusters. VRC is a more memory-efficient method for clustering selection than the commonly used ASW because it does not require a distance matrix of cells as input, making it more suitable for datasets with many cells. We use the fpc R package to perform the VRC analysis.

2.1.6 Lineage smoothing

For an MST corresponding to a selected clustering, Totem performs lineage smoothing using Slingshot (Street *et al.* 2018), which generates a directed trajectory along with pseudotime that quantifies cell differentiation continuously at the single-cell level. Slingshot fits principal curves (Hastie and Stuetzle 1989) for each lineage of the MST starting from a user-specified root node, which is initially selected randomly. The root node can be adjusted later by the user.

2.1.7 Trajectory visualization and interpretation

Totem includes visualization functions that aid trajectory inference, utilizing the dynplot R package (Saelens *et al.* 2019). The functions enable to visualize multiple MSTs and smoothed trajectories side by side over a two-dimensional embedding. The two-dimensional embedding can be provided by the user or generated using one of the methods included in the dyndimred R package, such as *t*-SNE, UMAP, or MDS. To assist in biological interpretation, the user can also visualize expression levels of genes over the embedding.

The cell connectivity can be visualized over the two-dimensional embedding to pinpoint milestones and branching points that could be otherwise missed or inaccurately modelled (Fig. 1). Local changes in the connectivity indicate milestone transitions that are relevant to include in the trajectory. The connectivity levels can be used to locate branching points, where a cell population is encircled by other cell populations whose connectivity is relatively lower (Fig. 4).

Although the clustering selection is performed based on the VRC of the cell connectivity, the resulting MST can still be suboptimal in terms of the connections that the milestone network comprises. In addition, small milestones can be missing or the trajectory can seem over-complicated. Therefore, the user should not rely upon a single, automatically generated trajectory. Instead, the trajectory should be validated using gene markers to ensure that the model is biologically sensible, and compared with the cell connectivity by visualization to ensure that the milestone network is in line with the connectivity profile. If the trajectory requires adjustments, Totem allows to easily test different clustering results until a trajectory is obtained that meets both requirements.

2.2 Benchmarking

To benchmark Totem, we repeated the comprehensive benchmarking of scRNA-seq trajectory inference methods using the dynverse framework (Saelens *et al.* 2019). We included trajectories that have a tree-like structure, which were labelled as linear, bifurcation, multifurcation, or tree in the original comparison. These comprise 69, 44, 16, and 87 linear, bifurcation, multifurcation, and tree trajectories, respectively. The datasets are of synthetic and real origin. The real datasets include 26 gold standard datasets of which ground-truth includes cell types and their differentiation order at the cluster level (discrete pseudotime). The 54 real, silver standard datasets have a ground-truth that includes a continuous or discrete pseudotime. The synthetic, simulated datasets provide the most accurate ground-truth, and they were simulated using four different simulators: dyntoy (Saelens *et al.* 2019), dyngen (Cannoodt *et al.* 2021), PROSSTT (Papadopoulos *et al.* 2019), and Splatter (Zappia *et al.* 2017).

The dynverse benchmarking framework includes four main metrics. The accuracy of the cell differentiation order is measured using the correlation of pairwise geodesic distances between the ground-truth and predicted trajectories. The accuracy of the differentially expressed genes is measured using the weighted correlation of the random forest regression-derived feature importance lists. The features that are ranked higher in the ground-truth are given a relatively higher weight. The F1 branches value maps the cells to the closest branching points in the ground-truth and inferred trajectories and measures the clustering similarity between the two clustering results. Hamming-Ipsen-Mikhailov (HIM) (Jurman *et al.* 2015) measures the similarity of two topologies. For example, two linear trajectories would yield a perfect HIM value of 1, even if the two trajectories would be otherwise completely dissimilar. All four metrics range from 0 to 1 and the geometric mean of the metrics is used to assess the overall performance by penalizing small values in the metrics.

To benchmark Totem, we compared it with the dynverse implementation of the Slingshot method, which uses PAM (*k*-medoids) clustering algorithm and ASW to select the optimal number of clusters, as well as TinGa, which is based on a GNG model (Fritzke 1994). However, the preprocessing was standardized so that each method uses the 5-dimensional LMDS, which is the default feature extraction method of TinGa. All three methods also allow to choose how to perform the normalization, feature selection, and feature extraction steps.

The vanilla version of Slingshot available at Bioconductor takes as input a low-dimensional embedding, such as principal components or UMAP dimensions, and a clustering. There are no requirements for using a certain type of clustering or embedding. Slingshot models the trajectory topology as an MST and fits one or multiple principal curves, depending on how many lineages there are, using one of the clusters as the starting point. The distances between the clusters are calculated using a Mahalanobis-like distance metric, which can also be changed to a mutual nearest neighbour (MNN) based metric or the Euclidean distance metric. If no starting cluster is specified by the user, Slingshot selects one so that the number of shared clusters between the lineages is maximized before the first branching point.

In total, the GNG algorithm of TinGa has eight parameters in addition to the input embedding for which the trajectory is inferred. Of the eight GNG-related parameters, the maximum number of nodes (by default, 30) can be decreased if the trajectory has too many milestones (cell types, clusters) in the network. The maximum number of iterations (by default, 10 000) should not be tuned as the high iteration count ensures that the GNG algorithm converges. The six remaining parameters control how sensitively the graph nodes are linked and changed during the iteration, and their adjustment should be considered if tuning the maximum number of nodes does not improve the result. The original article of TinGa mentions that especially adjustment of the two epsilon parameters (ε_b and ε_n) affects the performance.

2.3 Comparing clustering selection methods

Trajectory inference methods such as Slingshot require a clustering that is used as the basis for constructing the MST, which is subsequently smoothed using the simultaneous principal curves algorithm to obtain a directed trajectory and pseudotime. In addition to the cell connectivity criterion of Totem for clustering selection, we tested three other methods: ASW, which is used by the dynverse implementation of Slingshot; VRC; and the random criterion that ranks the clustering results into a random order.

To compare the clustering selection methods, we ran the *k*-medoids clustering algorithm (CLARA) 10 000 times for each dataset, selected the top 100 clustering results with each method, performed the trajectory inference using Slingshot for each clustering, and ran the dynverse performance evaluation for the trajectories. For each benchmark dataset, we varied the number of evaluated trajectories from 1 to 100 and calculated the maximum and mean of the overall performance score. Finally, we calculated the mean of the overall scores across the datasets.

2.4 Robustness of Totem

We investigated how robust the performance of Totem is with respect to its parameters: the number of clustering results (by default, 10 000) and the range of the *k* values (by default, from 2 to 20) from which the number of clusters is sampled with replacement. We randomly selected 10 datasets from the top 50 datasets with the highest overall score in the benchmark (names listed in Supplementary Fig. S2). For each of the 10 datasets, we ran Totem with different combinations of the number of clusterings (100, 500, 1000, 2500, 5000, 7500) and the upper limit of the *k* range (5, 10, 15, 20, 30, 40). For each parameter configuration and dataset, we ran Totem 10 times using different random seeds. We calculated the dynverse overall score between the inferred and ground-truth trajectories (Saelens *et al.* 2019).

3 Results

3.1 Benchmarking Totem for trajectory inference

To benchmark Totem, we used the dynverse benchmarking framework (Saelens *et al.* 2019) that includes 216 tree-shaped trajectories. In our comparison, we included the dynverse implementation of Slingshot (Street *et al.* 2018, Saelens *et al.* 2019), which uses ASW for clustering selection, and TinGa, which is a GNG-based method (Fritzke 1994) that can also predict more complex trajectories than tree-shaped, such as acyclic and disconnected trajectories (Todorov *et al.* 2020).

As shown in the rightmost panel of Fig. 2, Totem achieved superior overall scores (Friedman test; Bonferroni-adjusted P-value $\leq$ 0.001; Wilcoxon signed-rank test; Bonferroni-adjusted P-value $\leq$ 0.01) for datasets with a nonlinear trajectory (bifurcating, multifurcating, or other nonlinear tree), whereas TinGa was the second-best method for the nonlinear datasets. The violin plots also show that Totem's distribution of the overall scores with the nonlinear trajectories was more concentrated to higher performance levels compared to TinGa and Slingshot. For linear trajectories, Slingshot achieved superior performance among the tested methods (Friedman test; Bonferroni-adjusted P-value $\leq$ 0.001; Wilcoxon signed-rank test; Bonferroni-adjusted P-value $\leq$ 0.05), while the overall scores of the linear datasets were, on average, similar for TinGa and Totem. The difference in the overall performance was mainly attributable to the differences in the topology accuracy (HIM) and the accuracy of the cell assignment onto the branches (F1 branches). The HIM score of Slingshot was close to perfect for most of the linear datasets but worse compared to TinGa and Totem for the nonlinear datasets.

3.2 Benchmarking Totem for clustering selection

Selection of an appropriate clustering for constructing the MST is an important step in the trajectory inference process. Therefore, we took a closer look at the clustering selection step to investigate which clustering selection approach gives the best results when we consider multiple top-ranking trajectories simultaneously. We included two popular clustering selection methods, ASW and VRC, as well as the Totem method that uses cell connectivity and VRC, and also a method that ranks the clustering results into a random order (Random).

When we tested multiple top-ranking clustering results and always selected the trajectory that gave the best performance in the evaluation, the result suggested that the random criterion gave the best performance (Fig. 3A). This is an expected result considering that the random criterion generates more dissimilar trajectories than the other methods. However, when we averaged the performance values of the multiple top trajectories (Fig. 3B), the random criterion was the worst-performing method, as expected. ASW achieved the worst performance among the methods in the analysis in which we selected the best-performing trajectory, and it was the second-worst method when the performance values of the trajectories were averaged. VRC achieved slightly better performance than ASW when selecting the best-performing trajectory, and it was tied with the Totem method as the best method by the average performance. In addition to good average performance, Totem had a relatively high-performance curve in the analysis that considers the performance of the best-performing trajectories, similar with the random criterion. In other words, the selection method of Totem enables to find better performing trajectories than the ASW and VRC without sacrificing as much on the average performance as the random criterion.

3.3 Examples of trajectory inference with Totem

To demonstrate the utility of Totem in practice, we considered two datasets from the dynbenchmark database. The datasets meet the prerequisites of Totem: they have a tree-shaped topology with a single starting point and no disconnected parts. Importantly, the cell types of the two datasets are connected in the correct way in their low-dimensional

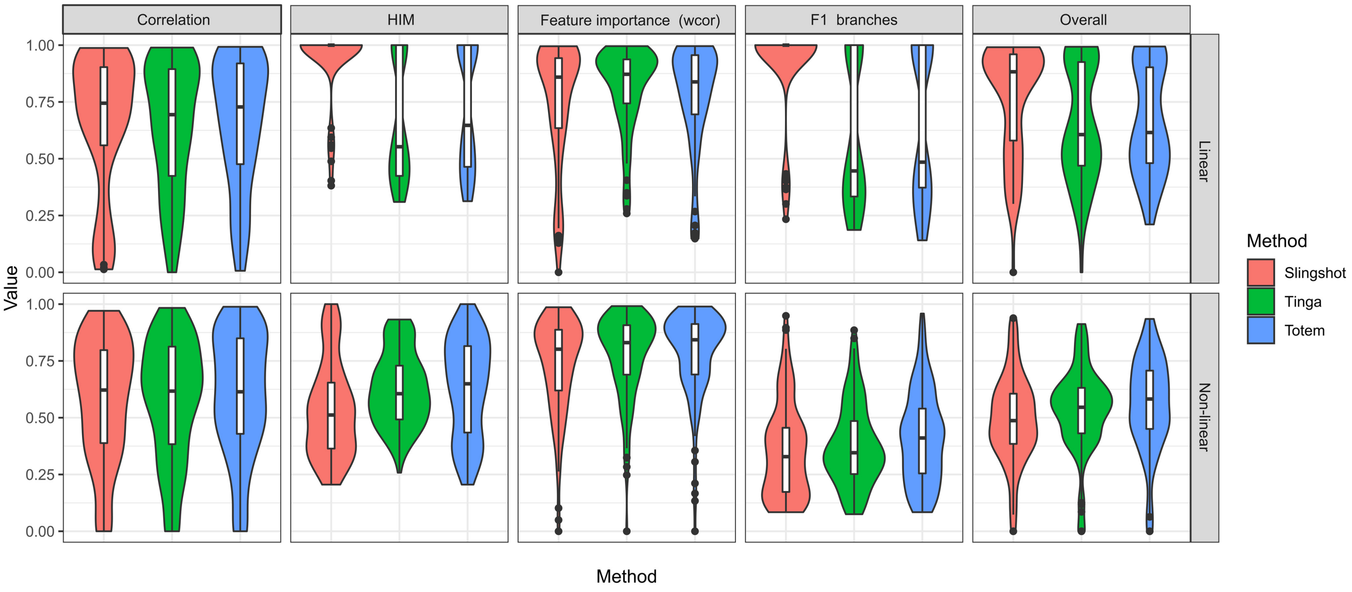

Figure 2. Results of the benchmarking for trajectory inference from scRNA-seq data. The overall score is the geometric mean of four performance metrics: the correlation of geodesic distances (correlation), which measures accuracy of cellular ordering, the Hamming-Ipsen-Mikhailov (HIM), which measures topological accuracy, the weighted correlation of feature importance lists, which measures accuracy of differentially expressed genes inferred from the trajectory, and the F1 branches, which measures accuracy of cell assignment onto branches. The results were grouped by the performance metric (columns) and whether the ground-truth topology is linear or nonlinear, i.e. bifurcating, multifurcating, or some other nonlinear tree (rows).

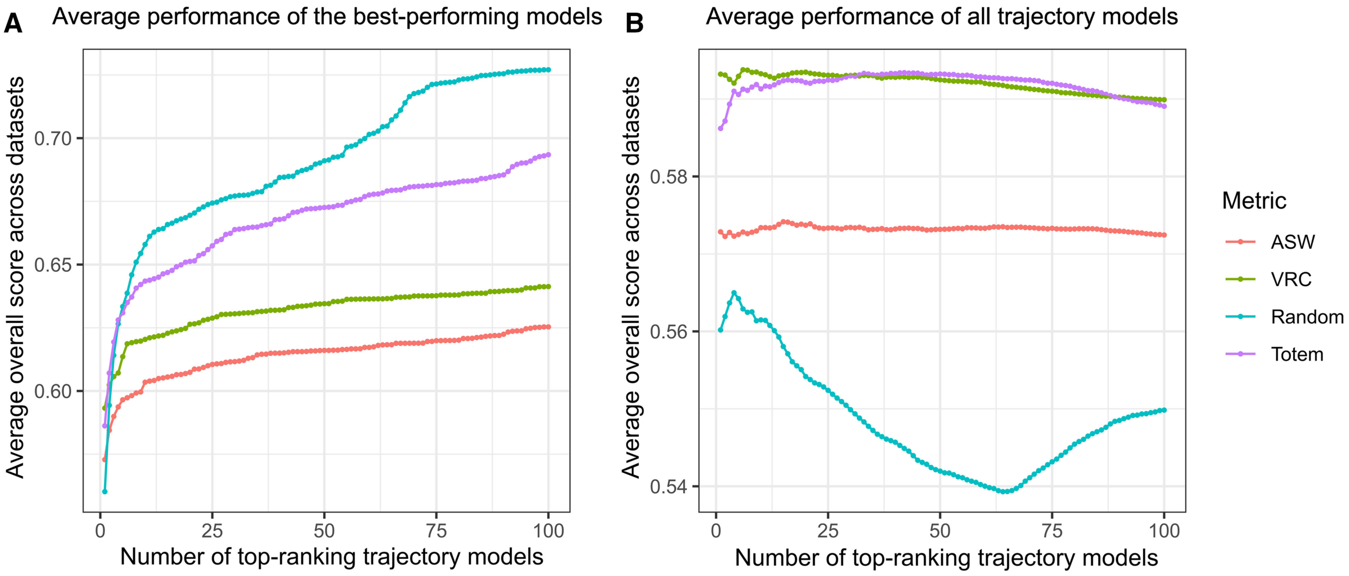

Figure 3. Comparing clustering selection methods in the dynverse benchmarking. We selected the top 100 clusterings from 10 000 k-medoids clusterings for each of the 216 dynverse benchmark datasets using four different methods: the average silhouette width (ASW), the variance ratio criterion (VRC), random selection, and the Totem method. For each clustering, the rest of the trajectory inference (Minimum Spanning Tree generation, smoothing) was performed using Slingshot. In (A), we varied the number of top-ranking trajectory models and selected the model for each dataset that gave the best overall score from all the models. In (B), we calculated the mean performance of all the models.

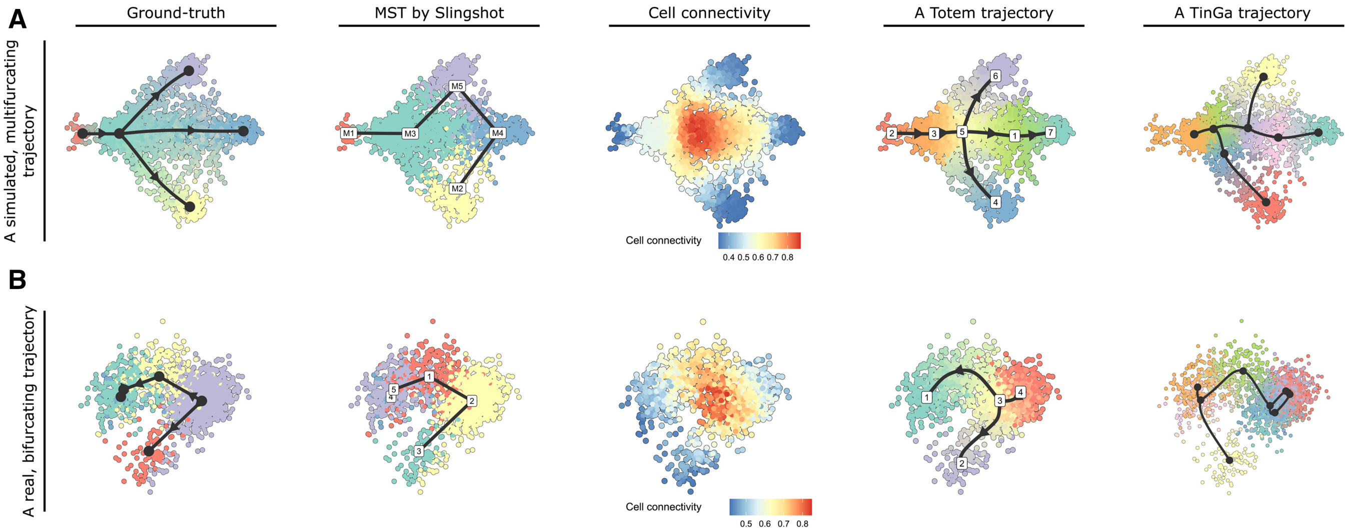

Figure 4. Example analyses with Slingshot, Totem, and TinGa. (A) A multifurcating trajectory simulated with dyntoy. (B) T cells from a mouse thymus with a bifurcating trajectory. The two-dimensional embeddings for visualization were generated with the Multi-Dimensional Scaling (MDS) method from the dyndimred R package.

embeddings, in this case 5-dimensional LMDS, a property that generally hinders trajectory inference of many of the dynverse benchmark datasets irrespective of the trajectory method used.

In the first example, we compared the trajectories (Fig. 4A) produced by Totem, Slingshot and TinGa for a simulated dataset that has a multifurcating trajectory (named multifurcating_4 in the dynverse benchmark data repository). With Slingshot, we used the ground-truth clustering of the simulated dataset as input. The MST produced by Slingshot was linear and did not correlate well with the true milestone network. The example shows how Slingshot can still produce inaccurate trajectories even if the ground-truth clustering is available because the cluster distances generate an MST with a wrong network. Similarly, TinGa's trajectory for the dataset was inaccurate because the multifurcation was not symmetrical. In contrast, Totem predicted trajectories that correlated well with the true milestone network. In addition, the cell connectivity measure of Totem provided, in general, accurate information about the location of the end points and the middle branching point.

The second example (Fig. 4B) shows similar results but for a bifurcating trajectory that includes T cells from a mouse thymus (Han *et al.* 2018, Saelens *et al.* 2019). Based on the trajectory inferred by Slingshot, the topology of this network could be linear, i.e. the starting point would be one of the end nodes, or bifurcating, i.e. the starting point would be one of the middle nodes, if we have no knowledge about the starting point of the trajectory. By selecting a trajectory with Totem that is in line with the cell connectivity profile of the data, we obtain a trajectory with the correct topology (bifurcating), where the bifurcation point is at the node in the middle, from which the bifurcation starts in the ground-truth trajectory. Both of these examples demonstrate, how the cell connectivity helped us to choose trajectories that exhibit the correct topology. TinGa's trajectory for this dataset was inaccurate because the two end points of the bifurcation were connected.

3.4 Robustness of Totem

We investigated the robustness of Totem to its parameters (Supplementary Figs S2 and S3). The results show that the robustness of Totem improved when the number of clusterings increased (Supplementary Fig. S2), and the variation was small when the number of clusterings was over 2500 (5000 or 7500). However, the upper limit of the cluster number range did not have a clear impact on the robustness, but it did affect the average performance (Supplementary Fig. S3). The lowest average performance was achieved when the upper limit of the cluster number range was only five (by default, 20), but otherwise the average performance was robust regardless of the upper limit of the number of clusters.

4 Discussion

While a large number of trajectory inference methods has been developed (Saelens *et al.* 2019) for constructing trajectories from single-cell data through mathematical modelling, the usability of these methods remains rather poor. One main reason is that the methods do not work optimally with every dataset, requiring the users to try different methods, tune the parameters, and adjust the preprocessing steps prior to trajectory inference. Oftentimes, the parameter tuning requires detailed knowledge of the underlying statistical and machine learning models, which many users are not familiar with, and the manuals of trajectory inference tools often do not provide clear instructions on how the parameters should be adjusted.

In this article, we introduced Totem, a tool designed to facilitate the inference of tree-shaped trajectories from single-cell data. Totem generates a large number of clustering results (by default, 10 000) with the *k*-medoids algorithm and calculates the cell connectivity measure, which acts as a useful baseline for finding the optimal trajectory. In our examples, we showed how the cell connectivity enabled us to select clustering results that generated accurate trajectories. Other trajectory inference methods, such as Slingshot (Street *et al.* 2018) and TinGa (Todorov *et al.* 2020), do not provide a metric like the cell connectivity that can be used as a reference to guide trajectory inference. With these methods the user needs to instead use the visualization to get a sense of the topology, use biological markers to ensure the model is biologically sensible, and change the parameters if the trajectory is not satisfactory, which can be arduous for users without advanced background in mathematics and machine learning.

In Slingshot, the between-cluster distances determine the MST and the milestone network. Our examples showed how even when the correct cell types were available, the MSTs predicted by Slingshot could still be inaccurate. One solution is to change the distance method, e.g. to a mutual neighbour-based method, but this will still not necessarily generate the correct network. Totem addresses this issue by providing a user-friendly interface to analyse MSTs generated based on different clustering results. With this interface the user can select an MST that best fits to the biological hypothesis and the cell connectivity profile without time-consuming and arbitrary parameter tuning. Totem then smoothens the MSTs of the selected clustering results using the simultaneous principal curves algorithm of Slingshot to obtain directed trajectories that include pseudotime. The trajectories can be exported as Slingshot or dynwrap objects to be used in downstream analysis, e.g. to perform differential expression analysis using tradeSeq (Van den Berge *et al.* 2020) or dyno (Saelens *et al.* 2019).

To benchmark Totem for inferring trajectories from scRNA-seq data, we used the dynverse benchmarking framework, which includes 216 tree-shaped trajectories. We compared Totem with the dynverse implementation of Slingshot and TinGa. Totem outperformed the other two methods for nonlinear trajectories (bifurcation, multifurcation, or some other nonlinear tree) and performed comparably with TinGa for linear trajectories. The results suggested that while Slingshot was the best method for linear trajectories among the tested methods, it was significantly worse than TinGa and Totem for nonlinear trajectories. This happened because ASW is generally known to select a small number of clusters, which will more likely generate a linear trajectory than a nonlinear trajectory when used for MST estimation. In contrast, TinGa is more likely to overcomplicate the trajectories with redundant cycles and disconnected parts than Slingshot and Totem, both of which can only infer tree-shaped trajectories. Indeed, the performance differences between the methods were mainly attributable to differences in the accuracy of the topology (HIM) and branch assignment (F1 branches).

To investigate what is the most optimal way to select the clustering that is used to generate the MST, we generated 10 000 *k*-medoids clustering results for each of the benchmark datasets and ranked them using different clustering evaluation methods, including ASW, VRC, and a Totem criterion that uses cell connectivity and VRC. When we investigated the performance scores of the top 100 clustering results, the results suggested that the Totem criterion and VRC provided the best average performance. However, the Totem criterion outperformed the VRC when we considered only the best trajectory from the set of trajectories. In other words, when testing multiple trajectories, it is more likely that we find a higher-performing trajectory with Totem than with the ASW and VRC methods, and the average performance of all the trajectories found by Totem will likely be at least as good as for the other methods.

Totem has the same limitations as Slingshot. In particular, it cannot be used to infer trajectories that have cycles or both diverging (cells diverge from a single point into several lineages) and converging (cells converge into a single point from several lineages) parts at the same time. In addition, unlike the updated Slingshot, Totem cannot currently handle disconnected trajectories. However, we do not consider this a major limitation because determining automatically which cell types belong to which disconnected sub-trajectories is not an easy task (Chen *et al.* 2022). A safer approach is to analyse the disconnected parts as separate trajectories by segregating the cell types manually before trajectory inference. Although there exist methods that can handle almost arbitrary topologies, such as PAGA (Wolf *et al.* 2019), Monocle3 (Cao *et al.* 2019), and

TinGa (Todorov *et al.* 2020), their issue is that they are more likely to overcomplicate the trajectory with extra cycles and branches than methods like Slingshot and Totem that are limited to tree-shaped trajectories. Similarly, methods like scShaper (Smolander *et al.* 2021), SCORPIUS (Cannoodt 2016), and Elpilinear (Albergante *et al.* 2020) that are limited to linear trajectories are still useful because they are guaranteed to provide the correct topology if the trajectory is expected to be linear, unlike the more complex methods.

An important consideration is how to perform the upstream analysis steps prior to trajectory inference, including quality control, normalization, and dimensionality reduction. The choice of the normalization method has been shown to affect the cell types that can be identified from single-cell data (Hafemeister and Satija 2019), and consequently the trajectory should also change to reflect the varying level of cell heterogeneity. Similarly, the choice of the feature selection and feature extraction methods can have an impact on the identifiable cell types. If analysing data with batch effects, the method used for data integration can also affect the composition of identifiable cell types (Luecken *et al.* 2022). Therefore, robustness to the upstream analysis steps should generally not be an objective of trajectory inference. Trajectory inference methods assume that the global distances between the cell types in the low-dimensional embeddings, such as principal components or UMAP dimensions, can be used to model the correct topology. If the correct topology cannot be inferred based on the cell type distances in the embeddings, alternative upstream analysis steps such as embedding refinement should be considered (Pandey and Zafar 2022).

In some cases, it is possible that the optimal trajectory, the one that is in line with the cell connectivity profile, is not readily obtainable from the top-ranking clustering results. This can happen when some cell types are extremely rare and some cell types are highly numerous. This cell number deviation makes it more difficult for the k-medoids to find a clustering result that creates the correct trajectory topology. To alleviate this issue, Totem's vignette explains that in situations such as these, it is a good idea to increase the number of clusters so that the rare cell types are captured in the clustering. If some of the highly numerous cell types are overclustered, these cell types can be merged using a function implemented in the Totem R package. This approach requires slightly more manual work but will guarantee a clustering that matches the cell connectivity profile.

As a potential future direction in Totem's development, the discrete cluster-level connectivity values could be turned into more precise, cell-level connectivity values by measuring the distance of each cell from the central/mass point of the cluster. However, the current implementation of Totem can also model the cell connectivity at the single-cell level by aggregating the connectivity vectors of a large number of clustering results (by default, 10 000) using the arithmetic mean. In our analyses, the arithmetic mean provided a continuous connectivity gradient, as opposed to the median-based approach.

To summarize, Totem is a tool designed to facilitate the inference of tree-shaped trajectories from single-cell data. It is built upon the popular Slingshot method, which uses a clustering to construct an MST and the simultaneous principal curves algorithm to obtain a directed trajectory along with pseudotime that quantifies cell differentiation at the single-cell level. The benefit of Totem over the available tools is that it is designed to provide a user-friendly interface for finding a clustering that is used as the basis of the trajectory. Although the clustering selection is highly critical for the success of the trajectory inference, performing it automatically in a way that will generate the correct milestone network in the trajectory remains challenging. To address this challenge, the analysis of different clustering results and their MSTs with Totem has been designed to be fast and easy without requiring complex parameter tuning and in-depth technical knowledge. As a notable difference compared to existing trajectory inference methods, Totem provides the cell connectivity measure, which aids the trajectory optimization by providing information about the location of the branching points and milestone transitions.

Acknowledgement

The authors thank the Elo lab for fruitful discussions and comments on the manuscript.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

L.L.E. reports grants from the European Research Council ERC [677943]; European Union's Horizon 2020 research and innovation programme [955321]; Academy of Finland [310561, 314443, 329278, 335434, 335611, 341342]; Sigrid Juselius Foundation, during the conduct of the study. This work was also supported by Turku Graduate School (UTUGS); Biocenter Finland; ELIXIR Finland.

Data availability

The benchmark data are available on Zenodo (Cannoodt *et al.* 2018). The Totem R package is available at https://github.com/elolab/Totem, and it includes a vignette that explains how Totem should be used. The codes that are relevant for repeating the benchmarking are available at https://github.com/elolab/Totem-benchmarking.

References

Albergante L, Mirkes E, Bac J *et al.* Robust and scalable learning of complex intrinsic dataset geometry via ElPiGraph. *Entropy* 2020;**22**: 296.

Butler A, Hoffman P, Smibert P *et al.* Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* 2018;**36**:411–20.

Calinski T, Harabasz J. A dendrite method for cluster analysis. *Comm Stat Theory Methods* 1974;**3**:1–27.

Cannoodt R. SCORPIUS improves trajectory inference and identifies novel modules in dendritic cell development. bioRxiv, 2016, preprint: not peer reviewed.

Cannoodt R, Saelens W, Todorov H *et al.* Single-cell-omics datasets containing a trajectory. *Zenodo*. 2018. https://doi.org/10.5281/zenodo.1443566.

Cannoodt R, Saelens W, Deconinck L *et al.* Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nat Commun* 2021;**12**:3942.

Cao J, Spielmann M, Qiu X *et al.* The single-cell transcriptional landscape of mammalian organogenesis. *Nature* 2019;**566**:496–502.

Chen W, Guillaume-Gentil O, Rainer PY *et al.* Live-seq enables temporal transcriptomic recording of single cells. *Nature* 2022;**608**:733–40.

Deconinck L, Cannoodt R, Saelens W *et al.* Recent advances in trajectory inference from single-cell omics data. *Curr. Opin Syst Biol* 2021;**27**:100344.

Fritzke B. A growing neural gas network learns topologies. In: *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press 1995.

Hafemeister C, Satija R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biol* 2019;**20**:296.

Han X, Wang R, Zhou Y *et al.* Mapping the mouse cell atlas by Microwell-Seq. *Cell* 2018;**172**:1091–107.e17.

Hastie T, Stuetzle W. Principal curves. *J Am Stat Assoc* 1989;**84**:502–16.

Ilicic T, Kim JK, Kolodziejczyk AA *et al.* Classification of low quality cells from single-cell RNA-seq data. *Genome Biol* 2016;**17**:29.

Jurman G, Visintainer R, Filosi M *et al.* The HIM glocal metric and kernel for network comparison and classification. In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015, 1–10.

Kaufman L, Rousseeuw PJ. Clustering large applications (program CLARA). In: *Finding Groups in Data*. New York: John Wiley & Sons 1990, 126–163.

Luecken MD, Büttner M, Chaichoompu K *et al.* Benchmarking atlas-level data integration in single-cell genomics. *Nat Methods* 2022;**19**: 41–50.

Maaten L, van der Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008;**9**:2579–605.

McInnes L, Healy J, Melville J. UMAP: uniform manifold approximation and projection for dimension reduction. arXiv preprint, arXiv:1802.03426, 2018.

Pandey K, Zafar H. Inference of cell state transitions and cell fate plasticity from single-cell with MARGARET. *Nucleic Acids Res* 2022;**50**: e86.

Papadopoulos N, Gonzalo PR, Söding J *et al.* PROSSTT: probabilistic simulation of single-cell RNA-seq data for complex differentiation processes. *Bioinformatics* 2019;**35**:3517–9.

Paradis E, Schliep K. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* 2019;**35**:526–8.

Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**:53–65.

Saelens W, Cannoodt R, Todorov H *et al.* A comparison of single-cell trajectory inference methods. *Nat Biotechnol* 2019;**37**:547–54.

Smolander J, Junttila S, Venäläinen MS *et al.* scShaper: an ensemble method for fast and accurate linear trajectory inference from single-cell RNA-seq data. *Bioinformatics* 2022;38(5):1328–1335.

Street K, Risso D, Fletcher RB *et al.* Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* 2018; **19**:477.

Todorov H, Cannoodt R, Saelens W *et al.* TinGa: fast and flexible trajectory inference with growing neural gas. *Bioinformatics* 2020;**36**: i66–74.

Van den Berge K, Roux de Bézieux H, Street K *et al.* Trajectory-based differential expression analysis for single-cell sequencing data. *Nat Commun* 2020;**11**:1201.

Wolf FA, Hamey FK, Plass M *et al.* PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol* 2019;**20**:59.

Wolf FA, Angerer P, Theis FJ *et al.* SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol* 2018;**19**:15.

Zappia L, Phipson B, Oshlack A *et al.* Splatter: simulation of single-cell RNA sequencing data. *Genome Biol* 2017;**18**:174.