RESEARCH

FORKS: Finding Orderings Robustly using K-means and Steiner trees

Mayank Sharma[1], Huipeng Li[2], Debarka Sengupta[3], Shyam Prabhakar[2*] and Jayadeva [1]

Abstract

Recent advances in single cell RNA-seq technologies have provided researchers with unprecedented details of transcriptomic variation across individual cells. However, it has not been straightforward to infer differentiation trajectories from such data. Here, we present Finding Orderings Robustly using K-means and Steiner trees (FORKS), an algorithm that pseudo-temporally orders cells and thereby infers bifurcating state trajectories. FORKS, which is a generic method, can be applied to both single-cell or bulk differentiation data. It is a semi-supervised approach, in that it requires the user to specify the starting point of the time course. We systematically benchmarked FORKS and 6 other pseudo-time estimation algorithms on 5 benchmark datasets, and found it to be more accurate, more reproducible, faster and more memory-efficient than existing methods for pseudo-temporal ordering. Another major advantage of our approach is that the algorithm requires no hyper-parameter tuning.

Keywords: scRNA-seq; Pseudo-temporal ordering; Steiner Trees

1 Introduction

Single cell RNA-seq (scRNA-seq) is a novel technique that allows measurement of transcriptomes at single cell resolution [1, 2, 3, 4, 5, 6, 7, 8]. Such a measurement is a snapshot of ongoing cellular processes. In contrast, bulk-sample methods only provide population-averaged measurements of gene expression, and therefore fail to accurately reflect the underlying diversity of expression phenotypes in the presence of cellular heterogeneity. For example, only single cell analysis can differentiate between high expression in a subset of cells and moderate expression in all cells. Thus, scRNA-seq is extremely valuable in unraveling complex biological phenomena such as cell differentiation, gene expression "noise" and gene regulatory interactions [9]. In particular, one common analytical approach has been to assume that transcriptomic variation across cells within a population approximates variation across time within a single lineage [10, 11, 12].

The problem of ordering single cell transcriptomes along continuous trajectories in state (expression) space, known as pseudo-temporal ordering, is typically solved in a lower-dimensional space that approximates proximity relationships between cells in the full space of transcript abundances [13]. Multiple methods have been used for dimension reduction in this context, including Principal Component Analysis (PCA) [14], Independent Component Analysis (ICA)

*Correspondence: prabhakars@gis.a-star.edu.sg

[2]Computational and Systems Biology, Genome Institute of Singapore, Singapore
Full list of author information is available at the end of the article
†Equal contributor

[15], Diffusion maps [16] and t-Stochastic Neighborhood Embedding (t-SNE) [17] are popular choices among others.

Multiple tools have been developed for pseudo-temporal ordering, some of which are capable of discovering branched trajectories. Here, we provide a brief overview of the techniques currently being used to achieve the same. Initial release of Monocle [18] used ICA as dimensionality reduction technique. Their latest release Monocle2 uses Discriminative Dimensionalty Reduction Tree (DDRTree) [19] technique to reduce the dimensionality. DDRTree solves the dual objective of finding the linear projection such that clusters are well separated. Wishbone [20] uses diffusion maps to create a nearest neighbor graph whereas Diffusion Pseudo-time (DPT) [21, 22] uses diffusion maps to find the transition probabilities and hence order the cells based on the diffusion distance, Waterfall [23] uses PCA, DPT uses diffusion maps, GPfates [24] uses Gaussian process Latent variable model (GPLVM) to reduce the dimension and overlapping mixture of gaussian processes (OMGP) to identify the bifurcations, TSCAN [25] uses a combination of PCA to reduce the dimension and model based clustering to find the cluster centers, SCUBA [26] models the developmental trajectory using a stochastic dynamical model and works in original space or in some cases reduce the dimension using t-SNE. DeLorean [27] uses Gaussian process to learn the gene expression profiles and the pseudo-time associated with each cell. SLICER [28] first select the genes then computes alpha hull of the data to find the optimal nearest neighbors for Locally Linear Embedding (LLE) [29]. Embeddr [30] reduces the dimension of data using Laplacian Eigenmaps [31], then fits principal curves to the manifold and projects onto the curve to estimate the pseudo-time. Embeddr cannot faithfully recover bifurcating trajectories if present in the data. Mpath [32] generates a branched trajectory by joining the cluster centers found using Hierarchial clustering and then projecting onto the Minimum Spanning Tree (MST).

Cannondt et. al. [33], provides a broad overview of many existing pseudo-time estimation algorithms. While evaluating these methods, we discovered that most suffered from a major limitation: the default parameter settings were not robust. As a consequence, they required adjustment of multiple hyper-parameters for each new dataset. We also found that some of the algorithms designed to detect branching trajectories performed poorly when the underlying biological process was unbranched. Moreover, non-linear embeddings were not necessary for inferring differentiation trajectories. In fact, linear methods such as PCA yielded at par or better results than non-linear methods.

In order to address the above challenges in pseudo-temporal ordering, we developed FORKS, a method that accurately infers bifurcating trajectories when present, and linear trajectories otherwise with no hyper-parameter tuning. The key idea behind FORKS is that similar cell types form clusters when embedded in a low dimensional manifold. As one traverses this manifold from a starting point or origin, gene expression changes and a pseudo-time can be assigned to individual cells based on their distance from the origin. FORKS relies on generalization of MST known as Steiner Trees to create robust and bifurcating trajectories. As previously noted [25], the MST of the entire set of cells may not be robust, since it could be substantially altered by a small perturbation in the data. FORKS therefore reduces the complexity of the problem and ameliorates measurement noise by finding the Steiner Points which can be thought of as cluster centers in case of k-means but connected via an MST. Another advantage of this approach is that FORKS is scalable to thousands of cells and genes. In order to compare the performance of FORKS against existing methods, we performed the first systematic benchmarking exercise for pseudo-temporal ordering algorithms, involving 7 algorithms in total and 4 transcriptomic datasets for which the true time stamps known.

2 Results

2.1 Trajectory Inference

In order to robustly infer single cell differentiation trajectories FORKS assumes the following facts about the data:

1 Differentiation is a continuous process and the data points represent samples on the manifold.
2 Different cell types form clusters in low dimensional space.
3 Cells are distributed normally around the line joining the two Steiner Points.

2.1.1 Data preprocessing

scRNA-seq gene expression matrix suffers from high technical variability which might result from various factors with the prime cause being the amplification process from a small quantity of RNA content in the cell. Thus, it is imperative to discard low quality cells and genes in order to achieve a robust ordering. After the preprocessing step, we reduce the dimensionality of data using PCA and finally, order the cells on the reduced dataset using FORKS.

2.1.2 Dimensionality reduction

Even after preprocessing, we are left with thousands of genes, thus leaving us with the task of solving the ordering problem in high dimension. Despite the high dimensional nature of data, most of the information lies close to a low dimensional manifold. We reduce the dimensionality of the original data matrix denoted by $X_o \in \mathbb{R}^{N \times D}$ to $X_r \in \mathbb{R}^{N \times d}$ using PCA, where N is the number of cells or data points, D is the number of genes after preprocessing and d is the reduced dimension. We compare various manifold learning/dimensionality reduction algorithms like PCA [14], ISOMAP [34], Multi-Dimensional Scaling (MDS) [35, 36], Spectral Embedding (SE) [37, 38], Random Forest (RF) [39] and t-SNE [17] to learn a reduced dimensional structure in the data. Contrary to prevailing biases towards using non-linear dimensionality reduction methods like Diffusion maps [22], our experiments demonstrate the eminence of PCA over these techniques. This empirical study indicates that most of the interactions between the genes tend to be linear in nature and only a few non-linear interactions may be required for differentiation process. Other advantages of PCA includes its computation speed, scalability and the fact that the number of dimensions to project the data on can be calculated using the energy or the variance captured in each component. Other techniques are highly sensitive to other parameters like kernel width as in case of Gaussian kernel for kernel PCA [40] or diffusion maps, perplexity in case of t-SNE and number of nearest neighbors in case of ISOMAP, Locally Linear Embedding (LLE) or any other method based on k-nearest neighbor graph.

Once a lower dimensional embedding has been found, we use approximate Eucledian Steiner Minimal Tree (ESMT) to find the Steiner points or cluster centers and then join them by a Minimum Spanning Tree (MST). The ESMT problem, as it is called, finds the shortest tree connecting N data points with K Steiner points, where $K \ll N$. ESMTs have been successfully applied to the areas of very large integrated circuits (VLSI), printed circuit boards, in the field of agriculture, telephony and in optimizing the building layouts. Application of ESMTs in the domain of genomics is the bridge we try to fill through our work.

Steiner Minimal Trees (SMTs) are hard to find as the number of Steiner points are not known a priori and for a given number of Steiner points, one has to determine the correct topology with respect to the other data points. It has been shown that computation of SMTs is NP-complete [41]. A work around these problems is to first set the number of Steiner points and then solve an optimization problem to determine the approximate location of Steiner points. ESMT problem has a strong connection with K-means and Minimum Spanning Tree (MST). We propose a gradient based optimization approach to solve the NP-complete problem. Though we may not find the optimal solution to the problem, we try to reach as close to the optimal solution. ESMT finds Steiner points or cluster centers such that total length of spanning tree formed by joining the Steiner Points is minimum. In many cases its solution is very similar to the solution of k-means but in case of noisy data, it finds robust solutions as compared to K-means. Once the MST joining the cluster centers is found, we project the cells onto the tree backbone or MST joining the cluster centers. FORKS uses a "starting cell" to commence the ordering. It finds the cluster center closest to the 'starting cell' and start the ordering from that center which we call as root of the tree. A starting cell is necessary because a completely unsupervised method may choose any of the points of MST to start the ordering, which might result in an incorrect ordering. Distances along the tree backbone starting from the root Steiner point now acts as a proxy for pseudo-temporal ordering.

Using the Steiner points to construct the MST, our algorithm thus can find bifurcations using small linear segments. When cross validated with multiple stratified folds of the datasets, we find that FORKS is superior to the other eight competing algorithms. It requires $O(NK)$ memory for storage of distance matrix hence can be scaled to large datasets. The name FORKS is an abbreviation for Finding Ordering Robustly using K-means and Steiner tree as it aptly finds a robust pseudo-temporal trajectory with the combination of K-means and Steiner Tree.

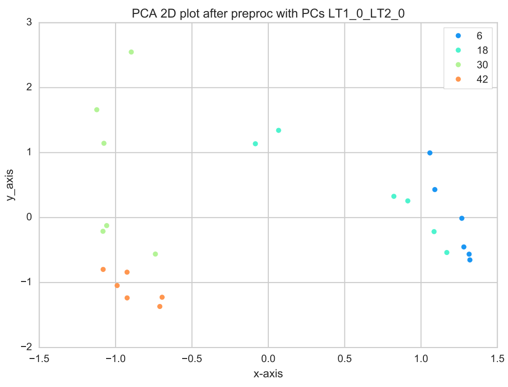
2.2 Datasets

We use 6 different datasets in the paper namely Arabidopsis [42], Deng_2014 [43], Guo_2010 [44], Klein [45], LPS [46] and Preimplant [47]. Among these datasets [42] is microarray dataset, [44] is Reverse transcription polymerase chain reaction (RT-PCR) dataset, and the rest are single cell RNA-seq datasets. These datasets cover a whole spectrum of techniques that have been used in gene expression analysis. The knowledge of observed cell time for each of the cell enables us to benchmark FORKS with other state-of-the-art algorithms that infer the pseudo-time trajectory. The details of datasets are presented in Table 1. To our knowledge this is the first comprehensive benchmarking with such a varied set of algorithms and datasets. The projection of individual datasets on the the first two principal components after the preprocessing step is shown in Figure 1 (1a-1f)

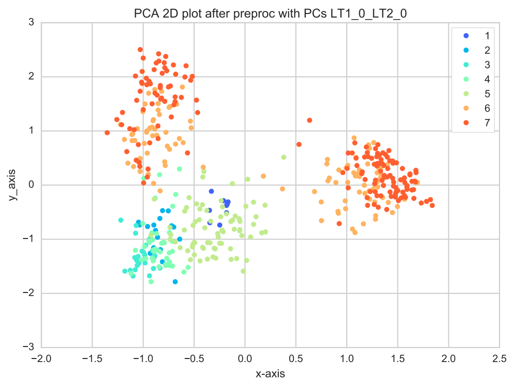
2.3 Comparison with other algorithms

We compared FORKS with eight other state-of-the-art algorithms namely DPT, GPfates, kmeans-R (used in [25]), Monocle2, SCUBA, SLICER and waterfall. The description of the algorithms used, viz., the dimensionality reduction techniques, trajectory modeling framework, scalability and their ease of use is mentioned in Tables 2 and 3. As a preprocessing step we use pQ normalization [57] for scRNA-seq datasets and data spherization (zero mean and unit variance transformation) for microarray and RT-PCR datasets. The data was then divided using stratified k-fold technique used in scikit [48] such that each fold contains similar distribution of true cell times as original data. The value of k varies with dataset due to difference in number of cells sequenced. Keeping the folds and the dataset same, all the algorithms were benchmarked. We calculate the Spearman correlation [49] between the pseudo-time computed by algorithm and the true time for each fold. Table **??** shows the (mean $\pm$ standard deviation) Spearman correlation for each of the algorithm. Monocle2, SCUBA, TSCAN, kmeans-R and waterfall do not require information about the "starting cell" of the pseudo-time. Hence, we run all these algorithm for a forward pass and a backward pass from their internal random starting point. For providing a more competitive environment to FORKS, we take the absolute values of correlation for these algorithms. [1]

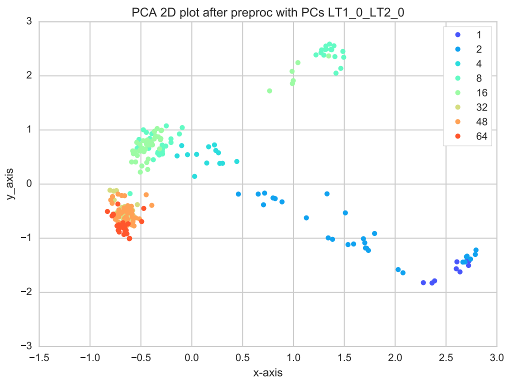
[1] All the tests were performed on a machine with Intel core-I5 2nd Gen processor with 6GB DDR2 RAM



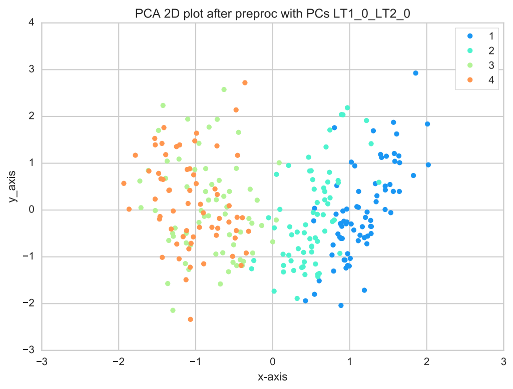
(a) 2D PCA plot of Arabidopsis data



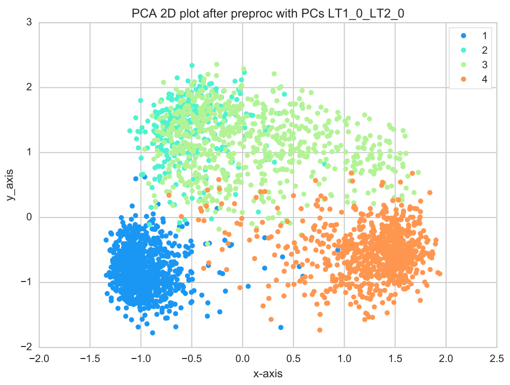
(b) 2D PCA plot of Guo_2010 data



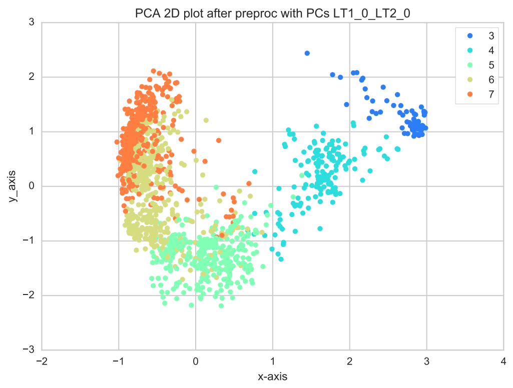
(c) 2D PCA plot of Deng_2014 data after preprocessing



(d) 2D PCA plot of LPS data after preprocessing



(e) 2D PCA plot of Klein data after preprocessing



(f) 2D PCA plot of Preimplant data after preprocessing

Figure 1: 2D PCA plot of Arabidopsis (microarray), Guo_2010 (RT-PCR), Deng_2014 (sc_RNA), Klein (sc_RNA), LPS (sc_RNA) and Preimplant (sc_RNA), the legend shows the various time points present in the datasets

Index	Dataset Name	Data type	Original dimension (cells X genes)	Dimension after preprocessing (cells X genes)	Time points	Dataset description	Bifurcations
1	Arabidopsis	microarray	24 × 150	24 × 117	4	Microarray data of the response of *Arabidopsis thaliana* to infection by nectropic fungal pathogen Botrytis cinerea	No
2	Deng 2014	scRNA-seq	293 × 1000	253 × 994	9	scRNA-seq embyonic developmental data from mouse preimplantation ranging from zygote to blastocyst	No
3	Guo 2010	RT-PCR	438 × 48	438 × 48	7	RT-PCR data quantifying developmental stages (1-cell to 64-cell blastocyst) from early stage mouse embryo	Yes
4	Klein	scRNA-seq	2717 × 24174	2717 × 323	4	Droplet sequencing of mouse embryonic stem cells upon leukemia inhibitory factor (LIF) removal	Yes
5	LPS	scRNA-seq	306 × 27723	268 × 3910	4	scRNA-seq bone marrow derived dendritic cell samples after simulating with lipopolysaccharide (LPS)	No
6	Preimplant embryo	scRNA-seq	1529 × 24444	1329 × 5490	5	scRNA-seq human preimplantation embryo dataset collected from 88 embryos ranging from day 3 to day 7	No

Table 1: Six datasets from multiple methods involving microarray, RT-PCR and scRNA-seq are used in the experiments

Figure 2 (2a-2f) shows the box plots of Spearman correlation for various folds on different datasets. Figure 2 show that FORKS outperform other algorithms in terms of Spearman correlations with known cell times in which median spearman correlation is greater than 0.9 in 5 out of 6 datasets.

Running times of various algorithms were also compared. Figures 3 (3a-3f) displays the box plots of run times in seconds, for various algorithm. Although it may not be fair to compare the run times of various algorithms as they are written in multiple programming languages. FORKS is completely coded in Python [50], and uses NumPy [51], SciPy [52] and Pandas [53] internally. FORKS has a higher run times than algorithms whose back-end is written in C or C++ for certain datasets. Still considering all these factors, FORKS is competitive in run times to other algorithms and in multiple instances where the dataset sizes are large it is faster than most. Figure 5 (5a-5b) shows the mean and standard deviation run times of all algorithms. We find that FORKS is third fastest in terms of run time and fourth in terms of robustness with respect to run times.

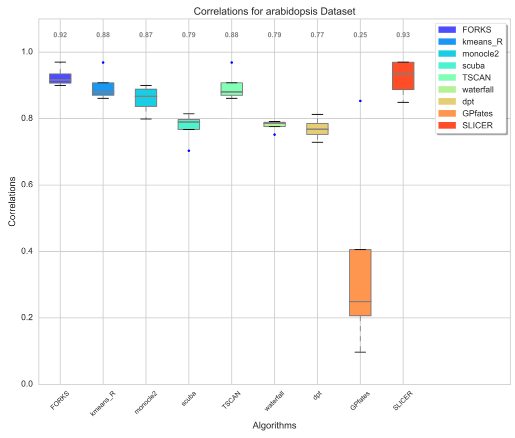
In order to check the accuracy and robustness of each method, we compute the mean and standard deviation of mean Spearman correlations for multiple instances of re-sampling of datasets which we call as data folds. Knowing the true cell times, we use stratified k-fold techniques to generate folds for each dataset. For an accurate and robust algorithm, its mean correlation across folds should be close to 1 and its standard deviation should be close to 0. The results are shown in Figure 4(4a-4b). FORKS has the highest median of the mean correlation among all the algorithms. Figure 4 indicate that FORKS has 10% higher median than its nearest competitor kmeans-R which is at 0.81. Using such an algorithm whose median correlation lies near 0.9 can faithfully recover the true trajectory for most of the datasets. FORKS also exhibits robust behavior across multiple datasets as the mean standard deviation achieved by the algorithm is the smallest at 0.02.

3 Discussion and Conclusion

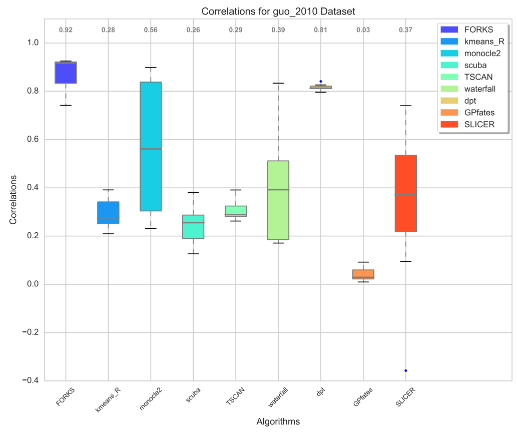
We developed FORKS (Finding Orderings Robustly using K-means and Steiner trees), a method that infers cellular trajectories from various gene expression datasets. Having the highest average Spearman correlation and smallest mean standard deviation of Spearman correlation with known cell times, than other algorithms, FORKS is able to robustly and accurately discover branching trajectories if present in the data. One of its biggest strength is minimal user interference in tuning all the parameters, an advantage which is not possessed by its competitor algorithms.

3.1 Using PCA as manifold learning algorithm of choice

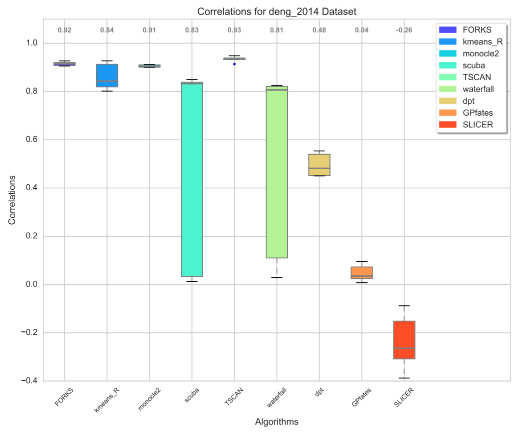
An interesting observation we found which is contrary to what has been previously established is that fact that non-linear embeddings are not



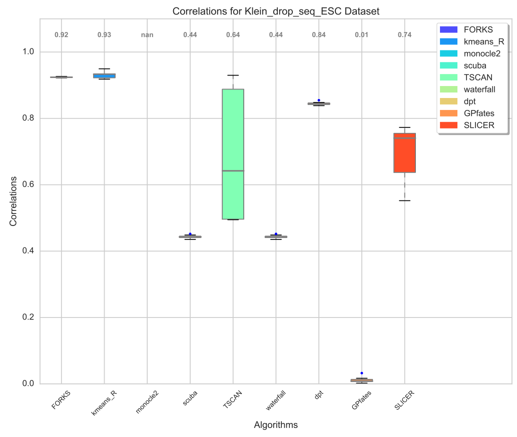
(a) Box plot of Spearman correlations for Arabidopsis



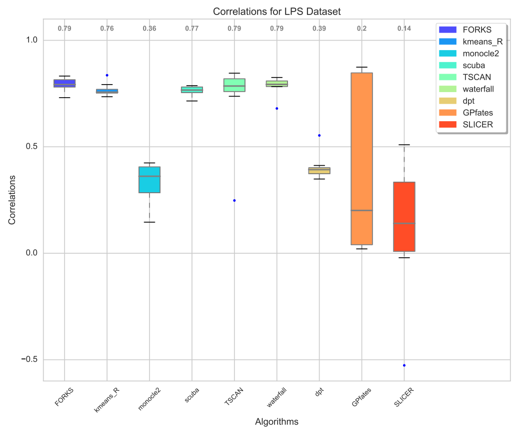
(b) Box plot of Spearman correlations for Guo_2010



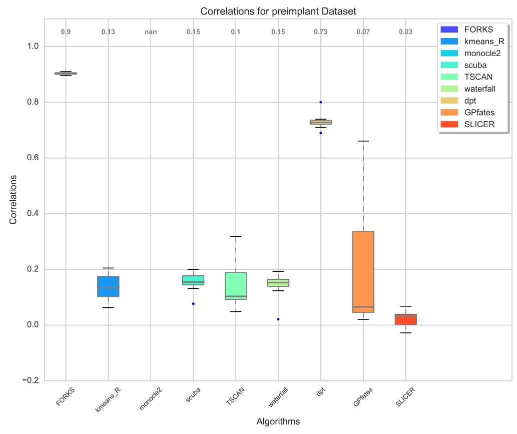
(c) Box plot of Spearman correlations for Deng_2014



(d) Box plot of Spearman correlations for Klein

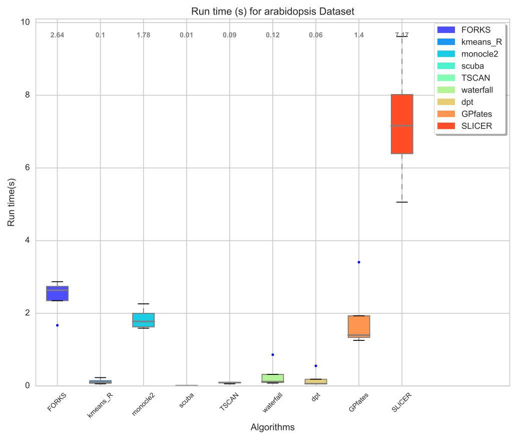


(e) Box plot of Spearman correlations for LPS

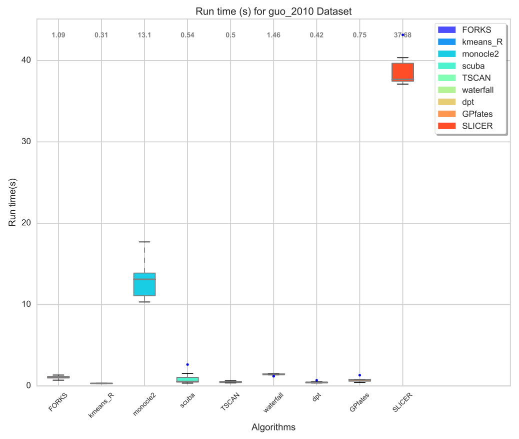


(f) Box plot of Spearman correlations for Preimplant

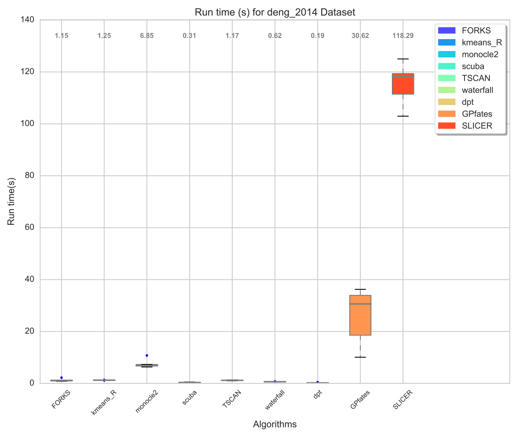
Figure 2: Box plots showing the correlations with given cell times for various algorithms for Arabidopsis (nfolds=4), Guo_2010 (nfolds=8), Deng_2014 (nfolds=5), Klein (nfolds=10), LPS (nfolds=8) and Preimplant (nfolds=10) datasets, the legend shows the various algorithms being compared. Values at the top of each figures are the median values



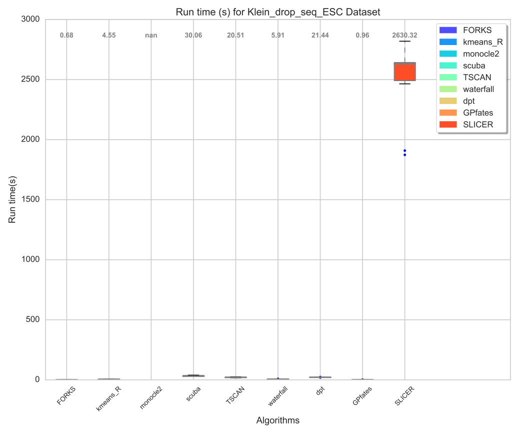
(a) Box plot of Run time (s) for Arabidopsis



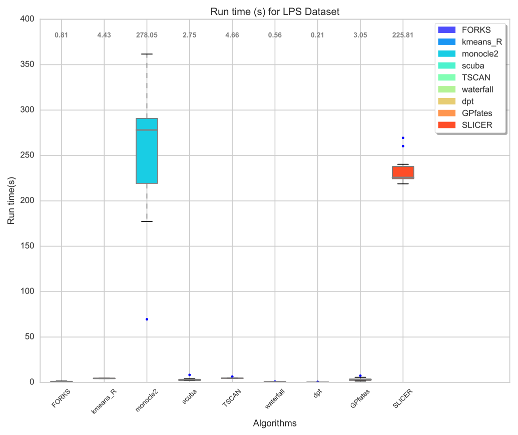
(b) Box plot of Run time (s) for Guo_2010



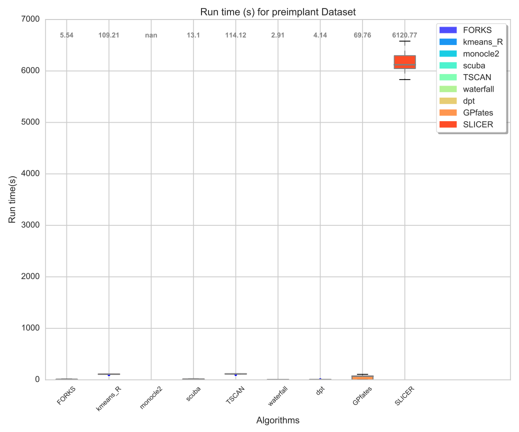
(c) Box plot of Run time (s) for Deng_2014



(d) Box plot of Run time (s) for Klein

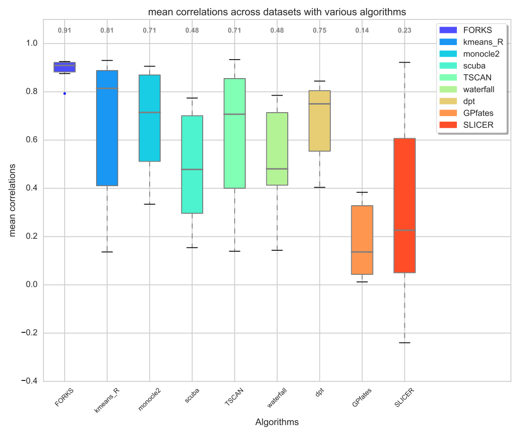


(e) Box plot of Run time (s) for LPS

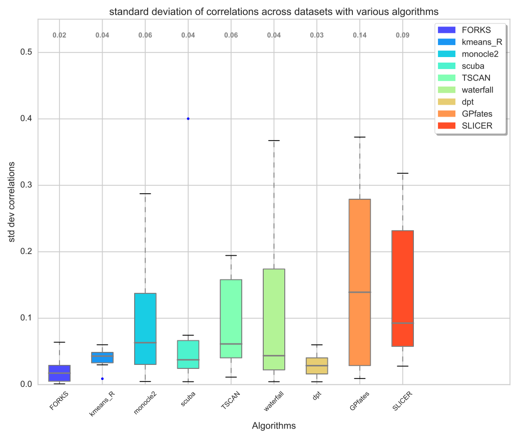


(f) Box plot of Run time (s) for Preimplant

Figure 3: Box plots showing the run times (s) with given cell times for various algorithms for Arabidopsis (nfolds=4), Guo_2010 (nfolds=8), Deng_2014 (nfolds=5), Klein (nfolds=10), LPS (nfolds=8) and Preimplant (nfolds=10) datasets, the legend shows the various algorithms being compared. Values at the top of each figures are the median values

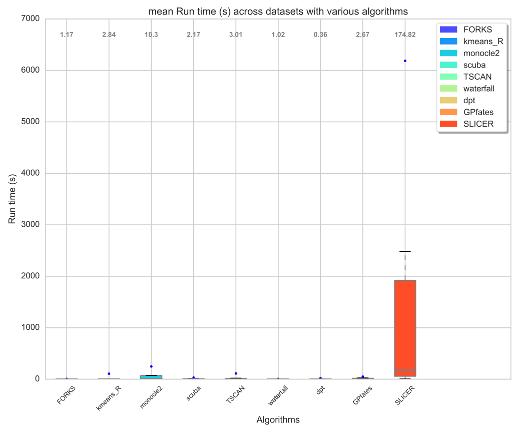


(a) Box plot of mean of Spearman correlations

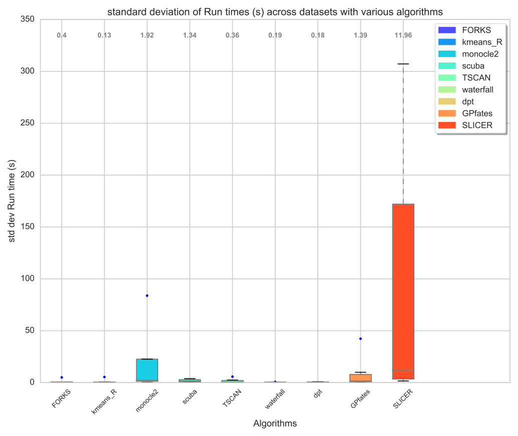


(b) Box plot of standard deviation of Spearman correlations

Figure 4: Box plots of means and standard deviation of Spearman correlations among all the datasets shows highly accurate and robust behavior of FORKS to change in folds and datasets. Values at the top of each figures are the median values



(a) Box plot of mean of run times (s)



(b) Box plot of standard deviation of run times (s)

Figure 5: Box plots of means and standard deviation of run times among all the datasets shows FORKS is highly competitive. Values at the top of each figures are the median values

Index	Name	Dimensionality reduction			Pseudo-time Trajectory	
		Manifold Learning	Clustering	Graph	Path finding	Cell ordering
1	SCUBA	t-SNE			Principal Curve	Projection on Principal curve
2	SLICER	LLE		k-NN	Shortest path from origin found by algorithm	Detect branches using geodesic entropy
3	TSCAN	PCA	Mclust		Longest path in MST	Project cells onto MST connecting cluster centers
4	waterfall	PCA	kmeans		MST connecting the cluster centers from origin found by algorithm	Project cells onto MST connecting cluster centers
5	kmeans-R	PCA	kmeans		MST connecting the cluster centers from origin found by algorithm	Project cells onto MST connecting cluster centers
6	Diffusion Pseudo-Time	Diffusion maps			Diffusion pseudo-time from origin using accumulated transition probabilities	Detect branches using random walks over cells
7	GPfates	B-GPLVM			B-GPLVM	OMGP
8	Monocle 2	DDRTree			Principal curves for each branch	Project cells onto principal curves
9	FORKS	PCA	Steiner tree		MST from starting cell	Projects cells onto path

Table 2: Pseudo-temporal ordering methods have many commonalities in their framework. As with all the methods, the first step being dimensionality reduction, then clustering or graph based analysis helps to easily model the trajectory, various methods are then utilized to find the ordering.

Index	Name	Branching	Extra inputs reg. data	Hyper-parameters to be tuned by user	Scalability w.r.t. cells
1	SCUBA	No	No	No	Yes
2	SLICER	Yes	Yes	Yes	No
3	TSCAN	Yes	No	No	Yes
4	waterfall	Yes	No	Yes	Yes
5	kmeans-R	Yes	No	No	Yes
6	Diffusion PseudoTime	Yes	Yes	Yes	yes
7	GPfates	Yes	No	No	Yes
8	Monocle 2	Yes	Yes	Yes	No
9	FORKS	Yes	Yes	No	Yes

Table 3: Table describes the scalability and amount of user interference required

necessarily required. For comprehensive empirical analysis we compared pseudo-temporal ordering on multiple datasets using various manifold learning like ISOMAP, MDS, PCA, Random Forest and t-SNE along with clustering algorithms like k-means, k-medoids, FORKS (Steiner tree with k-means as warm start) and Steiner tree with k-medoids as warm start.

The effect of various embeddings on average Spearman correlation for various datasets using k-means clustering are shown as box plots in Supplementary Figure 1 (1a-1f). We find that PCA has the highest median correlation on 5 out of 6 datasets. The run times for k-means clustering are shown in Supplementary Figure 2 (2a-2f). In this case PCA has the fastest median time for 3 out of 6 datasets.

Supplementary Figures 3 (3a-3b) and 4 (4a-4b) displays that PCA has the highest median correlation for all the datasets. It is highly robust with smallest standard deviation of median correlation and is comparable to ISOMAP and MDS in run times.

The correlations and time for k-medoids clustering are presented respectively, in Supplementary Figures 5 (5a-5f) and 6 (6a-6f).

Supplementary Figures 7 (7a-7b) and 8 (8a-8b) displays that ISOMAP has the highest median correlation for all the datasets. A pitfall with ISOMAP is the fact that median correlation is small and the embedding is non robust.

For FORKS (Steiner tree with k-means warm start), the correlations and run time are shown in Supplementary Figures 9 (9a-9f) and 10 (10a-10f).

Supplementary Figures 11 (11a-11b) and 12 (12a-12b) displays that PCA has the highest median correlation for all the datasets. It is highly robust with smallest standard deviation of median correlation.

Finally, for Steiner tree with k-medoids warm start, the correlations and run times are presented in Supplementary Figures 13 (13a-13f) and 14 (14a-14f).

Supplementary Figures 15 (15a-15b) and 16 (16a-16b) shows that PCA has the highest median correlation for all the datasets while ISOMAP has fastest running time. Although t-SNE displays the most robust behavior it lags considerably in median correlation and run-times.

We find that PCA has the best average Spearman correlations for k-means clustering, Steiner tree with k-medoids warm start and FORKS (Steiner tree with k-means warm start). ISOMAP has the best average Spearman correlation for k-medoids clustering. The average spearman correlation for PCA as an embedding is much higher than that of ISOMAP which becomes the basis of choosing PCA as the choice of manifold learning.

3.2 Using Steiner-Tree with k-means as clustering algorithm of choice

From Supplementary Figures 21 (21a -21d), we see that k-means and FORKS have similar average Spearman correlations, but FORKS has a smaller median standard deviation (0.058 compared to 0.065) across various algorithms and datasets. Hence we have a method that performs robustly.

3.3 Conclusion

In this paper we presented FORKS, an algorithm that finds a robust pseudo-temporal ordering of cells with no tuning of hyper-parameters. We compared FORKS with several state-of-the-art algorithms and showed its supremacy over them. We demonstrated that FORKS can be used with any kind of gene expression dataset till the dataset is in a tabular numeric form of cells $\times$ genes. We empirically proved our claim that the task of pseudo-temporal ordering does not require non-linear embeddings as is the general notion prevalent in the bio-informatics community. Finally, we incorporated the ideas from VLSI domain in the form of steiner tree and solved its approximate version using gradient descent to device FORKS. With ever increasing dataset size and quality, methods that scale up will have a certain edge over algorithms that do not. FORKS was created with such an ideology. Being closely related to k-means and using PCA as dimensionality reduction technique both of which can be solved stochastically, FORKS can be scaled up to large datasets. There is always scope of improving the current algorithm. Gene selection is an important step in any downstream analysis. For the purposes of this paper, we have used only the highly expressed and varying genes. For the problem at hand, using the marker genes can certainly improve the ordering. Also, to make the run times more smaller, the code can be cythonized or be completely coded in hardware friendly languages like C or C++. Also, one can try other clustering algorithms like Gaussian Mixture models (GMMs) [54], DBSCAN [55] and kernel-kmeans [56] to find the cluster centers and measure its performance.

4 Online Methods

4.1 Data Preprocessing:

Such dissimilitude of data presents us with the challenge of adopting the right preprocessing for each kind of dataset. We first highlight the common steps followed for all the datasets. The genes and cells containing all zero elements and the duplicated genes and cells present in the data were removed. Since marker gene information is not available in all the datasets and in the light of making the algorithm largely free from human intervention, we incorporated a gene selection step. Only a few genes who are expressed faithfully are responsible for the trajectory inference in the cells. We find the median and standard deviation of all the non-zero genes present in the data, by forming a one dimensional array. We discard low quality genes whose individual median gene expression in the expressed cells is below a certain multiple of the overall median. For the single-cell RNA sequencing datasets namely, (Klein [45], LPS [46], Deng_2014 [43], Preimplant [47]), we performed pseudo-counted Quantile (pQ) normalization [57]. It is well known that single cell RNA-sequencing datasets have inherent technical variability and high biological noise [58, 59, 60]. pQ normalization is a recently proposed technique build upon Quantile (Q) normalization [61] that reduces the technical bias, which is empirically found to be directly proportional to the number of detected genes [62]. pQ normalization homogenizes the expression of all genes below a fixed rank in each cell. Finally, for non-single cell RNA seq datasets (Guo_2010 [44] and Arabidopsis [42]), we found that sphering the dataset improves the pseudotime estimation. However pQ normalization degrades the performance.

4.2 Overview of FORKS algorithm:

It is known that only a few interactions are responsible for a particular problem like cell cycle or differentiation. In case of scRNA-seq, lowly expressed genes contribute largely to technical bias. Hence, it is imperative to reduce the dimension to capture the most meaningful interactions.

Dimensionality reduction: The algorithm begins by dimensionality reduction. Reducing the dimension, diminishes the complexity of problem at hand and reduces the noise present in higher dimensions. We use Principal Component Analysis (PCA) [14] to reduce the dimension of the data. PCA offers a graceful solution to problem of selecting the required number of dimension, by using the energy content in each of the principal components (PCs). We select the PCs such that the total energy content is at least 90%. There are other approaches to do so, for example, by eyeballing the curve to get the knee point. However, due to the lack of mathematical justification, this approach was discarded. PCA is a linear manifold learning technique many other pseudo-temporal ordering methods use various non-linear manifold learning techniques, and pay the cost of tuning multiple hyper-parameters, thereby increasing the complexity of training. If the parameters are not chosen correctly, one might end up learning erroneous manifold thereby deducing incorrect pseudo-time. PCA in this regards provides an excellent solution.

Finding number of clusters: Initial release of Monocle used minimum spanning tree (MST) based on the complete data and then found the longest path present in the MST. One major issue with this approach was the fact that MST on such a large dataset is not stable. Clustering the cells first and choosing cluster centers as the proxy for data is one way to mitigate problem. Once the dimension is reduced, next we choose the number of clusters that might be present in the data using Silhouette scores [63]. When the data labels are not known, the cluster evaluation should be performed using the model itself. The Silhouette Coefficient is composed of two scores for each sample:

1 a : The mean distance between a sample and all other points in the same class.
2 b : The mean distance between a sample and all other points in the next nearest cluster

The Silhouette Coefficient (s) per sample is calculated as:

$$s = \frac{b-a}{\max(a,b)} \tag{1}$$

The Silhouette Coefficient for a set of samples is given as the mean of the Silhouette Coefficient for each sample. The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster. We calculate the silhouette scores for the cluster centers ranging from 4 to 10. The number of cluster centers which gives the best average Silhouette Coefficient are selected.

ESMT: Let the number of cluster centers/steiner points be $K \ll N$. The cluster centers/steiner points are denoted by $x \in \mathbb{R}^d$ and data points by $p \in \mathbb{R}^d$. Let $r \in \mathbb{R}^{N\times K}$ be the matrix denoting the cluster center data point is assigned to. The problem of finding the euclidean steiner minimal tree (ESMT) can be written as:

$$\min_x \sum_{i,j \in MST} \|x_i - x_j\|^2 + C \sum_{n=1}^{N} \sum_{k=1}^{K} (r_{nk}\|p_n - x_k\|^2) \tag{2}$$

where, MST denotes the minimum spanning tree formed by steiner points.

The ESMT problem given as eq. (2) is non-convex as the second part of the objective function is the same as that of k-means, which is not a convex function [64]. Hence, optimization leads to a local minima. We use gradient descent to update the steiner points. As mentioned in the section 2, finding the Steiner tree is a NP-hard problem. We use approximate ESMT where we fix the number of clusters a priori. The initial number of cluster centers are calculated using Silhouette method and their initial values are computed using the solution of k-means. The hyper-parameter C is set to 1 in all our experiments. FORKS, which is mathematically written as eq. (2) can be interpreted as finding the steiner points such that the total length of minimum spanning tree formed joining the steiner points is minimum. We run the algorithm for 100 epochs keeping track of the best cost function values. The values of steiner points for which the cost function is minimum is returned.

Ordering: We divide the points belonging to each cluster and then use the methodology mentioned in TSCAN [25] for pseudo-temporal ordering. We first select a "starting cell" based on user input. For our experiments in order to demonstrate a proof of concept, we select the first cell belonging to the initial time as the starting cell.

Selecting a starting cell is important and this is where we have an edge over TSCAN or other methods which do not have a provision of selecting a starting cell. The proposed method of finding the longest path [25] in the MST can lead to erroneous results as it can be seen through the Figure 2 where TSCAN does not preform well. For bifurcating trajectories as in case of Figure 1b, where initial stage cell population may lie in the between the later stages of cell population, selecting the starting cell for ordering as one of the ends of the longest path is incorrect.

The ordering begins from the steiner point k_1 closest to the starting cell. Cells belonging to steiner point k_1 are assigned to edge $k_1 - k_2$. For, the intermediate steiner points $k_i, (i = 2, \ldots, K-1)$, the cells belonging to cluster k_i are divided into t parts where, t is the number of immediate neighbors of k_i in the MST. Once the assignment of cells is finished for each edge $e_j \in MST$ in MST, the cells are projected on their respective edges. The cell closest to k_1 is assigned a pseudo-time of 0. All the other cells are assigned values equal to the order of sorted projection value onto the edge e_i added with the last value of its preceding edge.

Code
The code for FORKS along with three of the datasets used in the paper can be found at the following github repository `https://github.com/macsharma/FORKS`.

Competing interests
The authors declare that they have no competing interests.

Author's contributions

Acknowledgements
Authors would like to acknowledge Tim Xiaoming for his feedback on scRNA-seq datasets and improvements to FORKS.

Author details
[1]Department of Electrical Engineering, Indian Institute of Technology, Delhi, India. [2]Computational and Systems Biology, Genome Institute of Singapore, Singapore. [3]Computer Science Engineering, Computational Biology, Indraprasth Institute of Information Technology, Delhi, India.

References
1. Bacher, R., Kendziorski, C.: Design and computational analysis of single-cell rna-sequencing experiments. Genome biology **17**(1), 1 (2016)
2. Yan, L., Yang, M., Guo, H., Yang, L., Wu, J., Li, R., Liu, P., Lian, Y., Zheng, X., Yan, J., *et al.*: Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. Nature structural & molecular biology **20**(9), 1131–1139 (2013)

3. Shalek, A.K., Satija, R., Adiconis, X., Gertner, R.S., Gaublomme, J.T., Raychowdhury, R., Schwartz, S., Yosef, N., Malboeuf, C., Lu, D., *et al.*: Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. Nature **498**(7453), 236–240 (2013)
4. Jaitin, D.A., Kenigsberg, E., Keren-Shaul, H., Elefant, N., Paul, F., Zaretsky, I., Mildner, A., Cohen, N., Jung, S., Tanay, A., *et al.*: Massively parallel single-cell rna-seq for marker-free decomposition of tissues into cell types. Science **343**(6172), 776–779 (2014)
5. Arsenio, J., Kakaradov, B., Metz, P.J., Kim, S.H., Yeo, G.W., Chang, J.T.: Early specification of cd8+ t lymphocyte fates during adaptive immunity revealed by single-cell gene-expression analyses. Nature immunology **15**(4), 365–372 (2014)
6. Zeisel, A., Muñoz-Manchado, A.B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., *et al.*: Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. Science **347**(6226), 1138–1142 (2015)
7. Treutlein, B., Brownfield, D.G., Wu, A.R., Neff, N.F., Mantalas, G.L., Espinoza, F.H., Desai, T.J., Krasnow, M.A., Quake, S.R.: Reconstructing lineage hierarchies of the distal lung epithelium using single-cell rna-seq. Nature **509**(7500), 371–375 (2014)
8. Moignard, V., Woodhouse, S., Haghverdi, L., Lilly, A.J., Tanaka, Y., Wilkinson, A.C., Buettner, F., Macaulay, I.C., Jawaid, W., Diamanti, E., *et al.*: Decoding the regulatory network of early blood development from single-cell gene expression measurements. Nature biotechnology **33**(3), 269–276 (2015)
9. Paul, F., Arkin, Y., Giladi, A., Jaitin, D.A., Kenigsberg, E., Keren-Shaul, H., Winter, D., Lara-Astiaso, D., Gury, M., Weiner, A., *et al.*: Transcriptional heterogeneity and lineage commitment in myeloid progenitors. Cell **163**(7), 1663–1677 (2015)
10. Shalek, A.K., Satija, R., Shuga, J., Trombetta, J.J., Gennert, D., Lu, D., Chen, P., Gertner, R.S., Gaublomme, J.T., Yosef, N., *et al.*: Single cell rna seq reveals dynamic paracrine control of cellular variation. Nature **510**(7505), 363 (2014)
11. Trapnell, C.: Defining cell types and states with single-cell genomics. Genome research **25**(10), 1491–1498 (2015)
12. Grün, D., Lyubimova, A., Kester, L., Wiebrands, K., Basak, O., Sasaki, N., Clevers, H., van Oudenaarden, A.: Single-cell messenger rna sequencing reveals rare intestinal cell types. Nature **525**(7568), 251–255 (2015)
13. van der Maaten, L.J.P., Postma, E.O., van den Herik, H.J.: Dimensionality Reduction: A Comparative Review (2008)
14. Jolliffe, I.: Principal Component Analysis. Wiley Online Library, ??? (2002)
15. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis vol. 46. John Wiley & Sons, ??? (2004)
16. Coifman, R.R., Lafon, S.: Diffusion maps. Applied and computational harmonic analysis **21**(1), 5–30 (2006)
17. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008)
18. Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N.J., Livak, K.J., Mikkelsen, T.S., Rinn, J.L.: The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nature biotechnology **32**(4), 381–386 (2014)
19. Mao, Q., Wang, L., Goodison, S., Sun, Y.: Dimensionality reduction via graph structure learning. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 765–774 (2015). ACM
20. Setty, M., Tadmor, M.D., Reich-Zeliger, S., Angel, O., Salame, T.M., Kathail, P., Choi, K., Bendall, S., Friedman, N., Pe'er, D.: Wishbone identifies bifurcating developmental trajectories from single-cell data. Nature biotechnology **34**(6), 637–645 (2016)
21. Haghverdi, L., Buettner, M., Wolf, F.A., Buettner, F., Theis, F.J.: Diffusion pseudotime robustly reconstructs lineage branching. bioRxiv, 041384 (2016)
22. Haghverdi, L., Buettner, F., Theis, F.J.: Diffusion maps for high-dimensional single-cell analysis of differentiation data. Bioinformatics **31**(18), 2989–2998 (2015)
23. Shin, J., Berg, D.A., Zhu, Y., Shin, J.Y., Song, J., Bonaguidi, M.A., Enikolopov, G., Nauen, D.W., Christian, K.M., Ming, G.-l., *et al.*: Single-cell rna-seq with waterfall reveals molecular cascades underlying adult neurogenesis. Cell Stem Cell **17**(3), 360–372 (2015)
24. Lönnberg, T., Svensson, V., James, K.R., Fernandez-Ruiz, D., Sebina, I., Montandon, R., Soon, M.S., Fogg, L.G., Stubbington, M.J., Bagger, F.O., et al.: Temporal mixture modelling of single-cell rna-seq data resolves a cd4+ t cell fate bifurcation. bioRxiv, 074971 (2016)
25. Ji, Z., Ji, H.: Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. Nucleic acids research, 430 (2016)
26. Marco, E., Karp, R.L., Guo, G., Robson, P., Hart, A.H., Trippa, L., Yuan, G.-C.: Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. Proceedings of the National Academy of Sciences **111**(52), 5643–5650 (2014)
27. Reid, J.E., Wernisch, L.: Pseudotime estimation: deconfounding single cell time series. bioRxiv, 019588 (2015)
28. Welch, J.D., Hartemink, A.J., Prins, J.F.: Slicer: inferring branched, nonlinear cellular trajectories from single cell rna-seq data. Genome biology **17**(1), 1 (2016)
29. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
30. Campbell, K., Ponting, C.P., Webber, C.: Laplacian eigenmaps and principal curves for high resolution pseudotemporal ordering of single-cell rna-seq profiles. bioRxiv, 027219 (2015)
31. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: NIPS, vol. 14, pp. 585–591 (2001)
32. Chen, J., Schlitzer, A., Chakarov, S., Ginhoux, F., Poidinger, M.: Mpath maps multi-branching single-cell trajectories revealing progenitor cell progression during development. Nature Communications **7** (2016)
33. Cannoodt, R., Saelens, W., Yvan, S.: Computational methods for trajectory inference from single-cell transcriptomics. European Journal of Immunology (2016)
34. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. science **290**(5500), 2319–2323 (2000)
35. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika **29**(1), 1–27 (1964)

36. Borg, I., Groenen, P.J.: Modern Multidimensional Scaling: Theory and Applications. Springer, ??? (2005)
37. Luo, B., Wilson, R.C., Hancock, E.R.: Spectral embedding of graphs. Pattern recognition **36**(10), 2213–2230 (2003)
38. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence **22**(8), 888–905 (2000)
39. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine learning **63**(1), 3–42 (2006)
40. Schölkopf, B., Smola, A., Müller, K.-R.: Kernel principal component analysis. In: International Conference on Artificial Neural Networks, pp. 583–588 (1997). Springer
41. Garey, M.R., Graham, R.L., Johnson, D.S.: The complexity of computing steiner minimal trees. SIAM journal on applied mathematics **32**(4), 835–859 (1977)
42. Windram, O., Madhou, P., McHattie, S., Hill, C., Hickman, R., Cooke, E., Jenkins, D.J., Penfold, C.A., Baxter, L., Breeze, E., *et al.*: Arabidopsis defense against botrytis cinerea: chronology and regulation deciphered by high-resolution temporal transcriptomic analysis. The Plant Cell **24**(9), 3530–3557 (2012)
43. Deng, Q., Ramsköld, D., Reinius, B., Sandberg, R.: Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. Science **343**(6167), 193–196 (2014)
44. Guo, G., Huss, M., Tong, G.Q., Wang, C., Sun, L.L., Clarke, N.D., Robson, P.: Resolution of cell fate decisions revealed by single-cell gene expression analysis from zygote to blastocyst. Developmental cell **18**(4), 675–685 (2010)
45. Klein, A.M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D.A., Kirschner, M.W.: Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell **161**(5), 1187–1201 (2015)
46. Amit, I., Garber, M., Chevrier, N., Leite, A.P., Donner, Y., Eisenhaure, T., Guttman, M., Grenier, J.K., Li, W., Zuk, O., *et al.*: Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses. Science **326**(5950), 257–263 (2009)
47. Petropoulos, S., Edsgärd, D., Reinius, B., Deng, Q., Panula, S.P., Codeluppi, S., Reyes, A.P., Linnarsson, S., Sandberg, R., Lanner, F.: Single-cell rna-seq reveals lineage and x chromosome dynamics in human preimplantation embryos. Cell **165**(4), 1012–1026 (2016)
48. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.*: Scikit-learn: Machine learning in python. Journal of Machine Learning Research **12**(Oct), 2825–2830 (2011)
49. Spearman, C.: The proof and measurement of association between two things. The American journal of psychology **15**(1), 72–101 (1904)
50. Van Rossum, G., Drake Jr, F.L.: Python Reference Manual. Centrum voor Wiskunde en Informatica Amsterdam, ??? (1995)
51. Van Der Walt, S., Colbert, S.C., Varoquaux, G.: The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering **13**(2), 22–30 (2011)
52. Jones, E., Oliphant, T., Peterson, P., et al.: Open source scientific tools for Python. Scipy (2001)
53. McKinney, W., *et al.*: Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference, vol. 445, pp. 51–56 (2010)
54. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological), 1–38 (1977)
55. Ester, M., Kriegel, H.-P., Sander, J., Xu, X., *et al.*: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol. 96, pp. 226–231 (1996)
56. Dhillon, I.S., Guan, Y., Kulis, B.: Kernel k-means: spectral clustering and normalized cuts. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 551–556 (2004). ACM
57. Sengupta, D., Rayan, N.A., Lim, M., Lim, B., Prabhakar, S.: Fast, scalable and accurate differential expression analysis for single cells. bioRxiv, 049734 (2016)
58. Tarazona, S., Furió-Tarí, P., Turrà, D., Di Pietro, A., Nueda, M.J., Ferrer, A., Conesa, A.: Data quality aware analysis of differential expression in rna-seq with noiseq r/bioc package. Nucleic acids research, 711 (2015)
59. Kolodziejczyk, A.A., Kim, J.K., Svensson, V., Marioni, J.C., Teichmann, S.A.: The technology and biology of single-cell rna sequencing. Molecular cell **58**(4), 610–620 (2015)
60. Stegle, O., Teichmann, S.A., Marioni, J.C.: Computational and analytical challenges in single-cell transcriptomics. Nature Reviews Genetics **16**(3), 133–145 (2015)
61. Bolstad, B.M., Irizarry, R.A., Åstrand, M., Speed, T.P.: A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics **19**(2), 185–193 (2003)
62. Hicks, S.C., Teng, M., Irizarry, R.A.: On the widespread and critical impact of systematic bias and batch effects in single-cell rna-seq data. bioRxiv, 025528 (2015)
63. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
64. Ng, A.: CS229 machine learning (2011)