Supplemental Material to Reconstructing differentiation networks and their regulation from time series single cell expression data

Jun Ding¹, Bruce J. Aronow², Naftali Kaminski³, Joseph Kitzmiller⁴, Jeffrey A. Whitsett⁴, and Ziv Bar-Joseph *¹

¹Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA.
²Division of Biomedical Informatics, Cincinnati Children’s Hospital Medical Center, Cincinnati, OH 45229, USA.
³Section of Pulmonary, Critical Care and Sleep Medicine, Yale School of Medicine, New Haven, CT 06520, USA.
⁴Section of Neonatology, Perinatal and Pulmonary Biology, Perinatal Institute, Cincinnati Children’s Hospital Medical Center, Cincinnati, OH 45229, USA.

Contents

Supplemental Methods

Initial clustering of single cells ........................................ 1
Reassigning clusters and initial model construction .................. 2
Predicting TFs regulating differentiation pathways .................. 4
A Kalman filter model for differentiation progression .............. 5
Model refinement and cell re-assignments ............................ 9

Supplemental Results

Performance comparison .................................................. 12
Performance of the alternative initial clustering (PCA+KMeans) strategy 12
Analysis of gene filtering strategy ..................................... 13
Supplemental Tables ..................................................... 14
Supplemental Figures ..................................................... 16

*To whom correspondence should be addressed. Ziv Bar-Joseph, Email:zivbj@cs.cmu.edu
**Supplemental Methods**

**Initial clustering of single cells**

We start by clustering the single cells at each individual time point to get the initial cell assignment. Here we prefer a correlation-based distance metric over the commonly used Euclidean distance, which is particularly sensitive to noise and thus not suitable for high dimensional and noisy data [1]. Here, we used the Spearman correlation to calculate the distance metric for clustering as it’s based on the ranks of the data thus more robust to the noises if compared with Pearson correlation [2]. Spectral Clustering was used to cluster those single cells based on the correlation affinity between them. Spectral Clustering has been reported to be widely used for the clustering of high dimensional data [3, 4] and even single cells [5, 6]. Spectral Clustering can be regarded as a dimension reduction and a K-Means afterward [7]. Although Spectral Clustering has many advantages especially on clustering high-dimensional data, it’s not scalable. The time complexity for Spectral Clustering is $O(n^3)$, where $n$ is total number of cells. To overcome this problem, we used an alternative initial clustering strategy PCA+K-means to deal with large dataset (enabled by "-l" option of scdiff). PCA is firstly used to project the high-dimensional data to lower dimensional space (In this study, we used the first 10 components). Then, K-means is used to cluster the cells in the large dataset.

A few widely used clustering quality assessment scores including Silhouette Score [8], Davis-Bouldin index [9] and AIC [10] (Akaike information criterion) were used here to determine the best number of clusters for each individual time point. First, Silhouette score was defined as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$  \hspace{1cm} (1)

$$a(i) = \frac{1}{|\text{Cluster}_i|} \sum_{x \in \text{Cluster}_i} [1 - \text{spearman}(x, \text{centroid}_i)]$$  \hspace{1cm} (2)

$$b(i) = \min_{i \neq j \in \text{Clusters}} [1 - \text{spearmanr}(\text{centroid}_i, \text{centroid}_j)]$$  \hspace{1cm} (3)

Where $i, j$ denotes Cluster $i$ and $j$. $b(i)$ is the smallest inter-cluster dissimilarity of cluster $i$ and $a(i)$ is a measure of intra-cluster dissimilarity of cluster $i$. For each cluster $i$, the silhouette score $s(i)$ is close to 1 (the best clustering) if $b(i) >> a(i)$. If $b(i) << a(i)$ then $s(i)$ will be close to -1 (the worst clustering). The average silhouette score of all clusters can be used to assess the overall clustering quality. Second, Davis-Bouldin index was defined as:

$$D(i) = \max_{i \neq j \in \text{Clusters}} \frac{a(i) + a(j)}{b(i, j)}$$  \hspace{1cm} (4)

$$b(i, j) = 1 - \text{spearman} (\text{centroid}_i, \text{centroid}_j)$$  \hspace{1cm} (5)

Where, $a(i)$ denotes the intra-cluster dissimilarity for cluster $i$ as defined above. $b(i, j)$ was used to represent the dissimilarity between cluster $i$ and $j$. $D(i)$ denotes
the Davis Bouldin index for cluster $i$, which is close to 0 if intra-cluster dissimilarity $a(i), a(j)$ is far smaller than inter-cluster dissimilarity $b(i, j)$. In other words, smaller Davis Bouldin index represents better clustering. We used the average Davis Bouldin index over all clusters to represent the overall clustering quality. Third, AIC score was defined as:

$$A(i) = RSS + 2KD$$

$$RSS = \sum_{i \in Cells} (x(i) - centroid_i)^2$$

Where, $x(i)$ represents the expression of cell $i$ and $centroid_i$ denotes the corresponding centroid of cell $i$. $K$ is the number of clusters for clustering and $D$ is the dimension of the expression data $x(i)$. $A(i)$ can be regarded as the RSS (Residual Sum of Squares) penalized by the number of clusters $K$. Smaller AIC represents better clustering quality.

With the aforementioned 3 scores, we used an ensemble strategy similar to random forest to determine the optimal number of clusters $K$ for each time point, which would be much more robust than just using individual clustering metric. The following is the procedure of determining the clustering parameter $K$. First, we randomly choose 90% of genes for each cell to get a sampled dataset $X$. Second, we calculate the Silhouette score, Davis Bouldin score and AIC scores for different $K$ candidates ($K \in [2, 20]$). Third, we determine the best $K$ based on 3 clustering scores. For each score, we will get a few candidates (Local optimal: peaks or negative peaks). The reason why to consider all possible local optimal instead of just the global optimal (the maximal or minimal values) is: the local optimal $K$ represents the best clustering at a specific level and it’s more robust that we consider all possible levels than just the level represented by the global optimal. For example, assume we have a group of animals with 5 different species. If we want to cluster those animals, we might end up with multiple local optimal such as 2 (representing sex) and 5 (representing species). $K = 2$ might be the global optimal as it’s relatively easier to just cluster the sex. But actually what we want to is to cluster these animals into more-specific species and thus $K = 5$ will present much more useful clustering results. The local optimal $K$ voted by a majority of these 3 scores will be kept as the best $K$ for current sample $X$. If we have multiple $K$ with the same number of max votes, we will choose the $K$ with best scores (Max for Silhouette, Min for Davis Bouldin index and AIC). The obtained optimal $K$ represents the vote of sample $X$. Finally, we generated 100 samples (voting trees) and we choose the $K$ voted by a majority of these voting trees. If we have multiple best $K$, we choose the one with best scores.

### Reassigning clusters and initial model construction

The single cell measurement for gene expression provided important insights for us to study the cell differentiation. However, single cells might have different cell cycles and growth rate [11], which means the gene expression measured at the same time point is probably from different developmental stages. With that said, it’s very important to take the heterogeneity into consideration if we want to study the differentiation process using the single cell expression data.
In this work, we represent the differentiation stage for each cell using a ‘Similarity To Ancestor-STA’ strategy. This was based on the following assumption. During the differentiation process, the differentiated cells are becoming more and more dissimilar to the same common ancestor (cluster of undifferentiated cells). Therefore, similarity to the same common ancestor can be used to represent the differentiation stage of a cell regardless of its measure time. The STA of given cluster was a vector of STA for all cells within the cluster. \( STA_{\text{cluster}} = [STA_{\text{cell}} \ \forall \text{cell} \in \text{cluster}] \) and the STA of cells was calculated by Spearman correlation. We re-group all the clusters based on the calculated STA values. First, we calculate the average gene expression for the ancestor cluster (the differentiation starting point, e.g. E14.5 in the lung single cells dataset). Second, we calculate \( STA = \text{Spearman}(\text{cell}.E, \text{Ancestor}.E) \) for each cell of every cluster. Where, \text{cell}.E and Ancestor.E represent the gene expression for the cell and the ancestor cluster respectively. The expression of the cluster was initially estimated as the mean value of the expression of all cells within the cluster. Third, we sort all clusters based on the mean value of the cluster STA vector. Fourth, we compare the STA vectors of every two neighboring cluster \( X \) and \( Y \) using ranksums test \( p_v = \text{ranksums}(X,Y) \). If they are statistically similar to each other, they will be grouped into the same ‘differentiation stage’. If they are statistically different \( (p - value < 0.05) \), they will be assigned to different ‘differentiation stages’. Besides the p-value, we also compared the differentiation difference between \( X \) and \( Y \) \( \text{diff}(X,Y) = \text{Avg}(X) - \text{Avg}(Y) \), where \( X \) and \( Y \) are STA vectors for two adjacent clusters. If \( \text{diff}(X,Y) \geq 2\text{Avg}(\text{all diffs}) \), \( X \) and \( Y \) will be also regarded as significantly different and these two clusters will be assigned to different groups. This was mainly used to deal with the clusters with limited number of cells (e.g. 3 ciliated cells in lung data set) and we do not have sufficient samples to calculate p-value. With the aforementioned strategy, all clusters will be re-grouped into different ‘differentiation levels’, which represents the relative differentiation stage of each cluster from the common ancestor (the starting undifferentiated cells).

Once we determined the set of levels in the model (differentiation stages) and the clusters associated with each level, we next connect clusters in each level to its parent cluster at the preceding level (except for the first level / time point cluster(s)). Here, we used the following statistical strategy to find out the parent for a specific cluster \( X \). First, calculate the similarity vector between the cluster \( X \) and any other cluster \( Y \) at level(X)-1 as: \( S(X,Y) = [\text{Spearman}(\text{cell}, Y) \ \forall \text{cell} \in X] \). Second, compare \( S(X,Y) \) for all possible \( Y \) at level(X)-1 using ranksums test \( p - value = \text{ranksums}(S(X,Y_0), S(X,Y_1)) \), \( Y_0 \) is the closet and \( Y_1 \) is the second closest cluster to \( X \) at level(X)-1. If they are significantly different, it suggests that level(X)-1 is the parent level of \( X \) and \( Y_0 \) is the parent of \( X \). If not, it indicates that level(X)-1 might be just a elder brother level (\( X \) are similar to all clusters at level(X)-1, which suggests that the clusters at level(X)-1 are from the same parent as cluster \( X \) and they are older). In this case, we need to go back one more level to search the parent. We keep doing this until reaching the parent level or the ancestor level (the very first level). By connecting all clusters to its parent, we get a graph (clusters as Nodes, parent-child relationship as Edges) to represent the model.
Predicting TFs regulating differentiation pathways

We can use a graph to represent the differentiation paths. The cluster was represented as a node and the connection between the cluster and its parent can be regarded as an edge. With the defined graph, we will try to associate a set of regulating Transcription Factors (TFs) for each individual edge. We have a collected set of commonly used TF-DNA interactions. With the TF targets information from the interaction file, we are able to identify a list of enriched TFs for each individual edge. First, we identify a differentially expressed (DE) gene list-DEG for each edge. The DEG is determined by student t-test and fold change.

\[
DEG = \{ g \mid p_{\text{t-test}}(X_g, Y_g) < 0.05 \land \text{abs}(fc(X_g, Y_g)) < 1.5 \} \tag{8}
\]

\[
f_{c}(X_g, Y_g) = \text{Avg}(Y_g) - \text{Avg}(X_g) \tag{9}
\]

Where \(X_g\) denotes the expression vector of gene \(g\) for all cells of the starting node (cluster) of the edge and \(Y_g\) denotes the expression vector of gene \(g\) for ending node (cluster) of the edge. As we mentioned earlier, the gene expression was in log-space, the fold change can be easily calculated as the difference of average between \(X_g\) and \(Y_g\). Second, the enriched TFs for given edge is statistically determined.

\[
p-value(x) = 1 - \sum_{0 \leq k < a} \binom{n}{k} p(x)^k (1 - p(x))^{n-k} \tag{10}
\]

\[
p(x) = \frac{\text{Target}(x)}{G} \tag{11}
\]

Where \(a\) is the number of targets of TF \(x\) in DE genes, \(n\) is the number of all DE genes, \(G\) is the number of all genes considered, and \(\text{Target}(x)\) denotes the number of targets of TF \(x\) in all considered genes. TFs with \(p-value < 0.1\) was kept as the candidates of regulating factors. The reason why we used 0.1 (a relatively larger cutoff) here is that we want to keep as many regulating TFs as possible for analysis purpose. Besides the p-value, we also require that the regulating TF must be expressing, which means it has to be expressing in at least 20% cells of the cluster. A ranking was also provided besides the p-value for each regulating TF to demonstrate the relative regulating power at each specific edge.

With those regulating TFs, we further explored the transition relationship for each edge to demonstrate how those TFs are driving the starting cluster to ending cluster of the edge. Here, we utilized a Lasso regression \([12]\) to learn the transition relationship for all edges of the graph. We used 3 classes (Up-regulated ↑, down-regulated ↓, None-regulated ≈) to label the transition of all genes from the starting cluster to the ending cluster of each edge. A gene was claimed as 'up-regulated' on specific edge if the fold change is larger than 1.5, 'down-regulated' if the fold change is smaller than -1.5, and 'None-regulated' otherwise. Here, we used a one-vs-rest multi-class logistic regression strategy, which can be interpreted as a set of binary logistic regressions (in our case, 2 binary logistic regressions as we have 3 different classes). The cost
function $J$ for the binary logistic regression was described as:

$$J(\theta) = -\frac{1}{n} \sum_{i=1}^{n} [(Y_i \log(h_\theta(X_i)) + (1 - Y_i)\log(1 - h_\theta(X_i))] + \frac{\lambda}{2n} \sum_{j=1}^{p} |\theta_j| \quad (12)$$

$$h_\theta(X_i) = \frac{1}{1 + e^{-\theta^T X_i}} \quad (13)$$

$Y_i$ denotes the corresponding class label ($\uparrow \downarrow \approx$) for gene $i$. Note, $Y_i$ is binary as it’s only showing whether the gene is belonging to a specific class and we have 2 sets of these binary logistic regressions. $n$ is the total number of all genes considered. $X_i$ represents the vector of all TFs, which target gene $i$. It’s a vector of binary 0/1. $\theta_j$ represents the weight for TF $j$. The cross-validation was used to determine the best $\lambda$. In this study, we used the LogisticRegressionCV module from scikit-learn [13] for above regression analysis. Using the above Lasso regression, the weight for some TFs might shrink to zero and we only keep those non-zeros as our regulating TFs.

A Kalman filter model for differentiation progression

To model expression changes and regulation during single cell differentiation we use a Kalman Filter model. Similar to Hidden Markov Model (HMMs), in Kalman filters we need to estimate transition and emission models, though unlike the unconstrained version in HMMs these take a specific, linear, form. Our Kalman Filter model assumes the gene expression at cluster $s$ is related to the expression at its parent cluster $P_s$ based on the following transition model:

$$X_s = A_s X_{P_s} + B_s + w_s \quad (14)$$

$$w_s \sim N(0, Q) \quad (15)$$

Where $X_s$ denotes the gene expression vector at cluster $s$, $X_{P_s}$ denotes the gene expression at parent cluster of $s$. $w_s$ is the process noise, which is assumed to be drawn from a zero mean Gaussian noise with co-variance $Q$. We need to learn Transition matrix $A$ and transition offset matrix $B$ for the transition model. The matrix $B$ is used in the transition model of the Kalman Filter. A Kalman Filter assumes the true state (e.g. true expression of a cell sub-type) at time $k$ evolved from the state at time $k-1$ using the transition model described in eqn. (14). As we discuss, while only using the expression data (without regulatory information), we rely on $A_s$, which is the identity matrix as we assume the gene expression profile of consecutive states should be similar. The unique aspect of our method is the addition of the matrix $B_s$, which relies on the regulatory information. Specifically, $B_s$ is determined by the logistic regression classifier. Since we assume the independence between genes (transition matrices are diagonal) we can decompose the above equation to individual genes as $X_s(g) = A_s(g) X_{P_s}(g) + B_s(g) + w_s(g)$. Given regulatory list of TFs predicted to regulate gene $g$, we can determine the expected level for $g$ in state $s$ based on the expression of its parents in state $P_s$ using a three way logistic classifier (up-regulated $\uparrow$, down-regulated $\downarrow$ or non-changing $\approx$). Using this we can get the $B_s(g)$ value as:
\[ B_s(g) = \begin{cases} 
U & \text{if } g \uparrow \\
D & \text{if } g \downarrow \\
0 & \text{if } g \approx 
\end{cases} \quad (16) \]

As the gene expression in study is in log space, therefore, we have the following relationship:

\[
X_s(g) = X_{Ps}(g) + U + w_s(g), \quad \text{if } g \uparrow \quad (17)
\]
\[
X_s(g) = X_{Ps}(g) - D + w_s(g), \quad \text{if } g \downarrow \quad (18)
\]
\[
X_s(g) = X_{Ps}(g) + w_s(g), \quad \text{if } g \approx \quad (19)
\]

Where, \( U \) is the average log fold change for up-regulated genes. \( D \) is the average log fold change for down-regulated genes. \( X_s(g), A_s(g), w_s(g) \) denote the expression \( g \), transition factor and noise of gene \( g \) at cluster \( s \). From the above analysis, \( A \) is an Identity matrix \( I \) and \( B \) is a diagonal matrix with 3 alphabets \([U,D,0]\) depending on the regulation situation of the specific gene. Here, we assumed the independence of the expression of all genes. There are a few reasons: (1) Although some genes might be co-regulated and thus their expression is correlated, we are not able to accurately depict such relationship. Therefore, it’s better that we just estimate the expression of specific gene \( g \) based on the expression of \( g \) at the parent cluster and the observation of \( g \) expression at the current cluster. It might be true that gene \( g \) and gene \( a \) are correlated, but it’s problematic if we want to estimate the expression of gene \( g \) based on the information of gene \( a \) as we can’t guarantee that such correlation is true. (2) In this study, we consider more than 15k genes and thus 2 dimensional transition matrix is huge. It’s both memory and time efficient if we assume the independence of gene expression as all the matrix will be simplified to diagonal and can be treated as 1 dimensional vector. The emission (or measurement) model of the Kalman Filter can be represented as:

\[
Y_s = H_s X_s + v_k \quad (20)
\]
\[
v_k \sim N(0, R) \quad (21)
\]

Where \( Y_s \) is the observed expression for all genes at cluster \( s \). \( X_s \) represents the real expression of all genes at cluster \( s \). \( v_k \) is the measurement noise, which is zero mean Gaussian noise with variance \( R \). \( H_s \) here is a identity matrix.

As we assumed the independence of gene expression, we can estimate the expression of each gene separately using Gaussian models and thus the Kalman Filter was much simplified in our study (one-dimensional Kalman Filter) and we do not need any matrix manipulation, which significantly improves the time and memory efficiency as we discussed above. Actually, Kalman Filter here can be regarded as a weighted average of prediction value (Based on the transition from parent cluster) and observation value (Based on observation on the current cluster). Since all the noise in the system was assumed to be Gaussian, we can write the probability of the expression \( g \) based on transition model and emission model respectively.
\[ p_{tr}(g) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(g-g_{tr})^2}{2\sigma_1^2}} \]  \hfill (22)

\[ p_{em}(g) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(g-g_{em})^2}{2\sigma_2^2}} \]  \hfill (23)

Where \( g \) represent specific gene expression value, \( p_{tr}(g) \) calculates the probability of \( g \) is being the true expression value of the specific gene based on the transition model. \( p_{em}(g) \) calculates the probability based on the emission model. \( \sigma_1^2 \) and \( \sigma_2^2 \) are the variance of gene \( g \) for transition (variance at parent state plus process noise variance) and emission model (measurement noise variance) respectively. The overall probability can be written as:

\[ p(g) = p_{tr}(g)p_{em}(g) = Ce^{-\frac{1}{2}(g-g_{tr})^2/\sigma_1^2-\frac{1}{2}(g-g_{em})^2/\sigma_2^2} \]  \hfill (24)

It can be also written as the following form:

\[ p(g) = C'e^{-\frac{1}{2}\frac{\sigma_1^2+\sigma_2^2}{\sigma_1^2+\sigma_2^2}(g-g_{tr}\sigma_2^2+g_{em}\sigma_1^2)^2} \]  \hfill (25)

Where \( C \) and \( C' \) are constant. From this result, the most probable expression \( \hat{g} \) is: \( \hat{g} = g_{tr}\sigma_2^2+g_{em}\sigma_1^2 \) and the combined variance is: \( \hat{\sigma}^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2} \). If we define a gain factor \( K \) as \( K = \sigma_1^2/(\sigma_1^2+\sigma_2^2) \), we can rewrite the best estimate as: \( \hat{g} = g_{tr} + K(g_{em} - g_{tr}) \) and the variance: \( \hat{\sigma}^2 = (1-K)\sigma_1^2 \). This is the general form of the classic Kalman Filter. Based on the above analysis, we can easily find that 1 dimensional Kalman Filter is actually a weighted sum of predictions from transition and emission models, which can be further illustrated by the following equations:

\[ \hat{g} = \frac{g_{tr}\sigma_2^2+g_{em}\sigma_1^2}{\sigma_1^2+\sigma_2^2} \]  \hfill (26)

\[ = \frac{w_1}{w_1+w_2}g_{tr} + \frac{w_2}{w_1+w_2}g_{em} \]  \hfill (27)

\[ \hat{\sigma}^2 = \frac{1}{w_1+w_2} \]  \hfill (28)

\[ w_1 = \frac{1}{\sigma_1^2} \]  \hfill (29)

\[ w_2 = \frac{1}{\sigma_2^2} \]  \hfill (30)

Here we used 2 predictors (1 transition model and 1 emission model). We can further expand the 1 dimensional Kalman Filter into the case with multiple (n) predictors.

\[ \hat{g} = \frac{\sum_{i=1}^{n} w_i g_i}{\sum_{i=1}^{n} w_i} \]  \hfill (31)

\[ \hat{\sigma}^2 = \frac{1}{\sum_{i=1}^{n} w_i} \]  \hfill (32)

\[ w_i = \frac{1}{\sigma_i^2} \]  \hfill (33)
where \( g_i \) denotes the prediction from the i-th predictor, \( w_i \) is the weight for the i-th predictor and \( \sigma_i^2 \) is the variance for i-th predictor. The proof for the above equation is fairly simple. Based on the attribute of Gaussian model, if we combine 2 Gaussian models, we will still have a Gaussian model with mean and variance as we derived above. Here, combining n Gaussian estimators can be regarded as combining 2 Gaussian models for n-1 times. Using equations for combining 2 Gaussian models, we can obtain the above equations for n Gaussian models.

In this study, we also used the Kalman smoothing strategy, which used 2 runs (Forward pass and Backward pass) of Kalman Filter to get a smoothed estimation of the gene expression at specific cluster \( s \). First, we run the Kalman Filter forward to estimate the expression of each cluster from root to leaf of the graph. Second, we run the Kalman Filter backward to estimate the expression at each cluster from leaf to root of the graph. Please note, in the forward pass, there is always one parent for each cluster and thus it’s the 2 Gaussian model case we discussed above. But, one cluster can be the parent cluster for multiple children clusters in backward pass. Whatever the case, we can always use the above weighted average to estimate the expression and variance.

\[
g(\hat{s}) = \text{weightedAverage}_{i=1}^{P_s} g_i(s)
\]

Where \( P_s \) is the number of predictors for cluster \( s \) and \( g_i(s) \) is the estimation of i-th predictor at cluster \( s \). For example, cluster \( s \) have 3 children clusters. Then, we will have 4 predictors in the backward pass for cluster \( s \) (3 from 3 children cluster and 1 from its own observation). Please note the transition offset of each gene for the backward pass has different signs compared with the forward pass. Once we got the forward pass estimation and the backward pass estimation of gene expression at cluster \( s \), the final estimation would be the weighted sum of the estimation from forward and backward passes.

We used a similar strategy to estimate the ‘differentiation time’ for each cluster. As we introduced above, the ‘differentiation time’ was indicated by ‘STA-Similarity To Ancestor’ value. Similarly, we need to define the transition model and emission model for the Kalman Filter.

\[
STA_s = STA_P_s + STEP_{STA} + q
\]
\[
OSTA_s = STA_s + r
\]
\[
g_s \sim N(0, \sigma_q^2)
\]
\[
r_s \sim N(0, \sigma_r^2)
\]

Where \( OSTA_s \) is the measurement of STA for cluster \( s \) (Mean value of STA for all cells within the cluster). \( STEP_{STA} \) denotes the transition offset from parent cluster to cluster \( s \), it was calculated as the average difference of STA between cluster and parent cluster in the graph. \( g_s \) represents the Gaussian process noise and \( r_s \) represents the Gaussian measurement noise at cluster \( s \). With the transition and emission model described above, we can estimate the value and variance of STA for cluster \( s \) using 1 Dimensional Kalman Filter.
In order to use the Kalman Filter on gene expression or STA, we need to know the process noise (Q for gene expression, \( \sigma_q^2 \) for STA) and measurement noise (R for gene expression, \( \sigma_r^2 \) for STA). Here, we do not have the prior knowledge about the process and measurement noise. Therefore, we need to estimate them just based on the measurement. We can follow the definition to estimate the process and measurement noise.

\[
R(g) = \frac{1}{|\text{cells}|} \sum_{i \in \text{cells}} (Y_i(g) - \mu(g))^2
\]  

(40)

\[
\mu(g) = \frac{1}{|\text{cells}|} Y_i(g)
\]  

(41)

\[
Q(g) = \frac{1}{|\text{cells}|} \sum_{i \in \text{cells}} (Y_i(g) - \mu'(g))^2
\]  

(42)

\[
\mu'(g) = \text{Estimation}_{\text{transition}}(g)
\]  

(43)

Where, g represent gene g, \( Y_i(g) \) represents the measurement of gene g at cell i. \( \text{Estimation}_{\text{transition}}(g) \) denotes the estimation of gene g based on the transition from its parent. \( \sigma_q^2 \) and \( \sigma_r^2 \) were estimated similarly.

**Model refinement and cell re-assignments**

Once we learn the initial transition and emission parameters for the model, we can determine a global likelihood based on the assignment of cells to different states in the model.

\[
\log(\text{Likelihood}(c_1, c_2, ..., c_n, A|M)) = \sum_{i=1}^{n} \log P(c_i, s_i|M)
\]  

(45)

\[
= \sum_{i=1}^{n} [\log(P(s_i)P(c_i|s_i))]
\]  

(46)

\[
= \sum_{i=1}^{n} [\log(P(s_i)) + \log(P(\text{STA}_{c_i}|s_i)) + \log(P(G_i|s_i))]
\]  

(47)

\[
= \sum_{i=1}^{n} \{\log(P(s_i)) + \log(P(\text{STA}_{c_i}|s_i)) + \sum_{g_k \in g'} \log(P(g_k|s_i))\}
\]  

(48)

\[
\log(P(s_i)) = \log(\prod_{q \in Q_i} p(q|q_p))
\]  

(49)

\[
= \sum_{q \in Q_i: \text{path to } s_i} \log(q|q_p)
\]  

(50)
Here $n$ is number of all cells, $A$ represents the current assignments of cells to states, $g^i$ is the set of all genes for cell $i$ and $s_i$ is the state to which cell $i$ is assigned. $P(STA_c|s_i)$ is the time probability of $c_i$, which indicates the agreement of STA values between $c_i$ and state $s_i$. $P(g_k|s_i)$ is the expression probability of gene $g_k$, which represents the probability that $g_k$ is from state $s_i$. $Q_i$ is the path from the root node to state (node) $s_i$, including the root node: $P(root|root_{parent}) = P(root|None) = P(root)$. $\log(q|q_p)$ modeled the transition relations and was estimated based on the current assignment: $P(q|q_p) = \frac{|C_q|}{|C_{Pq}|}$. $|C_q|$ is number of cells at state q. $CP_{q_p}$ denotes number of cells, which are from all children states of $q_p$ (parent state of $q$).

When we calculate the probability of expression, we used a mixture model, not just the Gaussian probability. The rationale behinds this is that there are many dropouts for single cell RNA-Seq because of the drawback the current technique. Therefore if the expression of gene is 0, it might be the dropout instead of the real gene expression. We can take this into consideration by utilizing the following mixture model.

$$P(g_k|s_i) = w_{g_k}p_1(g_k|s_i) + (1 - w_{g_k})p_2(g_k|s_i)$$ \hspace{1cm} (51)

$$p_1(g_k|s_i) \sim N(\mu_i(g_k), \sigma_i^2(g_k))$$ \hspace{1cm} (52)

$$p_2(g_k|s_i) = \begin{cases} k, & \text{if } g_k = 0. \\ 0, & \text{otherwise.} \end{cases}$$ \hspace{1cm} (53)

$$p_1(g_k|s_i) = \frac{1}{2\pi\sigma_i^2(g_k)} e^{-\frac{g_k - \mu_i(g_k)}{2\sigma_i^2(g_k)}}$$ \hspace{1cm} (54)

$$\mu_i(g_k) = \frac{[A_i(g_k)\mu_{p_i}(g_k) + B_i(g_k)] \ast \sigma^2_{em,i}(g_k) + g_i(g_k) \ast \sigma^2_{tr,i}(g_k)}{\sigma^2_{em,i}(g_k) + \sigma^2_{tr,i}(g_k)}$$ \hspace{1cm} (55)

$$\sigma_i(g_k) = \frac{\sigma^2_{em,i}(g_k)\sigma^2_{tr,i}(g_k)}{\sigma^2_{tr,i}(g_k) + \sigma^2_{em,i}(g_k)}$$ \hspace{1cm} (56)

$$\sigma^2_{tr,i}(g_k) = A_i^2(g_k) \ast \sigma^2_{p_i}(g_k) + \sigma^2_{pr,i}(g_k)$$ \hspace{1cm} (57)

Where $\mu_i(g_k)$ is the mean of expression of $g_k$ at cluster $i$ and $\sigma^2_i(g_k)$ denotes the variance of gene $g_k$ at cluster $i$. $w_{g_k}$ denotes the weight for Gaussian probability, which will be estimated as the ratio of none-zero cells (non-zero expression for $g_k$) at cluster $s$. $A_i, B_i$ specifies the transition model to state $i$ as described above and $A_i(g_k)$ and $B_i(g_k)$ specify the corresponding transition coefficients for $g_k$. $\mu_i(g_k)$ and $\sigma_i(g_k)$ denotes the mean and variance of $g_k$ at state $i$. $\sigma_{em,i}$ and $\sigma_{pr,i}$ are the measurement noise and process noise for the Kalman Filter, which need to be estimated as discussed above.

Similarly, we can calculate the time probability $P(STA_c|s_i)$

$$\log(P(STA_c|s_i)) = \left[ -\frac{1}{2} \log(2\pi\sigma^2_{STA_c}) - \frac{STA_c - \mu_{STA_c}}{2\sigma^2_{STA_c}} \right] \ast SF$$ \hspace{1cm} (58)

$$\log(P(STA_c|s_i)) = \left[ -\frac{1}{2} \log(2\pi\sigma^2_{STA_c}) - \frac{STA_c - \mu_{STA_c}}{2\sigma^2_{STA_c}} \right] \ast SF$$ \hspace{1cm} (59)

Where $\mu_{STA_c}$ and $\sigma_{STA_c}$ denote the mean and Variance of STA at state $s_i$, which can be estimated similarly as $\mu_i(g_k), \sigma_i(g_k)$ discussed above. $SF = |g^i|$ is a scaling
factor, which is used to scale up the time probability to a comparable level of gene expression probability.

We next attempt to improve the likelihood of the model by refining the model structure (i.e. changing parent - descendant assignments) and reassignment of cells to states in the model. To reassign cells, we compute the maximal probability for cell $c_i$, $P(c_i|s)$ for all possible states $s$ in the model. Specifically we find:

$$Assign(c_i) = \arg \max_s P(c_i, A|M)$$

$$= \arg \max_s P(c_i, s)$$

$$= \arg \max_s P(s)P(STA_i|s)P(G_i|s)$$

$$= \arg \max_s P(s)P(STA_i|s) \prod_{g \in G_i} P(g_k|s)$$

$$= \arg \max_s \log(P(s)) + \log(P(STA_i|s)) + \sum_{g_k \in g^i} \log(P(g_k|s))$$

After re-assigning cells to states, we will refine the model by updating nodes (states) and edges (parent relationship). We remove states that become empty and re-compute the edges (fromNode, toNode, regulating TFs) by updating the parent for each remaining state.

We iterate between cell assignments and model refinement until no changes are observed at the global likelihood. For the lung data studied in this model the number of iterations is 4. The interaction for mouse embryonic fibroblasts reprogramming single cells is 6.
Supplemental Results

Performance comparison on additional datasets

To show the performance of our method, we have compared our method to DPT [14], which has been reported to have a very good performance in different types of datasets, including some complex ones (with thousands of single cells) [15][16]. We first compared our method to DPT on the Treutlein et al. lung single cell dataset. As shown in Figure 4 panel a, our method is having a much better cell assignment: For our method, 10/11 Club cells were assigned to cluster E2.18.2, 3/3 ciliated cells were assigned to cluster E1.18.3, 12/12 AT2 cells were assigned to cluster E13.18.1. 28/41 AT1 cells were assigned to E2.18.0 and 13/41 AT1 cells were assigned to E3.18.1. On the other hand, for DPT, 2/1/1/7 out of 11 Club cells were assigned to cluster 0/1/2/3 respectively, 3 out of 3 ciliated cells were assigned to cluster 1, 18/4/2/17 out of 41 AT1 cells were assigned to cluster 0/1/2/3 respectively, 7/1/4/0 out of 12 AT2 cells were assigned to cluster 0/1/2/3 respectively. Actually, the terminal cell types such as AT1, AT2, Club are pretty spread out in the DPT 2D projection, which also indicates that the DPT is hard to cluster those cells properly. We also compared our method to DPT on the Treutlein et al. mouse embryonic fibroblasts reprogramming single cell dataset. Similarly, as shown in Figure 4 panel b and c, the cell assignment of our method is more consistent with the true assignment compared with the DPT assignment. DPT is not able to distinguish d2,d5 from MEF cells. There are 66 d2 and d5 cells (induced and intermediate cells) mixed with MEF cells. While in our model, those d2 and d5 cells were clearly separated. Besides, from the DPT model, Neuron cells are closer to d2 cells than d5 cells, which is not supported by the original study [17]. Besides the cell assignment, our model is also able to predict the differentiation trajectories and key factors driving the differentiation process, which can’t be done by other methods such as DPT and Monocle. Please refer to the result section for supports of the predicted models and regulators. We have also compared our model with DPT on Olsson et al. mouse bone marrow dataset [18] (Supplemental Figure 16), Treutlein et al. mouse embryonic fibroblasts reprogramming dataset setting 1 [17] (Supplemental Figure 19) and mouse lung single cell datasets from both Treutlein et al and Du et al. [19][20] (Supplemental Figure 21).

Performance of the alternative initial clustering (PCA+KMeans) strategy

scdiff can be divided into two parts, the first is the initialization and the second is the iterative refinement. The second part scales linearly with the number of cells and is usually very fast so it can handle tens of thousands of cells. However, the Spectral Clustering based initialization method is cubic in the number of cells and indeed does not scale well. Therefore, to address the issue of scalability, we added a new option for initialization using a clustering method based on PCA and K-means. Users can specify which of the methods to use by using “-l” option in the scdiff software. Please refer to the scdiff GitHub page [https://github.com/phoenixding/scdiff] for more details.

We have also tested this option both, to determine run time and to test the impact.
on the learned model. Up to 2500 cells, we can use scdiff with the spectral option (it runs within 1 hour on a 8G RAM, Intel(R) Xeon(R) CPU E3-1225 v5@3.3GHz machine). For larger dataset, we recommend the use of the scdiff “-l” option. Run times for for PCA+Kmeans initialization are presented in Supplemental Figure 23. As can be seen, using the “-l” option, leads to a runtime of less than 1 hour for 15000 cells.

To test the performance of the method with the new initialization strategy on a large dataset, we created 100 version of each of the initial cells in the Treutlein et al. 2014 lung dataset by adding random noise to all genes in the cell (leading to roughly 15K cells). The resulting model learned from this data is presented in Supplemental Figure 24. This model has more states than the original model (each with many more cells given the size of the input). However, it was still able to correctly assign the earlier time point (E14 and E16), to group BPs with AT1 and AT2 and to separate these two cell types. It was also able to identify a unique state for ciliated cells. The Club cell (Clara) is also present though it now contains a (small) set of ciliated cells as well (The ciliated cells are close to club cells). Overall, the model is quite similar to the original model and quite accurate from the biological point of view.

Analysis of gene filtering strategy

As we discussed in the main text, we filtered all the genes with zero variances as suggested in [19]. This reduced the number of genes in the analysis from 23K to 15K. We used all the remaining 15K genes in this study. We also tested a stricter filtering criterion in which we removed all genes that are not present in at least 20% of all cells (note that this can lead to missing key genes for cell types that account for less than 20% of total cells, including Club (Clara) and ciliated, which is why we kept all the non-zero variance genes). After the 20% filtering, 5384 genes are left. We re-run scdiff using only those genes and the results can be found in Supplemental Figure 22. As can be seen, the results are very similar to the results obtained using the full set of genes with the only difference observed for the placement of Club (clara) cells which are now branching later than in the model based on the entire set of genes.
Supplemental Tables

Supplemental Table 1: GO term enrichment analysis (p-value) of predicted regulating TFs

| Edge                      | GO:0060487 (lung epithelial cell differentiation) | GO:0042127 (Regulation of cell proliferation) | GO:0050678 (Regulation of epithelial cell proliferation) |
|---------------------------|--------------------------------------------------|-------------------------------------------------|---------------------------------------------------|
| E0 \(\rightarrow\) E1     | \(8.40 \times 10^{-2}\)                        | \(3.46 \times 10^{-2}\)                        | \(6.93 \times 10^{-4}\)                          |
| E0 \(\rightarrow\) E1     | \(8.65 \times 10^{-2}\)                        | \(7.11 \times 10^{-3}\)                        | \(5.83 \times 10^{-5}\)                          |
| E0 \(\rightarrow\) E1     | \(3.94 \times 10^{-2}\)                        | \(1.09 \times 10^{-2}\)                        | \(1.99 \times 10^{-3}\)                          |
| E0 \(\rightarrow\) E1     | \(7.95 \times 10^{-2}\)                        | \(4.10 \times 10^{-3}\)                        | \(1.18 \times 10^{-3}\)                          |
| E0 \(\rightarrow\) E1     | \(9.99 \times 10^{-3}\)                        | \(8.07 \times 10^{-3}\)                        | \(3.99 \times 10^{-4}\)                          |
| E0 \(\rightarrow\) E1     | \(3.98 \times 10^{-2}\)                        | \(4.05 \times 10^{-3}\)                        | \(1.95 \times 10^{-3}\)                          |
| E0 \(\rightarrow\) E1     | \(3.98 \times 10^{-2}\)                        | \(4.05 \times 10^{-3}\)                        | \(1.95 \times 10^{-3}\)                          |
| E0 \(\rightarrow\) E1     | NA                                               | \(1.00 \times 10^{-3}\)                        | \(3.19 \times 10^{-3}\)                          |
| E0 \(\rightarrow\) E1     | \(2.02 \times 10^{-2}\)                        | \(1.26 \times 10^{-3}\)                        | \(7.39 \times 10^{-4}\)                          |
| E0 \(\rightarrow\) E1     | \(1.30 \times 10^{-2}\)                        | \(1.25 \times 10^{-3}\)                        | \(7.39 \times 10^{-4}\)                          |
| E0 \(\rightarrow\) E1     | NA                                               | \(1.00 \times 10^{-3}\)                        | \(3.19 \times 10^{-3}\)                          |
| E0 \(\rightarrow\) E1     | \(2.02 \times 10^{-2}\)                        | \(1.26 \times 10^{-3}\)                        | \(7.39 \times 10^{-4}\)                          |
| E0 \(\rightarrow\) E1     | \(1.30 \times 10^{-2}\)                        | \(1.25 \times 10^{-3}\)                        | \(7.39 \times 10^{-4}\)                          |

Supplemental Table 2: Predicted TFs for Du et al lung single cell dataset

HSF1, HSF2, MAFK, NKX2-1, MAFF, MAFG, ETS2, GATA5, RARG, NFE2L1, MYC, BRCA1, TFE3, SP2, SP3, CREB1, GATA6, SETD2, GATA3, UBE4A, MAX, MAZ, RFC1, TCF7L2, RFC3, AHR, ELK1, ELK4, APC, NR3C1, NFATC3, POU2F1, CDC5L, HIF1A, CNTN2, IL6, ZBTB7A, FOXA1, SOX11, SOX12, EGR1, JUN, MEIS1, NF1, NFIL3, GTF2I, NR1H2, UBP1, TCF4, FOXJ2, TCF3, FOS, TOPORS, DSP, PAX6, DBP, TBPL1, IRF6, TBP, MEF2A, CREM, HLF, TFDP1, YY1, CLOC, SMAD4, FOSB, ATM, SMAD2, STAT6, STAT3, NR2F2, USF2, USF1, FOXL2, NFKB1, GABPB2, E2F7, GABPB1, NRF, HMGAL, RXRA, EP300, ARID5B, ARNT, MTM1, SREBF1, RARB, STAT5A, SREBF1, GABPA, JUNB, SOX8, SOX9, GTF2A1, GTF2A2, SOX2, TEAD1, ELF2, SOX4, XBP1, CEBPB, CEBPA, CEBPG, CEBPD, RB1, HNF1B, ETS1, SOX5, POU3F1, ESRRG, PLAU, E4F1, POU4F1, HES1, PBX1, NFIC, E2F5, E2F4, E2F3, E2F2, E2F1, BACH2, ATF4, ATF6, ATF1, BACH1, ATF3, ATF2

Supplemental Table 3: Cell assignment on Du et al lung single cell data with simulated dropouts

| Dropout rate | 5% cell assignment agreement | 10% cell assignment agreement | 20% cell assignment agreement | 30% cell assignment agreement | 40% cell assignment agreement | 60% cell assignment agreement | 80% cell assignment agreement |
|--------------|-----------------------------|-------------------------------|--------------------------------|-------------------------------|--------------------------------|-------------------------------|-------------------------------|
|              | 174 /174=99.43%             | 162 /162=93.1%               | 147 /147=84.48%               | 132 /132=75.86%               | 121 /121=69.54%               | 119 /119=68.31%               | 107 /107=61.49%               |
Supplemental Table 4: Cell assignment on Du et al lung single cell data with simulated Gaussian Noises

| Noise Level | 5%       | 10%      | 20%       |
|-------------|----------|----------|-----------|
| cell assignment agreement | 174/174=100% | 174/174=100% | 171/174=98.28% |

Supplemental Table 5: Cell assignment on different randomly sampling of Du et al lung single cell dataset

|                | 80%       | 82.5%     | 85%       | 87.5%     | 90%       |
|----------------|-----------|-----------|-----------|-----------|-----------|
| cell assignment agreement | 126/139=90.6% | 141/143=98.6% | 146/147=99.3% | 146/151=96.7% | 156/156=100% |

Supplemental Table 6: Hyper-geometric test between predicted DE genes for each edge and NKX2-1 targets from the CHIP-CHIP experiment

| Edge            | p-value |
|-----------------|---------|
| E0_14_0 → E1_16_3 | 0.363   |
| E0_14_0 → E1_16_1 | 0.510   |
| E0_14_0 → E1_16_2 | 0.00695 |
| E0_14_0 → E1_16_0 | 0.583   |
| E1_16_1 → E2_16_4 | 0.0683  |
| E1_16_1 → E2_18_1 | 0.259   |
| E1_16_1 → E2_18_4 | 0.093   |
| E1_16_4 → E3_18_2 | 0.653   |
| E1_16_4 → E3_18_0 | 0.780   |
| E1_18_0 → E4_18_3 | 0.0783  |
Supplemental Figure 1: Demonstration of scdiff procedure using 2014 Treutlein et al. lung single cell dataset.
Supplemental Figure 2: **The expression of house-keeping gene ACTB.** The datasets are largely compatible. ACTB was reported to be the most stable house-keeping gene in [19] and our analysis shows that the expression of ACTB is stable across single cells at different time points, including the new E16.5 cell (E16_JW).
Supplemental Figure 3: Predicted model on Du et al. lung single cell data with 20% simulated dropouts
Supplemental Figure 4: Predicted model on Du et al. lung single cell data with 30% simulated dropouts
Supplemental Figure 5: Predicted model on Du et al. lung single cell data with 40% simulated dropouts
Supplemental Figure 6: Predicted model on Du et al. lung single cell data with 60% simulated dropouts
Supplemental Figure 7: Predicted model on Du et al. lung single cell data with 80% simulated dropouts
Supplemental Figure 8: Predicted model on Du et al. lung single cell data with 20% simulated Gaussian noises. $expression_{\text{noise}} = expression_{\text{original}} \cdot (1 + \text{NoiseLevel} \cdot \text{GaussianNoise}(0,1))$, where GaussianNoise(0,1) is a random value drawn from Gaussian distribution with $\mu = 0$ and $\sigma = 1$. 
Supplemental Figure 9: Predicted model on 80% randomly sampling of Du et al. lung single cell dataset.
Supplemental Figure 10: Predicted model on 82.5% randomly sampling of Du et al. lung single cell dataset.
Supplemental Figure 11: Predicted model on 85% randomly sampling of Du et al. lung single cell dataset.
Supplemental Figure 12: Predicted model on 87.5% randomly sampling of Du et al. lung single cell dataset.
Supplemental Figure 13: Predicted model on 90% randomly sampling of Du et al. lung single cell dataset.
Supplemental Figure 14: Predicted model on Du et al. lung single cell dataset with STA cutoff 0.1.
Supplemental Figure 15: Predicted model on Du et al. lung single cell dataset with STA cutoff 0.01.
Supplemental Figure 16: **Comparison of DPT and our model on Olsson et al. mouse bone marrow single cell dataset.** Top left: true cell assignment. Top right: cell assignment by DPT. Bottom: predicted results by our model.
Supplemental Figure 17: Predicted model on Du et al. lung single cell dataset without using TF information.
Supplemental Figure 18: Decreasing expression of SOX9 and phosphohistone H3 with advancing gestation. SOX9 was expressed highly in peripheral buds (acinar buds) (E16.5) and decreased dramatically as sacculcation proceeded at E18.5. Cell proliferation was marked by phosphohistone H3 in epithelial cells at E16.5, and was marked decreased at E18.5. Scale bars are 100 microns. Patterns of staining are representative of n=2 at each gestational age.
Supplemental Figure 19: Comparison of DPT and our model on 2016 Treutlein et al. mouse embryonic fibroblasts reprogramming single cell setting 1 (treated with ASCL1) Top left: true cell assignment. Top right: cell assignment by DPT. Bottom: predicted results by our model. DPT is not able to distinguish d2 and d5 cells.
Supplemental Figure 20: The comparison between the predicted model and the ‘studied’ model by Olsson et al.
Supplemental Figure 21: Comparison of DPT and our model on lung single cell dataset from both Treutlein et al. and Du et al. Top Left: true cell assignment. Top right: cell assignment by DPT. Bottom: predicted results by our model. DPT is not able to distinguish PAT1/PAT2/PBP/NAT1/NAT2 those precursor cells. In our model, earlier precursors such as PAT1/PAT2/PBP cells were separated from later precursors NAT1/NAT2 cells.
Supplemental Figure 22: Predicted model on Treutlein et al. dataset after filtering out genes with zero variance or expressing in less than 20% of cells.
Supplemental Figure 23: **running time performance of scdiff (-l option)**
Supplemental Figure 24: Predicted model of scdiff (-l option) on a dataset with 15k cells
References

[1] A. Zimek, E. Schubert, and H.-P. Kriegel, “A survey on unsupervised outlier
detection in high-dimensional numerical data,” *Statistical Analysis and Data
Mining*, vol. 5, no. 5, pp. 363–387, 2012.

[2] J. Hardin, A. Mitani, L. Hicks, and B. VanKoten, “A robust measure of
correlation between two genes on a microarray,” *BMC bioinformatics*, vol. 8,
o. 1, p. 1, 2007.

[3] S. Wu, X. Feng, and W. Zhou, “Spectral clustering of high-dimensional
data exploiting sparse representation vectors,” *Neurocomputing*, vol. 135,
pp. 229–239, 2014.

[4] H. Zare, P. Shosohtari, A. Gupta, and R. R. Brinkman, “Data reduction for
spectral clustering to analyze high throughput flow cytometry data,” *BMC
bioinformatics*, vol. 11, no. 1, p. 1, 2010.

[5] M. Hedegaard, C. Matthäus, S. Hassing, C. Krafft, M. Diem, and J. Popp,
“Spectral unmixing and clustering algorithms for assessment of single cells by
raman microscopic imaging,” *Theoretical Chemistry Accounts*, vol. 130,
o. 4-6, pp. 1249–1260, 2011.

[6] A. Ocone, L. Haghverdi, N. S. Mueller, and F. J. Theis, “Reconstructing
gene regulatory dynamics from high-dimensional single-cell snapshot data,” *Bioinformatics*, vol. 31, no. 12, pp. i89–i96, 2015.

[7] Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and
M. Ouimet, “Spectral dimensionality reduction,” in *Feature Extraction*,
pp. 519–550, Springer, 2006.

[8] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation
and validation of cluster analysis,” *Journal of computational and applied
mathematics*, vol. 20, pp. 53–65, 1987.

[9] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE
transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227,
1979.

[10] H. Akaike, “Information theory and an extension of the maximum likelihood
principle,” in *Selected Papers of Hirotugu Akaike*, pp. 199–213, Springer, 1998.

[11] A. I. Goranov, M. Cook, M. Ricicova, G. Ben-Ari, C. Gonzalez, C. Hansen,
M. Tyers, and A. Amon, “The rate of cell growth is governed by cell cycle
stage,” *Genes & development*, vol. 23, no. 12, pp. 1408–1422, 2009.

[12] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the
Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] L. Haghverdi, M. Buettner, F. A. Wolf, F. Buettner, and F. J. Theis, “Diffusion pseudotime robustly reconstructs lineage branching,” *Nature Methods*, vol. 13, no. 10, pp. 845–848, 2016.

[15] S. Nestorowa, F. K. Hamey, B. P. Sala, E. Diamanti, M. Shepherd, E. Laurenti, N. K. Wilson, D. G. Kent, and B. Götgens, “A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation,” *Blood*, vol. 128, no. 8, pp. e20–e31, 2016.

[16] L. Velten, S. F. Haas, S. Raffel, S. Blaszkiewicz, S. Islam, B. P. Hennig, C. Hirche, C. Lutz, E. C. Buss, D. Nowak, et al., “Human haematopoietic stem cell lineage commitment is a continuous process,” *Nature Cell Biology*, vol. 19, no. 4, pp. 271–281, 2017.

[17] B. Treutlein, Q. Y. Lee, J. G. Camp, M. Mall, W. Koh, S. A. M. Shariati, S. Sim, N. F. Neff, J. M. Skotheim, M. Wernig, and S. R. Quake, “Dissecting direct reprogramming from fibroblast to neuron using single-cell RNA-seq,” *Nature*, pp. 1–15, 2016.

[18] A. Olsson, M. Venkatasubramanian, V. K. Chaudhri, B. J. Aronow, N. Salomonis, H. Singh, and H. L. Grimes, “Single-cell analysis of mixed-lineage states leading to a binary cell fate choice,” *Nature*, 2016.

[19] B. Treutlein, D. G. Brownfield, A. R. Wu, N. F. Neff, G. L. Mantalas, F. H. Espinoza, T. J. Desai, M. A. Krasnow, and S. R. Quake, “Reconstructing lineage hierarchies of the distal lung epithelium using single-cell rna-seq,” *Nature*, vol. 509, no. 7500, pp. 371–375, 2014.

[20] Y. Du, M. Guo, J. A. Whitsett, and Y. Xu, “lunggens: a web-based tool for mapping single-cell gene expression in the developing lung,” *Thorax*, vol. 70, no. 11, pp. 1092–1094, 2015.