Probabilistic Single Cell Lineage Tracing

Chieh Lin

March 2020
CMU-ML-20-100

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Ziv Bar-Joseph, Chair
Roni Rosenfeld
Jian Ma
Darrell Kotton
Killian Hurley

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was supported by National Institutes of Health under grant numbers 1R01GM122096, U01 HL122626, and OT2OD026682. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Time-series single-cell RNA-Seq, Graphical models, Regulatory networks, Developmental trajectories, Maximum likelihood, Bayesian hierarchical clustering

Abstract

Cell lineage tracing is a long-standing open problem in biology. To solve this problem, new technologies that can profile single-cells have been introduced in the last decade. Currently, studies attempt to construct lineage relationships using time-series single-cell RNA sequencing (scRNA-Seq) data or by utilizing artificial mutations for marking cells. The former studies rely on pseudo-time ordering which suffers from shortcomings that can impact their accuracy. The latter often apply phylogeny-based methods which often lead to hundreds of candidate trees. There is no current method to combine single-cell lineage trees from different individuals of the same organism to reconstruct a single invariant lineage for the same species.

In this thesis, we present a set of machine learning models that focus on reconstructing single-cell lineages. We developed a probabilistic model based on Continuous-State Hidden Markov Model (CSHMM) to reconstruct trajectories and branchings from time series scRNA-Seq data. The model is then extended by learning the dynamics of regulatory interactions that take place during the process being sutdied (CSHMM-TF). We next present a method that integrates sequence and expression data, In addition, we developed LinTIMaT, a statistical model for reconstructing single-cell lineage trees using both artificial mutations and scRNA-Seq data and for constructing a general invariant lineage tree from multiple cell lineage trees of the same species. Finally, we apply CSHMM to a new dataset and show that it is capable of reconstructing lineage relationships and provides important novel insights for studying lung development.

Acknowledgments

I would like to thank my advisor Ziv Bar-Joseph first for the help and mentorship I received from him. Ziv's insight and guidance in both machine learning and biology help me a lot when deciding my research direction. Ziv also makes me understand that how the results of our research may help biologists and this will be my lifetime treasure.

I would like to thank my committee members, Roni Rosenfeld, Jian Ma, Darrell Kotton, and Killian Hurley for being part of my committee and providing feedback on my thesis. I would also like to thank my co-authors Siddhartha Jain, Jun Ding and Hamim Zafar, without them and Ziv's help my PhD would not be possible.

I would also like to thank all the members from Ziv's System Biology Group. I have benefited a lot from the conversations and advice from Jun Ding, Matthew Ruffalo, Hamim Zafar, Ye Yuan, Jose Lugo-Martinez, Dongshunyi (Dora) Li, Amir Alavi, Sabrina Rashid, and Siddhartha (Sid) Jain.

Lastly, I would like to thank my partner Po-Wei Wang, my sisters Hsiu (Ringo) Lin and Min (Koshi) Lin, and my parents Mao-Chang (Clement) Lin and Shu-Ling (Shirine) Tu for their love and support for my graduate study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Understanding cellular lineage development is a long-standing and open problem in biology. Elucidating the lineage relationships among the diverse cell types can provide key insights into the fundamental processes underlying normal tissue development as well as valuable information on what goes wrong in developmental diseases [100, 186, 223].

In recent years, new technologies for single-cell RNA sequencing (scRNA-Seq) have been introduced. This has the potential to greatly increase the our ability to model biological systems compared to prior methods that rely on bulk RNA-Seq data. Using single cell expression data (scRNA-Seq) researchers can better identify cell specific pathways and genes which are often missed when profiling cell mixtures. Single cell analysis of developmental programs, various tissues and perturbations has already identified new cell types, new pathways and new marker genes for a variety of biological systems and conditions [180, 206, 208]. Current studies often profile thousands of individual cells in a single experiment [79, 83, 149] enabling researchers to curate catalogs of cellular identities across tissues [85, 149, 213].

Several scRNA-Seq studies focus on time-series data, most notably during development of various organs and systems [180, 185, 208, 209]. These are often used to construct lineage relationships at the single cell level. In all cases cells are usually sampled at specific intervals, RNA is extracted and sequenced, and expression profiles are determined. Using these expression profiles researchers then aim to reconstruct branching and cell fate decision models that underlie developmental processes. While useful, to date it has been challenging to use this technology to trace the expression of a cell at different times because cells are fully consumed when we measure their expression. A key question that emerges in time-series single-cell studies is the ability to connect different cell types over time by their expression profiles. Unlike experiments that profile bulk samples (or population of cells), in which a sample at time point $t+1$ is assumed to arise from the sample at time t [11], in single cell studies it is not always clear what cell type in time t led to a cell being profiled in time $t+1$. Since scRNA-Seq studies fully consume the cell (which effectively makes it a snapshot), it is not possible to trace it over time which make it difficult to connect progenitor cells to their descendents, or to follow the response of specific cell types over time. Another problem with this data is that cells collected at the same

time point are not completely synchronized; that is, the cells measured at the same time point could be at different developmental stages because of individual differences so cells measured at a specific time point may be more similar to cells at other time point (in terms of their perceived developmental or differentiation time) which may require the reassignment of cells between the measured time points.

1.1.1 Single-cell pseudo time orderings

To address these issues, a number of methods, often referred to as pseudotime inference methods, have been proposed [151, 159, 179, 206]. These methods order cells along a transcriptomic trajectory in embedded space such that cells that are close in that space are also assumed to be close in terms of their biological states. By tracing paths and trajectories of pseudotemporally ordered cells these methods determine the set of states leading from the starting point to the (often differentiated) final cell fate. Pseudotime and other tools developed for the analysis of time-series scRNA-Seq data can be largely divided based on the method they used (deterministic or probabilistic) and the representation they provide (continuous vs. discrete cell assignments).

The first (deterministic methods) utilize dimensionality reduction (reducing the input data to 2D or 3D for visualization purposes). Next, these methods attempt to identify a structure (either a graph or variant of a Minimum Spanning Tree (MST)) and use it to infer ordering for the cells. Examples methods using variants of this strategy are DPT [84], scTDA [159], PCA analysis [208], Monocle 2 [206], Wanderlust [13]), GPLVM [120, 156], Slingshot [189], and PAGA [222]. The advantage of such methods is that the resulting cell trajectories can be easily visualized since the dimension is reduced to 2D or 3D. The disadvantage is also obvious: a lot of information is lost in the process of dimension reduction since thousands of genes are reduced to 2 or 3 dimensions. Another problem of these methods is that their reconstructed lineage trajectories are highly dependent on what dimensionality reduction method they use. For example, one method may work on reduced t-SNE dimension but does not work well on PCA dimension. Thus, how to choose the correct dimensionality reduction method remains another problem to solve. Also, some of these methods cannot infer more than two branches in the trajectory which is often not enough for developmental and other studies.

The second type of methods (probabilistic methods) usually construct a probabilistic model of tree-like graph structure with (usually small number of) discrete states that cells can be assigned to and determine trajectories based on the graph structure. Each state is associated with an emission probability and cells are assigned to one of the discrete states based on maximum likelihood. Examples that use this strategy include SCUBA[124], TASIC [155], and SCDIFF [51]. One possible advantage of these methods is that they can relatively optimize their model parameters and cell assignments iteratively with Expectation-Maximization methods to improve the likelihood of their model. Besides, the methods can run on the full dimension of the dataset instead of reduced dimension. The disadvantage is that the small number of discrete states cannot account for possibly large number of biological states so the models are not able to generate continuous trajectory of cells. The model can forces cells that can be pretty distant in the time they represent to the same state. Also, cells that have similar biological states may be separated to different discrete states so that they are distant in the resulting model.

1.1.2 Regulator information for single cell pseudo time trajectories

Identifying the activation time for key regulators or transcription factors (TF) for each development stages is also an important problem. Among the methods discussed above, some can identify regulatory relationships (TF-DNA or TF-target relationships) between transcription factors and target genes. However, most of them only perform such identification as a post-analysis on their model parameters [42, 81, 207] so the integration of TFs with the scRNA-Seq data has not reached its full potential. This make it hard to utilize the information for improving model reconstruction and assignments. Only a few [51] utilize the identified TF information to further iteratively improve the model and TF identification. However, these methods use a discrete state model in which TFs can only be assigned to a specific (pre-defined) time. This makes it hard to identify the exact activation time of these TFs, to infer combinatorial activity of TFs and the dynamics of TF complexes assembly.

1.1.3 Artificial genetic marker technologies for lineage reconstruction

Traditionally, heritable markers have been utilized for prospective lineage tracing by first introducing them in a cell and then using them to track its descendants [100]. Such studies resorted to using diverse markers such as viral DNA barcodes [137], fluorescent proteins [12], mobile transposable elements [194], *Cre*-mediated tissue-specific recombination [146] and more. Other methods relied on retrospective lineage tracing by using naturally occurring somatic mutations [99, 236], microsatellite repeats [69] or epigenetic markers [132]. While these approaches provided valuable insights, they are often limited to a small number of markers and cells and due to the lack of coupled gene expression information, they cannot characterize the diverse cellular identities of the tracked cells and their relation to the lineage branching [223].

Very recently, new experimental techniques that simultaneously recover transcriptomic profiles and genetic lineage markers from the same cell have been introduced [4, 154, 187]. One of the earliest methods using such approach is scGESTALT [154] which combines the CRISPR-Cas9-based lineage tracing method termed GESTALT [129] with droplet-based single-cell transcriptomic profiling. scGESTALT inserts Cas9-induced stochastic (random) mutations to a genomic CRISPR barcode array at multiple time points. Next, the edited barcodes are then sequenced and used for reconstructing a lineage tree based on phylogeny-based methods such as Maximum Parsimony (MP) criterion [63]. Cell types are independently inferred based on scRNA-seq data. Another method is ScarTrace [4], which utilizes identical target sites located on separate transgenes for introducing CRISPR-Cas9 mutations followed by SORT-seq sequencing to capture the transcriptome. Lineage trees are then reconstructed by using the Maximum Parsimony principle on the mutation data. While these and similar methods have been successfully applied to a number of organisms [154, 187], they encompass several computational challenges. First, the random mutation data used for reconstructing the MP lineage is noisy and often saturated making it difficult to separate different cell types, especially at later stages. Even though expression information is collected for all genes in each cell, to date the reconstruction of the lineage tree solely depends on the stochastic Cas9-induced mutations. As a result, the resulting lineage tree sometimes fails to separate different types of cells and places similar cell types on distant branches. It might also build a lineage tree with unnecessary branching points

because the mutations are randomly introduced and this might separate the cells that belongs to the same biological stage. Further, hundreds of possibly very different tree topologies can have similar parsimony score based on mutations making the reconstruction more challenging.

1.1.4 Invariant lineage tree for single-cell lineage trees

Another challenge of single cell lineage tracing is that the resulting lineage trees differ between individuals of the same species. In addition, the random nature of the induced mutations makes it impossible to directly combine mutation data from multiple individuals for inferring a single invariant lineage tree based on multiple experiments. In phylogenetics, a common way to solve this problem is to use consensus tree methods to build lineage trees based on the percentage of agreement of clades. However, this cannot be applied here because the methods require that the number/label of leaves be exactly the same for different lineage trees. This does hold for single cell lineage trees because the number/population of sampled cells is different, and, as mentioned above, there is no way to map the cells in different individuals based on the mutations alone.

1.2 Thesis goals

In this thesis, we propose the following to enable to reconstruction of models that reconstruct developmental trajectories and trees:

1. **Develop a probabilistic model on which cells can be assigned continuously**
   We propose a pseudo-time method that orders cells for scRNA-Seq studies which combines the advantages of both dimension-reduction-based methods and probabilistic methods and minimizes the disadvantages of both. We developed the model based on Continuous-State Hidden Markov Model (CSHMM) and published the paper "Using Continuous-state Hidden Markov Models (CSHMMs) to model time-series scRNA-Seq data and reconstruct continuous cell trajectories" on *Bioinformatics*. See Chapter 2 for details.
2. **Infer the activation time for TF on the continuous cell trajectories**
   We propose to extend the CSHMM formulation to CSHMM-TF which can be used to determine the regulators and their time of activation for each of the reconstructed models. We publish the paper "Inferring the continuous TF activation time for the reconstructed continuous cell trajectories (CSHMM-TF)" on *PLOS Computational Biology*. See Chapter 3 for details.
3. **Use both CRISPR-Cas9 gene modification data and scRNA-Seq data to build a better lineage tracing model, and Develop a method for constructing invariant lineage tree from single cell lineage trees of different individuals**
   We propose a new statistical model, LinTIMaT, for reconstructing cell lineages using a maximum-likelihood framework by integrating both mutation and expression data. We also propose to extend this method to enable learning an invariant lineage tree for a species. We have finished the paper "Single-cell Lineage Tracing by Integrating CRISPR-Cas9 Mutations with Transcriptomic Data" and the paper is under revision in *Nature Communica-*

tions. See Chapter 4 for details.

4. **Apply CSHMM to a new biological dataset**
   Finally, apply CSHMM to a newly-generated lung developmental time-series scRNA-seq dataset to improve the protocol for differentiating pluripotent stem cells (PSCs) to lung cells. The reconstructed continuous cell trajectories from CSHMM are used to identify key factors that result in different cell fates. CSHMM was used to predict the precise time that the cell fates diverge and can help biologists decide the best time for interventions in the protocol. We publish the paper "Reconstructed Single-Cell Fate Trajectories Define Lineage Plasticity Windows during Differentiation of Human PSC-Derived Distal Lung Progenitors" on *Cell Stem Cell*. See Chapter 5 for details.

1.3 Introduction to relevant biological technologies and concepts

While the computational methods used in this thesis are discussed in details in each of the chapters, the work relies on some key biological concepts and on data from a few recent experimental methods. For completeness, we first introduce these concepts and methods. Later, when presenting the actual work in each of the chapters, we mention which data we used and how.

1.3.1 Single-Cell RNA-Sequencing (scRNA-Seq)

Single-Cell RNA-Sequencing, first presented in [200], is a method for profiling the quantity of RNAs from tens of thousands of genes in single cells. Overall, the experimental protocols for scRNA-seq are similar to the protocols for bulk RNA-seq, except that an isolation step is needed to be performed for scRNA-Seq data (Figure 1.1). For time-series scRNA-Seq data, a set of time points to be profiled is pre-determined. For each time point, the gene expression for cells of interest are measured. Note that the cells need to be consumed in order to get their gene expression so we cannot trace the same cell at different time points. Currently, profiling tens of thousands of single cells is possible. The details of strategies and challenges of analyzing single cell data will not be covered here but is reviewed in [235]. All of the gene expression data used in this thesis are scRNA-Seq or time-series scRNA-Seq (time-series scRNA-Seq in Chapter 2, 3 and 5, scRNA-Seq in Chapter 4).

1.3.2 Gene Ontology (GO) Analysis

Gene ontology (GO) associate genes with their biological functions [7], including biological process and cellular component. GO enrichment analysis for a group of genes is a useful way to check the biological meaning for the genes. This method is used to validate the results of our studies in this thesis (See Chapter 2-4). See Figure 1.2 for an example gene ontology.

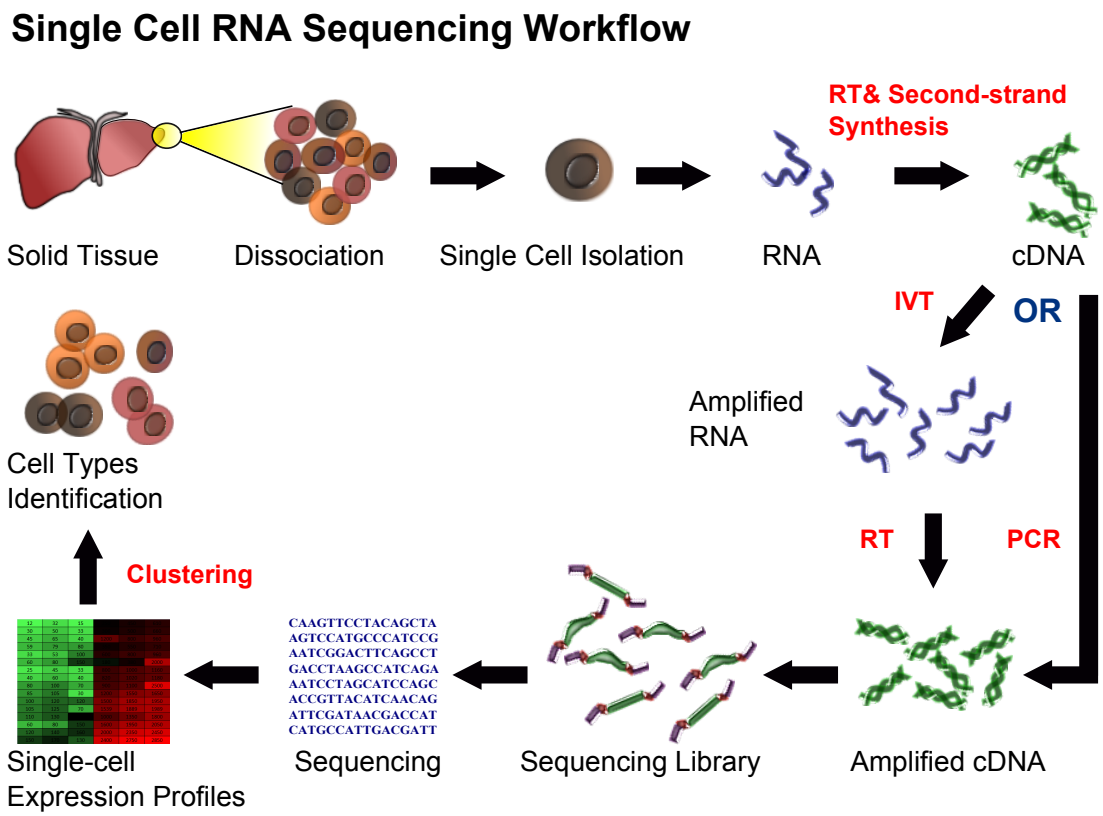

Figure 1.1: Current single-cell RNA sequencing protocols mostly involves the following steps: isolation of single cell and RNA, reverse transcription (RT), amplification, library generation and sequencing. Figure taken from Wikipedia (`https://en.wikipedia.org/wiki/Single_cell_sequencing`).

1.3.3 Transcription-Factors and target gene interaction (TF-DNA or TF-target interactions)

Transcription factors (TF) are proteins that bind to specific DNA sequences and regulate transcription. Each TF activates or represses the transcription of a specific set of genes if the TF binds to the DNA location related to the genes. It has been a challenging task to identify the protein-DNA relationships for an organism. In this thesis we use the information of potential targets of a set of transcription factors for human and mouse [59, 175] in Chapter 3.

This information is used to identify potential key regulators for each developmental processes. The details of how this data is obtained is described in [175]. Briefly, this data is constructed from 3 parts. In the first part, the human ChIP-Seqencing data is downloaded from ENCODE [40]. This data contain aggregated binding peaks for 148 human TFs across diverse cell lines. For each human gene, all the TFs that have transcription start sites near the gene were considered to regulate the gene. For the second part, ranked human PWM-gene predictions were obtained from [60] and each PWM was mapped to correspond TFs by using TRANSFAC [127]

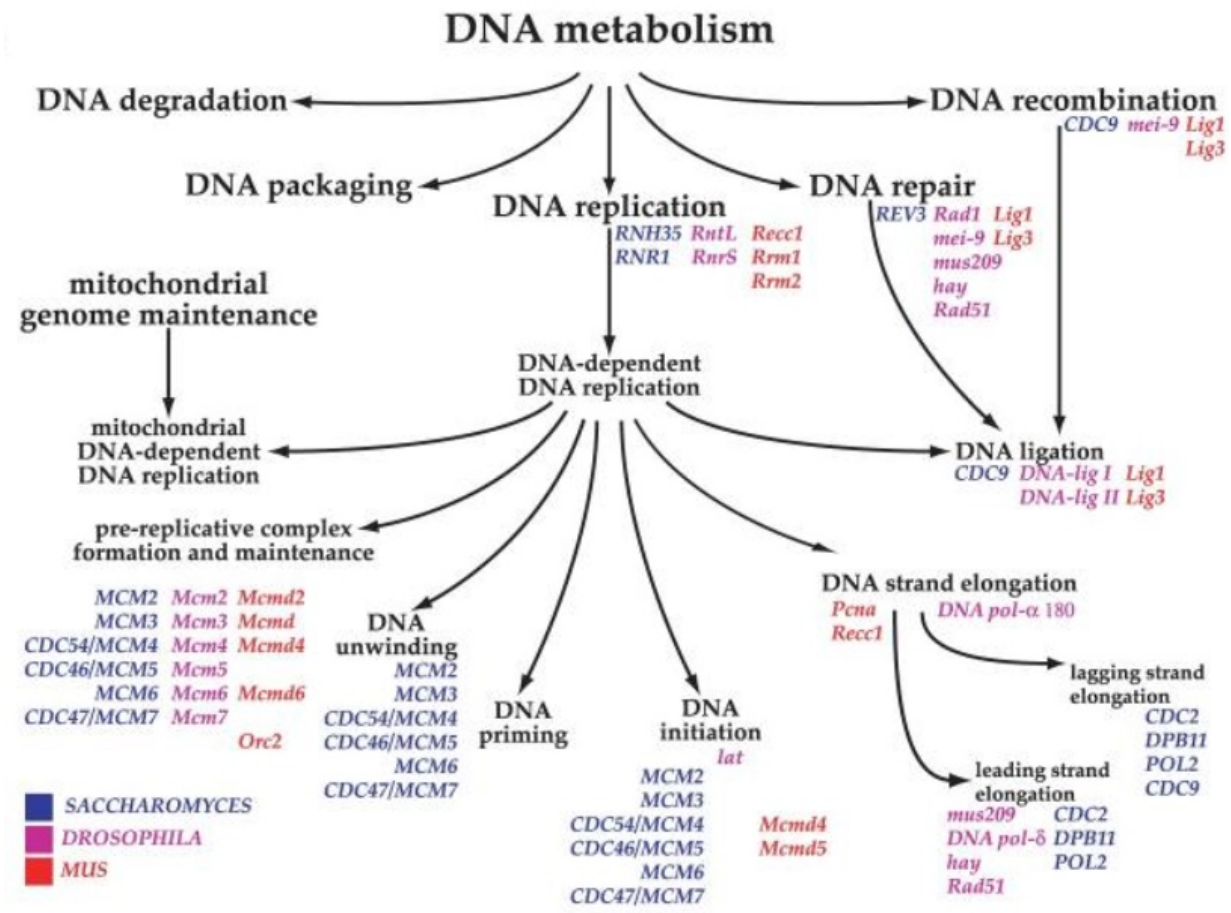

Figure 1.2: An example of Gene Ontology. The ontology is built from a structured vocabulary. Genes are associated with nodes in the ontology. A subset of genes are shown for simplicity. This figure shows a biological process ontology that describe DNA metabolism. Genes are colored based on different organisms. Figure taken and modified from [7].

and JASPAR [215]. For a gene, a protein-DNA interaction was identified if the gene is in the top 100 predictions in any of the PWM for TFs. The last part is for mouse TFs, the proten-DNA interaction is derived from the second part except that a top 1000 threshold is used instead of top 100. Human gene ids were translated to mouse gene ids based on Mouse Genome Database (MGD) [18] and HUGO Gene Nomenclature Committee (HGNC) database [178].

1.3.4 CRISPR-Cas9 genome editing in lineage tracing

CRISPR-Cas9 approach opens the possibilities to freely modify DNA. Details on the mechanism of this approach are out of the scope of this thesis. Great reviews on this topic are [54, 170]. Sequences that are suitable can be provided to guide CRISPR-Cas9 to its target. Once it matches to its target, this will result in small insertions or deletions. (Figure 1.3(a)) These edits in DNA are often called ”scars” and could serve as markers of cell lineage (Figure 1.3(b)). The changes made are irreversible because Cas9 cannot bind to the changed target sequence.

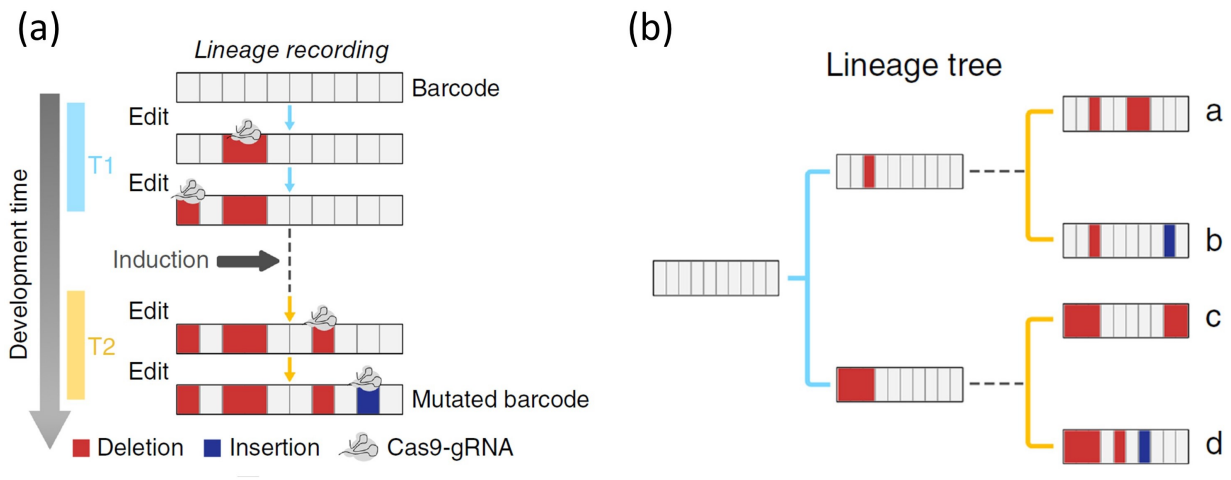

Figure 1.3: CRISPR-Cas9 gene editing technology in single cell lineage tracing. (a) During cell development process, CRISPR-Cas9 technology is applied to mutate barcodes and recode cell lineage information. Edits can be applied at different times (like T1 and T2). (b) The mutated barcodes can be used to reconstruct cell lineage tree. Figure taken and modified from [154].

1.4 Introduction to computational methods

As mentioned above, the details for each of the computational methods developed in this thesis are provided in each of the chapters. Here we provide very brief background for these methods.

1.4.1 Hidden Markov Model (HMM)

Hidden Markov Models (HMM) [152] is a statistical Markov model for modeling a Markov process with hidden states. HMM allows us to model both observed events and hidden events that are the causal factors for the observed events in the probabilistic model. An HMM is defined by following components (Table 1.1).

Table 1.1: The parameter definition for HMM

symbol	**definition**
V	the set of discrete observation alphabet
O	the sequence of T observations, each one drawn from the observation alphabet V
S	the set of N states: s_0,s_1,s_2 ... to s_N
π	the initial probability for each state, π_0,π_1,π_2 ... to π_N
A	the transition probability defined on any pair of states, A_{ij} denotes the probability from s_i to s_j
E	the parameters associated with emission probability for a given state and observation

A first-order HMM has two following assumptions to simplify the model.

- **Markov Assumption:** $P(s_t|s_1 s_2 ... s_{t-1}) = P(s_t|s_{t-1})$. This means that the probability of going to which next state only depends on the current state.

- **Output Independence** $P(o_i|s_1 s_2 ... s_t, o_1 o_2 ... o_t) = P(o_i|s_i)$. This means that the probability of an output observation on depends on the state that produce the observation, not on any other earlier or later states or observations.

Details for how to learn HMM can be found in [152]. We can see that HMM is applicable to discrete set of observation alphabet and finite number of states. However, for modeling continuous expression of scRNA-Seq data and possibly many number of states, HMM is not enough. Therefore, we will look into Continuous-State Hidden Markov Model (CSHMM) in the next session to solve this problem.

1.4.2 Continuous-State Hidden Markov Model (CSHMM)

Continuous-State Hidden Markov Model extends the state space to a continuous domain. For the continuous state, $S \subset \mathbb{R}^d$ instead of $\{s_0, s_1, s_2 \ ... \ s_N\}$. The most basic continuous-state version of HMM is linear Gaussian Markov Model (Kalman filter). $s_t = Cs_{t-1} + \epsilon_{t-1}$, $x_t = Ds_t + \xi_t$, where $s_t \in \mathbb{R}^d$ is a continuous-state hidden Markov process, $x_t \in \mathbb{R}^d$ is a continuous-valued observation, and $\epsilon_t \sim N(0, A)$, $\xi_t \sim N(0, B)$ are process noise and measurement noise respectively. Kalman Filter has many applications like tracking moving objects and stock modeling. For modeling cell lineage tracing with scRNA-Seq dataset, TASIC [155], and SCDIFF [51] adopt Kalman Filter. However, as we have mentioned before, scRNA-Seq datasets usually have very limited time points (often less than 5 time points with large interval). Due to limited observation, Kalmen Filter has small number of steps with big gaps and cannot account for possibly larger number of biological states with non-linear changes. One of our goal is to assign cells to the lineage tree continuously between the major states (with observation) and shows the differentiation stage of the cells so Kalmen Filter is not applicable here. For how we use CSHMM to model scRNA-Seq dataset, see Chapter 2 for more details.

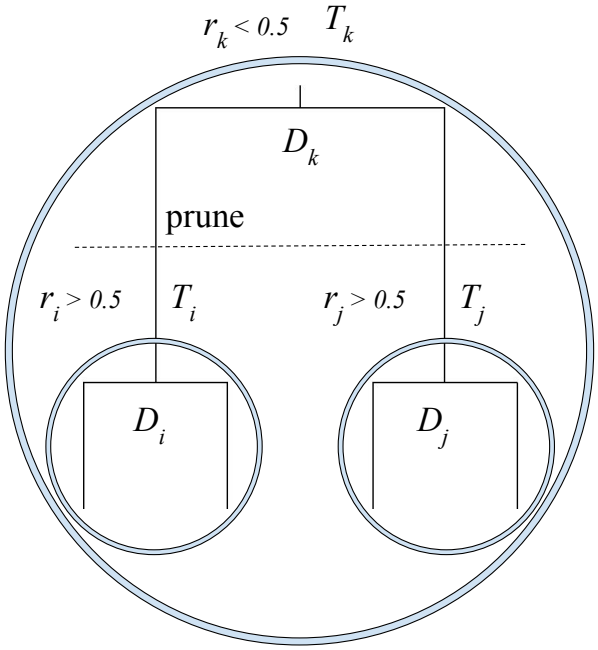

Figure 1.4: An example BHC tree, where T_i and T_j are merged into T_k. The corresponding D_i and D_j are merged into D_k. Each vertical line is a cluster and horizontal lines connecting vertical lines represents the merge of clusters into new cluster. r_k is defined as the probability that the two clusters under T_k should be merged. If $r_k < 0.5$ and both $r_i > 0.5, r_j > 0.5$ then T_i and T_j should be separate clusters so BHC will prune the tree accordingly to output the final clustering. See text for details.

1.4.3 Bayesian Hierarchical Clustering (BHC)

Bayesian Hierarchical Clustering (BHC) [93] is a bottom-up agglomerative clustering method that initializes each data sample as a cluster and iteratively choose two cluster to merge into a new cluster. Figure 1.4 shows an example of the BHC merge process. Let $\mathbf{x}^{(i)}$ denotes the observation of sample i and $\mathcal{D} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(k)}\}$ be the entire dataset. $\mathcal{D}_k \subset \mathcal{D}$ is the set of all the data at leaves under subtree T_k. Each iteration two clusters (T_i and T_j) will be merged into a new cluster and the dataset associated with the cluster will be $\mathcal{D}_k = \mathcal{D}_i \bigcup \mathcal{D}_j$. The choice of which two clusters to be merged is based on the highest value of r_k, which is defined as the probability that the two clusters should be merged. After the whole tree was constructed, BHC will output the final clustering by pruning the tree at where $r_k < 0.5$. To calculate r_k, BHC formed two hypotheses for each merge. For the first hypothesis $\mathcal{H}_1$, BHC assumes that each data are independently generated from a mixture model and each cluster corresponds to a distribution component. This means that, data points $\mathbf{x}^{(i)}$ in a cluster $\mathcal{D}_k$ are independently and identically generated from a probabilistic model $P(\mathbf{x}|\theta)$ with parameter θ. The conjugate prior of θ is $P(\theta|\beta)$ where β is the

hyperparameters of the prior. This way, the marginal likelihood of $\mathcal{D}_k$ can be expressed by

$$P(\mathcal{D}_k|\mathcal{H}_1) = \int P(\mathcal{D}_k|\theta)P(\theta|\beta)d\theta \tag{1.1}$$

$$= \int \big[\prod_{\mathbf{x}^{(i)} \in \mathcal{D}_k} p(\mathbf{x}^{(i)}|\theta) \big] P(\theta|\beta)d\theta \tag{1.2}$$

The alternative hypothesis $\mathcal{H}_2$ is that there are two or more clusters in $\mathcal{D}_k$. To make the calculation tractable, BHC restricts the clustering of T_k to be consistent with sub-trees T_i and T_j, this way, we can write the formula for $\mathcal{H}_2$:

$$P(\mathcal{D}_k|\mathcal{H}_2) = P(\mathcal{D}_i|T_i)P(\mathcal{D}_j|T_j) \tag{1.3}$$

By combining the two hypotheses and weighting them by a prior π_k for $\mathcal{H}_1$, we can obtain the recursive definition of the marginal probability of the $\mathcal{D}_k$ in tree T_k:

$$P(\mathcal{D}_k|T_k) = \pi_k P(\mathcal{D}_k|\mathcal{H}_1) + (1 - \pi_k)P(\mathcal{D}_i|T_i)P(\mathcal{D}_j|T_j) \tag{1.4}$$

π_k is also recursively defined by:

$$\pi_k = \frac{\alpha\Gamma(n_k)}{d_k} \tag{1.5}$$

$$d_k = \alpha\Gamma(n_k) + d_i d_j \tag{1.6}$$

The π_i and d_i for each initial cluster are set to $\pi_i = 1$ and $d_i = \alpha$, where α is another hyperparameter, n_k is the number of data points under sub-tree T_k and Γ is the Gamma function. The probability of deciding whether the two clustering should be merged r_k can be obtained by using Bayes rule:

$$r_k = \frac{\pi_k P(\mathcal{D}_k|\mathcal{H}_1)}{P(\mathcal{D}_k|T_k)} \tag{1.7}$$

In Chapter 4, we use the likelihood definition from BHC to model the expression likelihood on cell lineage tree.

1.5 Structure of this thesis

The CSHMM method of modeling time-series scRNA-Seq data for cell lineage tracing will first be introduced in Chapter 2. Then, in Chapter 3, we will introduce the new CSHMM-TF formulation by adding parameters for TFs. In Chapter 4, we will discuss LinTIMaT, the methods for combining CRISPR-Cas9 data and scRNA-Seq data to build cell lineage tree, and constructing invariant lineage tree for single cell lineage tracing. For Chapter 5, we will show results of applying CSHMM method to a newly generated lung development dataset and show how CSHMM can help biologists to improve the cell differentiation protocols. These chapters are adapted from our finished papers submitted to journals. Chapter 6 is for the conclusion and future works.

Chapter 2

CSHMM: Continuous-State HMMs for Modeling Time-Series Single-Cell RNA-Seq Data

This chapter describes CSHMM, which is the model that learns the lineage tree and can assign cells continuously on a lineage tree. The chapter has been adapted with changes from our paper [115] published in *Bioinformatics*: Lin, Chieh, and Ziv Bar-Joseph. "Continuous-state HMMs for modeling time-series single-cell RNA-Seq data." Bioinformatics 35.22 (2019): 4707-4715.

As we have mentioned in Chapter 1, the major two types of methods for analyzing time-series single-cell RNA-Seq studies have their own limitations. Deterministic methods are highly dependent on dimensionality reduction methods and the 2D/3D reduced dimension lose a lot of information from the dataset, while the probabilistic methods suffer from limited number of states to represent the possibly very large number of biological state. Here we present a new method for ordering cells in scRNA-Seq studies which combines the continuous representation offered by the deterministic methods and the ability to handle the full gene expression profile provided by the probabilistic methods. Our algorithm is based on the use of Continuous State HMMs (CSHMMs) [2]. Unlike standard HMMs which are defined using a discrete set of states, continuous state HMMs can have infinitely many states and so cells can be assigned to a much more detailed trajectory. We discuss how to formulate the CSHMMs for scRNA-Seq data and how to perform learning and inference in this model. Once we learn a CSHMM model, all cells are assigned to specific locations along paths which allows users to associate cells with specific fates and to reconstruct continuous developmental trajectories for the genes along each path. We applied our CSHMM to several scRNA-Seq datasets. As we show, the method was able to correctly assign cells to paths in order to reconstruct developmental trajectories for these processes improving upon the models obtained by both the deterministic and prior probabilistic models. Using the learned cell assignment we were also able to identify several novel genes for the different cell fate trajectories.

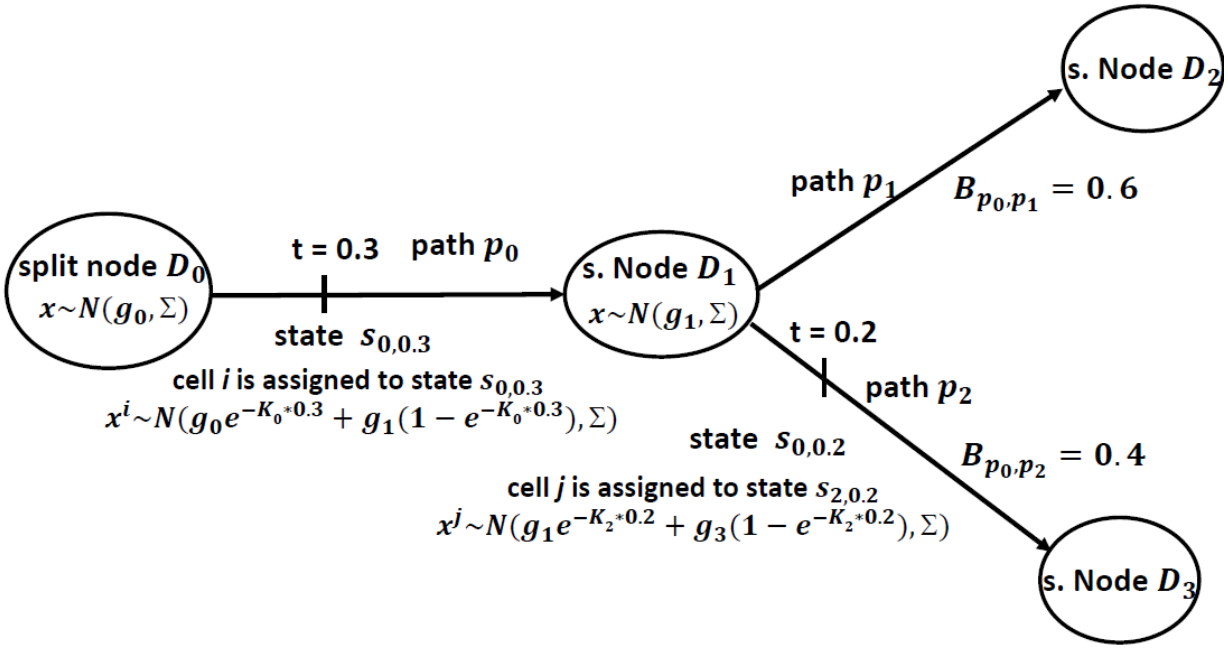

Figure 2.1: CSHMM model structure and parameters. Each path represents a set of infinite states parameterized by the path number and the location along the path. For each such state we define an emission probability and a transition probability to all other states in the model. Emission probability for a gene along a path is a function of the location of the state and a gene specific parameter k which controls the rate of change of its expression along the path. Split nodes are locations where paths split and are associated with a branch probability. Each cell is assigned to a state in the model. See text for complete details

Table 2.1: The parameter definition for CSHMM

symbol	definition
V	the observation alphabet $\subset \mathbb{R}^G$
π	the initial probability for each state
S	the set of states
B	the branch probability $\subset \mathbb{R}^{P*P}$
A	the transition probability defined on any pair of states and branch probability B
$E = (K, g, \Sigma)$	the parameters associated with emission probability for a given state
K	$K = \{K_1, ..., K_{\|P\|}\} \subset \mathbb{R}^G$
g	$g = \{g_1, ..., g_{\|D\|}\} \subset \mathbb{R}^G$
$\Sigma \subset \mathbb{R}^{G \times G}$	the covariance matrix with off-diagonal element to be 0 and diagonal term σ_j^2
D	the set of split points
P	the set of paths
G	the number of genes (dimension of data)

2.1 CSHMM model formulation

Figure 2.1 presents the CSHMM model structure. HMMs define a transition probability between states and emission probability for each state. CSHMMs defines the same set of parameters. However, since they have infinite many states (in our case corresponding to continuous time) both transition and emission probabilities are a function of the specific *path* a state resides on. Split points represent time points where we allow cells to split into different lineages and paths are defined as the collection of (infinitely many) states between two such split events. Note that in our model we learn the location of the splits from data and while these are initialized with the sampling rate (i.e. initially we use the sampled time points to define the split locations) as we discuss below the model can add splits between two time points to account for the asynchronous nature of cells in some studies.

Each cell is assigned to a specific state along one of the paths which corresponds to both, the time inferred for it by the algorithm and the cell type it belongs to. In addition to the state assignment and transitions at split nodes the model also encodes emission probabilities. Following prior work on modeling expression with HMMs [119] we use a Gaussian emission model and assume independence for gene specific expression levels conditioned on the state. To define an emission probability for a state we use the *relative* location of a state along a specific path. We define a state by the path number and the relative time for this path. We denote by $s_{p,t}$ the state representing time $0 \leq t \leq 1$ on path $p(D_a \rightarrow D_b)$, where a, b are the indices of the split nodes. Let i be a cell assigned to $s_{p,t}$. We denote by x_j^i the expression of gene j. The emission probability for gene j in cell i assigned to state $s_{p,t}$ is thus assumed to be
$x_j^i \sim N(\mu_{s_{p,t}}, \sigma_j^2) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp(-\frac{(x_j^i - \mu_{s_{p,t}})^2}{2\sigma_j^2})$. Where

$$\mu_{s_{p,t}} = g_{aj}\exp(-K_{p,j}t) + g_{bj}(1-\exp(-K_{p,j}t))$$
$$= g_{bj} + (g_{aj} - g_{bj})\exp(-K_{p,j}t) \quad (2.1)$$
$$\sigma_j^2 \text{ is the variance of gene } j \quad (2.2)$$

Here, g_{aj} is the mean expression for gene j at split node a. We assume a continuous change in expression for a subset of the genes along a path. However, we want to learn the specific shape of the change curve and so we add a parameter $K_{p,j}$ which controls the rate of change for gene j on path p , allowing different genes to change at different rates.

Using these notations we next define the following parameters that are required to specify a CSHMM: $\lambda = (V, \pi, S, A, E)$, where all the symbol definitions are presented in Table 2.1.

Each cell i is associated with an expression vector $X^i \in \mathbb{R}^G$, and a (hidden) state $y^i = s_{p,t}$. The observation alphabet $V \subset \mathbb{R}^G$, is thus a real value vector with dimension $|G|$, where G are the set of genes in our input set. We associate a root state $s_{0,0}$ with each HMM with initial probability of 1 ($\pi_{p,t} = 1$ for state $s_{0,0}$ and $\pi_{p,t} = 0$ for all other states). The transition probability $A(s_{p_1,t_1}, s_{p_2,t_2})$ for each pair of states $s_{p_1,t_1}, s_{p_2,t_2} \in S$ is defined as follows:

$$A(s_{p_1,t_1}, s_{p_2,t_2}) = 0, \text{ if } s_{p_2,t_2} \text{ is not reachable from } s_{p_1,t_1} \quad (2.3)$$
$$A(s_{p_1,t_1}, s_{p_2,t_2}) = 1/Z_{p_1,t_1}, \text{ if } p_2 = p_1 \text{ and } t_2 > t_1 \quad (2.4)$$
$$A(s_{p_1,t_1}, s_{p_2,t_2}) = \prod_{\substack{q \in \text{branch probability} \\ \text{from } p_1 \text{ to } p_2}} \frac{q}{Z_{p_1,t_1}}, \text{ if } p_2 \neq p_1, p_2 \text{ reachable from } p_1 \quad (2.5)$$

Where Z_{p_1,t_1} is a normalizing factor for the transition probability going out of state s_{p_1,t_1} i.e..

$$Z_{p_1,t_1} = 1 - t_1 + \sum_{\substack{\text{path } p \\ \text{reachable from } p_1}} \prod_{\substack{q \in \text{branch probability} \\ \text{from } p_1 \text{ to } p}} q. \quad (2.6)$$

The branch probability is defined on split nodes as shown in Figure 2.1. The second term in equation 2.6 is the product of all branch probabilities of the paths from p_1 to p. For example, assume that there are two paths in between states p_1 and p: p_a and p_b. Then the second term will be $B_{p_1,p_a} * B_{p_a,p_b} * B_{p_b,p}$, where B_{p_a,p_b} refers to the branch probability for cells to transition from p_a to p_b. The use of branching probabilities leads to lower likelihood for cell assignments to later (more specific) paths in the branching tree. This is similar to prior probabilistic methods for reconstructing branching trajectories [155]. The idea here is that earlier stages are often less specific (higher entropy [201], while later stages (representing specific fates) have a tighter expression profile. Thus, cells that represent specific cell types will still be assigned to their correct (late) stage based on their expression profile while noisier cells would be assigned to the earlier stages.

To see that this is indeed a Continuous-State Hidden Markov Model (CSHMM) model we note that the model contains a continuous set of states with well defined emission and transitions probabilities (transition probabilities integrate to 1 for each state). Transitions and emissions only depend on the current state. Each observation is assumed to have been emitted from one of

the states in the model.
Since we cannot assume that the time stamp associated with each cell is the correct time (to account for asynchrony) we need to determine cell assignments. In addition, we do not know the structure of the model in advance. We thus developed an Expectation Maximization (EM) algorithm which can jointly infer the model structure, parameters and cell assignments.

2.2 Likelihood function for the CSHMM model

Since CSHMMs are probabilistic models, to determine the optimal structure and parameters we first need to define the likelihood function that the model is trying to optimize. Denote by X^i the expression profile of cell i. Let $s^i_{p,t}$ denote the (unobserved) state (path p time t) which 'emitted' the expression of cell i (i.e. the state to which cell i is assigned to). Given an expression input matrix $X = \{X^1, ..., X^N\}$ and hidden variables $Y = \{y^1, ..., y^N)\}$ where $y^i = s^i_{p,t}$ is the state for cell i we can write the log likelihood as follows:

$$l(X, Y|\lambda) = \log P(X, Y|\lambda) = \sum_{i=1}^{N} \log P(X^i, y^i|\lambda) \tag{2.7}$$

Which can be can further decomposed using the parameters described above as:

$$P(X^i, y^i|\lambda) = P(X^i, s^i_{p,t}|\lambda) = P(X^i|s^i_{p,t}, \lambda)P(s^i_{p,t}|s^i_p, \lambda)P(s^i_p|\lambda) \tag{2.8}$$

$$P(s^i_p|\lambda) = \prod_{\substack{q \in \text{branch probability} \\ \text{from root to } p}} q \quad \text{(the branch probability)} \tag{2.9}$$

$$P(s^i_{p,t}|s^i_p, \lambda) = 1 \quad \text{(assume uniformly random on every t)} \tag{2.10}$$

$$P(X^i|s^i_{p,t}, \lambda) = \prod_{j=1}^{G} P(x^i_j|s^i_{p,t}, \lambda) \quad \text{(the emission probability)} \tag{2.11}$$

$$= \prod_{j=1}^{G} \frac{1}{\sqrt{2\pi\sigma^2_j}} \exp(-\frac{(x^i_j - \mu_{s_{p,t}})^2}{2\sigma^2_j}) \tag{2.12}$$

$$= \prod_{j=1}^{G} \frac{1}{\sqrt{2\pi\sigma^2_j}} \exp(-\frac{(x^i_j - g_{bj} - (g_{aj} - g_{bj})\exp(-K_{p,j}t))^2}{2\sigma^2_j}) \tag{2.13}$$

Thus, the complete log likelihood for N input cells is:

$$l(X, Y|\lambda) = \sum_{i=1}^{N} (\sum_{j=1}^{G} \log P(x^i_j|s^i_{p,t}, \lambda) + \log P(s^i_{p,t}|\lambda)) \tag{2.14}$$

Note that in equation 2.10, $P(s^i_{p,t}|s^i_p, \lambda)$ is a probability density function over path p with domain $0 \leq t \leq 1$. The normalization performed in equations 2.3 - 2.5 guarantees that all transition probabilities for a state integrates to 1.

2.3 Constraining expression changes along a path

Similar to prior pseudotime ordering methods our algorithm relies on the assumption that cells that are close to each other along the developmental trajectory have a similar (though not identical) expression profile. This implies that for most genes we would expect to see relatively small changes in expression whereas for a few genes (which may define the changes that the cell undergoes during the process) we expect larger changes. Thus, we expect differences between the expression profiles of consecutive split nodes to be sparse. To encode our assumption about the sparseness of the difference vector Δg we use L_1 regularization on the difference. Minimizing the L_1 regularization term for negative log-likelihood (NLL) is equivalent to maximizing the complete likelihood multiplied by the Laplace prior distribution [202]. The Laplace prior distribution on Δg and parameter h is:

$$f(\Delta g; h) = \prod_{i=1}^{G} \frac{1}{2h} \exp(-\frac{|(\Delta g)_i|}{h}) \tag{2.15}$$

where $h > 0$ is the scale of the distribution.
Adding this regularization, the log likelihood function changes to:

$$l(\lambda|X, Y) = \log P(X, Y|\lambda) + \log(\text{sparse probability}) \tag{2.16}$$

$$= \sum_{i=1}^{N} \log P(X^i, y^i|\lambda) + \sum_{\Delta g \text{ for each path}} \log(\prod_{j=1}^{G} \frac{1}{2h} \exp(-\frac{|(\Delta g)_j|}{h})) \tag{2.17}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{G} \log P(x_j^i|s_{p,t}^i, \lambda) + \sum_{i=1}^{N} \log P(s_{p,t}^i|\lambda) + \sum_{\Delta g \text{ for each path}} \sum_{j=1}^{G} -\frac{|(\Delta g)_i|}{h}) \tag{2.18}$$

2.4 Model initialization

For model initialization we slightly modify the strategy used in [51]. We construct an initial cell differentiation tree by clustering the cells, and then compute the distance of each of the clusters to the root of the tree (cells in first time point). Using this distance function clusters are assigned to different levels in the tree (where clusters in each level are significantly more distant from the root than the preceding level). Finally, we connect each cluster (except the root cluster) at level i to a parent cluster in level $i - 1$ by selecting the closest cluster, in expression space, in level $i - 1$. See Appendix A Supporting methods for complete details. Following this initialization step each cluster is associated with a path (the edge connecting it to its parent). Finally, cells in each cluster are randomly assigned along the path for that cluster. Split nodes are defined for cases where two or more clusters at a specific level connect to the same cluster at the level above them.

2.5 Learning and Inference (EM algorithm)

We use an EM algorithm to learn the parameters of the model and to infer new cell assignment. Given initial cell assignments, the branching probabilities can be easily inferred using standard Maximum Likelihood Estimation (Appendix A supporting methods). In the Appendix A supporting methods we also discuss how to learn the emission probability parameters which, due to the K parameter requires an optimization of a non convex target function. As for cell assignment, given model parameters we assign each cell to a state $s_{p,t}$ which maximizes the log-likelihood of the resulting model. Again, since the likelihood function is not concave, determining a optimal value t for a cell assigned to path p is challenging. In the Appendix A supporting methods we discussed a sampling strategy for solving this problem which we use to assign cells.

2.6 Modifying the model structure

So far we assumed a fixed model structure. However, as part of the EM algorithm cells are re-assigned and so some paths that started with several cells may become empty while for others we may need to reassign their parents as their expression parameters change. To allow for structure changes during the learning process we do the following. Following each EM iteration, we test for two things: First, if a path has less than 3 cells assigned to it we remove it from the model and connect any following paths to the path parents. In addition, we allow the algorithm to connect split nodes to different parents in the level above them. For this, we try to connect every path at a certain level to all paths at the prior level it was not connected to. For each such new connection we re-compute the log-likelihood for all cells assigned on the path. If the log likelihood increases for this set of cells we keep the new relationship, otherwise we do not. This is repeated for every possible connection resulting in the structure that maximizes the likelihood for the current assignments we have.

2.7 Analysis of gene expression for specific cell fates

To determine the set of genes associated which specific fates (a set of paths from root to a leaf in the model), we calculate the Spearman correlation between their expression values and the ordering of the cells assigned to the set of paths leading to a specific fate. We use gprofiler [157] for GO of the top 300 genes. For plotting gene expression we use a 4 degree polynomial to interpolate expressions in the different cells assigned to a trajectory. For each leaf node, we scale all cell assignments between the root and the node to be between 0 and 1 so that all expression profiles are plotted with the same length.

2.8 Results

To test the CSHMM model and to compare the results to prior pseudo-time ordering methods we used several time series scRNA-Seq datasets. The first dataset is for mouse lung development

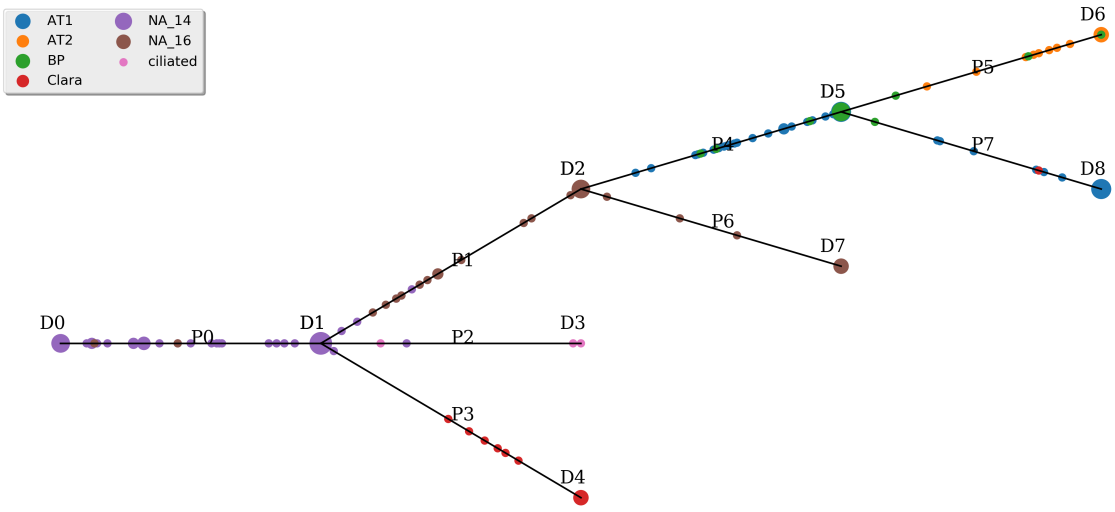

Figure 2.2: CSHMM model structure and continuous cell assignment for the lung developmental dataset. D nodes are split nodes and P edges are paths as shown in Figure 2.1. Each small circle is a cell assigned to a state on the a path. The bigger the circle the more cells are assigned to this state. Cells are colored based on the cell type / time point assigned to them in the original paper.

[208]. After preprocessing Methods, the lung dataset consisted of 152 cells with 15K genes, measured at 3 time points (14.5, 16.5, 18.5 days). Cells at time point 18.5 are labeled with one of the following cell types: alveolar type 1 (AT1), alveolar type 2 (AT2), bipotential progenitor (BP), Clara and Ciliated. Cells at earlier time points were not labeled in the original paper. We label these cells as NA_14 or NA_16 based on their time point. The second dataset profiled the process in which mouse embryonic fibroblasts (MEFs) are induced to become neuronal (iN) cells [209] This data contained 4 time points (0, 2, 5, 22 days) starting with MEF cells at day 0. Using known markers, Day 22 cells were labeled in the original paper with one of the following cell types: Neuron, Myocyte, Fibroblast. For the rest of the cells we used the assignments in the original papers for the plots, though they were not used by the CSHMM algorithm. Both datasets were processed in a similar way to the processing performed in the original paper: We removed genes with FPKM < 1 in all cells and genes with zero variance. Next expression values were transformed to log FPKM. In addition to these two well annotated, but rather small, datasets we also tested the CSHMM on a much larger zebrafish embryogenesis dataset [61]. This dataset has close to 40,000 cells profiled at 12 time points (from 3.3 to 12 hours). Cells in the last time point (only) were labeled with one of 25 cell types based on marker genes. This dataset is log TPM and genes expressed in less then 5% of cells are removed.

2.8.1 Application of CSHMM to lung developmental data

Figure 2.2 presents the resulting CSHMM branching model for the lung development data and the distribution of cells along its paths (based on the state assigned by the model). As can be seen, the CSHMM method was able to assign different cell types to different paths correctly,

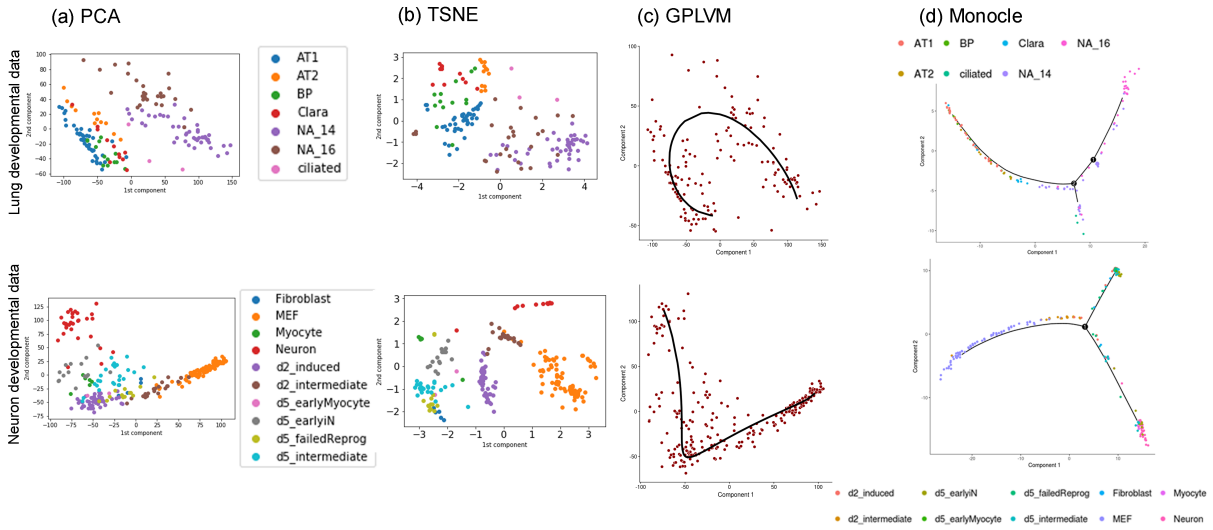

Figure 2.3: Analysis of lung development and MEF reprogramming data by prior methods. (a) PCA (b) TSNE (c) GPLVM (d) Monocle 2. Top row presents results for the lung dataset and the bottom for the neural developmental dataset. Colors correspond to cell fate assignments in the original papers.

for example, ciliated (path 2), Clara (path 3), AT1 (path 7), and AT2 (path 5) are all correctly associated with a terminal path. The bi-progenitor (BP) cells (path 4) are mostly assigned to the predecessors of the AT1 and AT2 paths in agreement with prior observations [208]. This highlights the ability of the method to assign cells measured at the same time point (E18.5) to different times in the model.

The ability to correctly reconstruct the branching trajectory for such *in vivo* data is not trivial. As we show in Figure 2.3 (a)(b)(c)(d), dimensionality reduction based methods that have been used in the past for pseudo time ordering, including PCA [208], TSNE (for which we used the optimal parameters, Appendix A Supporting Methods), GPLVM following PCA [32], and Monocle 2 [1] [151, 206] where unable to fully reconstruct the known developmental trajectory for this data. PCA is able to identify clusters for different cell types but the projection of the reduced dimensional cells cannot reconstruct the known trajectory over time. Similarly, TSNE was also unable to separate some cell types for the later time point and was mixing E14.5 and E16.5 cells. GPLVM correctly orders E14.5 and E16.5 cells, however, it is unable to determine branching models for the different cells types in E18.5 and is also unable to determine the relative earlier ordering of the BP cells. Monocle 2 was able to generate trajectories, associating cells with specific time points, however, for this data it finds only 1 split point and was also unable to correctly separate the E18.5 cells according to their types. To test if Monocle 2 is able to separate E18.5 cells for lung datasets, we try to expand the left branch and the result is in Appendix A Figure 2.7. As can be seen, Monocle 2 is still unable to separate E18.5 cell types. We also tried to compare to scTDA [159] , however that method requires a commercial software from Ayasdi Inc. that we did not have access to. We have also compared the results to prior probabilistic methods that use a discrete set of states [155]. For this we have re-run the CSHMM algorithm but this time

[1] which performs minimum-spanning-tree analysis on a reduced dimension of the data

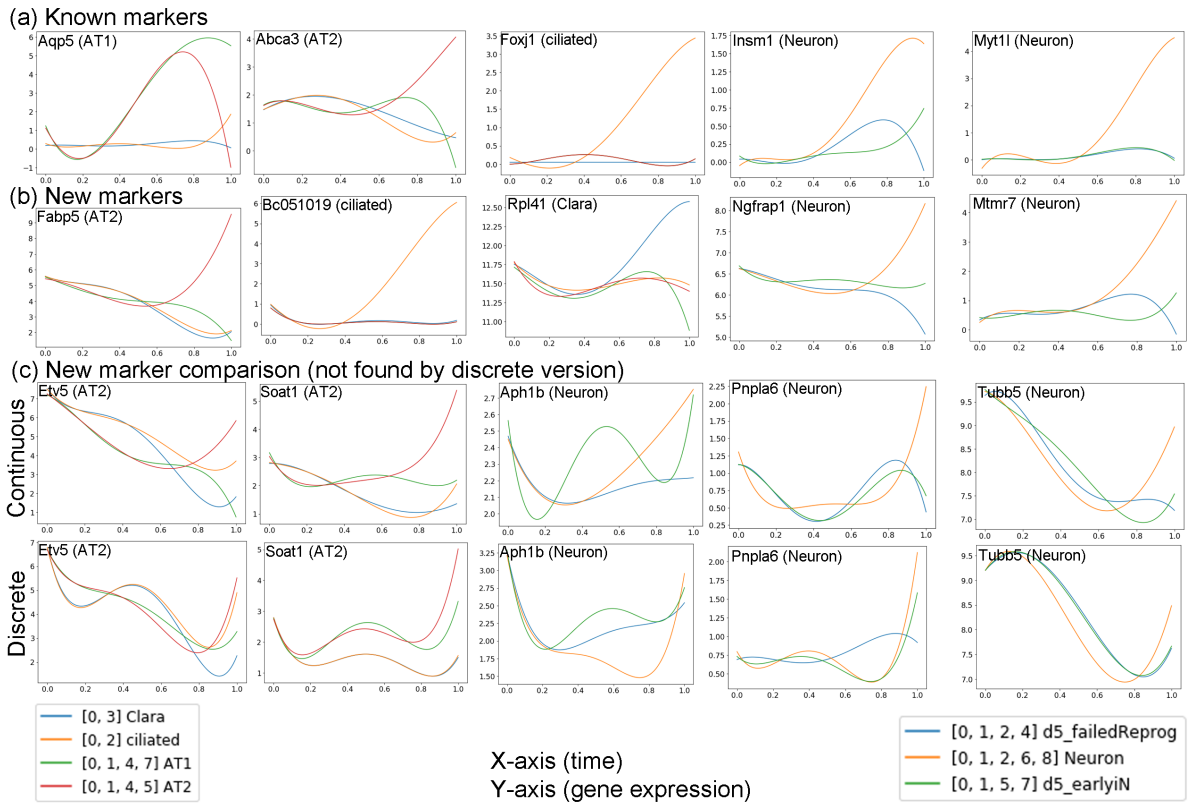

Figure 2.4: Reconstructed gene expression profiles for lung and neural development data. Each figure plots the expression profile of a gene along the different paths in the corresponding model. Each image includes the gene name and the cell type it was assigned to by the model (AT1, AT2, cillated and Clara from the lung model and Neuron from the neuron model). (Top row) Known markers for the different cell types. (Second row) Novel markers not identified in the original papers found by the CSHMM assignments. (Third and fourth rows) Comparison of reconstructed profiles using the CSHMM (top) and discrete HMM (bottom). Several genes has a unique path profile using the CSHMM but did not display such profile when using the discrete model.

allowing cells to be assigned only to the endpoints of paths themselves and not to intermediate points. Results are presented in Appendix A Figure 2.8. As can be seen, although the discrete version leads to good result in terms of cell assignments, there are some differences. Specifically, BP cells are mostly assigned to terminal paths in these models, rather than intermediate paths. Further, as we show below, the CSHMM model is better at identifying cell type specific genes when compared to prior probabilistic discrete models.

Identifying cell type specific genes in the lung dataset

The continuous nature of the CSHMM allows us to reconstruct the full gene expression trajectories for each path / cell type (ending at a leaf in our model). For this we use the ordering of cells from root to leaf for each of the leaves. To overcome noise in individual cell measurements we fit a continuous function (a polynomial of degree 4) to the set of values for each gene

and plot the resulting curve. We then use these curves to search for genes that are specifically correlated with a leaf (cell type, Methods). In addition, we can compare trajectories for genes between two leafs to identify genes that are uniquely associated with one cell type. To illustrate the advantages of such analysis we have plotted in Figure 2.4 (a) the trajectories of some of the known markers for the cell types in the data and additional genes (Figure 2.4 (b)) that, while not currently known as markers or cell type specific, are expressed in a similar manner to known markers and so are predicted to be novel markers for specific cell types. For example, Aqp5's expression is high for the AT1 path, but is strongly decreasing in the AT2 path after the split between these two paths. For the novel markers, Bc051019 displays very similar expression to the known cillated marker Foxj1. An independent study profiling scRNA-Seq of lung epithelial cells [55] has also identified it as a ciliated marker. See Appendix A Table 2.2 for a full list of genes that are significantly associated with each pair of paths. The ability to reconstruct the full trajectory of the genes along the paths based on the continuous state assignments is also an advantage of the CSHMM compared to prior methods. As we show in Figure 2.4 (c), for several genes the trajectory assigned by the CSHMM model is more accurate (based on known biology) than their trajectory in a discrete HMM model. For example, Etv5 and Soat1 are two AT1/AT2 markers found by CSHMM which are not identified by a regular HMM. Recent studies suggest that Etv5 is essential for the maintenance of AT2 cells [242], and Soat1 is expressed in AT2 cells [82]. We have also performed GO analysis on the set of genes that are identified for each leaf path (See supporting website). Several of the functions identified agree with known functions for the terminal paths. For example, the most significant GO category for genes correlated with path 2 *the cillated path* was cilium assembly (p-value = 1e-14) which is indeed the major function of cillated cells. Similarly, epithelium development was one of the top categories for path 3 *Clara path* (p-value = 4e-6). For path 7 (AT1 cells) the top categories were related to extracellular matrix (p-value = 3e-9), which is known to be associated with the development of this cell type [142].

2.8.2 Application of CSHMM to neural developmental data

We have also analyzed a slightly more complicated MEF cell differentiation dataset [209]. The resulting CSHMM and cell assignments are presented in Figure 2.5. As can be seen, similar to the lung data, for this data the assignment of cells to paths generally agrees with their known function. For example, the 0-1-2-6-8 set of paths lead from the embryonic MEF cells (day 0) to d2_intermediate, then d5_intermediate and finally to Neuron cells (day 22). In contrast, paths 0-1-3, while following the initial set of cells up to day 2, leads to a different outcome by day 5 (the d5_failedReprog fate). Other trajectories are likely representing the fact that cells are unsynchronized. For example, the 0-1-5-7 paths represent a slightly less mature set of cells along a reasonable trajectory (embryonic - d2_intermediate - d2_induced- d5_earlyN). Once again, most prior methods for the representation and analysis of time series scRNA-Seq data are unable to accurately represent this branching process (Figure 2.3 (a)(b)(c)(d)). Monocle 2 while doing a good job at identifying the major branching between failed and neuron cells, fails to separate the d5_earlyN and d5_failedReprog which are assigned to different paths in our model. Similarly, PCA and GPLVM do not clearly identify the trajectories and tend to mix the successful and unsuccessful differentiated cells. TSNE was also unable to clearly identify the trajectory from

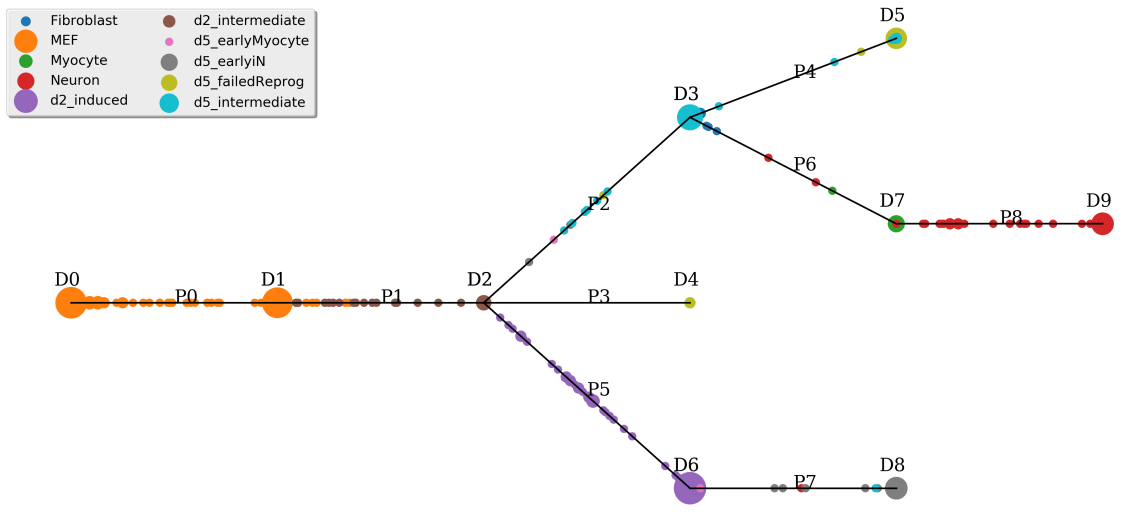

Figure 2.5: The CSHMM model structure and continuous cell assignment for the MEF reprogramming dataset. Notations, symbols and colors are similar to the ones discussed for Figure 2.1.

d2 and d5 to neurons. We also ran a discrete version of the CSHMM algorithm (Appendix A Figure 2.9). Similar to the results for the lung developmental data, the discrete model largely agrees with the continuous one in terms of the overall topology. However, they differ in some of the cell assignments (for example, the discrete model assigns some of the later d2_induced cells to path 0) and, as we show below it is also less able to identify cell type specific genes.

Identifying genes activated during neural cell development

Several known and novel genes can be identified using the continuous cell assignments (Figure 2.4 (a)(b)(c)). For example, Insm1 and Myt1l were identified in the original paper as known neuron markers and their CSHMM reconstructed trajectories agree with such roles. Several other genes not identified in the original paper appear to be highly correlated with successful differentiation. For example, Ngfrap1 (Bex3, Figure 2.4(b)) has been identified previously as contributing to nerve growth [29]. Similarly, prior studies have shown that Mtmr7 is highly expressed in the brain [131]. Other genes identified highlight the difference between the discrete and continuous models (Figure 2.4(c) and Appendix A Supporting Results). We also analyzed the top GO categories for the set of genes associated with specific fates. Enriched GO categories for genes correlated with each fate agrees well with known functions. For example, for the neuron path (path 8) the top categories are ”neuron part” (p-value 1e-29) and ”synapse” (p-value 1e-22). For the path that includes neural progenitors (earlyN, path 7) we see an enrichment for ”nervous system development” (p-value 5e-14) as well as for several categories and TFs related to cell proliferation (including E2F with a p-value of 3e-22). In sharp contrast, the ”failed reprogramming” path (Path 4) is not enriched for any neural activity and is instead enriched for various extracellular matrix categories (p-value 1e-20). See supporting website for the complete list of enriched categories.

2.8.3 Scalability and robustness of the CSHMM model

The datasets discussed so far include hundreds of cells with relatively high coverage. Some of the recent scRNA-Seq studies profile a higher number of cells (thousands) though usually with lower coverage. To test the ability of the method to scale to larger number of cells we used both simulated and additional data. For simulated data we re-run the analysis discussed above by replicating each of the 152 lung cells 100 times and adding 20% random dropout to the genes in each replicate (generating a total of 15K expression profiles). For the CSHMM learning and inference we used the top 1000 most variable genes and tested two versions of search for cell assignment and K, either using 10 or 100 values (which is what we used for the smaller dataset). For the 10 values variant, running on a desktop with 4 threads takes roughly 40 minutes to perform one iteration of the EM algorithm and since we usually require less than 10 iterations the total run time is still less than 7 hours even for this dataset. For the 100 values variant the total run time is 15h. The resulting models are presented in Appendix A Figure 2.10 (10 values) and Appendix A Figure 2.11 (100 values). As can be seen, even when using more than 15000 cells, both models reconstruct all the major paths that were recovered in the original (152 cells) lung model. In addition to testing the impact of the number of cells, and different values for the hyper-parameter representing the time points in each path, we also used the simulated data to test the impact of different choices for another hyper-parameter of our model, $\lambda_g = \frac{1}{h}$ which controls the L1 penalty used to select genes for the different paths. As we show (Appendix A section 2.10.1), we do not observe a large impact on the resulting model for a set of reasonable values for this parameter.

In addition to using simulated data we also tested the CHMM on the 40K cells zebrafish dataset mentioned above. Unlike the two datastes discussed above much less is known about the specific differentiation pathways for several of the last time point cell types. CSHMM analysis of this dataset required only 2 iterations and took 33 hours per iteration. Results are presented in Figure 2.6. To determine the success of the assignments focused on the leafs in the model (corresponding to the annotated cell types in the original paper). We calculated the adjusted random index (ARI) agreement between these two sets. We found that the ARI achieved by the CSHMM assignments is significantly better when compared to 1000 randomization tests for the cells ($p < 10^{-10}$ based on randomization tests, Appendix A Figure 2.30). Thus, the CSHMM method can scale to larger datasets with tens of thousands of cells. The number of parameters for CSHMM is $O(P * G)$ and the time complexity for CSHMM is $O(N * P * G * S)$, where N is the number of cells, P is the number of paths (edges), G is the number of genes, S is the number of sampled points for cell assignments and for learning K. Note that the time complexity of previous discrete probabilistic methods (e.g. SCDIFF [51]) is $O(N * P * G)$, so the key increase in time here refers to the need to sample from a much larger set of possible states. To reduce the model complexity or running time, users can reduce the number of genes, or reduce number of sampled points, or simplify the initial tree structure with fewer edges.

For how we determine the number of iteration needed for each dataset based on the change of cell assignment, please see Appendix A section 2.10.2.

Another issue that can impact the analysis of scRNA-Seq data is dropout. Due to the lower quantity of RNA obtained from single cells, and the amplification steps required, several genes

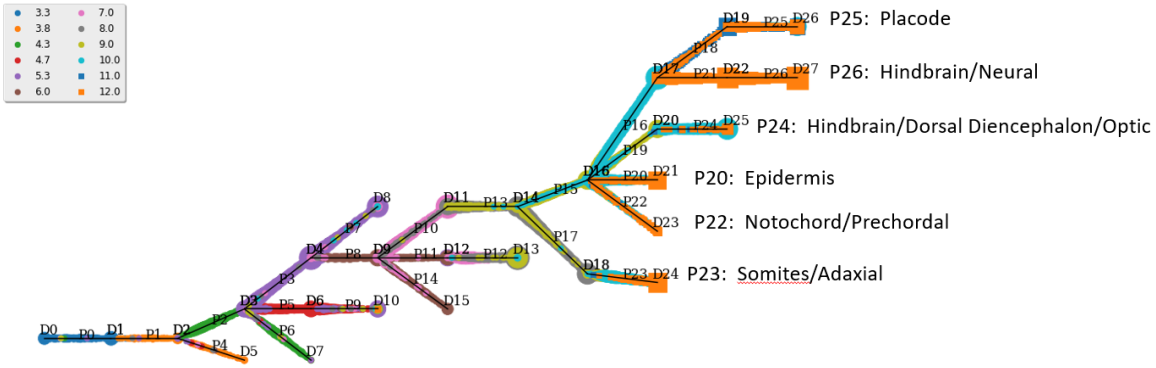

Figure 2.6: The CSHMM model structure and continuous cell assignment for zebrafish embryogenesis dataset. Notations, symbols and colors are similar to the ones discussed for Figure 2.1. Note that the leaf paths of 6-somite stage (time point 12.0) are labeled with one or more labels based on dominating cell types.

with low transcript numbers may appear to have 0 transcripts in scRNA-Seq data [101]. As we discuss in Appendix A Supporting Results, we performed extensive analysis of dropout impact on the CSHMM method. We observe that for rates which increase dropout by 5-20% results stay largely the same. Beyond 20% additional dropouts we observe a larger impact in which Clara and AT2 cells are merged in a single path though AT1 and AT2 cells are still separated and the initial parts of the model are also correct even for 40% additional dropouts.
To test the effect initialization of cell locations on each path, we run CSHMM with different random cell initialization on lung developmental dataset on top 1000 most variable genes. The results are in Appendix A Figures 2.39-2.43. We can see that with smaller set of genes and different random seed, the major structure stays the same. However, out of 5 random initialization, 3 models have deleted ciliated path and merge the ciliated cells into the Clara path. We thus conclude that when the number of genes is less, random initialization on cell locations on path can impact the results for rare cell types. We also further randomly drop 20% of genes/cells for the same set of random seeds for cell initializations and observe similar results (Appendix A Figures 2.44-2.53). Dropping 20% of genes does not change the model structure, while dropping 20% of cells increase the chance of dropping the corresponding path since this will be hard for CSHMM to separate the rare cell types if some of the rare cells are dropped. To test how much the learning can correct errors made in the initial clustering based assignment we create additional clusters as noise and attach the noise clusters to the original model as a different initialization profile. To create additional clusters, for each terminal path with more than 5 cells (P3, P5, P6, P7), we randomly sampled 20% of the cells to construct a new paths (P8, P9, P10, P11) and attached this new path to one of the original paths at random. Appendix A Figure 2.20 and 2.31 present the original model and the model with the additional 4 clusters. The result after training with iteration 1-4 is shown in Appendix A Figure 2.32-2.35. As can be seen, while the model very quickly trims most of the duplicate clusters (3 of the 4 are removed in the first iteration) cells continue to be re-assigned for 4 iterations until the model converges. The only added path that is not removed (and appears in the final model) is P11, which was initially connected after the path

with mostly the same cell types (AT1). While the learning algorithm did not remove this path, it used it to further refine the ordering and assignment of cells which, as can be seen, changes between the iterations. Thus, we conclude that while the initial assignments play an important role the method is able to correct errors introduced during this phase as part of the CSHMM learning procedure.

2.9 Discussion

Both major strategies for modeling developmental trajectories for time series scRNA-Seq data have advantages and disadvantages. Pseudotime ordering allows for continuous assignment of cells and the reconstruction of complete expression trajectories. However, the result of these methods often depends on the reduced dimension and the ordering is based on a very limited set of values for each cell. In contrast, probabilistic methods can handle the complete set of genes well, but do not provide a continuous representation for the expression profiles.

Here we show that Continuous State HMMs (CSHMMs) can provide a solution for both problems. On the one hand it is a probabilistic method and so can accommodate full expression profile while on the other hand it provides continuous assignment of cells to paths. We formally defined the CSHMM and discussed methods for learning and inference in such model. We applied our methods to simulated and real scRNA-Seq data. Analysis of the models constructed by the CSHMM method shows that it can accurately reconstruct the branching model for these differentiation processes, correctly assigns cells to the different paths and fates and reconstructs expression trajectories that identified known and novel marker for the different cell types.

While it is impossible to say if the continuous cell assignments orderings determined by the our model are correct (since we do not know the ground truth), a possible way to evaluate the accuracy of these assignments is to look at the resulting gene trajectories. Given a specific ordering, by any method, we can plot the resulting expression profiles for genes in these cells. This can be used to both, identify genes that are in agreement with a specific path in the model and to compare the ordering with orderings obtained by other methods. As we have shown in Figure 2.4, genes identified by the CSHMM ordering include several of the known markers for specific cell types, improving upon prior methods. This results provides some support to the accuracy of the cell assignment to paths. We also try to validate our cell orderings based on Spearman correlation between the pseudo time and sampled time for each full path (from P0 to all leaf paths). We found that for all three datasets (lung, neuron, zebrafish), the correlation shows a strong agreement. See Appendix A section 2.10.2 for full results.

While these initial results are encouraging, we would also like to test the ability to incorporate other types of data, including regulatory information, to aid in improving the model learning and cell assignment. In the next Chapter, we will present how we extend CSHMM to incorporate regulatory information. Also, we have applied CSHMM to a newly generated dataset for improving lung cell differentiating protocols. See Chapter 5 for details.

2.10 Appendix A: Supplement to Continuous State HMMs for Modeling Time Series Single Cell RNA-Seq Data

2.10.1 Supplementary Methods for CSHMM

Learning (M step)

Given initial cell assignments, the branching probabilities can be easily inferred using standard Maximum Likelihood Estimation (see below).

Next, we discuss learning the emission probability parameters. For genes that change along a path, we need to learn a mean value g for split nodes and the $K_{p,j}$ parameter which encodes for each path and each gene the rate of change between the start and end expression values for that gene on that path. For K, even with a fixed mean value g for each split node, it is difficult to compute it in close form because of non-convexity. We thus use a line search strategy to determine $K_{p,j}$. For this we compute the likelihood for 100 possible values between 0 to 5 (since $e^{-5} \approx 0$), and choose the value that achieves the maximum probability for $K_{p,j}$ (note of course that since this is a gene and path specific parameter it can be done independently for each gene / path).

As for g, let $w_j^i = \exp(-K_{p,j}t^i)$, and $\lambda_g = \frac{1}{h}$ be the L1 sparse parameter. Then, the negative log likelihood terms that depend on g are:

$$\begin{aligned} NLL &= \sum_i^N \sum_j^G \frac{1}{2\sigma_j^2}(x_j^i - \mu_{s_{p,t}}^i)^2 + \lambda_g \sum_{(g_1,g_2)\in path} \sum_j^G |g_{1,j} - g_{2,j}| \\ &= \sum_i^N \sum_j^G \frac{1}{2\sigma_j^2}(g_{pa,j}w_j^i + g_{pb,j}(1-w_j^i) - x_j^i)^2 \\ &\quad + \lambda_g \sum_{(g_1,g_2)\in path} \sum_j^G |g_{1,j} - g_{2,j}| \end{aligned} \tag{2.19}$$

where (g_{pa}, g_{pb}) and (g_1, g_2) refers to the mean gene expression of the split point at both ends of a path. Since the function is convex, we let $\lambda_g = 1$ and use CVXPY [47, 75, 76], a disciplined convex optimization toolkit utilizing cone-spliting interior point method, to solve the linear system.

As for the variance, since we assume that the variance σ_j of each gene j is the same across all the paths, once we have the g values we can use a standard MLE method to derive the closed-form solution for its estimation (see Appendix A section 2.10.1).

Inferring cell assignments (E-step)

Given model parameters λ, we would like to assign each of the cells in our input dataset expression matrix X to a state $s_{p,t}$ which maximizes the log-likelihood. Determining a optimal value t for a cell assigned to path p is hard to be performed in closed form because the likelihood function to t is not concave.

Instead, similar to the optimization of $K_{p,j}$ parameter, we use a sampling strategy to find the best time along a path for each cell. Specifically, for each path we sample 100 points uniformly and compute the likelihood of assigning the cell to each of these points. Since the likelihood function (when model parameters are known) decomposes based on cells, this process is efficient.

Details for MLE

Branch probability First, we have the constraint that $\sum_{p_2} B_{p_1,p_2} = 1 \quad \forall p_1, p_2 \in P$. Using Lagrange multipliers we can write:

$$L(X, Y, \theta, \lambda) = \left(\sum_{i=1}^{N} \sum_{\substack{q \in \text{branch probability} \\ \text{from } p_1 \text{ to } p_2}} \log(q) \right) + \theta^T (B\mathbb{1} - \mathbb{1}) \tag{2.20}$$

We obtain the update for B_{p_1,p_2} by setting gradient to 0

$$\frac{\partial L(X, Y, \theta, \lambda)}{\partial B_{p_1,p_2}} = 0 \Rightarrow \frac{N_{p_1,p_2}}{B_{p_1,p_2}} + \theta_{p_1} = 0 \tag{2.21}$$

$$\sum_{p_2} B_{p_1,p_2} = 1 \Rightarrow \sum_{p_2} \frac{-N_{p_1,p_2}}{\theta_{p_1}} = 1 \Rightarrow \theta_{p_1} = \sum_{p_2} -N_{p_1,p_2} \tag{2.22}$$

$$\Rightarrow B_{p_1,p_2} = \frac{N_{p_1,p_2}}{\sum_{p_2} N_{p_1,p_2}} \tag{2.23}$$

Where N_{p_1,p_2} is the number of cells assigned to path p_2 that comes from p_1, specifically, $N_{p_1,p_2} =$ the number of cells assigned to p_2 only if p_1 is the parent of p_2, otherwise $N_{p_1,p_2} = 0$. Note that the size of θ is $|P|$, the number of path. The size of B is $|P| * |P|$. $\mathbb{1}$ is the vector of size $|P|$ where every entry is 1.

Learning σ_j We compute the gradient of σ_j, the variance parameter for each gene:

$$\frac{\partial}{\partial \sigma_j} \log P(X, Y|\lambda) = \frac{\partial}{\partial \sigma_j} (\sum_{i=1}^{N} \sum_{j=1}^{G} \log P(x_j^i | s_{p,t}^i, \lambda)) \tag{2.24}$$

$$= \frac{\partial}{\partial \sigma_j} (\sum_{i=1}^{N} \log N(\mu_{s_{p,t}^i}, \sigma_j^2)) \tag{2.25}$$

$$= \frac{\partial}{\partial \sigma_j} (\sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp(-\frac{(x_j^i - \mu_{s_{p,t}^i})^2}{2\sigma_j^2})) \tag{2.26}$$

$$= \frac{\partial}{\partial \sigma_j} (\sum_{i=1}^{N} -\log(\sigma_j) - \log(\sqrt{2\pi}) - \frac{(x_j^i - \mu_{s_{p,t}^i})^2}{2\sigma_j^2}) \tag{2.27}$$

$$= \sum_{i=1}^{N} (-\frac{1}{\sigma_j} + \frac{(x_j^i - \mu_{s_{p,t}^i})^2}{\sigma_j^3}) \tag{2.28}$$

Setting gradient to 0 we have:

$$0 = \sum_{i=1}^{N}(-\frac{1}{\sigma_j} + \frac{(x_j^i - \mu_{s_{p,t}^i})^2}{\sigma_j^3}) \tag{2.29}$$

$$\Rightarrow \sigma_j^2 = \frac{\sum_{i=1}^{N}(x_j^i - \mu_{s_{p,t}^i})^2}{N} \tag{2.30}$$

Details for model initialization methods

To initialize the CSHMM model, we apply the same strategy as the SCDIFF tool [50]. Specifically, a spectral clustering algorithm with Spearman correlation distance metric is applied to each time point so that each time point has clusters of cells. Spectral clustering and Spearman correlation is selected because of their robustness to noise[249] and high dimensional data [91, 141, 226, 238]. However, for large dataset spectral clustering might be too slow because the time complexity is $O(n^3)$. Therefore, in case of large dataset (number of cells $>$ 2000), PCA with 10 components followed by K-means clustering is used. To select the best number of clusters in each time point, an ensemble strategy with 3 clustering assessment scores are using: Silhouette Score [164], Davis-Bouldin index [45] and AIC [3]. 100 randomly subsampled datasets are generated by selecting 90% of gene at random. Then, for each of the subsampled dataset, the abovementioned 3 clustering scores are computed for different of clusters. The final number of clusters is selected based on the majority of voting.

Though the above procedure could determine the number of clusters for each of the time point, the single cells might have different developmental stages at the same time point. It is not reasonable to assume that all the cells in the same time point represent the same developmental stages. Therefore, the following procedure is used to determine the proper time point (level in the differentiation tree) for each of the cluster. Basically, the level of each cluster is determined by how similar it is to the ancestor, where the ancestor is defined as the cells in the first time point. Spearman correlation is used to determine the similarity of clusters. Then all the clusters are sorted based on the similarity to ancestor. Each pair of the adjacent clusters will be assigned to different if their ranksum test or difference of mean value is significant (p-value < 0.05 or difference of mean value is greater than the average difference of mean values of all clusters), and vice versa. After that, the clusters are assigned to their parents based on the Spearman correlation. Thus the initial graph is constructed.

We treat each of the initial clusters as paths and all the cells in a cluster is also assigned to the path. Then the cells in each of the path are initialized with random time. The K parameter of each path are initialized with 1, σ parameter are initialized with 1. Branch probability is calculated based on the number of cells in each path.

Parameter selection for other methods

For Monocle and GPLVM, we adopt the default parameter suggested by their tutorial. For PCA we use the default parameter from sklearn package. For TSNE, we also tried the default parameter of sklearn package but it didn't work well (all the cells are in a small ball with different cell types mixed). We then follow the suggestions from the sklearn's tsne webpage (`http://`

scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html). We first use PCA to reduce the dimension to 50. It is hard to say what parameter set is the best since the dimension reduction is purely unsupervised and the results are different every time because TSNE is randomly initialized. We set the random seed to 1 to make the results more stable. Then we try different learning rates and select the relatively best one (learning rate = 10).

Hyperparameter selection for our CSHMM model

There are three hyperparameters for our CSHMM model. The first one is the range of K parameters. The second one is the number of time points that can be assigned on each path. The third one is the λ_g parameter for L1 that controls the sparsity of each path. For the first one, we already mentioned how we select the parameter for K in Chapter 2. For the second one, this is the parameter that can be changed by the user for resolution and time-complexity trade-off. We set this parameter to 100 time points so that we can see a nice continuous cell assignment. For the last one, we train our CSHMM model with λ_g to be 0.1, 0.5, 0.8, 1, 2 for both lung and neuron dataset (see Figure 2.22-2.29). We found that the cell assignments are very similar and larger λ_g makes more cells group together near endpoints. This result is reasonable because when we put more penalty to the length of each path, the length of the path will be smaller so the cells are more likely to be grouped together. As the result is robust to the hyperparameter λ_g in the range of 0.1, 0.5, 0.8, 1, 2, we arbitrarily set $\lambda_g = 1$.

2.10.2 Supplementary Results for CSHMM

Identifying genes activated during neural cell development Genes identified in (Figure 2.4 (c)) highlight the difference between the discrete and continuous models. For example, studies have shown that Aph1b is required for synaptic transmission [62], Pnpla6 encodes neuropathy target esterase (NTE), which is required for neuronal differentiation [114, 205], and Tubb5 is involved in embryonic neurogenesis [23] and neuronal differentiation[140].

Dropout analysis

Another issue that can impact the analysis of scRNA-Seq data is dropout. Due to the lower quantity of RNA obtained from single cells, and the amplification steps required, several genes with low transcript numbers may appear to have 0 transcripts in scRNA-Seq data [101]. Obviously, since we have analyzed real scRNA-Seq data above, the method can handle some amount of dropout. However, we also wanted to study the impact of even larger dropout percentage on the performance of the CSHMM method. For this, we used the lung data and randomly removed values for different % of genes (setting them to 0 instead). We tested the removal of an additional 5-40% of the values. Results are presented in Figures 2.12-2.19. We observe that between 5-20% results stay largely the same as the original analysis discussed above (similar overall branching process) with the main difference being the loss of the cillated path (path 2 in the original model) which is instead combined with the Clara path. This is caused by the fact that there are only 3 cillated cells and when more noise is added these are not unique enough to justify their own

path. Beyond 20% additional dropouts we observe a larger impact in which Clara and AT2 cells are merged in a single path though AT1 and AT2 cells are still separated and the initial parts of the model are also correct even for 40% additional dropouts. All the experiments above are performed on a single computer with 2 Intel(R) Xeon(R) E5-2670 (2.60GHz) CPU, which have 16 cores in total.

Cell assignment change analysis

We have ploted Figures that display the convergence of the algorithm as a function of the number of iterations. See Figures 2.36-2.38. As the figures show, while there is a relatively large change in the assignments in the first iteration (indicating that the initialization is not the final result) we see a rather quick drop in the number of changes in later iterations leading to convergence in only a few iterations (fewer than 5% of the cells change assignment after the 5th iteration for all datasets, including the large zebrafish dataset). We also plotted the average change in assignment for individual cells in each of the models as a function of the iteration. The values on the y axis denote the change in assignment where each path is measured with a unit (1) length. Thus, changes less than 1 mean that the cell does not move more than 1 path and changes that are much smaller than 1 (for example, 0.1) mean that the vast majority of cells move slightly on the same path. As can be seen, after the first few iterations cells do not change their location, even for models that have a very long potential path chain (for zebrafish the max change can be 11, but after 5 iterations average change is < 0.1). We thus conclude that for all datasets we analyze the model converges very rapidly.

Cell pseudo time alignment with real time analysis

To see how well our cell assignments agree with the real time, we calculate the Spearman correlation between the cell pseud time and sampled time for each full path. Specifically, for each full path from P0 to leaf paths, we calculate a vector ST which stores the sampled time of each cell on the full path, and a vector PT, which stores the pseudo time of each cell on the full path. The pseudo time is calculated as the level on the tree plus the continuous time assignment (could be $0\sim1$). Then the Spearman correlation is calculated for vectors ST and PT. See Tables 2.3-2.5 for the Spearman correlation values for all the full paths for lung, neuron and zebrafish datasets. We can observe that for most of the full path, we obtain a strong Spearman correlation (mean value =0.69 in lung dataset, 0.87 for neuron dataset, 0.84 for zebrafish dataset). We notice a relatively small value for the [0,2] full path in lung dataset. We found that this low value is caused by the fact that most of the cells in this full path have the same sampled time. Thus, one or two noisy cells with wrong pseudo time assignment can impact the correlation very much.

We have also attempted to test what we believe the reviewer suggested by generating a model with only three paths (no branching) on lung developmental dataset, where we assigned a subset of cells from the first (14.5 days) and last (18.5 days) time points to the first and last path respectively, and all other cells are assigned to the intermediate path (16.5 days cells, and a random subset of 14.5 and 18.5 days cells). Specifically, we randomly assigned 50% of cells in the first time point to the first path, and 50% of cells in the last time point to the last path. All the cells not assigned to the first or last path are assigned to the intermediate path. This way, the

initial Spearman correlation is 0.5542422025573547. After training, the Spearman correlation increased to 0.8700670257944979. Which shows that CSHMM has the ability to reconstruct the correct ordering of cell pseudo times.

Difference in PCA and TSNE figures

We agree that our PCA and TSNE results are not exactly the same as the paper in Treutlein et. al, 2014 and Treutlein et. al, 2016. We note that we applied the published method to the same data so this should not be an implementation issue. We believe that the main difference is the subset of genes used.

In their original 2014 paper (treutlein et. al. 2014), they describe the data pre-processing: "We performed principal component analysis (PCA) on all 80 single cell transcriptomes using genes expressed in more than two cells and with a non-zero variance (8578 genes)."

However, using the same criteria (genes expressed in more than two cells and with a non-zero variance) we arrive at 15K genes so the gene set is definitely different. Its impossible for us to determine which 8.5K of the 15K they actually used.

For treutlein et. al. 2016, their description about the data processing: "PCA was performed on cells using all genes expressed in more than two cells and with a variance in transcript level (log2(FPKM)) across all single cells greater than 2. This threshold resulted generally in about 8,000–12,000 genes" . Here we are getting results that are closer to the original paper (12K genes) though its not clear if the results they present are for the full set of 12K or for the subset of 8K. If its the latter it may explain the difference in results. Again, it is hard for us to obtain the exact subset they used based on the criteria they specify.

2.10.3 Supplementary Tables and Figures for CSHMM

Table 2.2: Top 20 genes for pairs of paths by analyzing absolute Spearman correlation difference of the continuous version of CSHMM

	AT1 v.s. AT2	**ciliated v.s. Clara**	**Neuron v.s. d5_failed**
rank	**gene**	**gene**	**gene**
1	Sftpc	Ptges3	Ccl27a
2	Napsa	Upf3b	Efha2
3	Slc34a2	6820408C15Rik	Pnpla6
4	Bex2	Rps26	Scg5
5	Sftpa1	BC051019	C330021F23Rik
6	Sftpb	Smek2	Sdr39u1
7	Cxcl15	Nudc	Gprasp1
8	Soat1	H3f3a	Mtmr7
9	Egfl6	Rpl41	Wdr6
10	Pi4k2b	1700016K19Rik	Ywhaz
11	Fabp5	Rsrc1	Ube2o
12	Timp3	Psmg2	Rabl2
13	Sdc4	Ttc18	Gm5148
14	Etv5	Mycbp	Aph1b
15	S100g	Rpl32	Jag1
16	Vegfa	Zfp330	Pfn2
17	Lamp3	Api5	Tubb5
18	Dbi	Slc23a1	Inpp5f
19	Scd1	0610010O12Rik	Ngfrap1
20	Ctsc	Tmsb4x	Iigp1

Table 2.3: The Spearman correlation of the alignment between pseudo time and real time for each full path in neuron reprogramming dataset

full path	**Spearman correlation**
[0, 1, 3]	0.7680191287339879
[0, 1, 2, 4]	0.9018068415674703
[0, 1, 5, 7]	0.8931473308855927
[0, 1, 2, 6, 8]	0.9383527975628075
mean	0.8753315246874646

Table 2.4: The Spearman correlation of the alignment between pseudo time and real time for each full path in lung developmental dataset

full path	Spearman correlation
[0, 2]	0.2887894814708995
[0, 3]	0.6210802183502201
[0, 1, 4, 5]	0.9033599759039195
[0, 1, 6]	0.7385161028042448
[0, 1, 4, 7]	0.9034242854649855
mean	0.6910340127988539

Table 2.5: The Spearman correlation of the alignment between pseudo time and real time for each full path in zebrafish developmental dataset

full path	Spearman correlation
[0, 1, 2]	0.8417471178271797
[0, 1, 3, 5]	0.7854446605088631
[0, 1, 3, 6]	0.8237440248843126
[0, 1, 3, 7, 8]	0.8898975289662838
[0, 1, 3, 4, 10]	0.670641814919371
[0, 1, 3, 4, 11]	0.5486626449471951
[0, 1, 3, 12]	0.9260671352222831
[0, 1, 3, 7, 14]	0.795485536900837
[0, 1, 3, 4, 13, 15]	0.8220065077837665
[0, 1, 3, 4, 13, 19]	0.8428645511724535
[0, 1, 3, 4, 9, 16, 18, 24]	0.8383287889234831
[0, 1, 3, 4, 9, 16, 18, 20, 23, 28]	0.9313388243270483
[0, 1, 3, 4, 9, 16, 18, 20, 23, 29]	0.9428715801149222
[0, 1, 3, 4, 9, 16, 17, 21, 22, 30]	0.9141078067198244
[0, 1, 3, 4, 9, 16, 18, 20, 26, 31]	0.9300799458627882
[0, 1, 3, 4, 9, 16, 18, 20, 23, 25, 32]	0.961651600331465
[0, 1, 3, 4, 9, 16, 18, 20, 23, 27, 33]	0.9700027220192817
mean	0.8491142818489034

Note that, in all the following CSHMM structure and cell assignments Figures, the D nodes are split nodes and P edges are paths as shown in Figure 2.1 in Chapter 2. Each small circle is cells assigned to a state on the tree structure. The bigger the circle the more cells are assigned to the position. The color of the circles represent different cell types. For lung developmental dataset, we change the NA cell type to NA_14 or NA_16 based on the observation time. For discrete models, cells are only allowed to be assigned to endpoints of each path. For the initial model, the cells are assigned to each path based on our initialization method described above (Appendix A section 2.10.1). In each path, cells are assigned to each time point randomly.

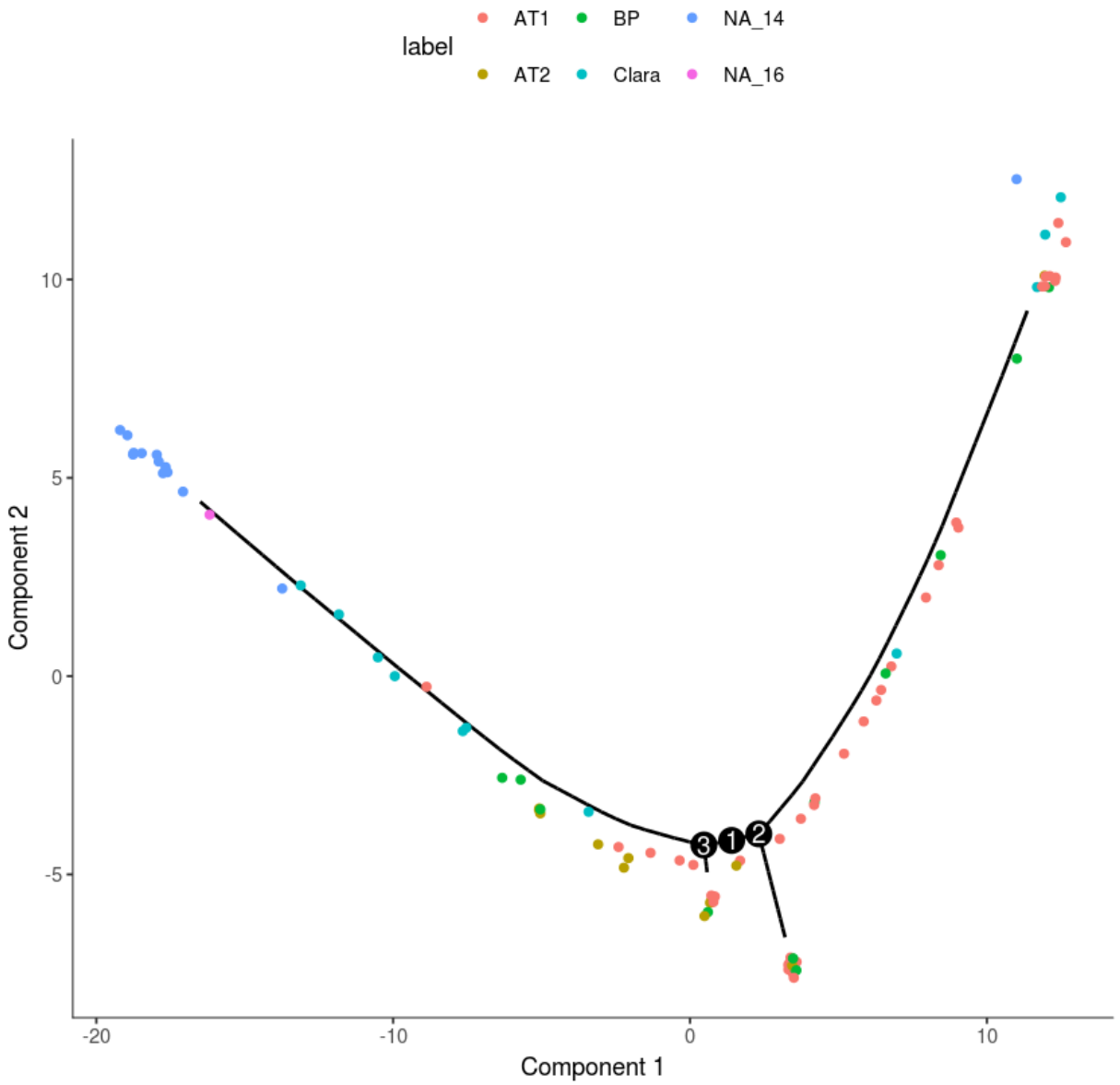

Figure 2.7: Monocle2 expansion on the left E18.5 branch for lung dataset in Figure 2.3. We can see that even with expansion Monocle2 is still unable to separate the E18.5 cell types.

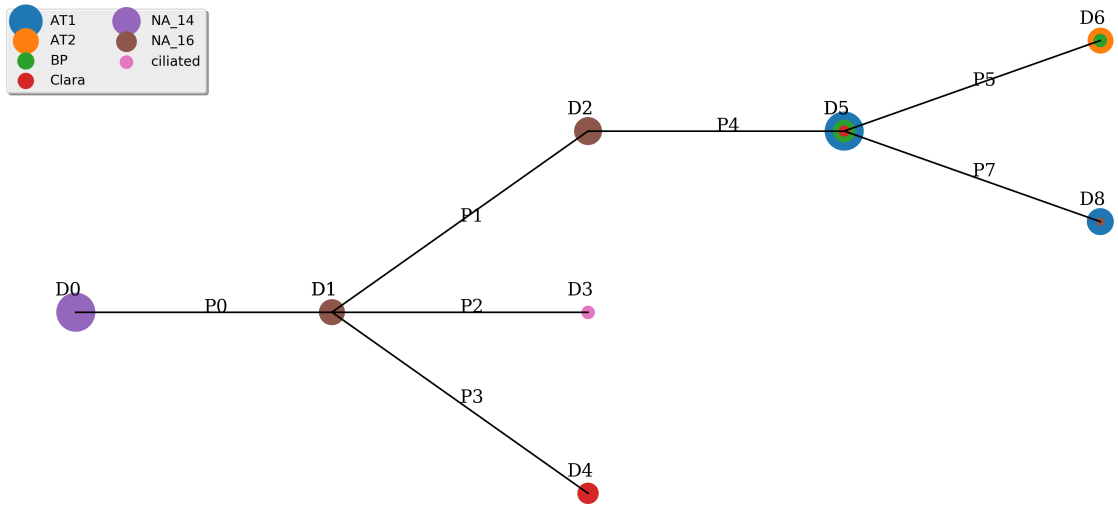

Figure 2.8: The CSHMM model structure and discrete cell assignment for lung developmental dataset.

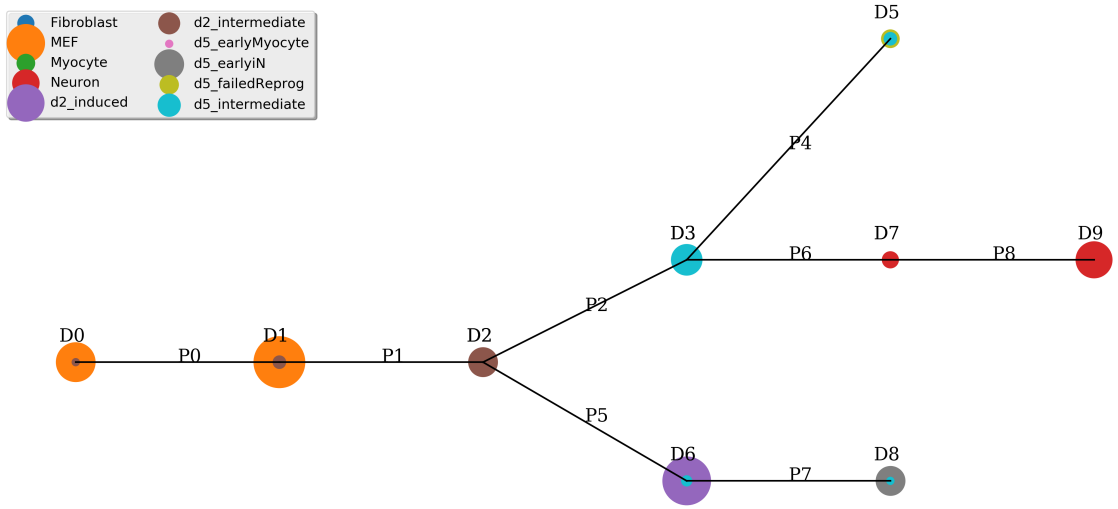

Figure 2.9: The CSHMM model structure and discrete cell assignment for neuron developmental dataset.

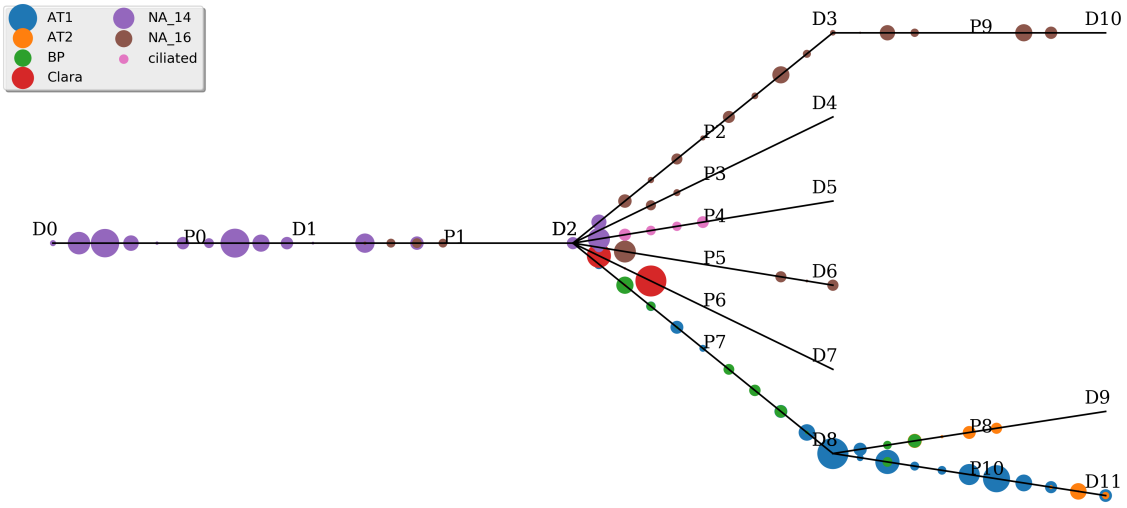

Figure 2.10: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 100 times cells, 20% dropout rate, and number of uniform sampled time is 10.

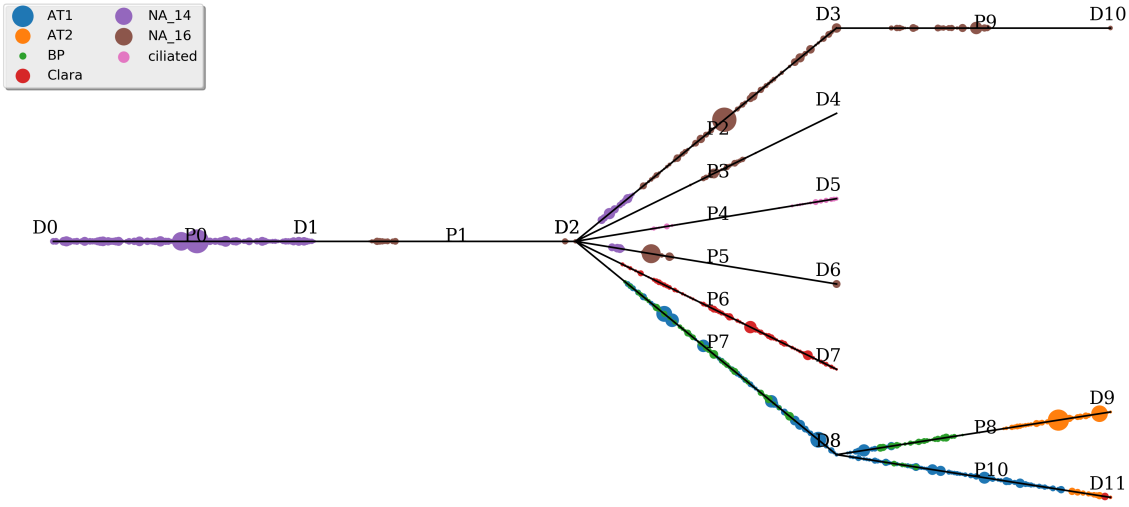

Figure 2.11: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 100 times cells, 20% dropout rate, and number of uniform sampled time is 100.

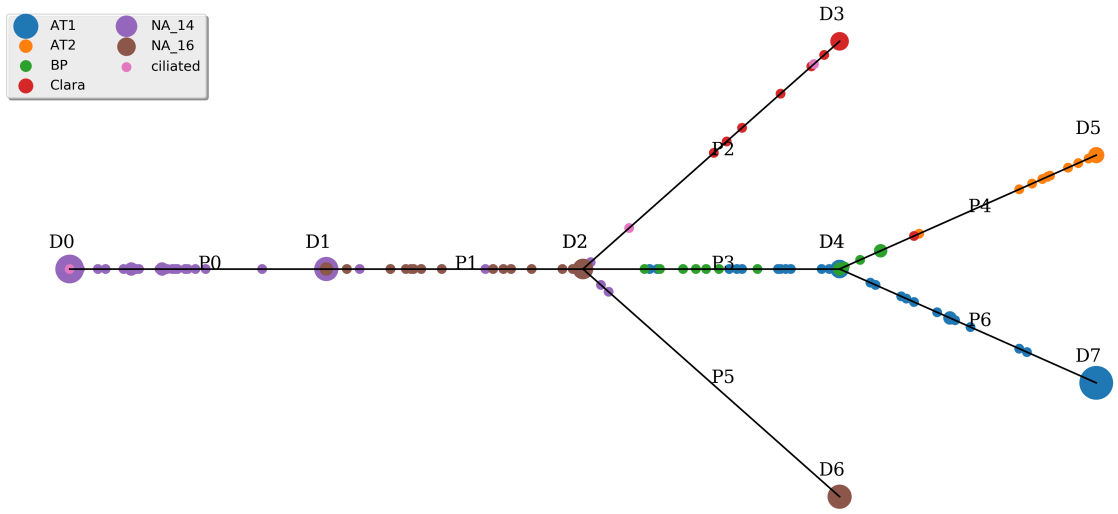

Figure 2.12: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 5% dropout rate.

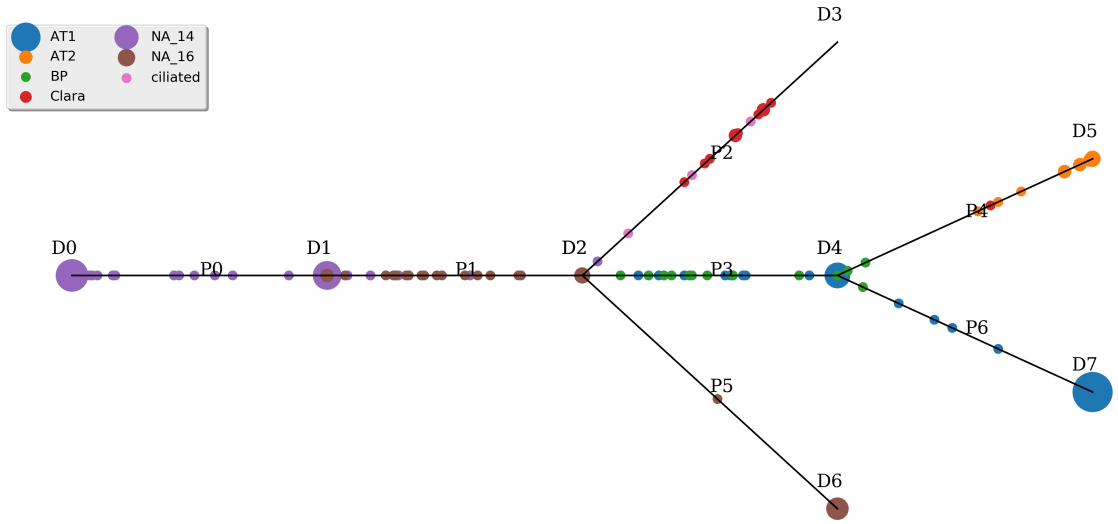

Figure 2.13: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 10% dropout rate.

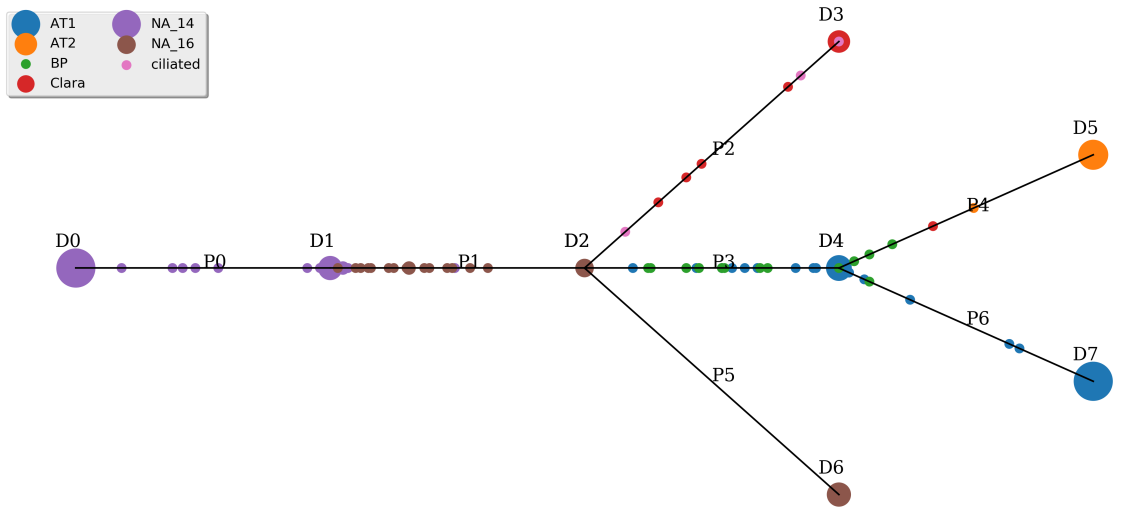

Figure 2.14: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 15% dropout rate.

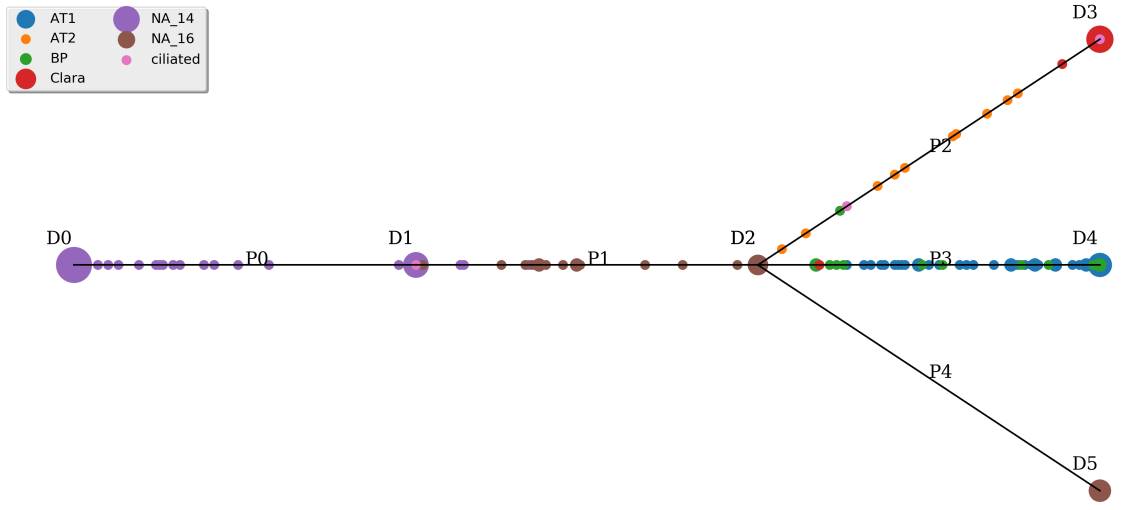

Figure 2.15: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 20% dropout rate.

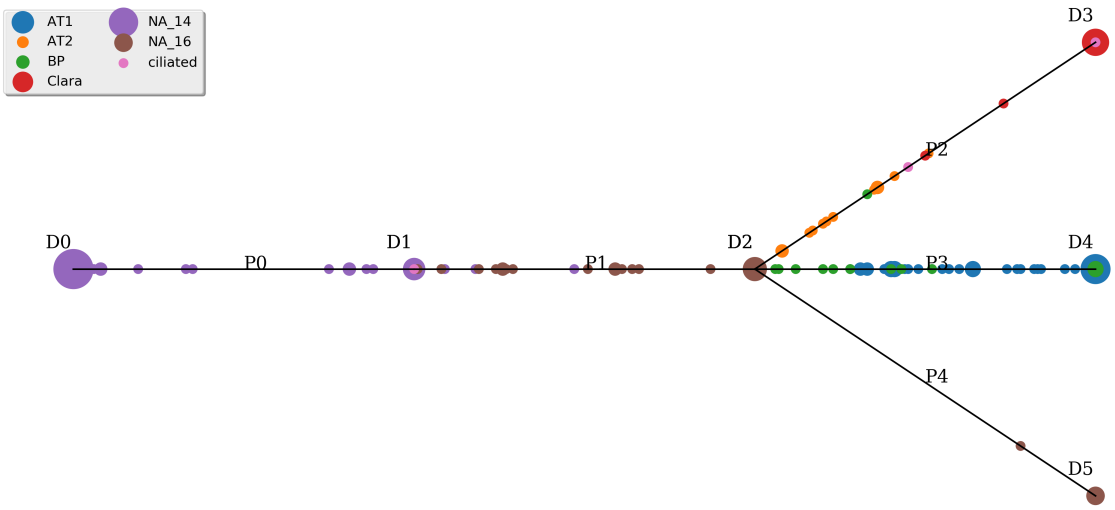

Figure 2.16: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 25% dropout rate.

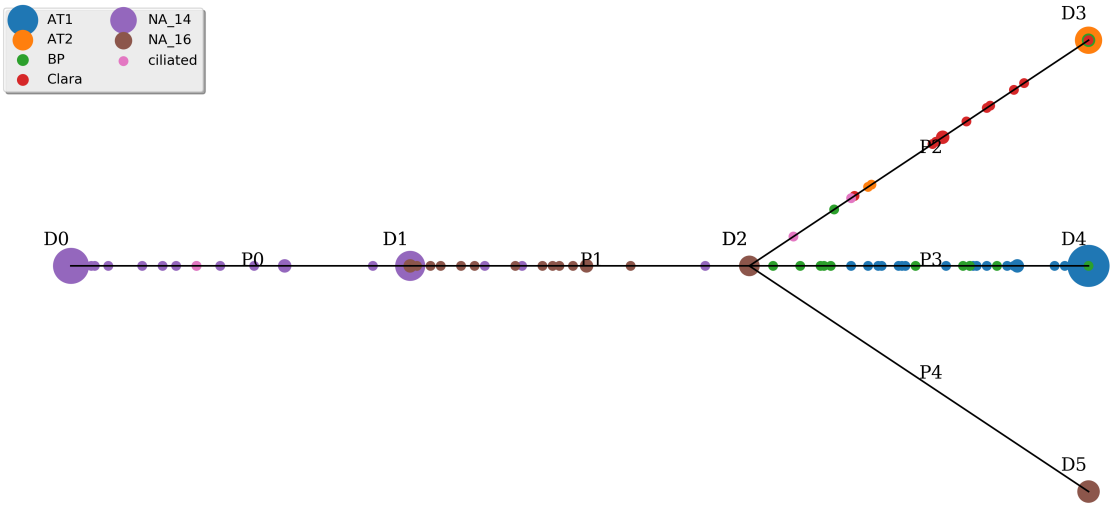

Figure 2.17: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 30% dropout rate.

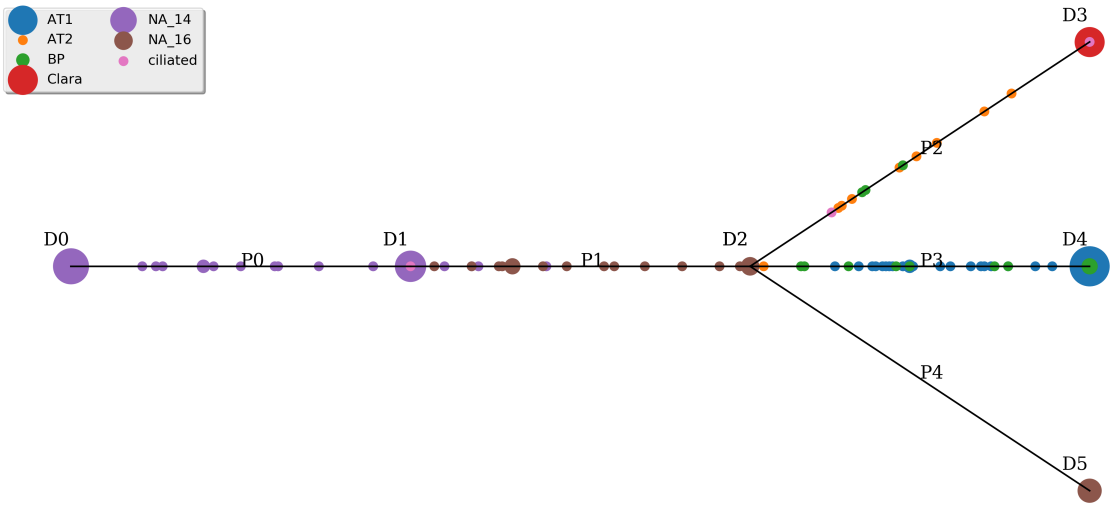

Figure 2.18: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 35% dropout rate.

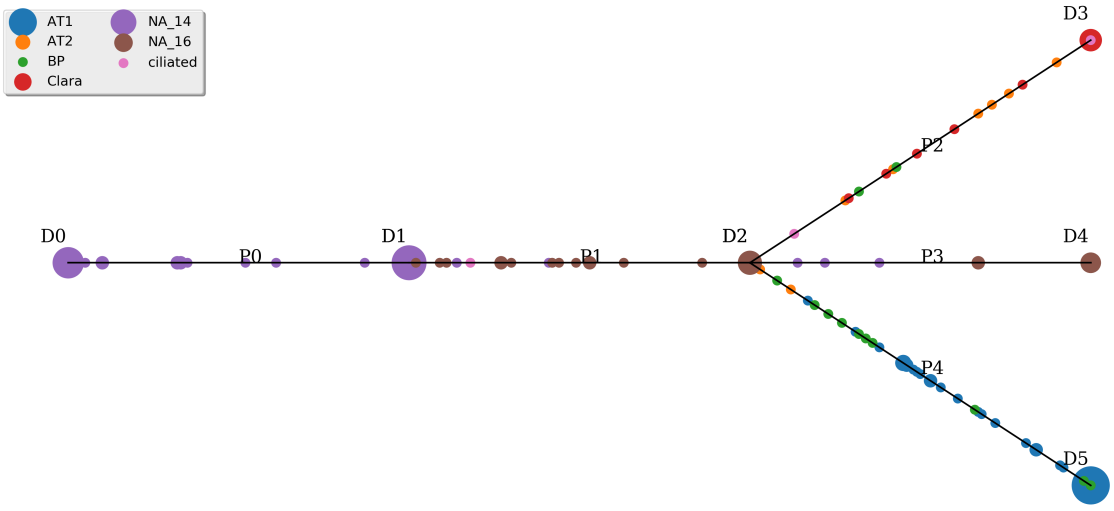

Figure 2.19: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 40% dropout rate.

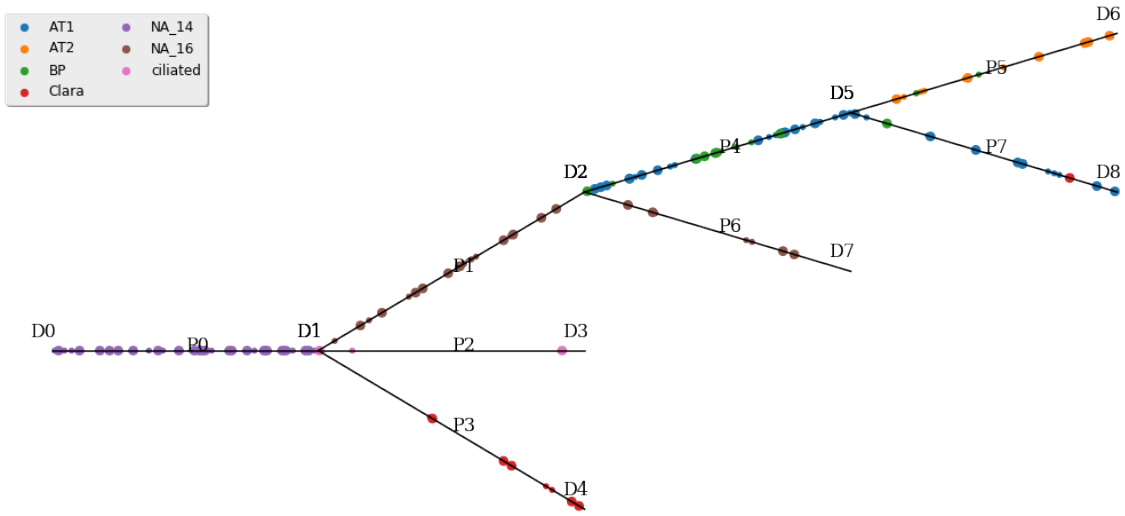

Figure 2.20: The initial CSHMM model structure and continuous cell assignment for lung developmental dataset.

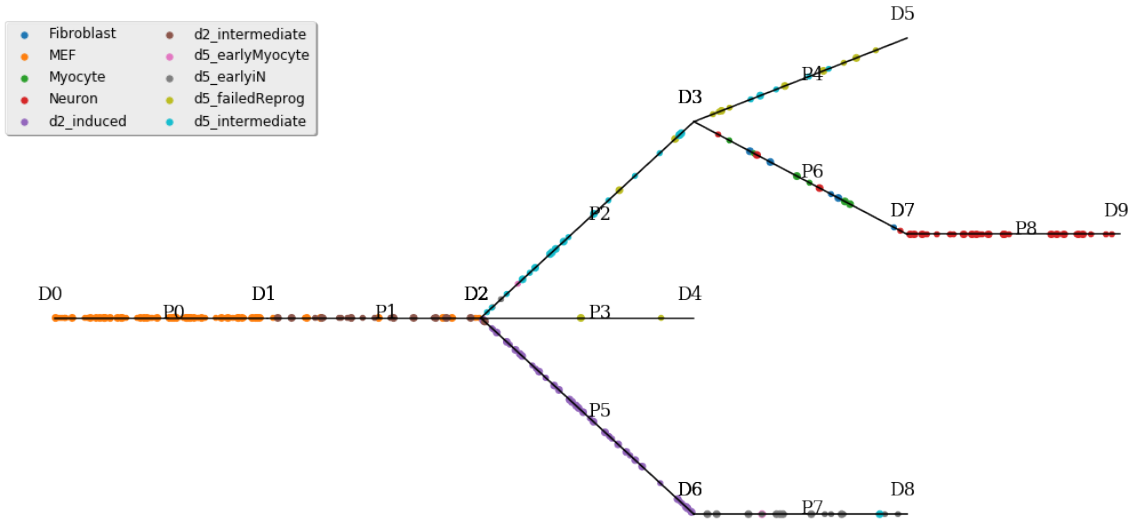

Figure 2.21: The initial CSHMM model structure and continuous cell assignment for neuron developmental dataset.

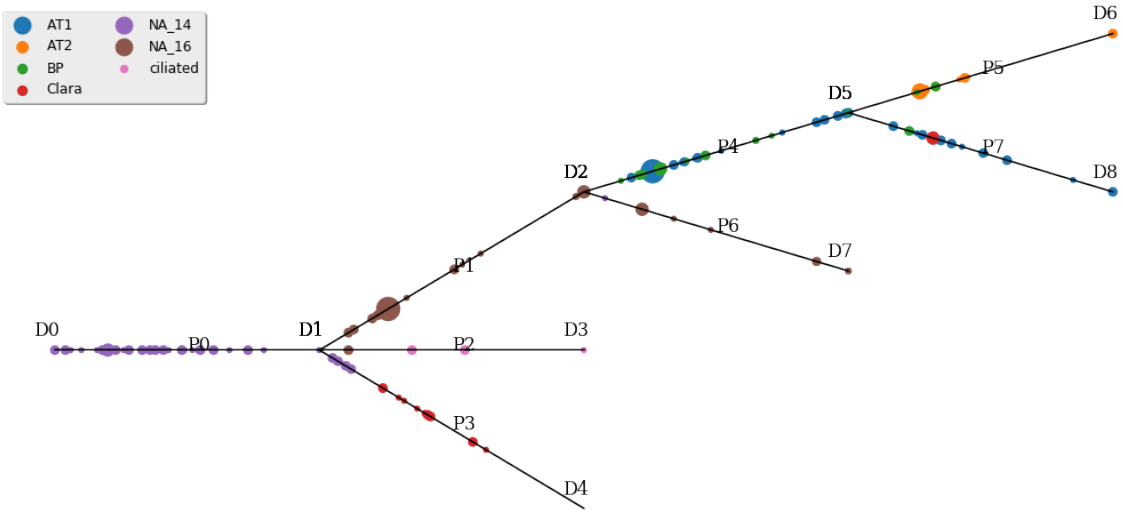

Figure 2.22: The CSHMM model structure and continuous cell assignment for lung developmental dataset with λ_g parameter 0.1.

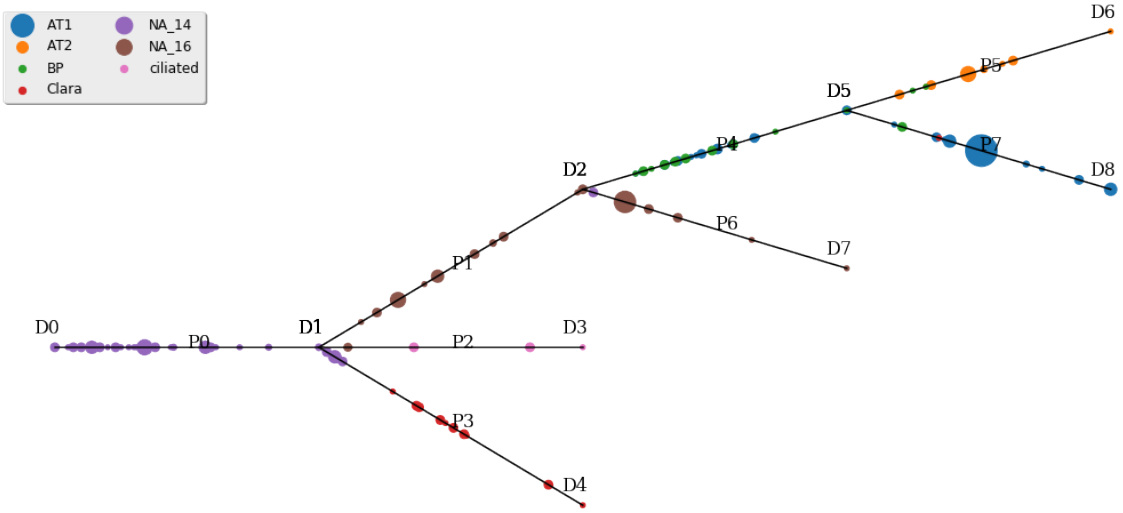

Figure 2.23: The CSHMM model structure and continuous cell assignment for lung developmental dataset with λ_g parameter 0.5.

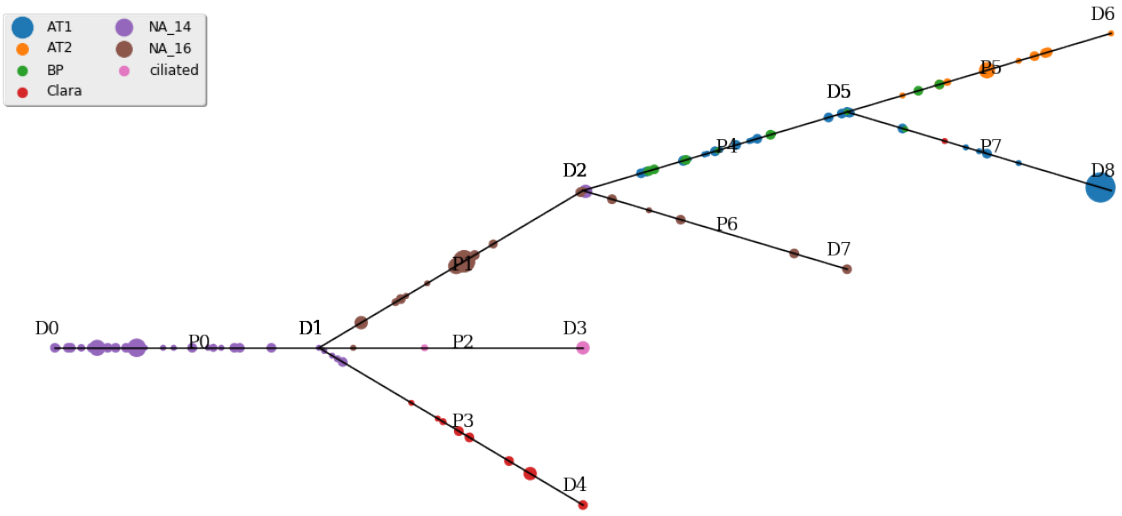

Figure 2.24: The CSHMM model structure and continuous cell assignment for lung developmental dataset with λ_g parameter 0.8.

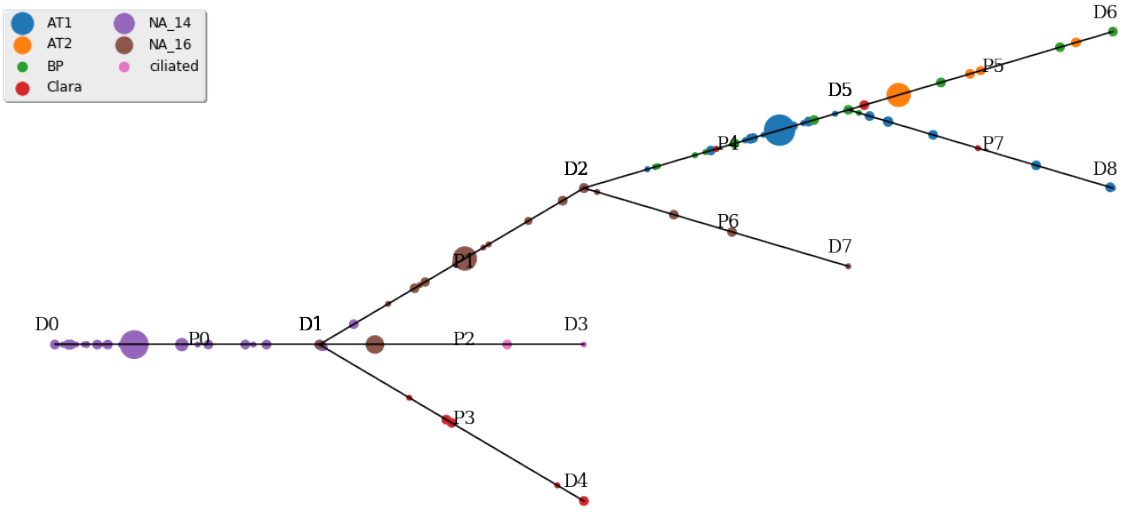

Figure 2.25: The CSHMM model structure and continuous cell assignment for lung developmental dataset with λ_g parameter 2.

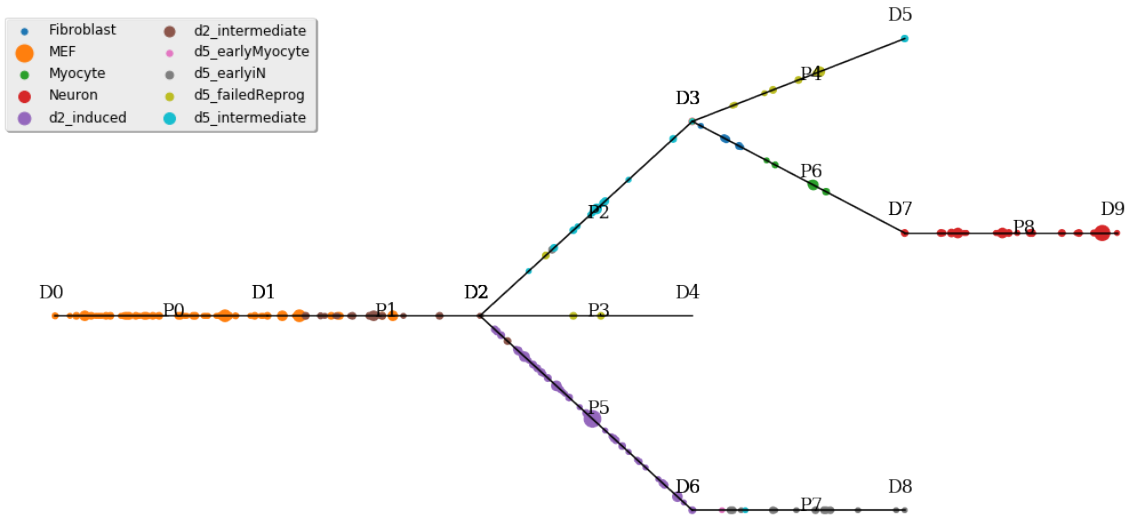

Figure 2.26: The CSHMM model structure and continuous cell assignment for neuron developmental dataset with λ_g parameter 0.1.

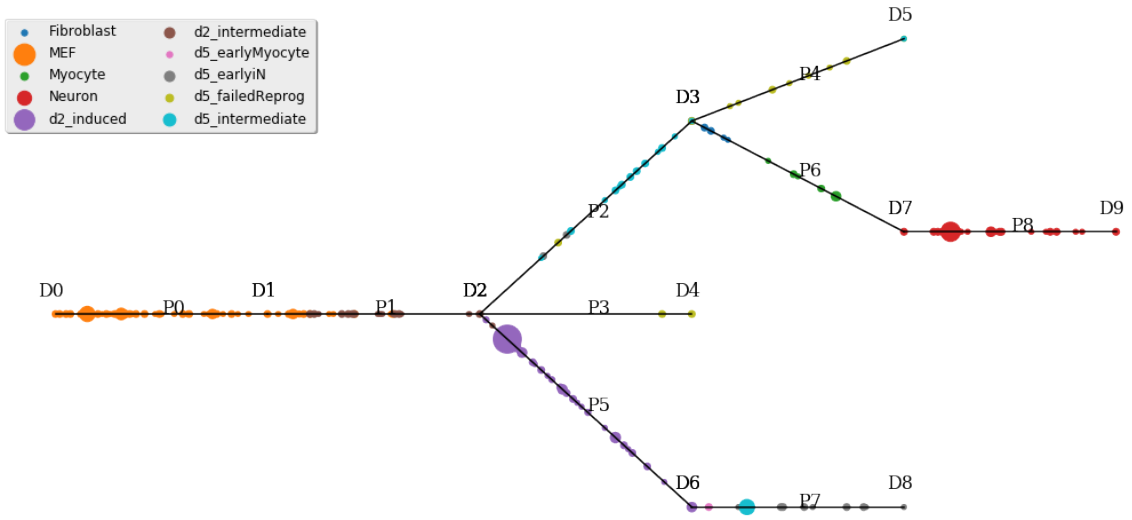

Figure 2.27: The CSHMM model structure and continuous cell assignment for neuron developmental dataset with λ_g parameter 0.5.

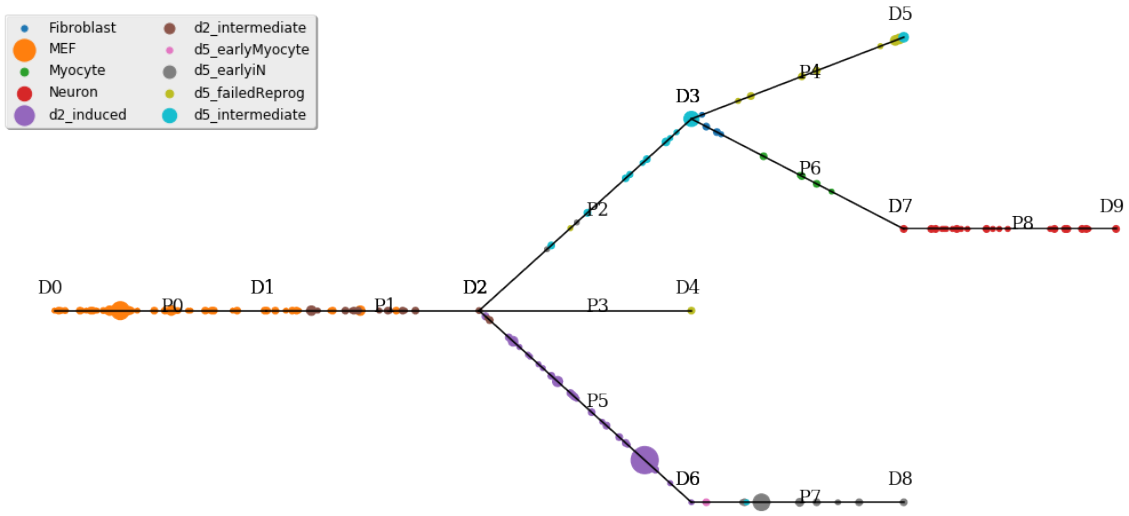

Figure 2.28: The CSHMM model structure and continuous cell assignment for neuron developmental dataset with λ_g parameter 0.8.

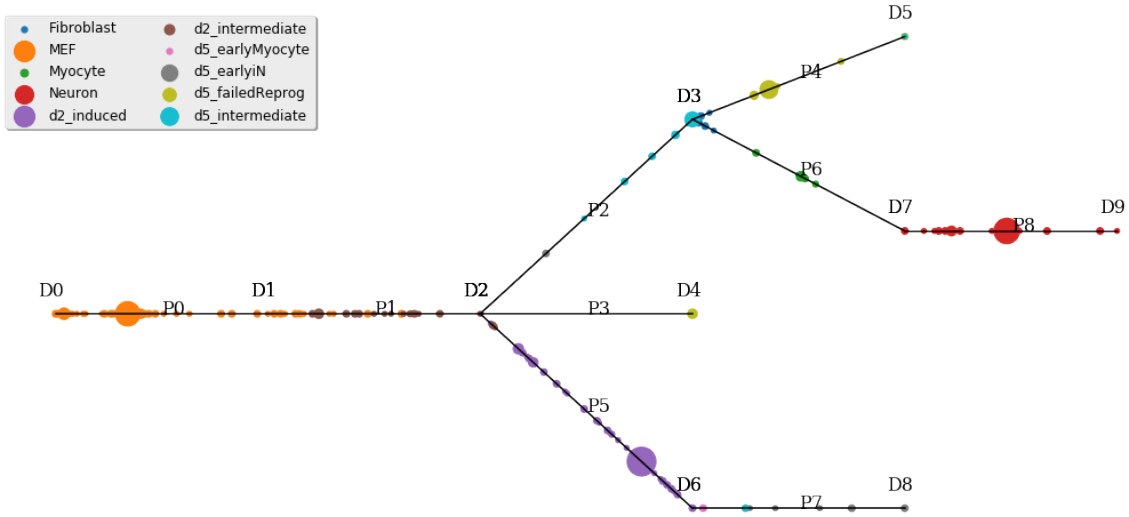

Figure 2.29: The CSHMM model structure and continuous cell assignment for neuron developmental dataset with λ_g parameter 2.

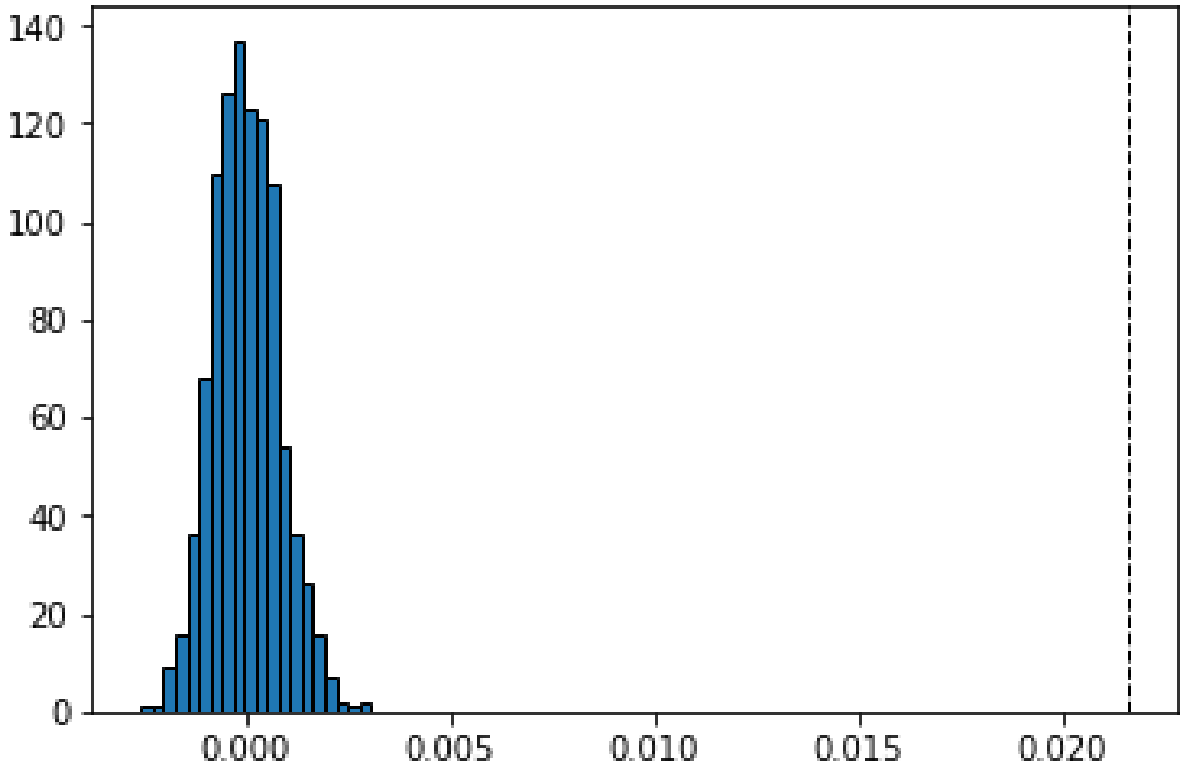

Figure 2.30: The histogram of adjusted random index (ARI) on 1000 random experiments of zebrafish dataset. The dashed line is the result of CSHMM. The p-value is less than 10^{-10}.

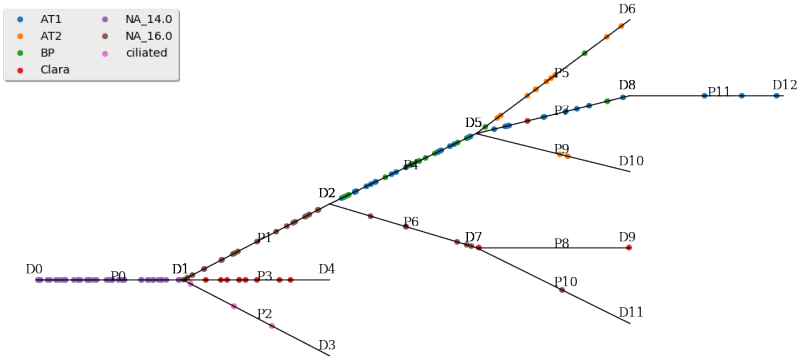

Figure 2.31: The CSHMM model structure and continuous cell assignment for lung developmental dataset with noisy initial structure. This is the structure after 0 iteration.

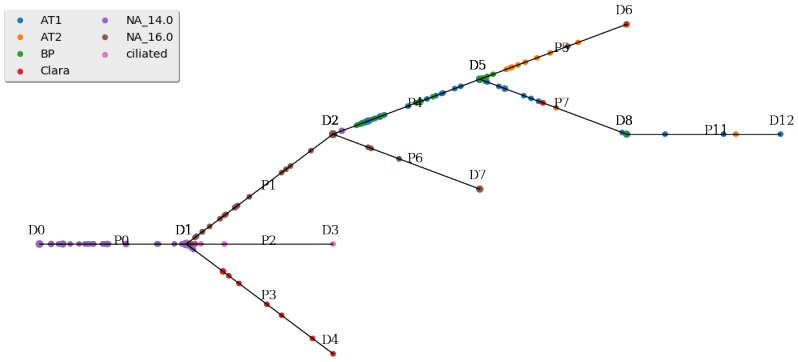

Figure 2.32: The CSHMM model structure and continuous cell assignment for lung developmental dataset with noisy initial structure. This is the structure after 1 iteration.

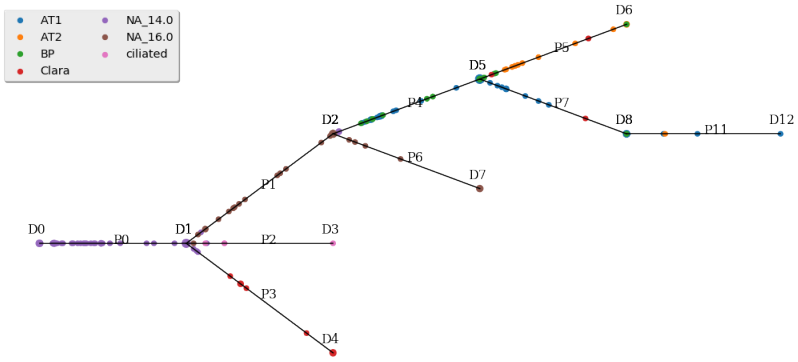

Figure 2.33: The CSHMM model structure and continuous cell assignment for lung developmental dataset with noisy initial structure. This is the structure after 2 iterations.

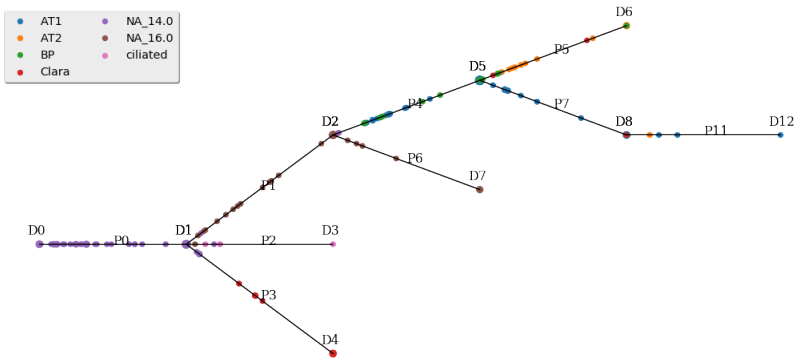

Figure 2.34: The CSHMM model structure and continuous cell assignment for lung developmental dataset with noisy initial structure. This is the structure after 3 iterations.

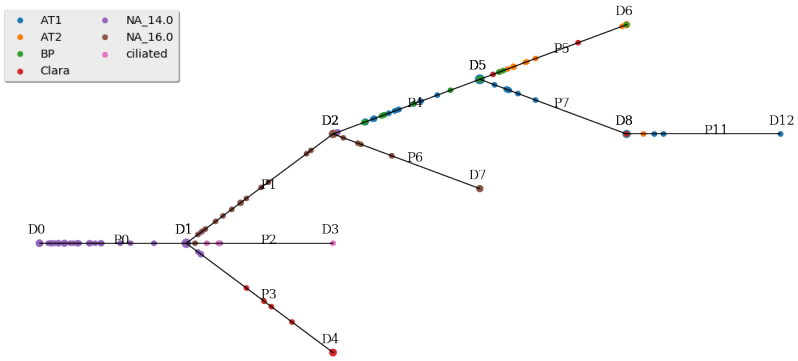

Figure 2.35: The CSHMM model structure and continuous cell assignment for lung developmental dataset with noisy initial structure. This is the structure after 4 iterations.

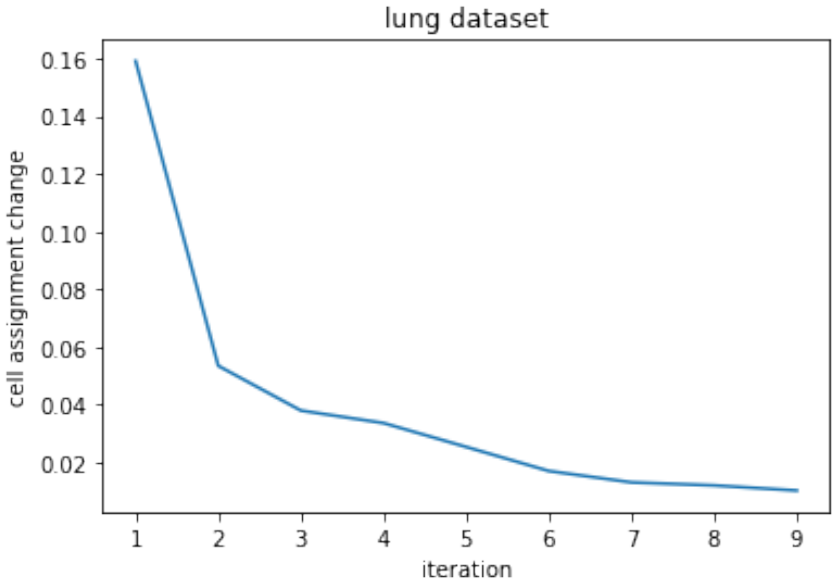

Figure 2.36: The average cell assignment change of lung development dataset for each iteration during training. We can observe an elbow shape happens around the second iteration

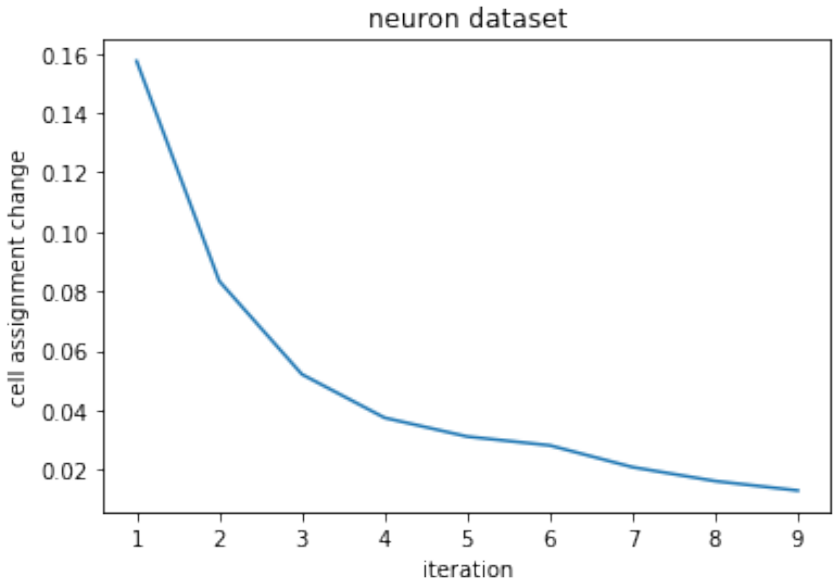

Figure 2.37: The average cell assignment change of neuron reprogramming dataset for each iteration during training. We can observe an elbow shape happens around the third iteration

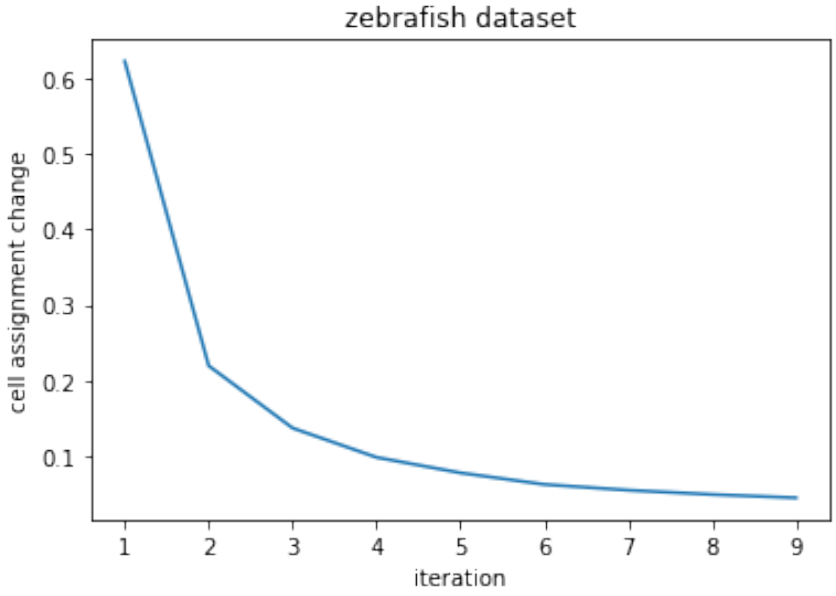

Figure 2.38: The average cell assignment change of zebrafish dataset for each iteration during training. We can observe an elbow shape happens around the second iteration

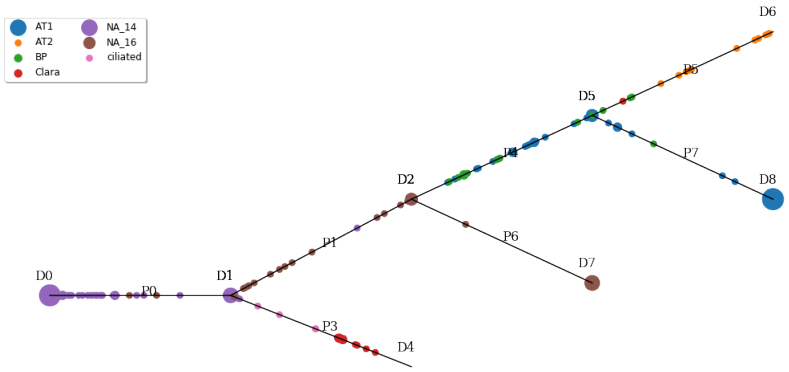

Figure 2.39: The CSHMM model structure and continuous cell assignment for lung developmental dataset with random seed 1.

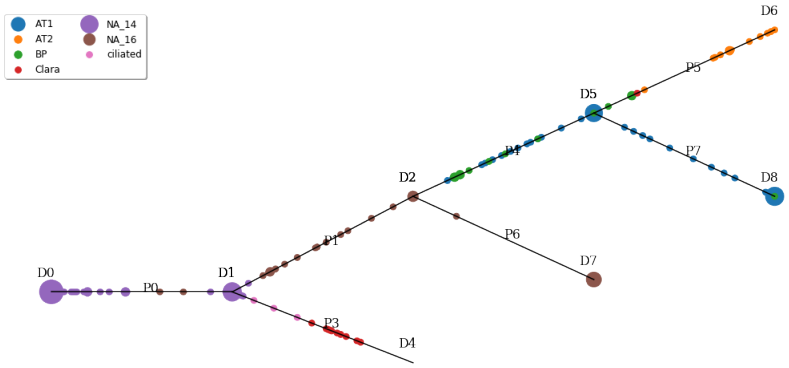

Figure 2.40: The CSHMM model structure and continuous cell assignment for lung developmental dataset with random seed 2.

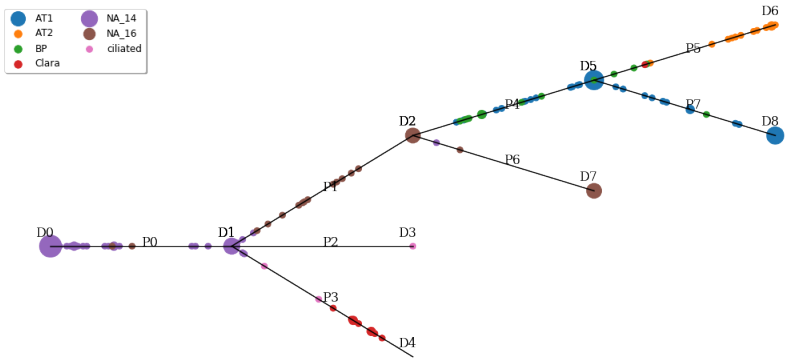

Figure 2.41: The CSHMM model structure and continuous cell assignment for lung developmental dataset with random seed 3.

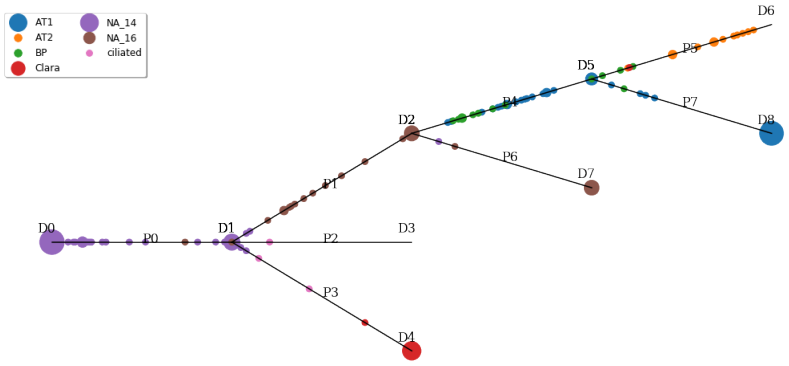

Figure 2.42: The CSHMM model structure and continuous cell assignment for lung developmental dataset with random seed 4.

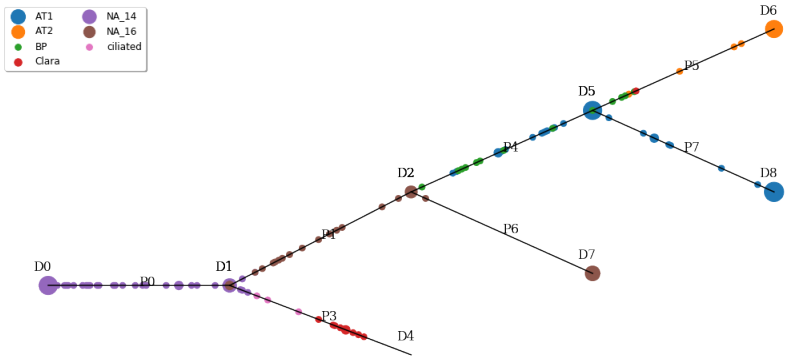

Figure 2.43: The CSHMM model structure and continuous cell assignment for lung developmental dataset with random seed 5.

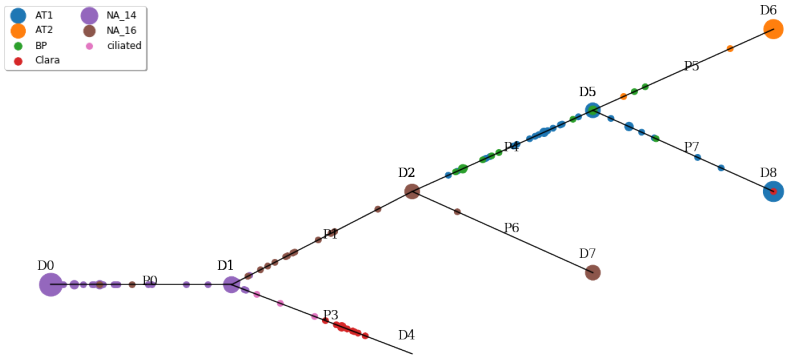

Figure 2.44: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of genes and random seed 1.

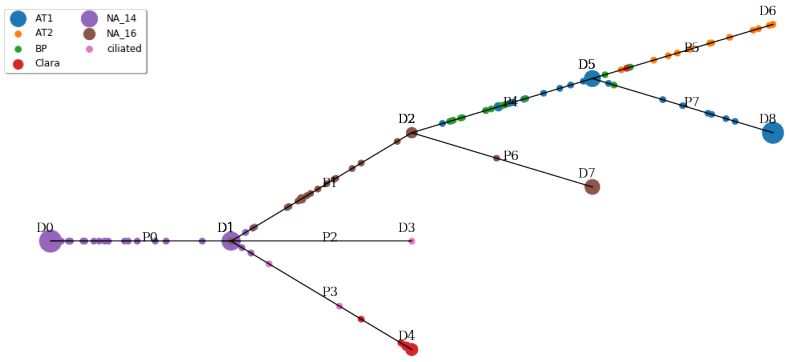

Figure 2.45: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of genes and random seed 2.

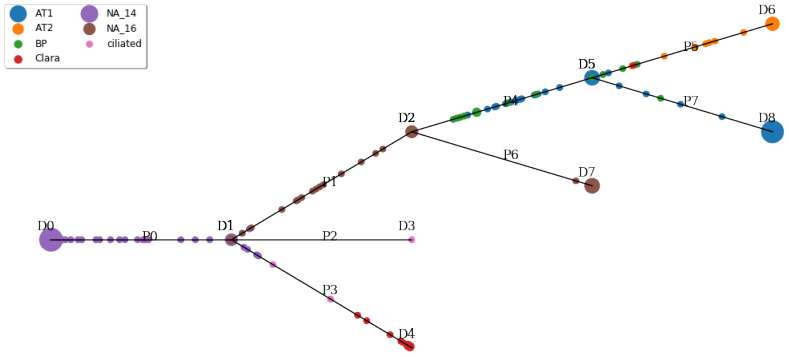

Figure 2.46: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of genes and random seed 3.

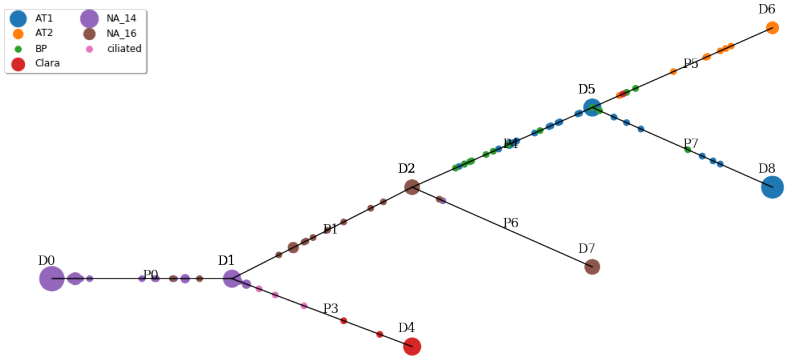

Figure 2.47: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of genes and random seed 4.

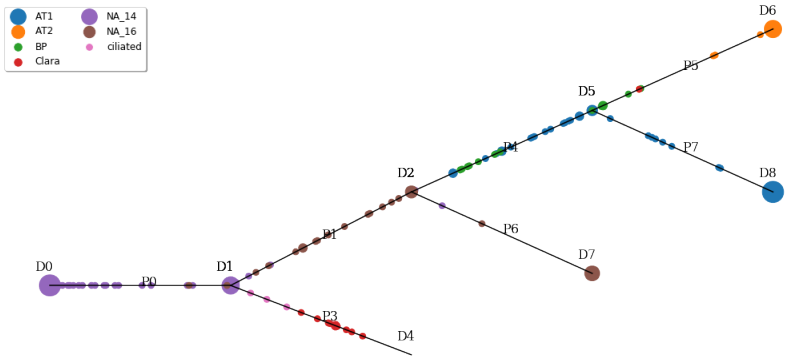

Figure 2.48: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of genes and random seed 5.

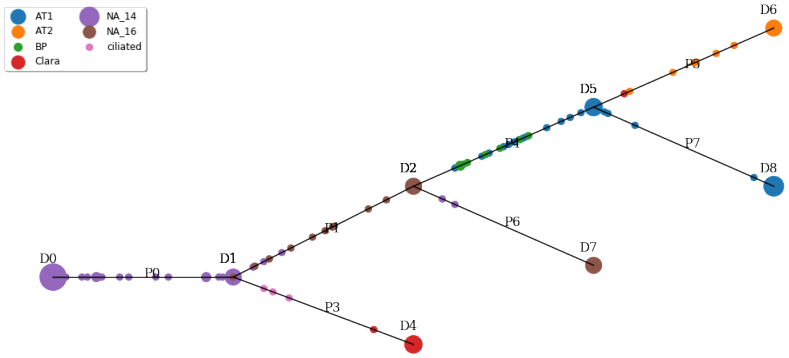

Figure 2.49: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of cells and random seed 1.

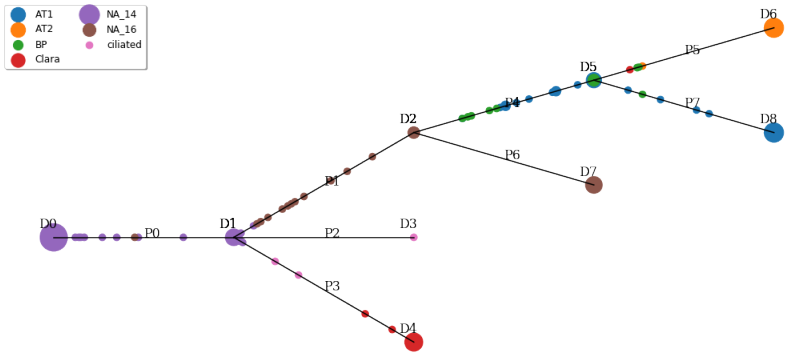

Figure 2.50: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of cells and random seed 2.

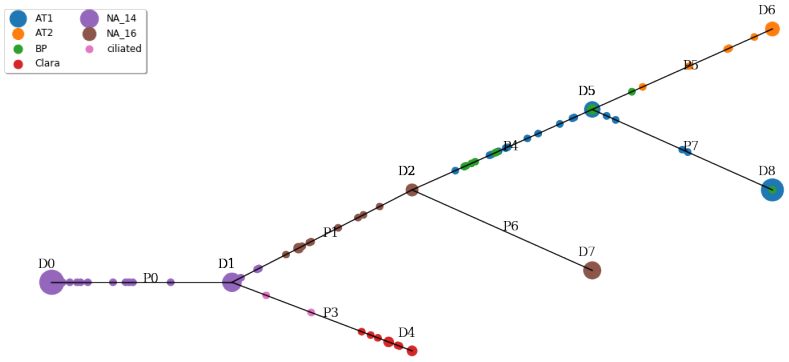

Figure 2.51: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of cells and random seed 3.

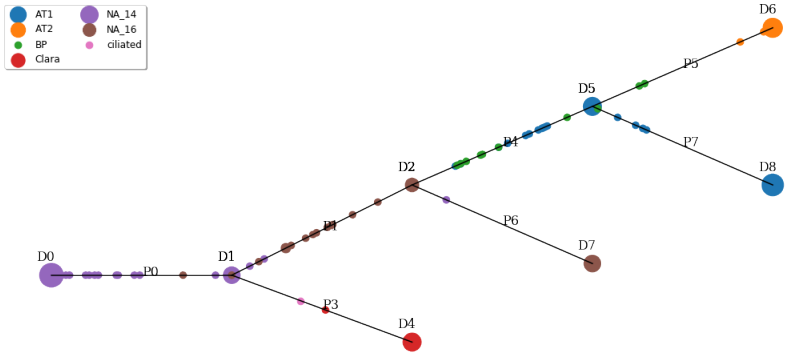

Figure 2.52: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of cells and random seed 4.

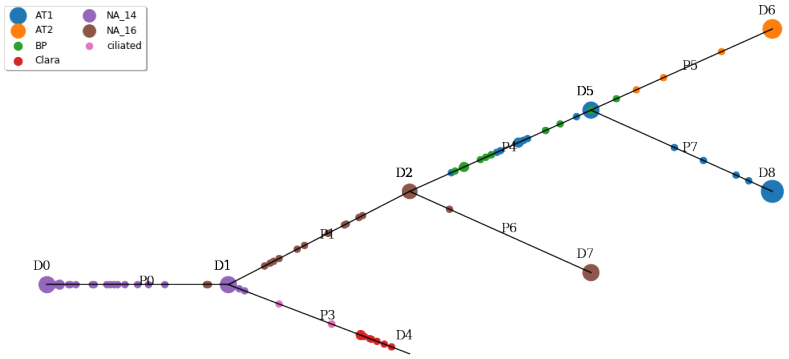

Figure 2.53: The CSHMM model structure and continuous cell assignment for lung developmental dataset with 80% of cells and random seed 5.

Chapter 3

CSHMM-TF: Inferring TF activation order in time series scRNA-Seq studies

While our CSHMM method is able to assign cells to their paths and time, it does not provide detailed information about the regulation of the activity in these cells. Such information can be obtained by using the TF-target information. We have extended the formulation of CSHMM to fit in the TF-target information and thus developed CSHMM-TF. This chapter describes CSHMM-TF and has been adapted with changes from our paper [117] Lin C, Ding J, Bar-Joseph Z (2020) "Inferring TF activation order in time series scRNASeq studies." PLoS Comput Biol 16(2): e1007644. https://doi.org/10.1371/journal.pcbi.1007644.

As we have mentioned in Chapter 1, previous methods of modeling regulatory relationships to improve cell developmental trajectories suffer from their own limitations. Most of them only perform post-analysis on fully-trained model so the information of regulating TFs are not used to improve the model. Some can utilize the identified TF information to further iteratively improve the model but these model only identify TFs on pre-defined discrete number of time points making it hard to determine the actual time for TF activation. To address these issues we extended the CSHMM method described in the previous chapter for modeling dynamic scRNA-Seq branching data to take into account TF-gene interaction as well. We formulate an new CSHMM model (termed CSHMM-TF) in which the regulation by TFs influences the emission probabilities of the different paths. Using the revised model we associate TFs with different model paths and identify a specific activation time along the path for the different TFs. Applying our CSHMM-TF to several mouse and human scRNA-Seq datasets, we show that by using this information the resulting models are more accurate compared to models that do not use TF-gene interaction information. We also discuss the combinatorial aspects of TF regulation and show that many of the TFs assigned to the same paths are indeed working together to regulate genes. Finally, we study the dynamic of TFs activation by looking at early and late TFs for the same path (or genes) and use this to raise novel hypotheses regarding TF activation order.

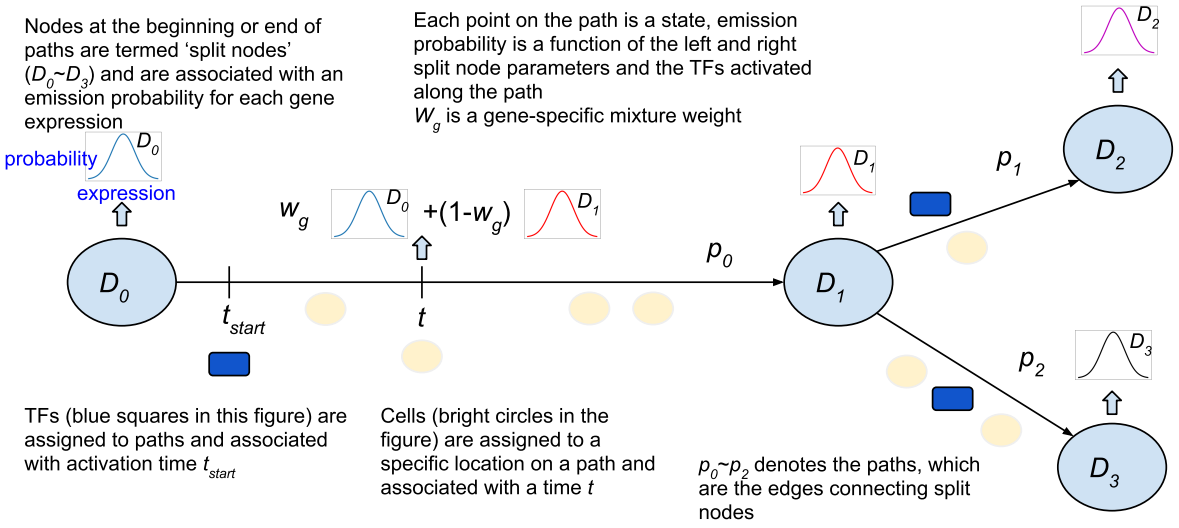

Figure 3.1: CSHMM-TF model structure and parameters. The figure presents the assignments of cells and TFs to the reconstructed branching model for the process studies. Each edge (path) represents a set of infinite states parameterized by the path number and the location along the path. We use a function based on parameters learned for the split nodes (nodes at the start and end of each path) and TF assignments to define an emission probability. Emission probability for a gene along a path is a function of the location of the state and prior TFs (t and t_{start}) and a gene specific parameter k which controls the rate of change of its expression along the path. Split nodes are locations where paths split and are associated with a branch (transition) probability. The t_start parameter defines the TF activation time for a specific TF associated with the path. Cell assignment to paths is determined by the emission probabilities and the expression of specific TF targets for the TFs associated with the path. w is a vector of *gene-specific* mixture weight, where the weights are a non linear function which depends on (t and t_{start}). See text for more details.

3.1 CSHMM-TF formulation

CSHMM-TF extends the formulation of CSHMM for time-series scRNA-Seq data (Chapter 1) by adding TF regulation information to each path (edge). In addition, the model also assigns the *time* at which a TF is impacting its targets. The model assigns both activators and repressor TFs. For simplicity we are using the term "TF activation" when discussing this assignment though the actual direction of the impact is calculated independently of the timing assignment and as mentioned above can be either positive or negative. Our method uses TF targets to infer TF activity since several prior studies have shown that the expression of many TFs does not adequately reflect their activation profiles as many of them are post-transcriptionally and post-transcriptionally regulated. In contrast, the activity of target genes is often a better proxy for TF activity [172]. The assignment of continuous activation time also allows the model to infer combinatorial regulatory relationships (if two TFs are assigned to regulate the same path) and in some cases to infer the order of the recruitment process for different TFs regulating the same gene. Figure 3.1 presents the CSHMM-TF structure. In the figure, we denote a few states as split nodes ($D_0 \sim D_3$ nodes). These are the states in which cells are allowed to split to two or

more branches and they represent important split stages for cell lineages. The edges between split nodes are denoted as paths ($p_0 \sim p_2$) and each contains infinitely many states such that each point on a path corresponds to an state. States are parametrized by their location w.r.t the two split nodes at the end of the path they reside on. Each of the split nodes is associated with a branch probability B. For each state (including split nodes), we define an emission probability by determining parameters for a multivariate Gaussian distribution which, following previous work, assumes independence for gene specific expression levels conditioned on the state [176]. The main difference between CSHMM-TF and CSHMM is that the formulation of CSHMM-TF utilizes TF-gene interaction information to change the likelihood function of cell assignments to paths. The assignment of a TF to path, and its inferred activation time (t_{start}) directly affects the emission probability of cells assigned to locations on the paths that follow the start time of the TF. To formulate the emission probabilities in CSHMM-TF we use $s_{p,t}$ to represent a specific state where $0 \leq t \leq 1$ is a pseudo time on path $p(D_a \rightarrow D_b)$, and a, b are the indices of split nodes. Denote by x_j^i the expression of gene j in cell i, the emission probability for gene j in cell i assigned to state $s_{p,t}$ is modeled as a Gaussian distribution with mean $\mu_{j,s_{p,t}}$ and variance σ_j: $x_j^i \sim N(\mu_{j,s_{p,t}}, \sigma_j^2), P(x_j^i | s_{p,t}^i, \theta) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp(-\frac{(x_j^i - \mu_{j,s_{p,t}})^2}{2\sigma_j^2})$. Where

$$\begin{aligned}\mu_{j,s_{p,t}} &= g_{aj} \exp(-K_{p,j}t') + g_{bj}(1 - \exp(-K_{p,j}t')) = g_{bj} + (g_{aj} - g_{bj}) \exp(-K_{p,j}t') \\ &= g_{bj} + (g_{aj} - g_{bj}) \exp(-K_{p,j} \max(0, t - t_{j,start})) \end{aligned} \tag{3.1}$$

Here, θ is the set of model parameters (see Table 3.1). g_{aj} is the mean expression for gene j at split node a. We assume a continuous change in expression for a subset of the genes along a path (from left split node g_a to right split node g_b with a mixture weight $w_j = \exp(-K_{p,j}t')$). Note that this weight is gene specific and depends in part on the TFs predicted to regulate that gene. To allow different genes to change non-linearly at different rates across the path (some at the beginning while others at the end) we use a gene specific parameter $K_{p,j}$ to denote the rate of change. For genes regulated by TFs that do not change at the start of the path we use $t' = \max(0, t - t_{start})$. Here, t is the time assignment of the cell, $t_{j,start}$ is the TF activation time for TF regulating gene j, which we discuss in more detail below. For genes not regulated by any TF assigned to this path, or those regulated by TFs that are activated at the start of the path, $t' = \max(0, t - t_{start})$ is equal to t. We also attempted to include dropout probability using a mixture weight model in the emission probability, however, this did not change the performance of CSHMM-TF much and so is omitted here. These notations are enough to define the parameters required to specify a CSHMM-TF: $\theta = (V, \pi, S, A, E')$. All symbol definitions are presented in Table 3.1. In Appendix B Supporting Methods we prove that our definition of CSHMM-TF leads to a valid continuous state HMM and also provide additional details of the definition of transition probabilities for CSHMM-TF.

3.2 Assigning regulating TFs to each path

To predict regulating TFs for each path we extend methods that only allow discrete time assignments to TF activity [51]. We first remove TFs that are expressed in less than 20% of cells in the path. Next, we determine differentially expressed (DE) genes by performing a t-test

Table 3.1: Parameters of the CSHMM-TF model: $\theta_{CSHMM-TF} = (V, \pi, S, A, E')$

symbol	definition
V	the observation alphabet $\subset \mathbb{R}^G$ (the possible input set)
π	the initial probability for each state, $\pi_{s_{0,0}} = 1$
S	the set of states (each path has infinitely many states) $s_{p,t}$ denotes the hidden state of path p, pseudo time t
B	the branch probability defined on each pair of paths, $\sum_{j \in P} B_{i,j} = 1, 0 \leq B_{i,j} \leq 1 \quad \forall i, j \in P$
A	the transition probability defined on any pair of states s_{p_i,t_i} and s_{p_j,t_j}
$E' = (K, g, \sigma^2, \Omega, \Phi)$	the parameters associated with emission probability for a given state
K	$K = \{K_1, ..., K_{\|P\|}\} \subset \mathbb{R}^G$, $K_{p,j}$ denotes the gene changing speed for gene j at path p
g	$g = \{g_1, ..., g_{\|D\|}\} \subset \mathbb{R}^G$, $g_{d,j}$ denotes the mean gene expression of gene j for split nodes d g_d denotes the mean gene expression vector of split node d
$\sigma^2 \subset \mathbb{R}^G$	the variance vector for genes
$\Omega \subset \mathbb{R}^{G \times \|F\|}$	the matrix where each entry $\Omega_{i,j}$ is 0 or 1 denoting whether gene i is regulated by TF j or not
$\Phi \subset \mathbb{R}^{\|P\| \times \|F\|}$	the matrix where each entry $\Phi_{i,j}$ denoting the relationship of path i and TF j. Where -1 means no relationship, $0 \leq t_{start} \leq 0.5$ means TF j is assigned to path i with time t_{start}
D	the set of split points
P	the set of paths
G	the number of genes (dimension of data)
F	the set of TFs
λ_g	the hyper parameter for the L1 regularization that controls the sparsity of Δg for every path p

between cells assigned to the current and parent path (Appendix B Supporting Methods). After we identify the set of DE genes, we use the TF-target information (Ω parameter) obtained from [59, 175] to calculate the p-value (based on hyper-geometric distribution) for each TF for this path. Details about the how the TF-target information is provided in Appendix B Supporting Methods. We keep TFs with a p-value ≤ 0.05 (p-value obtained by binomial test) with an upper bound of 10 TF for each path. The method for assigning TFs in each path is presented in Appendix B Supporting Methods (in the section "Assigning pseudo time to TF regulating a path").

3.3 Adjusting regularization parameters based on TF assignments

We assume that most genes do not change in a specific path (i.e. developmental branching is only affecting a subset of the genes). Based on this we regularize the gene expression difference vector (Δg) which represent the change in expression for each gene between the two nodes that define a path (start and end). We use a L1 regularization with parameter λ_g, where larger λ_g means more strict regulation. To incorporate TF information to this regularization (given our assumption that genes regulated by path specific TFs are more likely to change in that path) we use instead $\frac{\lambda_g}{1+\alpha_{p,j}}$ as the regularization term. Here $\alpha_{p,j}$ is the probability that the expression of gene j will change along path p (and so the higher the probability the lower the regularization for gene j). $\alpha_{p,j}$ is estimated by fitting a logistic regression model for all genes regulated by TFs on path p. Such changes in the regularization parameters allow genes that are targets of

assigned TFs to change more than other genes for which no explanation for change in expression is determined by the model.

3.4 Likelihood function for the CSHMM-TF model

We use the following notations: we assume we have N cells. Let X^i denote the expression profile of cell i and let $y^i = s^i_{p,t}$ be the hidden state denoting that cell i is assigned to path p with pseudo time t. Δg_p is the difference vector for the expression values at the endpoints of path p. Using notations defined above the log-likelihood with L1 regularization term is:

$$\begin{aligned} l(\theta|X,Y) &= \sum_{i=1}^{N} \log P(X^i, y^i|\theta) + \log(\text{L1 regularization term}) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{G} \log P(x^i_j|s^i_{p,t},\theta) + \sum_{i=1}^{N} \log P(s^i_{p,t}|\theta) + \sum_{p\in P}\sum_{j=1}^{G} -\frac{\lambda_g}{1+\alpha_{p,j}}|(\Delta g_p)_j| \end{aligned} \tag{3.2}$$

$$\text{Where, } P(s^i_{p,t}|\theta) = \prod_{\substack{q\in\text{branch probability}\\ \text{from root to } p}} q \quad \text{(the branch probability)} \tag{3.3}$$

$$\begin{aligned} &P(x^i_j|s^i_{p,t},\theta) \quad \text{(the emission probability)} \\ &= \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x^i_j - g_{bj} - (g_{aj}-g_{bj})\exp(-K_{p,j}\max(0, t-t_{j,start})))^2}{2\sigma_j^2}\right) \end{aligned} \tag{3.4}$$

Where (g_a, g_b) refers to the mean gene expression of the split point at both ends of a path. Briefly, the log-likelihood shown in equation 3.2 contains three terms. The first, further expanded in equation 3.4, represents the emission probability of each cell. Note that in this part we use a modified cell time t' as we have discussed previously. The second, expanded in equation 3.3, represents the penalty we use for cells assigned on later (more specific) paths. The idea is similar to prior probabilistic methods for reconstructing branching trajectories [155]: earlier stages are often less specific (higher entropy [201], while later stages (representing specific fates) have a tighter expression profile. Thus, cells that represent specific cell types will still be assigned to their correct (late) stage based on their expression profile while noisier cells would be assigned to the earlier stages. The last term in equation 3.2 is the new L1 regularization term, where the L1 parameter has been replaced as we have discussed previously.

3.5 Model initialization, learning and continuous cell assignments

For model initialization, the advantages of the SCDIFF initialization method[51] for CSHMMs have been previously discussed in[115]. Based on these results we use the same initialization for CSHMM-TF as well. Specifically, we first construct a discrete branching model based

on the time-series scRNA-Seq data only. This step includes performing clustering for each time point, adjusting the level of the clusters based on time point information, and constructing a tree-branching model from the clusters. While initial assignment is based on the time information, cells can be re-assigned to different tree branches (representing other time points) as part of the iterative learning of the model. In this model, which uses prior methods for pseudotime ordering (SCDIFF [51]) cells are assigned to discrete nodes rather than continuously to paths, and no TF information is used. Next, we assign cells in each internal node to a random location along the corresponding developmental path that is incoming to that node leading to an initial continuous model. Details about model initialization for CSHMM-TF are presented in Appendix B Supporting Methods. For model learning and continuous cell assignments, we adopt the Expectation-Maximization algorithm (EM), where in the E-step we do the continuous cell assignments; in the M-step we try to maximize the likelihood of CSHMM-TF with Maximum Likelihood Estimation (MLE) and sampling. We iterate between E-step and M-step to improve the likelihood of the model. Figure 3.2 presents a flowchart for the steps used when learning CSHMM-TF. Details about parameter learning for CSHMM-TF are also presented in Appendix B Supporting Methods.

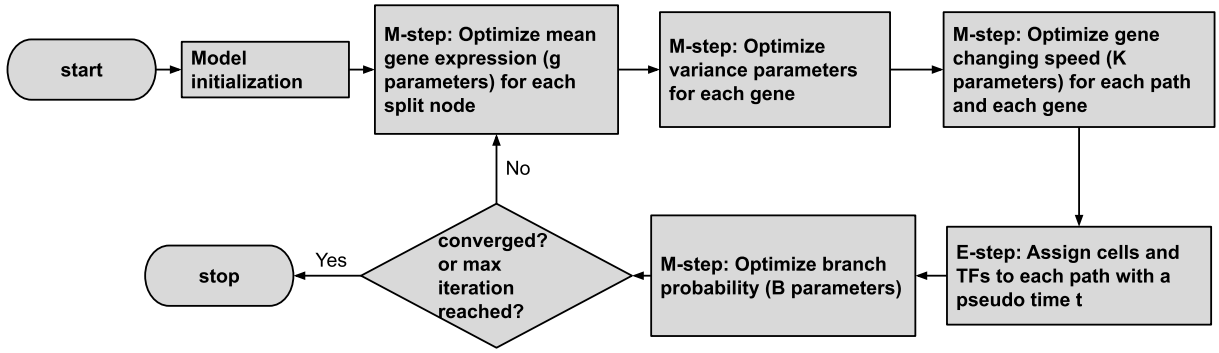

Figure 3.2: Flow chart of how to iteratively learn CSHMM-TF

3.6 Results

3.6.1 Application of CSHMM-TF to time series scRNA-Seq data

We applied CSHMM-TF to several time series scRNA-Seq datasets in human and mouse. The number of cells in the datasets ranged from 152 (mouse lung data) to $\sim$ 21K (mouse cortex data). Datasets were processed by removing genes with overall low expression (following[51]). Following this step the number of genes in the models ranged from 10-18K. Details about data processing information is available in the Appendix B Supporting methods. Details about how TF-gene interaction information is obtained is provided in Appendix B Supporting methods. The first is a human liver dataset with 765 cells, 19K genes, collected at 4 developmental stages [31]. The second studies human skeletal muscle myoblasts and contains 271 cells, 13K genes and 4 time points [206]. The third is from mouse and looks at differentiation of medial ganglionic eminences (MGC) to the Cortex [128]. This dataset contains $\sim$ 21K cells, $\sim$10K genes and 3

time points. The fourth is mouse embryonic fibroblasts (MEF) reprogramming to neurons [209]. It contains 252 cells, 12K genes and 4 time points. The fifth is a lung development dataset with 152 cells, 15K genes and 3 time points [208].

Figure 3.3-3.4 present the resulting CSHMM-TF models for the human liver data and the mouse lung developmental data with TF assignments. As can be seen in these figures, unlike prior methods that assign TFs to discrete branch points only [51, 78, 175, 214], CSHMM-TF can infer a more refined time for the activation of TFs. This helps improve the assignment of cells to different paths, to infer combinatorial TF regulation and to determine TF ordering as we show below. See also Appendix B Figure 3.6, 3.8, 3.10 and Table 3.5, 3.7 for results for the MEF reprogramming, myoblasts differentiation, and the cortex differentiation datasets, respectfully.

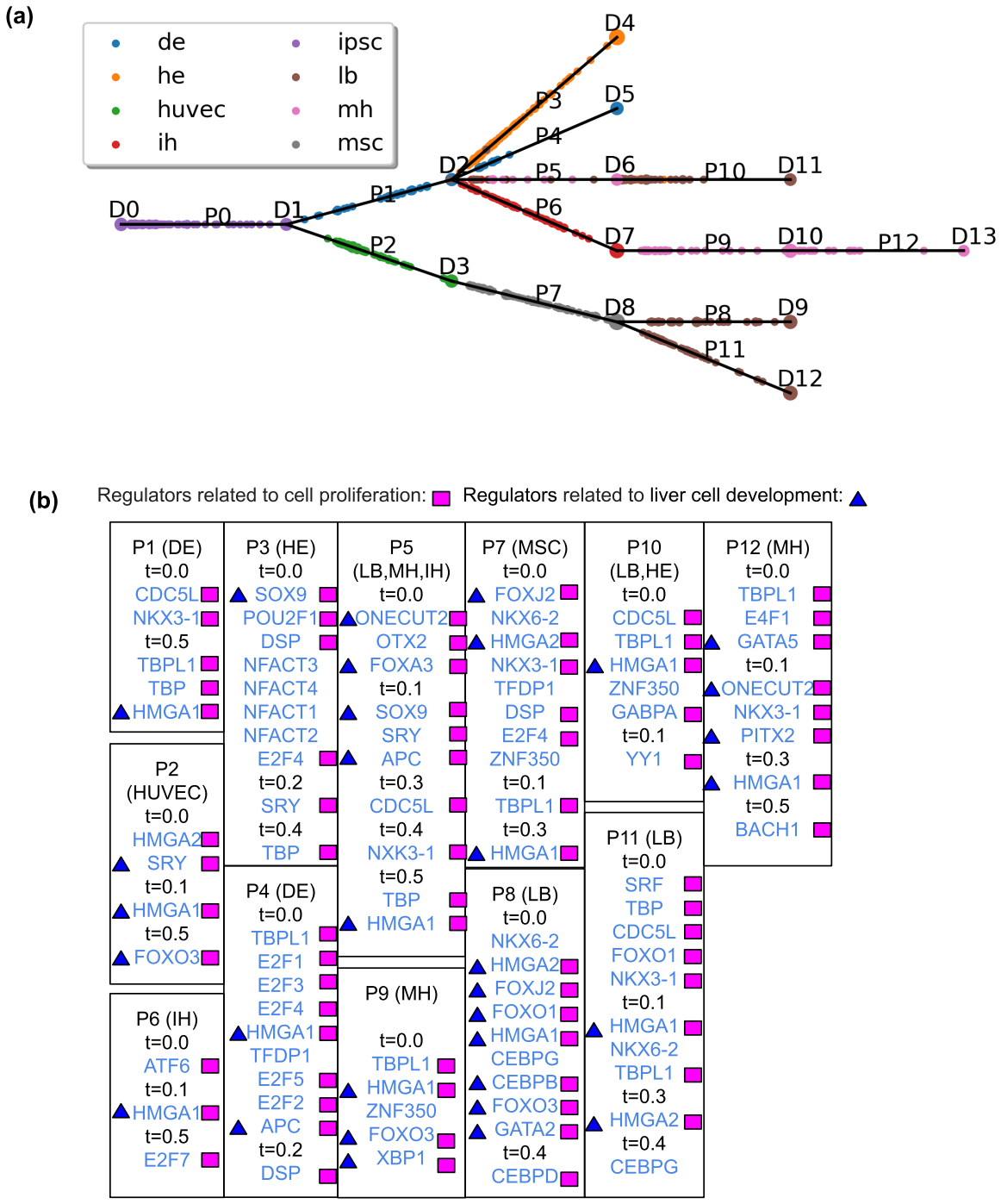

Figure 3.3: CSHMM-TF result for the liver dataset (a) CSHMM-TF structure and continuous cell assignment for the liver dataset. D nodes are split nodes and p edges are paths as shown in Figure 3.1. Each circle on a path represents cells assigned to a state on that path. The bigger the circle the more cells are assigned to this state. Cells are colored based on the cell type / time point assigned to them in the original paper. (b) TF assignments by CSHMM-TF for the liver dataset. We highlight known functional roles for several TFs. Path names (DE, LB etc.) are based on annotated cells assigned to that path in the figure above. Full names of cell types can be found on Appendix B Supporting methods of data collection and processing.

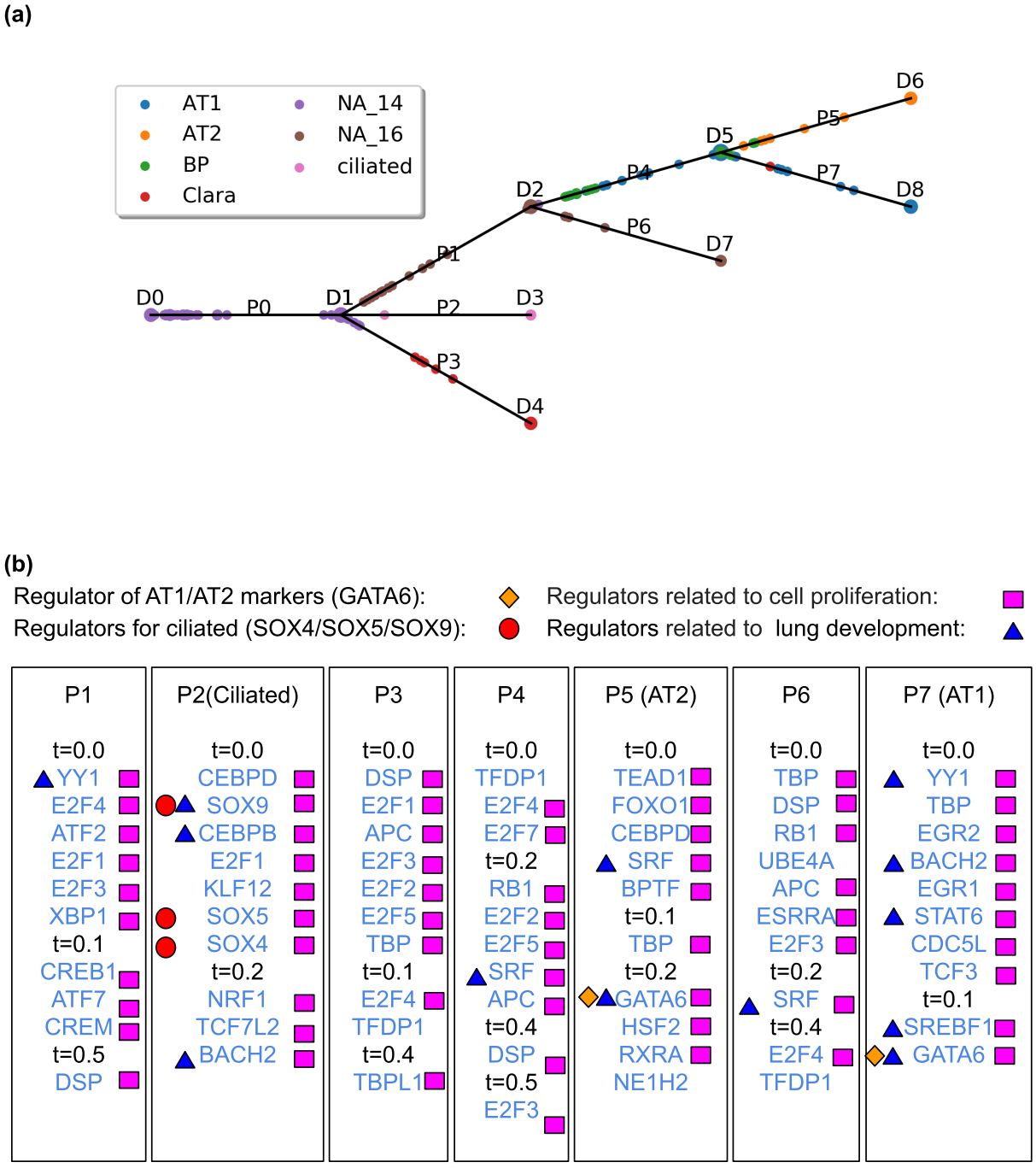

Figure 3.4: CSHMM-TF result for the lung development dataset (a) CSHMM-TF structure and continuous cell assignment for lung development dataset. Notations are similar to the ones described in Figure 3.3 (b) TF assignments to each path by CSHMM-TF. We highlight known functional roles for several TFs. Path names (Ciliated, AT1 etc.) are based on annotated cells assigned to that path in the figure above.

The reconstructed trajectories for the liver dataset (Figure 3.3 (a)), correctly reconstruct the

relationship of induced Pluripotent Stem Cells (iPSC) → DE (definitive endoderm) → HE (hepatic endoderm) and IH (immature hepatoblast-like) → MH (mature hepatocyte-like). For the lung dataset (Figure 3.4 (a)), CSHMM-TF correctly assigns cells, based on their known types, to terminal paths (ciliated, Clara, AT1 and AT2). Progenitor cells and BP cells are also correctly assigned to earlier paths.

3.6.2 Assigned TFs correctly match cell types in each path

Figure 3.3-3.4 (b) present TF assignment for CSHMM-TF for the liver and lung dataset. In the figures we highlight known functions related to development and the specific processes for several TFs. As can be seen, CSHMM-TF identifies known key regulators (Figure 3.3 (b)). For example, FOX family TFs are identified in several paths and are known to control the formation and function of the liver [108]. HMGA1 (identified in all path except P3) and HMGA2 (identified in P7, P8, P11) are known to be involved in several developmental processes [109, 244]. ONECUT2 regulates liver development and is required for liver bud expansion [125]. CEBPB, identified for path P8 which is the path for liver bud, is the marker of early liver development and expressed in the early liver bud[221]. GATA2 is important in hepatic cell fate decision [72]. SOX9 is also related to hepatogenic differentiation [143]. SRF is essential for hypatocyte proliferation and liver function [195]. PITX2 is related to the differentiation of induced hepatic stem cells [35]. See Appendix B Supporting Results for a full list.

For the lung dataset, several of the TFs assigned by the model to the lung dataset are known to play important roles in lung development. These include SOX9 [161, 211], which plays an important role in tracheal and lung epithelium development, GATA6 [66, 233], a regulator for AT1/AT2 cell type, SREBF1 which regulates the biological process of perinatal lung maturation [24], STAT6 which can serve as a therapeutic target for preventing pulmonary hypoplasia [147], YY1 [20], which is required in lung morphogenesis and CEBPB plays pivotal role in determining airway epithelial differentiation [162]. Others include SRF, a critical protein for pulmonary myofibroblast differentiation [169] and BACH2 which is required for the functional maturation of alveolar macrophages and pulmonary homeostasis [138]. Additionally, a number of cell type specific marker genes can be identified based on their expression profiles in paths identified by CSHMM-TF. For example, AQP5 is a known marker for type 1 cells (AT1, path P7) and SFTPC, SFTPA and NKX2-1 are known markers for type 2 cells (AT2, path P5). GATA6 is the regulator for these markers [66], and is assigned to both paths by CSHMM-TF. SOX4 and SOX9 control formation of primary cilia [148] and SOX5 activates the expression of ciliary genes. All 3 TFs are correctly detected for path (ciliated path).

For both the lung and liver datasets, CSHMM-TF has also identified several TFs related to cell proliferation, as expected for developing tissues and organs. Examples are shown in the figures and the Appendix B Supporting results. Similar results for the neuron reprogramming dataset are also available in the Appendix B Supporting results.

3.6.3 Verifying predicted TF activation time

While we observe the expression values for all genes and TFs, when learning the CSHMM-TF model we do not use the expression of the TFs. Instead, following past work [11] we deter-

mine TF activity and timing based on TF targets. This allows us to identify TFs that are post-transcriptionally regulated which are missed when only using expression data to infer activity. However, some TFs are transcriptionally regulated and we can thus use their expression profiles to validate model assignments. Specifically, since TF expression levels and protein-protein interactions are not used to infer their targets, we use them for model validation. Figure 3.5 presents expression profiles smoothed by 4-degree polynomial for top assigned TFs based on p-values from binomial test in the lung, neuron, and liver models. Each figure legend denotes the color and the time assignment for TFs.

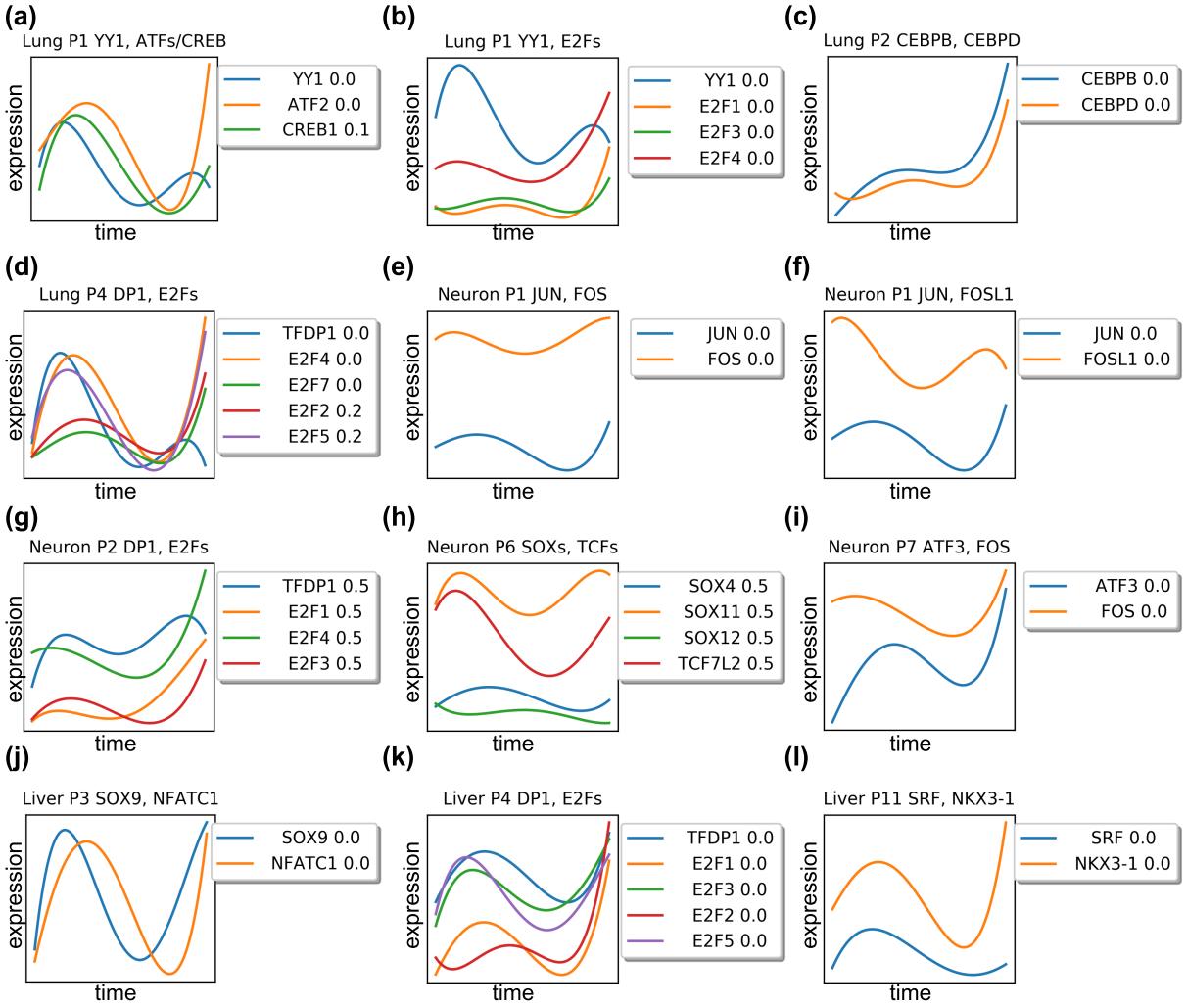

Figure 3.5: Expression profiles for top TFs assigned by the method to the lung, neuron, and liver reconstructed models. Each figure plots the expression TFs predicted to co-regulate a specific path. Each figure legend denotes the color and the *time* assignment for each TF. Profiles for TFs are the MLE estimates for these TFs expression values based on learned model parameters. (a-d) co-regulating TF expressions in lung paths. (e-i) co-regulating TF expressions in neuron paths. (j-l) co-regulating TF expressions in liver paths. See text for details

.

Several of these profiles agree with both their time assignment and their relationship to other TFs assigned to the same paths. For example, the transcriptional repressor protein YY1 is known to directly interact with members of the ATF/CREB family of transcription factors [247]. These TFs are all assigned to path P1 with YY1 being up-regulated earlier than ATF/CREB supporting model assignments (Figure 3.5 (a)). Similarly, interactions between YY1 and E2F genes was previously noted [173, 212] and indeed both are assigned to path P1 (Figure 3.5 (b)). CEBPB/CEBPD, known to form a heterodimers [33] are both correctly assigned to the same time (Figure 3.5 (c)). Similarly, E2Fs which are known to bind DP1 [134] are assigned to the same time and path (Figure 3.5 (d), (g), (k)).

FOS and JUN can form heterodimers [39] and are also assigned the same activation time (Figure 3.5 (e),(f)).

SOX genes are known to modulate beta-catenin/TCF activity [103]. Our model assigning all of them to the same time in path P6 of the neuron data, though expression analysis shows that sox11 is slightly ahead of TCF7 (Figure 3.5 (h)). ATF3 is a known co-factor of c-Fos and both are correctly assigned to the same time (Figure 3.5 (i)). In addition, SOX9 is known to be the downstream target of NFATC1 [36], and CSHMM-TF identified both of them in the same path and assign them at the same time point (Figure 3.5 (j)). Finally, SRF are known to form a physical complex with NKX3-1 [184], and both of them are assigned at the same path with same time (Figure 3.5 (l)). In Appendix B Table 3.8, we present the Spearman correlations for the expression of predicted TF pairs. As can be seen, overall the high correlations support the assignments of CSHMM-TF.

3.6.4 TF interactions further support TF assignment times

In addition to the support provided by the analysis of expression profiles we looked at known interactions between TFs to determine whether TFs assigned by CSHMM-TF to the same path (either at the same or different times) are indeed known to interact. For this, we determined the number of protein-protein interactions (PPI) or regulatory interactions in each paths and compared these to random TF sets of the same size. We have further divided the analysis to determine the significance of interactions within and between a specific time assignment (early-early, late-late, or early-late where early is defined as an assignment to the branching point (0) and late as everything after that).

We searched for interactions for all 5 models in the TcoF-DB database [174], which contains transcription factor interactions for human and mouse. Results are presented in Table 3.2. Each dataset is represented by 3 rows: The first displays the number of interactions in the TcoF-DB in all paths, divided by the number of all combinations in all paths. Take the lung data as an example, there are 257 TFs in the dataset, so there could be 257*256/2 = 32896 possible TF interactions, but only 960 of these interactions are found in the TcoF-DB database. For the #A vs A column, the numerator is the sum of the number of interactions found in TcoF-DB, while the denominator is the sum of all possible interactions in each path (in this dataset we have identified top 10 TFs in each path, so this number becomes 10*9/2 * (7 paths) = 315). For the second row of each dataset, we just calculated the ratio based on the numbers in the first row. For the third row, we calculated the p-value based on hypergeometric test compared to the #total column.

Overall, we see very significant enrichment for interactions between TFs assigned to the same

Table 3.2: Analysis of predicted TF-TF interactions based on the TcoF database. Abbreviations: total: all possible interactions in a dataset, A: all TFs assigned to each path, E: early TFs in each of the paths, L: late TFs. For each dataset we present 3 rows: number of combinations, ratio and p-value.

Dataset	#of TF	#total	#A vs A	#E vs E	#L vs L	#E vs L
Liver #comb	252	1021/31626	20/342	11/166	2/48	7 / 128
Liver ratio		0.032	0.058	0.066	0.042	0.055
Liver-p-value		X	3.99E-03	7.85E-03	2.02E-01	5.60E-02
Lung #comb	257	960/32896	30/315	8/119	5/47	17/149
Lung ratio		0.029	0.095	0.067	0.106	0.114
Lung p-value		X	4.56E-09	8.24E-03	2.35E-03	3.91E-07
Cortical #comb	157	423/12246	19/291	9/144	0/33	10 / 114
Cortical ratio		0.035	0.065	0.063	0.000	0.088
Cortical-p-value		X	2.72E-03	2.76E-02	X	1.93E-03
Neuron #comb	208	873/21528	30/351	16/90	8/85	6/176
Neuron ratio		0.040	0.085	0.17	0.094	0.034
Neuron p-value		X	4.47E-05	1.07E-07	7.47E-03	X
Myoblast #comb	230	875/26335	49/447	45/408	0/3	4/36
Myoblast ratio		0.033	0.109	0.111	0.000	0.111
Myoblast-p-value		X	7.18E-14	5.50E-13	X	6.42E-03

path. For most datasets we also see significant enrichment for interaction for 'early TFs'. These are TFs that are assigned to the initial part of the path (usually those that regulate a large number of genes in the path) and as shown above in many cases represent proteins that are involved in complexes that jointly regulate a large number of genes. However, interestingly we also find for some of the datasets (most notably the mouse lung data) a strong enrichment for early-late interactions. These interactions likely represent a late TF activation or recruitment by an earlier TF. The fact that many of them are known interactions indicate that our model, using scRNA-Seq data, is indeed able to identify the specific timing of the regulation of the different TFs which are usually all assigned to the same time.

3.6.5 Comparison to other methods

We compared CSHMM-TF with several prior methods for trajectory inference that do not utilize TF-gene interaction data. For this we looked at the accuracy of the reconstructed trajectories and cell assignments as well as on the inference of TFs and their order. Appendix B Figure 3.7 presents a comparisons for the lung and neuron datasets between CSHMM-TF and several prior methods for pseudo-time inference including PCA [208], TSNE, GPLVM following PCA [32], Monocle 2 [151, 206], Slingshot [189], and PAGA [222]. Note that, although PCA and TSNE are not cell trajectory reconstruction methods, a number of previous time series scRNA-Seq analysis papers have used these methods to discuss trajectories [42, 208]. In addition, several of the trajectory assignment methods only work on the reduced dimension representation (including GPLVM and slingshot) and so we plot the results for these methods as well.As the figure shows, for a number of cell types these methods were unable to fully reconstruct known developmental trajectories. For example, while PCA and TSNE, were able to identify clusters for some cell types in both the lung and neuron data, they were unable to reconstruct the correct trajectories and also mix a number of different cell types correctly assigned by CSHMM-TF. GPLVM correctly orders cells along a pseudotime, however, it is unable to determine branching models.

Monocle 2 is able to reconstruct cell trajectories, however it only found a single split for these datasets and also mixed cell types that CSHMM-TF correctly separated into unique branches. Slingshot is able to order cells along a pseudotime but it did not identify any branch point for the lung data. For the neural data it correctly separates the MEF and neuron cells, but is unable to infer a correct trajectory along the different cell types (in fact, one of its trajectories ends with d2_induced which is an intermediate cell type). As for PAGA, while it correctly clusters cell types, it does not seem to provide any clear trajectory for the cells or clusters. For both datasets PAGA produces a set of weakly connected cliques making it hard to infer the branching.

To compare the results of CSHMM-TF with CSHMM that does not utilize TF-gene interactions, we developed a quantitative measure which calculates the accuracy of the ordering inferred by the two methods (Appendix B Supporting Methods). We used this to compare the two methods on three of the datasets analyzed in this paper: lung, neuron and liver. Results are shown in Appendix B Table 3.10. As can be seen, CSHMM-TF assignments are in better agreement with known cell differentiation stages when compared to CSHMM for all three datasets. In some of them the improvement is small (1-2%) while for the lung dataset, the improvement is about 9%. To further study the usefulness of the TF-gene interaction information we have also compared CSHMM-TF to a version that uses random TF-gene assignments. Again, we see a decrease in performance when not using the correct TF-gene interactions (Appendix B Table 3.10). For the random assignments we also determined the number of significant TFs identified by CSHMM-TF. As can be seen in Appendix B Table 3.11, random TF-gene interactions lead to much fewer significant TFs indicating that, as we assumed in the model, several co-regulated genes are assigned to the same paths by CSHMM-TF.

As mentioned above, most prior methods do not attempt to model regulation by TFs. However, a few do, and so we next compared CSHMM-TF to two prior methods for TF assignments using the liver dataset. The first is SCDIFF[51], which, unlike our method does not provide continuous assignment for cells. The second is based on post-processing assignment of TFs following model reconstruction [42, 81, 207] . These methods perform t-test for the expressions of TFs between each path and its parent path and use a p-value cutoff to select differentially expressed (DE) TFs. Here, we use the DE method as a post processing step following CSHMM analysis for comparison. Appendix B Table 3.3 and Table 3.4 present the resulting TFs selected by SCDIFF and the DE method. For both methods we select the top 10 TFs for each path and compare these to the top 10 CSHMM-TF predictions. While we see some overlap (HMGA1, HMGA2 and PITX2) between TFs identified by the DE method, and those identified by CSHMM-TF, all other liver TFs identified by CSHMM-TF which were discussed are missed by the DE method. Similarly, we see a number of known liver development TFs that were identified by CSHMM-TF but missed by SCDIFF including ONECUT2 at P5 [125], APC [28] at P4, and SOX9 [143] at P3 and P5.

3.6.6 Scalability and robustness of CSHMM-TF

While some recent scRNA-Seq studies profile thousands of cells, very few large time series scRNA-Seq datasets are currently available. One of the datasets we analyzed, which studied mouse cortical development is quite large ($\sim$21K cells, $\sim$10K genes) [128]. As we have shown in Appendix B Figure 3.10 and Table 3.7, CSHMM-TF can be successfully applied to such

data. Total runtime for this dataset on a desktop with 4 cores was less than 3 days and since assignments of cells to paths are easy to parallelize, run time can be significantly reduced on a larger cluster. To test performance on slightly smaller, though better annotated, dataset we performed simulation analysis based on the liver scRNA-Seq data [31] using $\sim$10K cells. For this, we generated a new dataset with $\sim$10K cells based on the human liver data. We created 13 random cells from each original cell by randomly adding 20% dropouts (setting the expression of 20% random genes in each cell to zero). Results are presented in Appendix B Figure 3.9 and Table 3.9. Run time on a desktop is about 9 hours for one EM iteration with the total run time of less than 2 days. We have also compared the accuracy of the resulting model to the original model (based on a smaller data size) and found them to be comparable. See Appendix B Supporting Results for complete details.

3.7 Discussion

While several methods have been developed to reconstruct developmental models based on time series scRNA-Seq data, very few of these utilize information about TF-gene interactions to further improve the models. Such complementary information can aid in correctly reconstructing models for development and differentiation and can help explain the regulation of the process being studied.

Here we presented CSHMM-TF a continuous-state HMM model which combines cell assignments to a developmental model with TF assignments as regulators of the process. To learn the model the method iterates between cell assignments to branches and TF assignments to specific time points. Cells assigned to paths to which TFs are assigned are assumed to have that TF active. Based on the analysis of the targets of these cells we can both, identify the regulators and improve the assignments of cells to paths.

We applied the method to several scRNA-Seq datasets from both human and mouse. As we show, the method was able to reconstruct biologically sound models for all datasets, in most cases correctly grouping cells based on known types. In contrast, several other pseudo-time scRNA-Seq analysis methods were unable to correctly reconstruct models for at least some of these studies highlighting the advantage of integrating expression and regulation data.

Beyond the construction of the models and cell assignments to specific positions, CSHMM-TF identifies several TFs as regulating key aspects of the processes. Analysis of the TFs identified for the different biological systems studied supports these assignments since many of them are known to play important roles in those process while others represent novel predictions about the regulation of specific branching events. In addition to the list of TFs, CSHMM-TF provides information about potential combinatorial and causal relationships between TFs assigned to the same path. As we showed, TFs assigned to the beginning of paths are often interacting and in some cases early and late TFs are interacting as well. In these cases CSHMM-TF provides information on the dynamics of the assembly process of TF complexes which, without the detailed trajectories provided by scRNA-Seq would have been hard to do.

CSHMM-TF can also be complimentary to current analysis methods that are based on identifying DE TFs. For the liver data, we found that PITX2, a known liver development TF [35], appears in paths P6 for the DE while it appears as regulating a later path, P12, for CSHMM-TF.

This likely means that while PITX2 is first DE early, its impact and regulatory role are only observed later in the developmental process. Such joint analysis can further improve the confidence in the identified TFs.

While CSHMM-TF was successful in analyzing several biological systems, there are certainly many places where it can be improved. First, CSHMM-TF relies on a predefined list of TF-gene interactions, and this is likely incomplete preventing the method from identifying additional key TFs. In addition, while the method is able to identify interacting TFs, the model for their impact is additive and so it would be hard for this method to identify more complex relationships (for example, AND and OR types).

Besides TF information, there are still other types of data that are used for single-cell lineage tracing. As we have mentioned in Chapter 1, there are new studies that introduce CRISPR-Cas9 technologies for lineage reconstruction. These studies can insert artificial markers (mutations) to single-cells and at the same time profiles the expression of cells. This makes it possible to integrate both mutations and expression to improve single-cell lineage tracing. In the next chapter, we will present a new framework that combines both types of data.

3.8 Appendix B: Supplement to CSHMM-TF: Inferring TF activation order in time series scRNA-Seq studies

3.8.1 Supplementary Methods for CSHMM-TF

Data collection and processing We tested our method on five publicly available time-series scRNA-Seq datasets in human and mouse. These include human liver development study in which cells were followed from pluripotency in 2D culture and 3D liver buds [31], mouse lung development data [208] which profiles lung epithelial cell differentiation, mouse cortical development data [128], human skeletal muscle myoblast development data [206], and mouse neuron reprogramming data that studies the cell reprogramming trajectories from embryonic fibroblasts (MEFs) to neuron cells [209]. For the human data, we used the same processing method as TASIC [155], which keeps only the genes expressed in more than 25% of cells for further analysis. Mouse neuron and mouse lung datasets were processed as suggested by the original studies: We filtered genes with either FPKM < 1 in all cells or zero variance. Next, expression values were transformed to log space. For mouse cortical data we first removed genes expressed in less than 5% of cells. Then, for each cell we normalized its expression and log2-transformed all values. After these initial pre-processing, the human liver data contains 765 cells with 19K genes in 4 developmental stages (iPSC (induced pluripotent stem cells) $\rightarrow$ DE (definitive endoderm) $\rightarrow$ HE (hepatic endoderm), IH (immature hepatoblast-like) $\rightarrow$ MH (mature hepatocyte-like), LB (liver buds) and mesenchymal stem cell (MSC), Human umbilical vein endothelial cells (HUVEC)). The mouse lung dataset contains 152 cells with 15K genes measured at three time points E (Embryonic)14.5, E16.5, and E18.5. At timepoint E18.5 (though not at E14.5 and E16.5), cells were labeled with one of the following cell types: alveolar type 1 (AT1), alveolar type 2 (AT2), bipotential progenitor (BP), Clara, and Ciliated. We used the profiled time point to label cells in E14.5 and E16.5 as NA_14 and NA_16. For the mouse cortical data, we select the medial ganglionic eminences (MGE) cells and cortex cells, and this result in $\sim$21K cells with $\sim$10K genes and 3 time points (E (embryonic) 13.5, E18.5 and P (postnatal) 10). At E13.5 cells are labeled as MGE, which is the progenitor of cortex cells. At E18.5 and P10, all cells are labeled as cortex cells. The mouse MEF dataset is composed of 252 cells with 12K genes measured at 4 time points (0, 2, 5, 22 days). Cells were labeled with one of the following cell types: Neuron, Myocyte, and Fibroblast, MEF (mouse embryonic fibroblasts), and other progenitor-like cell types. The human skeletal muscle myoblast dataset has 271 cells with 13K genes and 4 measured time points (0, 24, 48, 72 hours). Please note that, although some of the datasets we used are well-labeled based on known markers, CSHMM-TF is purely unsupervised. Cell labels are only used for evaluation in the result section.

Details about the how the TF-target information data is obtained Transcription factors (TF) are proteins that bind to specific DNA sequences and regulate transcription processes. Each TF activates or represses the transcription of a specific set of genes if the TF binds to the DNA location related to the genes. It has been a challenging task to identify the protein-DNA relationships for an organism. In this paper we use the information of potential targets of a set of transcription factors for human and mouse [59, 175]. This information is used to identify potential key regu-

lators for each developmental processes. The details of how this data is obtained is described in [175]. Briefly, this data is constructed from 3 parts. In the first part, the human ChIP-Seqencing data is downloaded from ENCODE [40]. This data contain aggregated binding peaks for 148 human TFs across diverse cell lines. For each human gene, all the TFs that have transcription start sites near the gene were considered to regulate the gene. For the second part, ranked human PWM-gene predictions were obtained from [60] and each PWM was mapped to correspond TFs by using TRANSFAC [127] and JASPAR [215]. For a gene, a protein-DNA interaction was identified if the gene is in the top 100 predictions in any of the PWM for TFs. The last part is for mouse TFs, the proten-DNA interaction is derived from the second part except that a top 1000 threshold is used instead of top 100. Human gene ids were translated to mouse gene ids based on Mouse Genome Database (MGD) [18] and HUGO Gene Nomenclature Committee (HGNC) database [178].

CSHMM-TF is a valid continuous state HMM To show that the model defined above is indeed a Continuous-State Hidden Markov Model (CSHMM) we extend the argument used by [115]. There it is shown that without including the TF information, $\theta = (V, \pi, S, A, E)$ is indeed a Continuous-State Hidden Markov Model model (CSHMM) with a properly defined initial probability (π), a transition probability function (A) that covers all states (S) and an emission probability model for each state. The above are sufficient conditions to fully define a continuous-state HMM [1]. In the new model, TF information only effects emission probability (E') and so all other correctness claims for emission and initial probability stand. As for the emission, to show that the model indeed defines a unique emission probability for each state note that the TF model introduces two additional parameters: (Ω and Φ). Ω is the matrix encoding TF-target information with $\Omega_{i,j}$ denoting if gene i is regulated by TF j. This is part of the input and so defined for all states. Φ stores the information of TF activation time with $\Phi_{i,j}$ denoting if TF j is regulating path i and the value represents the activation time. Default values for all i, j entries in this matrix are null (no impact) which means that the TF-gene info for this TF is not used by the model for this path. Only TFs for which the timing is defined are used by the model. Since these are specifically assigned in the Expectation-Maximization (EM) steps (see below) we are guaranteed that they would satisfy the requirement for a valid emission probability as required by a HMM. We thus conclude that the new model is also a valid continuous-state HMM (CSHMM).

Definition of the transition probability (A) of CSHMM-TF CSHMM-TF adopts the same transition probability definition as CSHMM.
The transition probability $A(s_{p_1,t_1}, s_{p_2,t_2})$ for each pair of states $s_{p_1,t_1}, s_{p_2,t_2} \in S$ is defined as follows:

$$A(s_{p_1,t_1}, s_{p_2,t_2}) = 0, \text{ if } s_{p_2,t_2} \text{ is not reachable from } s_{p_1,t_1} \tag{3.5}$$

$$A(s_{p_1,t_1}, s_{p_2,t_2}) = 1/Z_{p_1,t_1}, \text{ if } p_2 = p_1 \text{ and } t_2 > t_1 \tag{3.6}$$

$$A(s_{p_1,t_1}, s_{p_2,t_2}) = \prod_{\substack{q \in \text{branch probability} \\ \text{from } p_1 \text{ to } p_2}} \frac{q}{Z_{p_1,t_1}}, \text{ if } p_2 \neq p_1, p_2 \text{ reachable from } p_1 \tag{3.7}$$

Where $s_{p,t}$ is the hidden state of cells assigned at path p with pseudo time t, Z_{p_1,t_1} is a normalizing factor for the transition probability going out of state s_{p_1,t_1} i.e..

$$Z_{p_1,t_1} = 1 - t_1 + \sum_{\substack{\text{path } p \\ \text{reachable from } p_1}} \prod_{\substack{q \in \text{branch probability} \\ \text{from } p_1 \text{ to } p}} q. \tag{3.8}$$

The branch probability (B) is defined on split nodes (D). The second term in equation 3.8 is the product of all branch probabilities of the paths from p_1 to p. For example, assume that there are two paths in between states p_1 and p: p_a and p_b. Then the second term will be $B_{p_1,p_a} * B_{p_a,p_b} * B_{p_b,p}$, where B_{p_a,p_b} refers to the branch probability for cells to transition from p_a to p_b. Note that transition probabilities integrate to 1 for each state. Also transitions and emissions only depend on the current state.

Supporting details on finding DE genes We first using t-test between each path and its parent path to find the genes that has p-value smaller than 0.05. After that, we uses a set of log2 fold change values (0.6, 1.0, 1.5) to get three DE genes list. The reason we use a set of fold change values is because that datasets usually have different expression changes between paths therefore using a set of fold change values we will be less likely to lose the DE genes information.

Assigning pseudo time to TF regulating a path

In addition to the assignment of TFs to paths, we would also like to use the scRNA-Seq data to fine tune the specific time at which the TF exerts their influence on genes in the path. This is a major advantage of the continuous scRNA-Seq data that cannot be obtained with time series bulk data given its discrete sampling nature. To determine t_{start} for each TF / path we use a modified pseudotime $t' = t - t_{start}$ to calculate the probability of the target genes being regulated by TF, which will thus make TF have an effect on the loss function (log-likelihood) $t_{j,start}$ is defined as the smallest activation time for the target gene j if it is regulated by multiple TFs. If no regulating TFs are detected for a gene, the $t_{j,start}$ will be defined as 0 which will have the same effect as CSHMM. The t_{start} of the TF is then set to the best value from 0 to 0.5 with the highest probability to its target genes by sampling 5 points uniformly. CSHMM-TF only allows the target gene expression starting to change after t_{start}, so setting t_{start} close to 1 will make the target gene expression not changing in the path. Therefore, we restrict $t_{start} \leq 0.5$ to make sure that the target gene expression have enough time to change. This information is then stored in parameter Φ. Note that, We have included the TF activation time t_{start} in the likelihood function so we only need to find the best value for t_{start} that makes the probability of target genes highest. The shape of the expression profile of gene j is now describe by parameter: $K_{p,j}$ (speed of changing of gene j on path p) and $t_{j,start}$ (starting time for gene j).

Model initialization

For model initialization we apply the same strategy used in SCDIFF tool [51], which construct an initial cell differentiation tree by clustering the cells at each time point, and then compute the distance of each of the clusters to the root of the tree (cells in first time point). Using

this distance function clusters are assigned to different levels in the tree (where clusters in each level are significantly more distant from the root than the preceding level). Finally, each cluster (except the root cluster) at level i is connected to a parent cluster in level $i-1$ by selecting the closest cluster, in expression space, in level $i-1$. Following the initialization step of SCDIFF, we associate each cluster associated with a path (the edge connecting it to its parent). Finally, cells in each cluster are randomly assigned along the path for that cluster. Split nodes are defined for cases where two or more clusters at a specific level connect to the same cluster at the level above them. The TFs are not assigned in the model initialization step. The effect of different model initializations and how they affect the final result has been tested in [115].

Model learning and inference

We use an Expectation-Maximization (EM) algorithm to learn the parameters of the model and to infer new cell assignment. Given initial cell assignments, the branching probabilities can be easily inferred using standard Maximum Likelihood Estimation. In the following sections we discuss how to learn the emission probability parameters which, due to the K parameter requires an optimization of a non convex target function. As for cell assignment, given model parameters we assign each cell to a state $s_{p,t}$ which maximizes the log-likelihood of the resulting model. Again, since the likelihood function is not concave, determining a optimal value t for a cell assigned to path p is challenging. We will discuss a sampling strategy for solving this problem which we use to assign cells in the following sections.

Inferring cell assignments (E-step)

Given model parameters θ, we would like to assign each of the cells in our input dataset expression matrix X to a state $s_{p,t}$ which maximizes the log-likelihood. Determining a optimal value t for a cell assigned to path p is hard to be performed in closed-form because the likelihood function to t is not concave. Instead, similar to the optimization of $K_{p,j}$ parameter, we use a sampling strategy to find the best time along a path for each cell. Specifically, for each path we sample 100 points uniformly and compute the likelihood of assigning the cell to each of these points. Since the likelihood function (when model parameters are known) decomposes based on cells, this process is efficient.

Model learning (M step)

Given initial cell assignments, the branching probabilities can be easily inferred using standard Maximum Likelihood Estimation (see below).

Next, we discuss learning the emission probability parameters. For genes that change along a path, we need to learn a mean value g for split nodes and the $K_{p,j}$ parameter which encodes for each path and each gene the rate of change between the start and end expression values for that gene on that path. For K, even with a fixed mean value g for each split node, it is difficult to compute it in close form because of non-convexity. We thus use a line search strategy to determine $K_{p,j}$. For this we compute the likelihood for 100 possible values between 0 to 10 (since $e^{-10} \approx 0$), and choose the value that achieves the maximum probability for $K_{p,j}$ (note of

course that since this is a gene and path specific parameter it can be done independently for each gene / path).

As for g, let $w_j^i = \exp(-K_{p,j}t'^i)$, λ_g be the L1 sparse parameter, and Δg_p is the difference vector for the expression values at the endpoints of path p. Using notations defined above, the negative log likelihood terms that depend on g are:

$$\begin{aligned} NLL &= \sum_i^N \sum_j^G \frac{1}{2\sigma_j^2}(x_j^i - \mu_{j,s_{p,t}}^i)^2 + \quad \sum_{p\in P}\sum_{j=1}^G -\frac{\lambda_g}{1+\alpha_{p,j}}|(\Delta g_p)_j| \\ &= \sum_i^N \sum_j^G \frac{1}{2\sigma_j^2}(g_{pa,j}w_j^i + g_{pb,j}(1-w_j^i) - x_j^i)^2 \\ &\quad + \sum_{p\in P}\sum_{j=1}^G \frac{\lambda_g}{1+\alpha_{p,j}}|(\Delta g_p)_j| \end{aligned} \tag{3.9}$$

where (g_{pa}, g_{pb}) refers to the mean gene expression of the split point at both ends of a path.

Since the function is convex, in CSHMM we let $\lambda_g = 1$ and use CVXPY [47, 75, 76], a disciplined convex optimization toolkit utilizing cone-spliting interior point method, to solve the linear system. Now for CSHMM-TF, we also provide another option that is glasso r package [68] to solve the L1 lasso problem because it is usually faster than CVXPY. As for the variance, since we assume that the variance σ_j of each gene j is the same across all the paths, once we have the g values we can use a standard MLE method to derive the closed-form solution for its estimation (see following supplementary section).

Details for MLE

Branch probability First, we have the constraint that $\sum_{p_2} B_{p_1,p_2} = 1 \quad \forall p_1, p_2 \in P$. Using Lagrange multipliers we can write:

$$L(X, Y, \alpha, \theta) = \left(\sum_{i=1}^N \sum_{\substack{q\in\text{branch probability} \\ \text{from } p_1 \text{ to } p_2}} \log(q) \right) + \alpha^T(B1 - 1) \tag{3.10}$$

We obtain the update for B_{p_1,p_2} by setting gradient to 0

$$\frac{\partial L(X, Y, \alpha, \theta)}{\partial B_{p_1,p_2}} = 0 \Rightarrow \frac{N_{p_1,p_2}}{B_{p_1,p_2}} + \alpha_{p_1} = 0 \tag{3.11}$$

$$\sum_{p_2} B_{p_1,p_2} = 1 \Rightarrow \sum_{p_2} \frac{-N_{p_1,p_2}}{\alpha_{p_1}} = 1 \Rightarrow \alpha_{p_1} = \sum_{p_2} -N_{p_1,p_2} \tag{3.12}$$

$$\Rightarrow B_{p_1,p_2} = \frac{N_{p_1,p_2}}{\sum_{p_2} N_{p_1,p_2}} \tag{3.13}$$

Where N_{p_1,p_2} is the number of cells assigned to path p_2 that comes from p_1

Learning σ_j We compute the gradient of σ_j, the variance parameter for each gene:

$$\frac{\partial}{\partial \sigma_j} \log P(X, Y|\theta) = \frac{\partial}{\partial \sigma_j} (\sum_{i=1}^{N} \sum_{j=1}^{G} \log P(x_j^i | s_{p,t}^i, \theta)) \tag{3.14}$$

$$= \frac{\partial}{\partial \sigma_j} (\sum_{i=1}^{N} \log N(\mu_{j,s_{p,t}^i}, \sigma_j^2)) \tag{3.15}$$

$$= \frac{\partial}{\partial \sigma_j} (\sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp(-\frac{(x_j^i - \mu_{j,s_{p,t}^i})^2}{2\sigma_j^2})) \tag{3.16}$$

$$= \frac{\partial}{\partial \sigma_j} (\sum_{i=1}^{N} -\log(\sigma_j) - \log(\sqrt{2\pi}) - \frac{(x_j^i - \mu_{j,s_{p,t}^i})^2}{2\sigma_j^2}) \tag{3.17}$$

$$= \sum_{i=1}^{N} (-\frac{1}{\sigma_j} + \frac{(x_j^i - \mu_{j,s_{p,t}^i})^2}{\sigma_j^3}) \tag{3.18}$$

Setting gradient to 0 we have:

$$0 = \sum_{i=1}^{N} (-\frac{1}{\sigma_j} + \frac{(x_j^i - \mu_{j,s_{p,t}^i})^2}{\sigma_j^3}) \tag{3.19}$$

$$\Rightarrow \sigma_j^2 = \frac{\sum_{i=1}^{N} (x_j^i - \mu_{j,s_{p,t}^i})^2}{N} \tag{3.20}$$

Quantitative measure for comparing CSHMM and CSHMM-TF models

We developed a distance function on the cell assignments of CSHMM and CSHMM-TF models based on partial orderings ($\mathbb{P}$) defined from literature. Specifically, assumes that from literature, we know that cell type A is the parent cell type of cell type B, we denote this relationship as A→B (partial ordering). For every pair of cells (c_i, c_j) that has A→B relationship, we calculate the number of cells between c_i and c_j that are neither type A nor type B. Therefore, we assume that there are no other cell types between cell type A and B. For A→B→C relationships, we instead calculate A→B and B→C and sum them together. The total distance is the summation of each pair of cells that belongs to each pair of partial orderings. That is:

$$\text{Distance} = \sum_{(A,B)\in\mathbb{P}} \sum_{c_i\in A, c_j\in B} \sum_{c_k\notin A\cup B} \mathbb{1}_{(c_k \text{ lies between } c_i \text{ and } c_j)} \tag{3.21}$$

3.8.2 Supplementary Results for CSHMM-TF

TFs for cell proliferation In Result, we mentioned that most of the TFs in the lung and neuron datasets are related to cell proliferation. Examples are as follows: E2Fs [92], YY1[227], ATF2/ATF7[104], XPB1 [89], CREB/CREM [46], DSP [216], TBP/ELK1 [245], TBPL1 [230], CEBPs [98], SOX9 [218], KLFs[15], SOX5[19], SOX4 [34], NRF1[97], TCF7L2 [183], BACH2 [130], SRF[80], APC [144], RB1 [96], TEADs [118], FOXOs [17], BPTF [224], GATA6 [14], HSF2 [191], RXRs [188], ESRRA [204], EGR1/EGR2 [58], STAT6 [25], CDC5L [219], TCF3 [145], SREBFs [220], FOS/FOSB/FOSL1/JUN/JUNB/JUND [67], GAPBs [234], EP300 [70], HSF1 [248], FOXJ2 [181], REST[239], NFIL3 [232], FLI1 [22], ETS1[166], SOX11/SOX12[112], SOX8 [231], HMGA2 [199], MAX [111], TFAP4 [44], NF1[37], ATF5 [126], ATF1 [88], NKX3-1[105], SRY[21], FOXO3[150], POU2F1[246], ONECUT2[122], OTX2[171], FOXA3[197] ATF6[87], FOXJ2[243], GATA2[210], FOXO1[217], GATA5[229], E4F1 [43], PITX2[110], BACH1[241].

TF assigned correctly for liver development dataset Besides the TFs mentioned in Result , CSHMM-TF also identified other TFs that is related to liver development. For example, APC is related to the WNT signaling pathway in liver development [28]. XBP-1 is a transcription factor essential for hepatocyte growth [158]. GATA5 is reported to be essential in liver development in other organisms [74, 90].

TF assigned correctly for neuron reprogramming dataset For the neuron reprogramming dataset, CSHMM-TF also identifies known key regulators for some of the cell types (Figure 3.6 (b)). For example, REST is identified for path 8, which is the neuron path, and REST is known to be required to repress neuronal gene expression in vivo [38]. ATF5/ATF7 are key regulators of nervous system development [77]. SRF, also identified by CSHMM-TF, has also been implicated in neuronal development [121]. TCF3 is a known repressor of Wnt–β-Catenin signaling and maintains neural stem cell population during neocortical development [106]. CREM is identified in path 8. Studies indicates that the lack of CREB/CREM genes leads to migration

abnormalities during brain development [48]. NF1 controls neural stem cell (NSC) proliferation [37], SOX4/SOX11/SOX12 have been reported to be essential for NPC proliferation and differentiation [112].

Supporting results of large simulated liver data See Figure 3.9 and Table 3.6 for the result cell trajectories and TF assignment. As for the results, comparing the structure for the real and simulated liver dataset, we observe that the structure and the temporal cell type assignments are overall similar, however, the larger and noisier simulated dataset does not contain some of the more detailed branching observed in the original model. This is likely the result of the increased dropout which makes it harder for the method to distinguish between similar cell types leading to them being merged in a single path. TF assignments are also pretty well conserved between the two models.

3.8.3 Supplementary Tables and Figures for CSHMM-TF

Note that, in all the following CSHMM structure and cell assignments Figures, the D nodes are split nodes and P edges are paths as shown in Figure 3.1 in Chapter 3. Each small circle is cells assigned to a state on the tree structure. The bigger the circle the more cells are assigned to the position. The color of the circles represent different cell types/ assigned time points based on original papers.

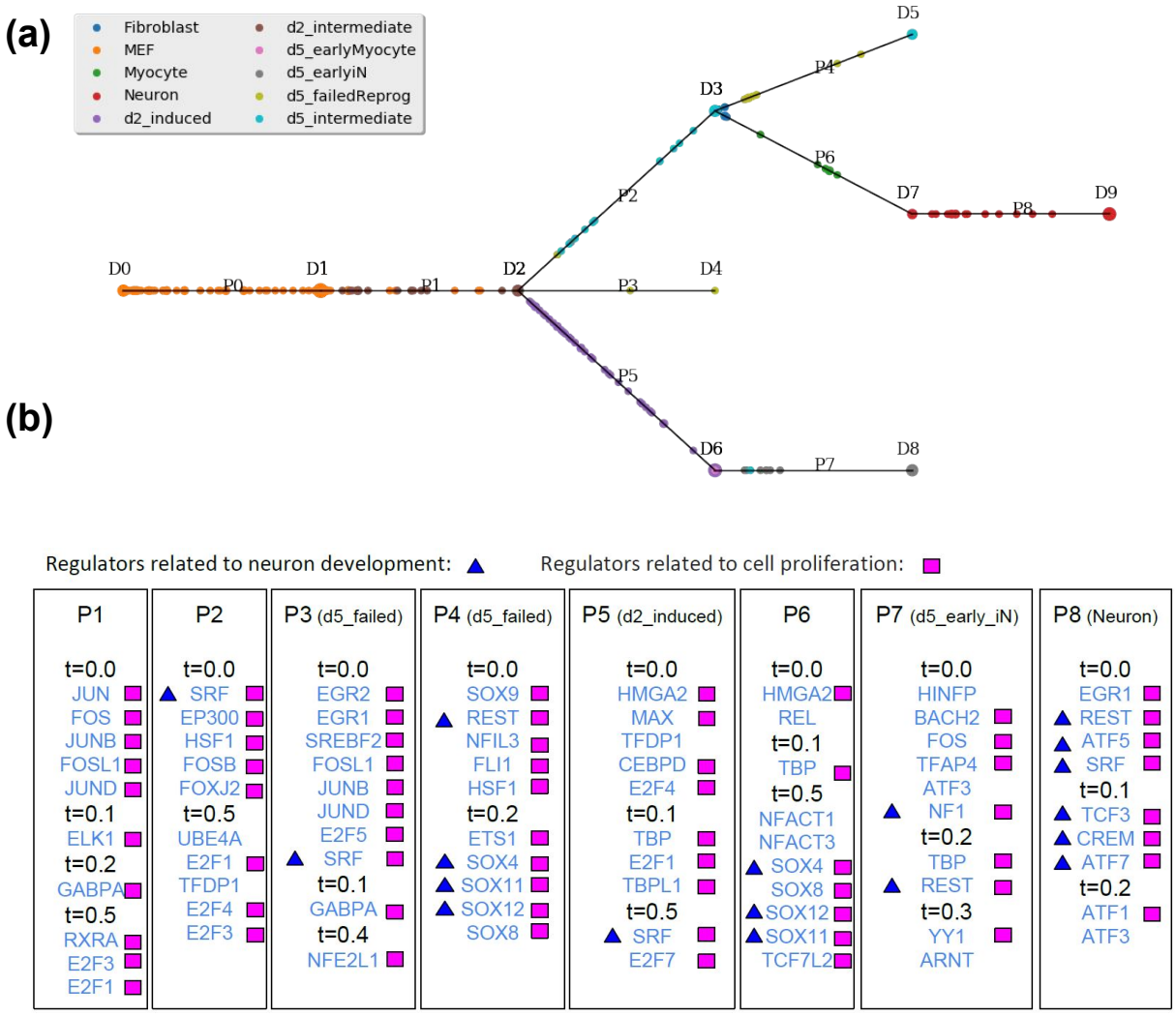

Figure 3.6: (a) CSHMM-TF structure and continuous cell assignment for the neuron reprogramming dataset. (b) TF assignments by CSHMM-TF for the neuron reprogramming dataset.

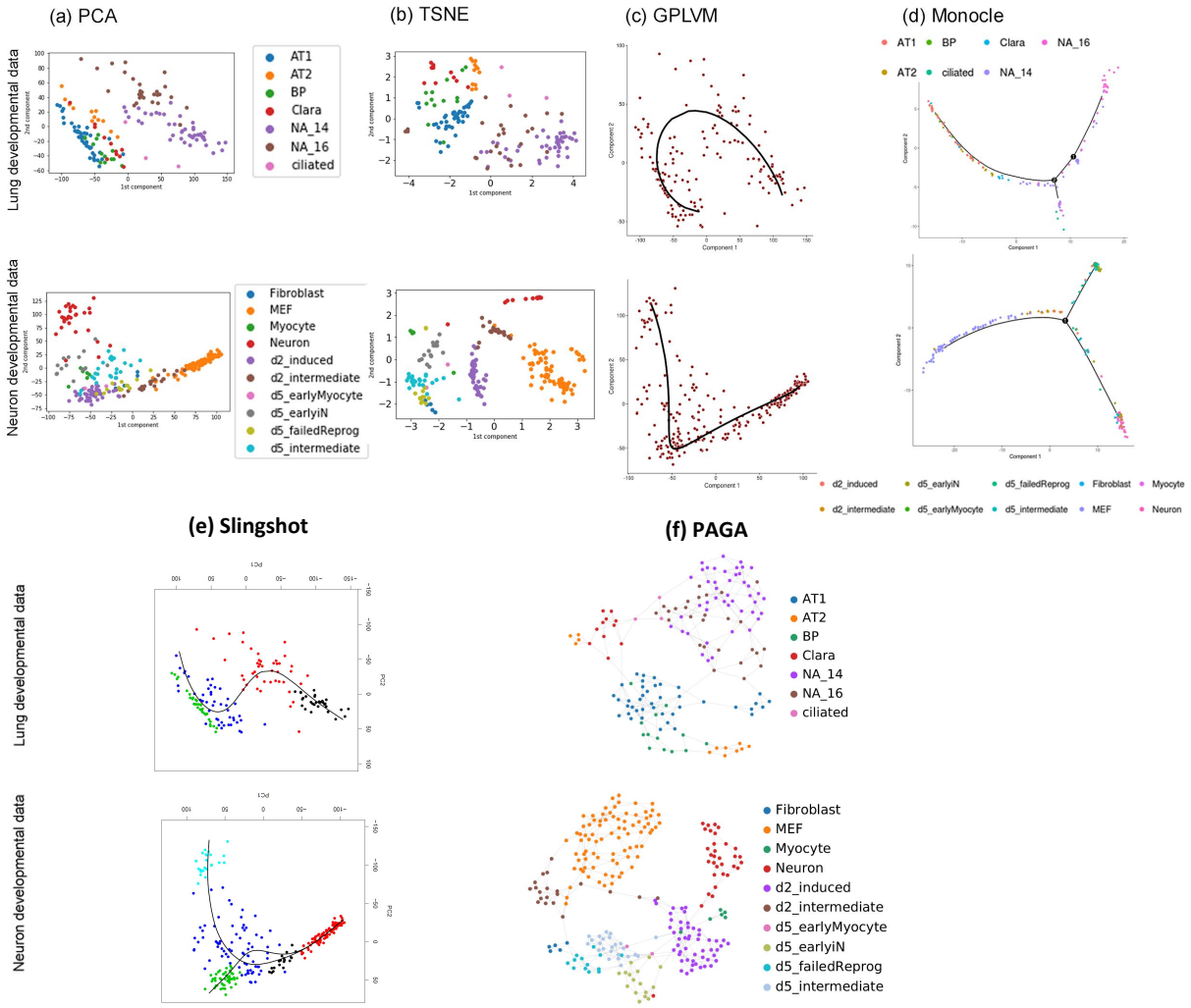

Figure 3.7: Analysis of lung development and MEF reprogramming data by prior methods. (a) PCA (b) TSNE (c) GPLVM (d) Monocle 2 (e) Slingshot (f) PAGA. The first and the third row presents results for the lung dataset and the second and the fourth rows are for the neural developmental dataset. Colors correspond to cell fate assignments in the original papers. We run GPLVM/Slingshot/PAGA on reduced dimension by PCA. The output of GPLVM/Slingshot does not have coloring for cell types but we can see part (a) for the cell types coloring. Note: The PCA plot of Slingshot is flipped both horizontally and vertically so we also flipped it here.

Table 3.3: The TF assignment of SCDIFF for liver dataset. Each column shows the top 10 TFs assigned to the path based on p-values.

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
HMGA1	FOXO3	HMGA1	TBPL1	TBP	HMGA1	HMGA1	NKX6-2	HMGA1	TBPL1	HMGA1	HMGA1
TBPL1	NFATC3	TCF7L2	HMGA1	E2F2	E2F7	TBPL1	HMGA1	TBPL1	HMGA1	TBP	RORA
CDC5L	NFATC4	TBP	RB1	E2F5	TOPORS	TFDP1	CEBPD	PITX2	YY1	CEBPB	TBPL1
RB1	HMGA1	POU2F1	E2F1	HMGA1	E2F4	E2F4	HMGA2	ZNF350	GABPA	CEBPG	TBP
E2F7	SOX9	MTF1	NFYA	PITX2	TFDP1	NKX3-1	CEBPB	FOXO3	FOSB	MEF2A	BACH1
FOXA3	E2F7	OTX2	NFYB	SOX5	UBE4A	E2F7	FOXJ2		STAT3	HMGA2	
FOXM1	HMGA2	NKX6-2		SRY	FOXO1	E2F2	FOXD1		ATF6	NKX6-2	
SOX9		EGR1		FOXO3	E2F3	APC	CEBPG		ZNF350	SRF	
SOX11		E2F4		RB1	ATF6	RB1	SRF		STAT1	CEBPD	
SOX12		UBE4A		CD40	DSP	HMGA2	ETS1		BACH1	CDC5L	

Table 3.4: The TF assignment for liver dataset based on the post-processing step of finding differently expressed TFs on CSHMM. Each column shows the top 10 TFs assigned to the path based on p-values.

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
POU5F1	POU5F1	PAX6		FOS	ZIC2	ERG	DSP	PBX1	TP53	RORA	GATA6
HNF4A	SOX11	ZIC2		HMGA2	HMGA2	DSP	ERG	NR6A1	NR6A1	MXI1	STAT3
GATA4	NR2F2	HMGA2		ZIC2	PAX6	ARID5B	FOXO1	JUND	HNF4A	RELA	EGR1
FOXA2	ZEB1	HNF1B		EGR1	OTX2	EGR1	ARID5B	HERPUD1	TCF12	CEBPD	TBPL1
PRDM1	ERG	ETV4		PAX6	HMGA1	NR2F2	STAT1	MAFF	NR1H3	CEBPB	NFE2L3
PAX6	HOXB7	OTX2		HES1	HERPUD1	FOXO1	HIF1A	CEBPD	SMAD3	TP53	
GATA6	NR6A1	HMGA1		HMGA1	MAF	TAL1	EGR1	SOX11	CREB1	STAT1	
GATA3	ELK3	PRDM1		ATF3	PITX2	STAT1	ETS1	STAT3	ETS2	ATF4	
STAT3	PBX1	GATA5		JUND	FOXM1	HMGA1	NR2F2	STAT4	IRF9	FOSB	
TP53	FLI1	GTF2I		SMAD3	NR5A2	NFE2L3	ETS2	JUN	DSP	GTF2A2	

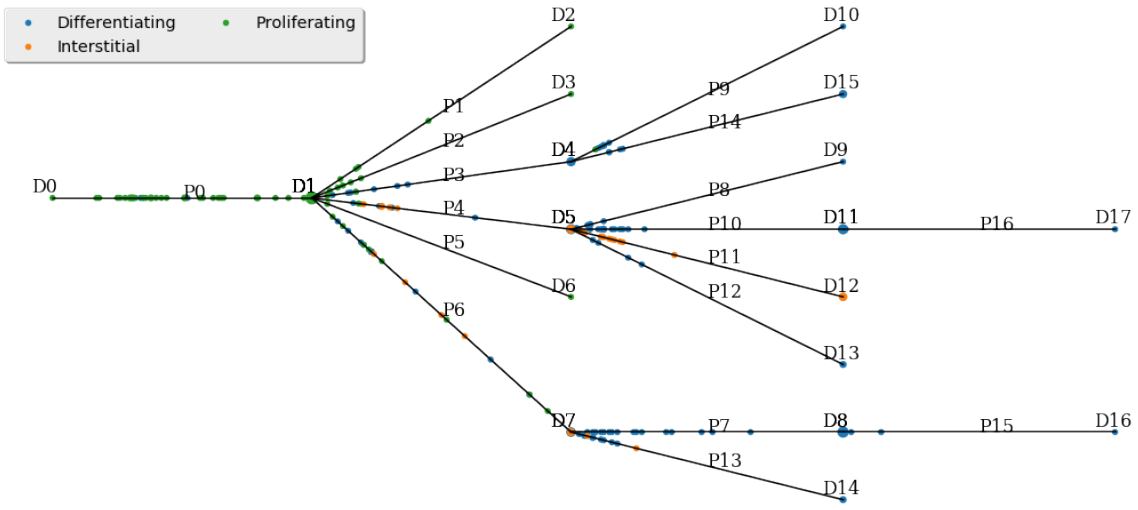

Figure 3.8: CSHMM-TF structure and continuous cell assignment for myoblast dataset.

Table 3.5: The TF assignment to each path for myoblast dataset. Each column shows the top 10 TFs assigned to the path with assigned activation time.

P1	P2	P3	P4	P5	P6	P7	P8
IRF2 0.0	POU3F2 0.0	SRF 0.0	CDC5L 0.0	E2F1 0.0	SRF 0.0	MTF1 0.0	BPTF 0.0
TBP 0.0	ZBTB7A 0.0	BACH2 0.0	MTF1 0.0	DSP 0.0	NKX3-1 0.0	CDC5L 0.0	RFX5 0.0
POU2F1 0.0	IRF3 0.0	E2F5 0.0	TBP 0.0	TBPL1 0.0	PAX6 0.0	SRF 0.0	RFXAP 0.0
	HMGA1 0.0	E2F2 0.0	FOXO1 0.0	E2F7 0.0	MTF1 0.0	TBP 0.0	RFXANK 0.0
	NFYA 0.0	TBP 0.0	ATF6 0.0	RB1 0.0	AR 0.0	CEBPG 0.0	ZNF350 0.0
	NKX3-1 0.0	ATF1 0.0	SRF 0.0	E2F4 0.0	ZNF350 0.0	FOS 0.0	PITX2 0.0
	POU2F1 0.0	NFKB1 0.0	HMGA1 0.5	HMGA1 0.0	POU2F1 0.0	FOSL1 0.0	
		FOXO1 0.0		TFDP1 0.0	HMGA1 0.3	JUNB 0.0	
		ATF3 0.0		E2F3 0.0	TBP 0.5	JUND 0.0	
		CD40 0.1		NKX3-1 0.0	FOXO3 0.5	MEF2A 0.0	
P9	P10	P11	P12	P13	P14	P15	P16
MAX 0.0	ZBTB6 0.0	FOXJ2 0.0	FOSL1 0.0	PITX2 0.0	CEBPD 0.0	YY1 0.0	GLI2 0.0
NFYA 0.0	TBP 0.0	MITF 0.0	JUNB 0.0	SETD2 0.0	TBP 0.0	NR2F1 0.0	GLI3 0.0
NKX3-1 0.0	HMGA2 0.0	MYC 0.0	JUN 0.0	TBP 0.0	HMGA1 0.0	NR2F2 0.0	SRF 0.0
NFIC 0.0	NKX3-1 0.0	UBE4A 0.0	JUND 0.0	HIF1A 0.0	SRF 0.0	CUZD1 0.0	JUNB 0.0
VDR 0.0	POU2F1 0.0	ATF6 0.0	HMGA1 0.0	PBX1 0.0	NFATC1 0.0	RARG 0.0	JUND 0.0
		HMGA1 0.0	NR1H2 0.0		NFATC3 0.0	RARB 0.0	PBX1 0.0
		ZNF350 0.0	NR1H3 0.0		NFATC4 0.0	RARA 0.0	JUN 0.0
		NR2F2 0.0				ATF2 0.0	FOSL2 0.0
		NR1H2 0.0				ATF4 0.0	FOS 0.0
		PBX1 0.0				ATF5 0.0	POU2F1 0.0

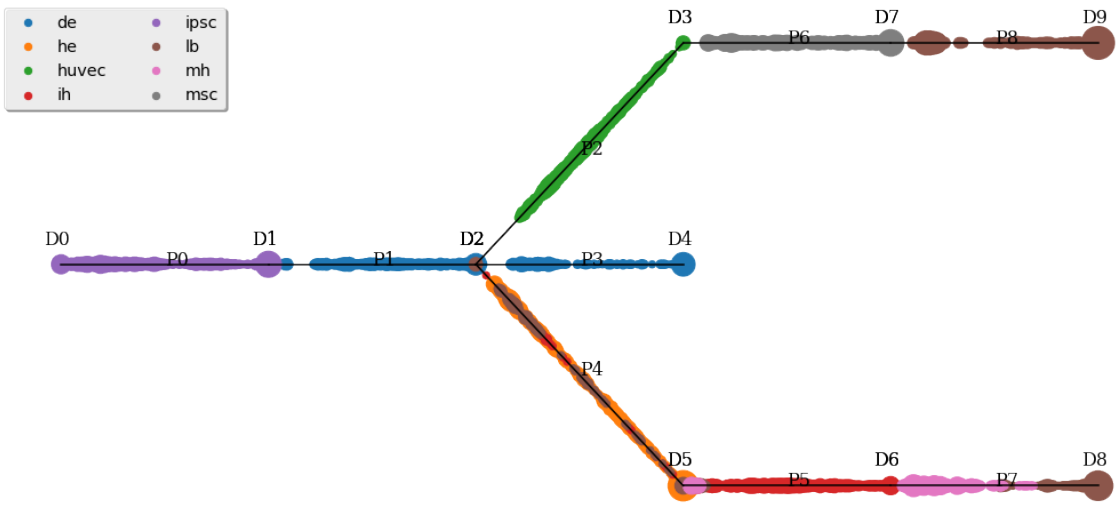

Figure 3.9: (a) CSHMM-TF structure and continuous cell assignment for the simulated liver dataset (∼ 10K cells, 20% dropout).

Table 3.6: The TF assignment to each path for simluated liver dataset (∼10K cells, 20% dropout). Each column shows the top 10 TFs assigned to the path with assigned activation time. Path names are based on annotated cells assigned to that path in the figure.

P1 (DE)	P2 (HUVEC)	P3 (DE)	P4 (HE/MSC)	P5 (IH)	P6 (MSC)	P7 (MH/LB)	P8 (LB)
CDC5L 0.0	FOXJ2 0.0	NFYA 0.0	E2F3 0.0	CDC5L 0.0	TBPL1 0.0	TBPL1 0.0	HMGA1 0.0
TBPL1 0.0	HMGA1 0.0	E2F4 0.0	NFATC2 0.0	HMGA1 0.1	NKX6-2 0.0	HMGA1 0.3	SRF 0.0
HMGA1 0.0	NFATC1 0.0	E2F7 0.0	NFATC1 0.0	TOPORS 0.1	CEBPB 0.0		CEBPD 0.0
NKX3-1 0.0	NFATC2 0.0	HMGA1 0.1	NFATC4 0.0		TFDP1 0.0		TBP 0.0
ZNF219 0.0	NFATC3 0.0	TBPL1 0.3	NFATC3 0.0		DSP 0.0		CEBPG 0.0
TBP 0.1	NFATC4 0.0	DSP 0.3	E2F4 0.0		E2F4 0.0		NKX6-2 0.0
	SOX5 0.0	RB1 0.3	UBE4A 0.0		HMGA2 0.0		CDC5L 0.0
		E2F5 0.3	TBP 0.1		FOXJ2 0.0		FOXJ2 0.1
		E2F2 0.3	CEBPG 0.1		E2F7 0.1		HMGA2 0.1
		E2F1 0.5	DSP 0.5		HMGA1 0.5		GATA6 0.2

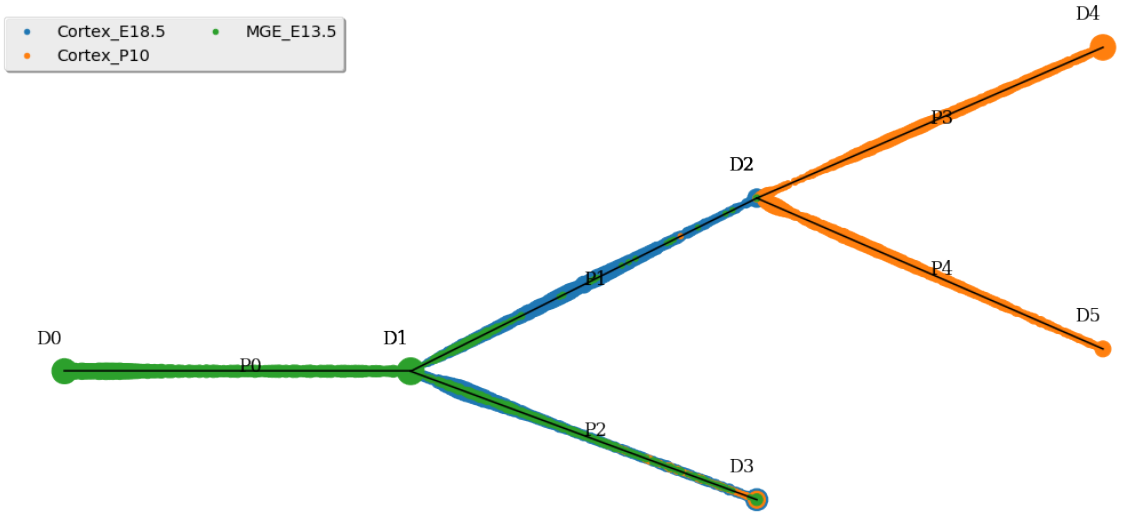

Figure 3.10: CSHMM-TF structure and continuous cell assignment for mouse cortical dataset. Cells are labeled based on cell types and sampled time. E means embryonic days and P means postnatal days. As can be seen, the model correctly assigns cells based on their biological order (MGE-E18-P1). The model also assigns several relevant TFs to these paths as shown in Table 3.7

.

Table 3.7: The TF assignment to each path for mouse cortical data (∼21K cells ∼10K genes). Each column shows the top 10 TFs assigned to the path with assigned activation time.

P1	P2	P3	P4
UBP1 0.0	CREB1 0.0	HLF 0.0	TBPL1 0.0
CLOCK 0.0	MYC 0.0	CLOCK 0.0	CLOCK 0.0
TFAP4 0.0	MAX 0.0	YY1 0.0	ARNT2 0.0
NFIL3 0.1	YY1 0.0	NFIL3 0.0	ELK1 0.1
ATF2 0.1	ELK1 0.0	ATF2 0.0	MEF2A 0.4
MAZ 0.2	CEBPG 0.0	SOX5 0.0	YY1 0.5
ELK1 0.2	PBX1 0.0	SOX11 0.1	
YY1 0.5	SOX11 0.0	SOX12 0.1	
	SOX12 0.0	SOX2 0.1	
	PATZ1 0.1	SOX4 0.1	

Table 3.8: The Spearman correlation for expression of TF interactions pairs identified in Figure 3.5 in Chapter 3.

Dataset	Path	TF1	TF2	Correlation
Liver	P3	sox9	nfatc1	0.52
Liver	P4	tfdp1	e2f1	0.95
Liver	P4	tfdp1	e2f3	0.74
Liver	P4	tfdp1	e2f5	0.74
Liver	P11	srf	nkx3-1	0.82
Lung	P1	yy1	atf2	0.57
Lung	P1	yy1	creb1	0.68
Lung	P1	yy1	e2f4	0.60
Lung	P2	cebpb	cebpd	0.89
Lung	P4	tfdp1	e2f4	0.67
Lung	P4	tfdp1	e2f7	0.47
Lung	P4	tfdp1	e2f2	0.44
Lung	P4	tfdp1	e2f5	0.75
Neuron	P1	jun	fos	0.18
Neuron	P1	jun	fosl1	0.31
Neuron	P2	tfdp1	e2f1	0.88
Neuron	P2	tfdp1	e2f4	0.55
Neuron	P2	tfdp1	e2f3	0.38
Neuron	P6	sox4	tcf7l2	0.43
Neuron	P6	sox11	tcf7l2	0.65
Neuron	P7	atf3	fos	0.30

Table 3.9: The partial order list of lung/neuron/liver dataset for calculating the quantitative distance measure

dataset	**partial order list**
lung	(BP,AT1),(BP,AT2)
neuron	(MEF,d2_intermediate),(d2_intermediate,d5_intermediate),(d2_induced,d5_earlyiN) ,(d5_earlyiN,Neuron),(d5_earlyMyocyte,Myocyte)
liver	(iPSC,DE),(DE,HE),(IH,MH)

Table 3.10: The quantitative distance measure reduction in % for lung/neuron/liver datasets. Larger values are better. The partial order list of each dataset are shown in Table 3.9

dataset	**CSHMM-TF vs. CSHMM**	**CSHMM-TF vs. CSHMM-randomTF**
lung	9.170305677	3.711790393
neuron	1.240238861	0.780891135
liver	1.963861879	5.076180997

Table 3.11: The comparison number of significant TF and the minimum p-value between CSHMM-TF and CSHMM-randomTF for lung/neuron/liver datasets (We define p-value ≤ 0.001 as significant here)

	CSHMM-TF		**CSHMM-randomTF**	
dataset	**# significant TF**	**min p-value**	**# significant TF**	**min p-value**
lung	44	4.74E-09	2	3.15E-04
neuron	14	3.64E-08	1	1.96E-04
liver	12	1.62E-07	6	1.44E-04

Chapter 4

Single-cell Lineage Tracing by Integrating CRISPR-Cas9 Mutations with Transcriptomic Data

In chapter 2 and 3, we have introduced CSHMM and its extension CSHMM-TF for modeling time-series scRNA-Seq dataset to reconstruct continuous single-cell developmental trajectories and these trajectories and help researchers to study cell lineage. In chapter 1, we have also mentioned that there are technologies that can put heritable marks on single cells to study lineage and recent studies even allow simultaneously profile scRNA-Seq dataset with artificial genetic markers. However, previous studies only analyze them separately and do not combine these two dataset together to learn the cell lineage tree. Here we present LinTIMaT, which is the model that learns a cell lineage tree based on both scRNA-Seq and mutation data. The chapter has been adapted with changes from our paper under review in *Nature Communications*: Zafar, Hamim, Chieh Lin, and Ziv Bar-Joseph. "Single-cell Lineage Tracing by Integrating CRISPR-Cas9 Mutations with Transcriptomic Data".

To improve the reconstruction of lineages from CRISPR-Cas9 mutations and scRNA-seq data, we developed a novel statistical method, LinTIMaT (**Lin**eage **T**racing by **I**ntegrating **M**utation **a**nd **T**ranscriptomic data) that integrates mutational and transcriptomic data for reconstructing lineage trees in a maximum-likelihood framework. LinTIMaT employs a novel likelihood function for evaluating different tree structures based on mutation information. It then defines a new likelihood optimization problem which combines the likelihood score for the mutation data with Bayesian hierarchical clustering [93], which evaluates the coherence of the expression information such that the resulting tree concurrently maximizes agreement for both transcriptomics and genetic markers from the same cell. The tree space is explored by a novel heuristic search algorithm that first infers a lineage tree based on mutation information and further refines it based on both mutation and expression information. Finally, LinTIMaT also employs an algorithm for integrating lineages reconstructed for different individuals of the same species for inferring an invariant lineage tree. We applied LinTIMaT to both, simulated mutation data where ground truth is known and to zebrafish datasets generated using two different technologies [4, 154]. As we show, by integrating transcriptomic and mutational data, LinTIMaT was able to improve the reconstruction of lineages when compared to MP method. In addition, we used LinTIMaT to

combine data from multiple individuals for reconstructing an invariant lineage. As we show, such invariant lineage further improved on each of the individual lineages in terms of both clade homogeneity and functional assignment for the cells residing on the leaves of the lineage tree.

4.1 Overview of LinTIMaT

To enable the accurate reconstruction of individual and invariant lineages, we developed LinTIMaT that integrates CRISPR-Cas9 mutations with transcriptomic data from single cells. An overview of the algorithm is shown in Figure 4.1.

We assume that the cell lineage tree is a rooted directed tree (Figure 4.1a). The root of this lineage tree denotes the initial cells that do not contain any marker (or editing event). The leaves of this tree denote the cells from which the mutated barcodes and RNA-seq data have been recovered. The CRISPR-Cas9 edits are acquired on the branches of the cell lineage tree as the single-cell zygote transforms into an adult organism. For expression data, the method assumes that cells under an internal node can either display similar expression profile (low variance owing to similar cell type) or two or more different expression profiles (high variance) if they later split into multiple cell types. The generative process assumed by LinTIMaT is presented in Figure 4.2.

LinTIMaT reconstructs the lineage tree by maximizing a likelihood function that accounts for both mutations and expression data. The likelihood function imposes a Camin-Sokal parsimony criterion for each synthetic marker. The probability associated with a transition of mutation state for a marker along a branch of the lineage tree is computed based on the abundance of the marker in the single cells. To compute the expression likelihood based on the transcriptomic data, the lineage is modeled as a Bayesian hierarchical clustering (BHC) [93] of the cells and the marginal likelihoods of all the partitions consistent with the given lineage tree are computed based on a Dirichlet process mixture model. To optimize the tree topology, we employ a heuristic search algorithm, which stochastically explores the space of lineage trees.

The above algorithm reconstructs trees for a specific CRISPR-Cas9 mutation set. To integrate trees resulting from repeat experiments of the same organism, LinTIMaT further reconstructs a species invariant lineage tree (Figure 4.1b). Our model assumes that a subset of the lineages (and cells) are conserved between different individuals of the same species. Our invariant lineage tree reconstruction algorithm attempts to identify such invariant groups of cells based on both, their similar expression pattern and their branching history. The method starts with an initial greedy matching and iterates to minimize an objective function consisting of two distance functions, the first is aimed at minimizing the disagreement between the topology of the invariant lineage tree and the individual lineage trees while the second distance is minimized for improving the matching of the preserved clusters.

4.2 Likelihood of a cell lineage tree

We assume that the cell lineage tree is a rooted directed tree $\mathcal{T}$. The root of this lineage tree denotes the initial cell that does not contain any marker (or editing event). The leaves of

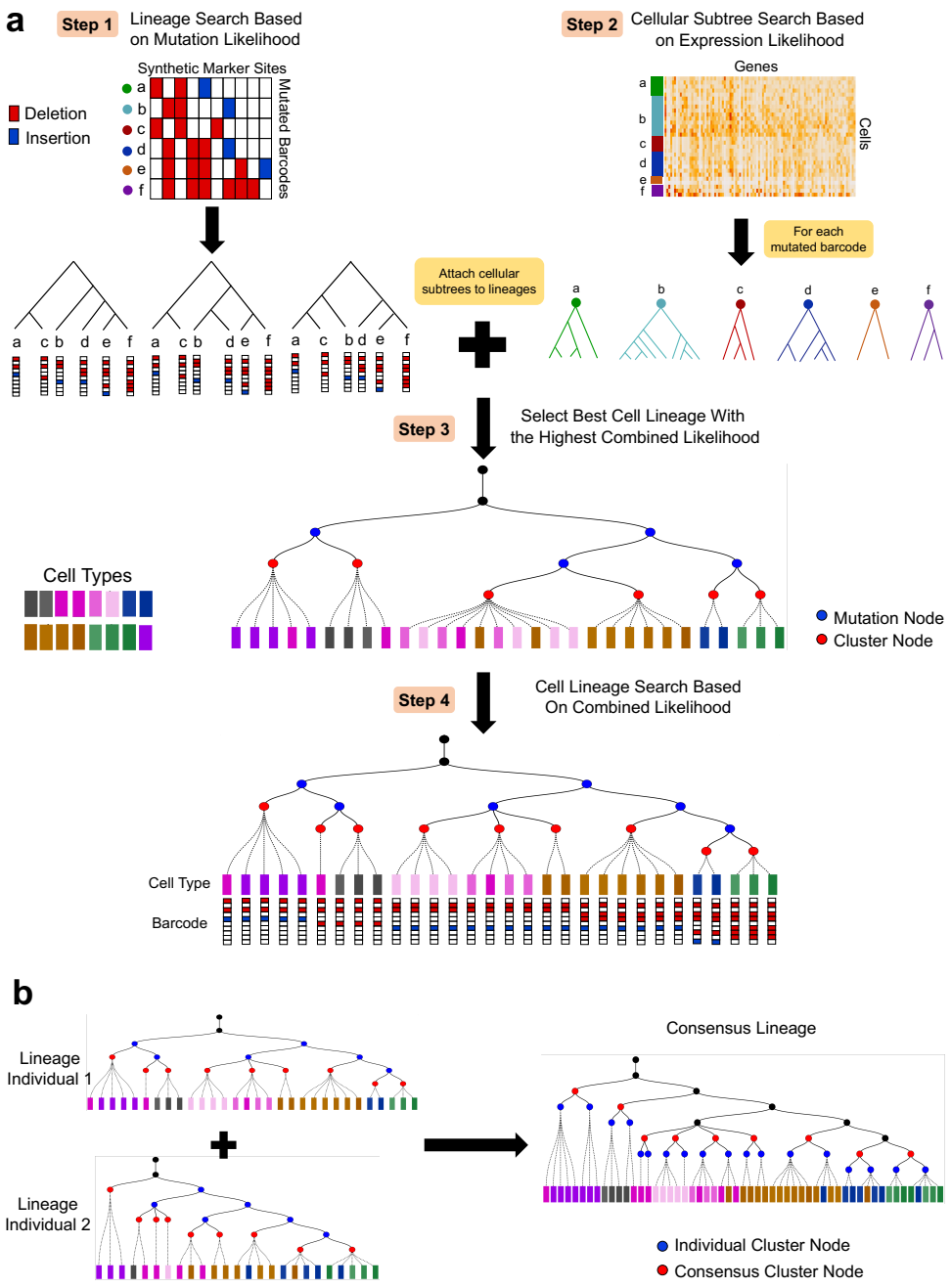

Figure 4.1: Overview of LinTIMaT. (a) LinTIMaT reconstructs a cell lineage tree by integrating CRISPR-Cas9 mutations and transcriptomic data. In Step 1, LinTIMaT infers top scoring lineage trees built on barcodes using only mutation likelihood. In Step 2, for all cells carrying the same barcode, LinTIMaT reconstructs a cellular subtree based on expression likelihood. In Step 3, cellular subtrees are attached to barcode lineages to obtain cell lineage trees and the tree with the best combined likelihood is selected. Finally, LinTIMaT uses a hill-climbing search for refining the cell lineage tree by optimizing the combined likelihood (Step 4). (b) To reconstruct a invariant lineage, LinTIMaT performs an iterative search that attempts to minimize the distance between individual lineage trees and the invariant tree topology. As part of the iterative process, LinTIMaT matches clusters in one individual tree to clusters in other individual tree(s) such that leaves in the resulting invariant tree contain cells from all individual studies. See Methods for complete details.

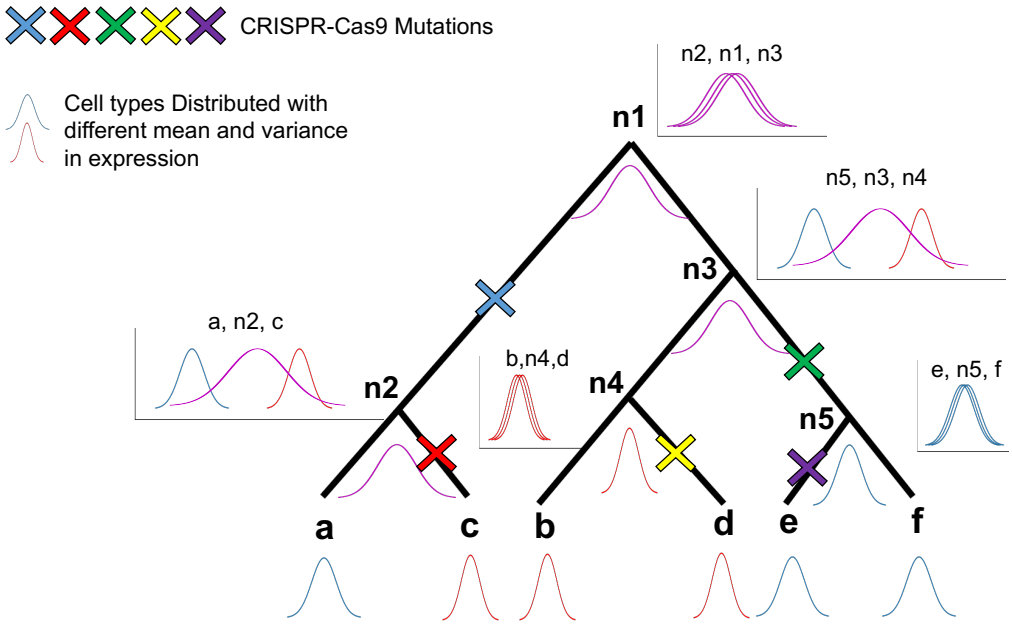

Figure 4.2: Generative process of LinTIMaT. Different CRISPR-Cas9 mutations are acquired on the branches of the lineage. a,b,c,d,e,f represent different barcodes. a, e and f contain cells from cell type 1 (blue); b, c and d contain cells from cell type 2 (red). The gene-expression at an internal node follows a Gaussian distribution based on the cells in the subtree rooted at the node. If children have similar distribution, then the internal node will also have similar distribution (e.g., n1, n4, n5). If children have different distribution, the internal node will have a distribution with larger variance (e.g., n2, n3). Cells with similar expression can occur in distant branches of the cell lineage. For example, c has similar expression profile as b and d; a has similar expression profile as e and f but because of their different mutation profile, LinTIMaT is able place them on distant branches.

this tree denote cells profiled in the experiments. Cells go through the differentiation process along the branches of the lineage tree and as part of this process acquire the synthetic mutations (edits). Some of the internal nodes in the cell lineage tree represent the unique mutated barcodes shared by the leaves (cells) under that specific internal node. For ease of computation, we first reconstruct a rooted binary lineage tree and later eliminate the internal branchings that are not supported by any synthetic mutations. As mentioned in the previous section, our method aims to reconstruct a cell lineage tree by combining two complementary types of data. For this, we defined a joint likelihood function for the two data types and then search the space of possible trees for a model that maximizes the likelihood function. We first describe the likelihood function for each of the data types and then discuss how to perform a search for maximizing the joint likelihood to reconstruct the most likely tree.

4.2.1 Mutation likelihood

The first component of the likelihood function evaluates the likelihood of the cell lineage tree based on the mutation data. The mutations induced by Cas9 are irreversible since the Cas9 protein cannot bind to the target sites once changed. To account for this, we impose a Camin-Sokal parsimony criterion [30] on each synthetic mutation. This criterion states that each synthetic mutation can be acquired at least once along the lineage but once acquired they are never lost. We also assume that the synthetic mutations are acquired independently and parsimoniously as higher number of mutations along the branches of the cell lineage indicates a more complex mutational history which is less likely. For a given cell lineage tree $\mathcal{T}$, we first use Fitch's algorithm [65] to assign ancestral states for each marker to each internal node of the tree satisfying maximum parsimony. Such an assignment, $\mathcal{A}$ results in the least number of mutations on the given tree. The mutation likelihood ($\mathcal{L}_M$) of the cell lineage tree is then given by

$$\mathcal{L}_M(\mathcal{T}) = P(\mathcal{E}|\mathcal{T}, \mathcal{A}) = \prod_{s=1}^{S} P(\mathcal{E}_{*s}|\mathcal{T}, \mathcal{A}_s) \tag{4.1}$$

where $\mathcal{E}_{*s}$ is the observed data for marker s which is a vector corresponding to N values for N cells. $\mathcal{A}_s$ denotes the parsimonious assignment of ancestral states for all internal nodes for marker s.

For an internal node v with children u and w, $L_s^v(\mathcal{A})$ denotes the partial conditional likelihood for marker s defined by

$$L_s^v(\mathcal{A}_s^v = x) = P(\mathcal{E}_s^v|\mathcal{T}, \mathcal{A}_s^v = x) \tag{4.2}$$

where $\mathcal{E}_s^v$ denotes the restriction of observed data for marker s, $\mathcal{E}_{*s}$ to the descendants of node v subject to the condition that $\mathcal{A}_s^v = x$ is the ancestral state for marker s assigned by Fitch's algorithm, $x \in \{0, 1\}$. L_s^v gives the likelihood for marker s for the subtree rooted at node v, given the assignment of ancestral states by Fitch's algorithm.

The likelihood for the full observed data $\mathcal{E}_{*s}$ for marker s is given by

$$P(\mathcal{E}_{*s}|\mathcal{T}, \mathcal{A}) = L_s^r(\mathcal{A}_s^r = 0) \tag{4.3}$$

where r is the root of the lineage tree. Since, the root of the tree does not contain any synthetic mutation, $\mathcal{A}_s^r = 0, \forall s \in \{1, 2, \ldots, S\}$. For any internal node v with children u and w, the partial

conditional likelihood satisfies the recursive relation

$$L_s^v = \left[P_{t_{\mathcal{A}_s^v \to \mathcal{A}_s^u}} L_s^u\right] \left[P_{t_{\mathcal{A}_s^v \to \mathcal{A}_s^w}} L_s^w\right] \tag{4.4}$$

$P_{t_{\mathcal{A}_s^v \to \mathcal{A}_s^u}}$ and $P_{t_{\mathcal{A}_s^v \to \mathcal{A}_s^w}}$ denote the transition probabilities on branches that connect v and u, and v and w respectively. For each synthetic mutation s, we define a transition probability matrix given by

$$P_t^s = \begin{bmatrix} 1 - m_s & m_s \\ 0 & 1 \end{bmatrix} \tag{4.5}$$

where m_s denotes the fraction of cells harboring s and $P_t^s(i, j)$ denotes the probability of transition from state i to state j along any branch of the tree. If a mutation assignment violates the Camin-Sokal parsimony criterion (i.e. a mutation is reversed), the log-likelihood is heavily penalized (-100000) so that LinTIMaT prefers the tree without such violation.

For each leaf l of the tree, the partial likelihood is set to $L_s^l = 1$. It is important to note that our mutation likelihood function does not explicitly model the editing rate at each CRISPR target.

4.2.2 Expression likelihood

For the expression data likelihood, we model the lineage as a Bayesian hierarchical clustering (BHC) [93] of the cells and used the likelihood formulation provided by BHC. BHC is a bottom-up agglomerative clustering method that iteratively merges clusters based on marginal likelihoods. Following several other methods we assume a diagonal matrix when computing gene expression variance for each internal and leaf node [115, 123]. Following BHC algorithm, we compute the marginal likelihoods of all the partitions consistent with the given lineage tree based on a Dirichlet process mixture model. The expression likelihood ($\mathcal{L}_E$) for the complete dataset is given by the marginal likelihood for the root of the tree and it essentially provides a lower bound on the marginal likelihood of a Dirichlet process mixture model.

$$\mathcal{L}_E(\mathcal{T}) = P(\mathcal{Y}|\mathcal{T}) = \mathcal{L}_{\mathcal{G}}^r \tag{4.6}$$

where $\mathcal{Y}$ is the $N \times G$ gene-expression matrix, $\mathcal{G}$ is the set of G genes and $P(\mathcal{Y}|\mathcal{T})$ is the expression likelihood for the lineage tree which is also the marginal likelihood ($\mathcal{L}_{\mathcal{G}}^r$) for the root of the tree.

For an internal node v with children u and w, $\mathcal{T}^v$ denotes the subtree rooted at v. Let $\mathcal{Y}^v \subset \mathcal{Y}$ be the set of gene expression data at the leaves under the subtree $\mathcal{T}^v$ and $\mathcal{Y}^v = \mathcal{Y}^u \bigcup \mathcal{Y}^w$. To compute the marginal likelihood for node v ($\mathcal{L}_{\mathcal{G}}^v$), we compute the probability of the data under two hypotheses of BHC. The first hypothesis, $\mathcal{H}_1^v$ assumes that each data point is independently generated from a mixture model and each cluster corresponds to a distribution component. This means that the data points $\mathbf{y}^{(i)}$ in the cluster $\mathcal{Y}^v$ are independently and identically generated from a probabilistic model $P(\mathbf{y}|\theta)$ with parameters θ. Thus, the marginal probability of the data $\mathcal{Y}^v$

under the hypothesis $\mathcal{H}_1^v$ is given by

$$\begin{aligned} P(\mathcal{Y}^v|\mathcal{H}_1^v) &= \int P(\mathcal{Y}^v|\theta)P(\theta|\beta)d\theta \\ &= \int \left[\prod_{\mathbf{y}^{(i)}\in\mathcal{Y}^v} P(\mathbf{y}^{(i)}|\theta)\right] P(\theta|\beta)d\theta \end{aligned} \tag{4.7}$$

The integral in equation 4.7 can be made tractable by choosing a distribution with conjugate prior, as discussed in Appendix C Supplementary Methods.

The alternative hypothesis $\mathcal{H}_2^v$ assumes that there are two or more clusters in $\mathcal{Y}^v$. Instead of summing over all (exponential) possible ways of dividing $\mathcal{Y}^v$ into two or more clusters, we follow the strategy in BHC [93] and sum over the clusterings that partition the data $\mathcal{Y}^v$ in a way that is consistent with the subtrees $\mathcal{T}^u$ and $\mathcal{T}^w$. This gives us the probability of the data under the alternative hypothesis

$$P(\mathcal{Y}^v|\mathcal{H}_2^v) = \mathcal{L}_\mathcal{G}^u\mathcal{L}_\mathcal{G}^w = P(\mathcal{Y}^u|\mathcal{T}^u)P(\mathcal{Y}^w|\mathcal{T}^w) \tag{4.8}$$

In equation 4.8, $P(\mathcal{Y}^u|\mathcal{T}^u)$ and $P(\mathcal{Y}^w|\mathcal{T}^w)$ represent the marginal likelihoods of subtrees rooted at nodes u and w respectively. Combining the two likelihoods of the two hypotheses leads to a recursive definition of the marginal likelihood for the subtree $\mathcal{T}^v$ rooted at the node v

$$\mathcal{L}_\mathcal{G}^v = P(\mathcal{Y}^v|\mathcal{T}^v) = \pi_v P(\mathcal{Y}^v|\mathcal{H}_1^v) + (1-\pi_v)P(\mathcal{Y}^u|\mathcal{T}^u)P(\mathcal{Y}^w|\mathcal{T}^w) \tag{4.9}$$

Where π_v is a parameter for weighting the two alternatives and is defined recursively for every node. The recursive definition of π_v for node v is given by

$$\pi_v = \frac{\alpha\Gamma(n_v)}{d_v}, \quad d_v = \alpha\Gamma(n_v) + d_u d_w \tag{4.10}$$

In equation 4.10, α denotes a hyperparameter, the concentration parameter of the Dirichlet process mixture model, n_v is the number of data points under the subtree $\mathcal{T}^v$ and $\Gamma(.)$ is the Gamma function. For each leaf l, we set the values $\pi_l = 1$ and $d_l = \alpha$. Also, for each leaf l, the marginal likelihood ($\mathcal{L}_\mathcal{G}^l$) is calculated based on only the first hypothesis

$$\mathcal{L}_\mathcal{G}^l = P(\mathcal{Y}^l|\mathcal{H}_1^l). \tag{4.11}$$

See Appendix C section 4.14.1 for discussion on how the prior is set for this model.

4.2.3 Combined likelihood

For a given lineage tree, the joint log-likelihood ($\mathcal{L}_T$) function for the mutation and expression data is a weighted sum given by

$$\mathcal{L}_T(\mathcal{T}) = \omega_1 \log \mathcal{L}_M(\mathcal{T}) + \omega_2 \log \mathcal{L}_E(\mathcal{T}) \tag{4.12}$$

The values of ω_1 and ω_2 are chosen so that the values of the two likelihood components stay in the same range. In our experiments, we have used $\omega_1 = 50$ and $\omega_2 = 1$ (see Appendix C Figure 4.27).

4.3 Search algorithm for inferring lineage tree

Searching for the optimal tree under a maximum-likelihood framework like ours is a NP hard problem [64]. We have thus developed a heuristic search algorithm which stochastically explores the space of lineage trees. The search algorithm consists of several stages as described below.

1. In the first step, we only focus on the barcodes and search for top scoring solutions. The search process starts from a random tree topology built on B leaves corresponding to B unique barcodes. In searching the barcode lineage tree, we employ the mutation likelihood function. In each iteration, a new barcode lineage tree, $\mathcal{T}_B'$ is proposed from the current tree $\mathcal{T}_B$ as we discuss below. If the proposed tree results in a higher likelihood, it is accepted, otherwise rejected. Instead of storing a single solution, we keep several of top scoring barcode lineage trees.

$$\mathcal{T}_B^{[1]}, \mathcal{T}_B^{[2]}, \ldots, \mathcal{T}_B^{[t]} = \underset{\mathcal{T}_B}{\operatorname{argmax}}\, \mathcal{L}_M(\mathcal{T}_B) = \underset{\mathcal{T}_B}{\operatorname{argmax}}\, P(\mathcal{E}_{B\times S}|\mathcal{T}_B, \mathcal{A}) \tag{4.13}$$

2. Next, we utilize the expression data. As mentioned above, a barcode can be shared between multiple cells. We thus next search for the best cellular subtree ($\mathcal{T}_b$) for the set of cells associated with each mutated barcode b. We employ hill-climbing to obtain single solution for each barcode that harbors more than 2 cells.

$$\mathcal{T}_b = \underset{\mathcal{T}}{\operatorname{argmax}}\, P(\mathcal{Y}_{(c|z_c=b)*}|\mathcal{T}) \quad \forall b \in \{1, \ldots, B\} \tag{4.14}$$

3. In the third step, we construct complete cell lineage trees by attaching cellular subtrees for each barcode to barcode lineage trees. To obtain the cell lineage tree $\mathcal{T}_i$ from a barcode lineage tree $\mathcal{T}_B^{[i]}$, for each barcode b harboring more than 2 cells, we choose the cellular subtree $\mathcal{T}_b$ inferred in step 2 and connect its root to the leaf in $\mathcal{T}_B^{[i]}$ that corresponds to b. For a barcode b shared by two cells, the cells are connected to the leaf representing b in $\mathcal{T}_B^{[i]}$ as children. This gives us t full binary cell lineage trees corresponding to t barcode lineage trees. Next, we evaluate the total log-likelihood of each of these cell lineage trees and choose the best one.

$$\mathcal{T}^+ = \underset{\mathcal{T}_i, i=1,\ldots,t}{\operatorname{argmax}}\, \mathcal{L}_T(\mathcal{T}_i) \tag{4.15}$$

   We also record the best mutation log-likelihood, $\mathcal{L}_M^{best}$ for the best cell lineage tree and define a threshold value for mutation log-likelihood

$$\mathcal{L}_M^{thr} = \mathcal{L}_M^{best} + thr \times \mathcal{L}_M^{best} \tag{4.16}$$

   where, thr is a user-defined value close to 0.

4. In the final step, we perform another hill-climbing search to optimize the cell lineage tree $\mathcal{T}^+$ inferred in step 3 in terms of the joint likelihood function. The search starts from $\mathcal{T}^+$ and in each iteration, we propose a new cell lineage tree $\mathcal{T}'$ from the current tree $\mathcal{T}$ as we discuss below. For the new tree, we first ensure that the mutation log-likelihood of the new tree does not go below $\mathcal{L}_M^{thr}$. If this condition is satisfied and the total likelihood is

improved then the new lineage tree is accepted. We stop the search if the total likelihood does not improve for a large number of iterations and return the best lineage tree achieved so far.

$$\mathcal{T}_{best} = \underset{\mathcal{T}}{\operatorname{argmax}}\, \mathcal{L}_T(\mathcal{T}) \tag{4.17}$$

4.4 Tree search moves

To explore the space of lineage trees, LinTIMaT employ two different types of moves that can make small and big changes in the tree topology. For this, we adopt two of the tree proposals described in [107] for efficient exploration of tree space for Bayesian phylogenetic inference. Both of these moves are branch-rearrangement proposals that alter the topology of the lineage tree.

The first tree proposal is a swapping move called Stochastic Nearest Neighbor Interchange (stNNI). In this move, we choose an internal branch as the focal branch and stochastically swap the subtrees attached to the focal branch. This type of move results in minimal topology change and is used only in the second step of our algorithm that infers cellular subtree for each mutated barcode.

The second tree proposal is a pruning-regrafting move, namely Random Subtree Pruning and Regrafting (rSPR). In this move, we first randomly select an interior branch, prune a subtree attached to that branch, and then reattach the subtree to another regrafting branch present in the other subtree. The regrafting branch is also chosen randomly. This type of move can introduce a larger amount of topology change in the tree and this is used in step 1 and 4 of our search algorithm.

4.5 Inferring clusters from cell lineage tree

To obtain cell clusters from the inferred lineage tree, we employ the statistical model comparison criterion provided by the BHC model for gene expression data. For an internal node v with children u and w, we compute the probability of the data under two hypotheses. The first hypothesis suggests that all the cells under the node v belongs to a single cluster. We compute the posterior probability (r_v) of this hypothesis using Bayes rule:

$$r_v = P(\mathcal{H}_1^v|\mathcal{Y}^v) = \frac{\pi_v P(\mathcal{Y}^v|\mathcal{H}_1^v)}{\pi_v P(\mathcal{Y}^v|\mathcal{H}_1^v) + (1-\pi_v)P(\mathcal{Y}^u|\mathcal{T}^u)P(\mathcal{Y}^w|\mathcal{T}^w)} \tag{4.18}$$

The lineage tree can be cut at the nodes where r_v goes from $r_v < 0.5$ to $r_v > 0.5$ to obtain clustering of cells.

4.6 Combining lineage trees from multiple individuals to reconstruct a invariant lineage tree

As mentioned in the Introduction, a key challenge when working with CRISPR mutation data is the fact that these are not the same across different experiments. Thus, standard phy-

logenetic invariant tree building cannot be applied to this data. Instead, given a set of lineage trees, $\{\mathcal{T}_1, \ldots, \mathcal{T}_I\}$ for I individuals, we construct a single lineage tree $\mathcal{T}_{cons}$ that jointly explains the differentiation of these individual organisms. Individual lineage trees that are input to the invariant lineage reconstruction method are built on a leaf set of different number of cells. $\mathcal{T}_{cons}$ is constructed by following the steps below.

1. For each individual lineage tree $\mathcal{T}_i$, we denote by C_{ij} cluster j (a leaf) in tree $\mathcal{T}_i$
2. We remove all clusters with less than a pre-determined number of cells, tc (here we use $tc = 3$)
3. Next, for each pair of remaining clusters from two individual trees, $C_{i1,j1}$,$C_{i2,j2}$, we calculate their distance d based on gene expressions and only keep to top $x\%$ of pairs with the smallest distance (here $x = 1$).
4. Using these score we perform greedy matching. We select the cluster pair with the smallest distance, if both clusters are not matched we match the clusters and continue until no more matches can be made. This process results in K matched cluster pairs which are used in the invariant linage tree.
5. For each individual lineage tree $\mathcal{T}_i$, we obtain the backbone tree $\mathcal{T}_i^c$ built using these K clusters.
6. $\mathcal{T}_{cons}$ is a lineage tree built on a leaf set of K clusters. We first define a cluster matching $\mathcal{M}$ as a matching where each cluster in each individual lineage tree $\mathcal{T}_i$ (or each leaf in $\mathcal{T}_i^c$) is matched with a leaf of $\mathcal{T}_{cons}$. We reconstruct $\mathcal{T}_{cons}$ and a cluster matching $\mathcal{M}_{cons}$ by minimizing an objective function given by

$$\mathcal{T}_{cons}, \mathcal{M}_{cons} = \underset{\mathcal{T}^*, \mathcal{M}^*}{\operatorname{argmin}}\, \omega_1 \sum_{i=1}^{I} \mathcal{S}(\mathcal{T}^*, \mathcal{T}_i^c) + \omega_2 \sum_{j=1}^{K} \mathcal{E}(c_j) \tag{4.19}$$

   where $\mathcal{T}^*$ is a candidate invariant lineage, $\mathcal{M}^*$ is a candidate cluster matching, $\mathcal{S}(\mathcal{T}^*, \mathcal{T}_i^c)$ denotes the sum of pairwise leaf shortest path distance between candidate invariant lineage $\mathcal{T}^*$ and individual lineage $\mathcal{T}_i^c$, $\mathcal{E}(c_j)$ denotes the sum of pairwise distance between the clusters of the individual lineage trees that match with cluster (or leaf) c_j in the candidate invariant lineage. The objective function for searching the invariant lineage and the optimal cluster matching is described below in detail. We employ a two-step heuristic search algorithm for optimizing the objective function (described below).

4.7 Objective function for searching invariant lineage tree

The objective function for reconstructing the invariant lineage attempts to balance two competing issues. The first is that the invariant tree should be as close as possible to each of the individual lineages. The second is that the agreement (in terms of expression) between nearby subtrees in the invariant tree would be high. We thus attempt to minimize two different distance functions to select the optimal tree. $\mathcal{D}_{\mathcal{S}} = \sum_{i=1}^{I} \mathcal{S}(\mathcal{T}^*, \mathcal{T}_i^c)$ computes the distance (or disagreement) between the topology of the invariant lineage and the individual lineage trees.

$\mathcal{D}_{\mathcal{E}} = \sum_{j=1}^{K} \mathcal{E}(c_j)$ is the other distance function which attempts to minimize disagreement between the gene expression values of matched clusters.

For computing $\mathcal{D}_{\mathcal{S}}$, we employ the sum of pairwise leaf shortest-path distance [163, 237] between two trees as a distance measure for comparing two tree topologies. The shortest path distance $\delta_{ij}(.)$ between two leaves c_i and c_j in a tree is given by the sum of the number of edges that separate them from their most recent common ancestor. Overall pairwise leaf shortest-path distance between two trees is obtained by summing up the absolute differences between the shortest-path distances of all unordered pairs of leaves in the two trees

$$\mathcal{S}(\mathcal{T}_1, \mathcal{T}_2) = \sum_{i=0}^{K-1} \sum_{j=1+1}^{K} |\delta_{ij}(\mathcal{T}_1) - \delta_{ij}(\mathcal{T}_2)| \tag{4.20}$$

For computing $\mathcal{D}_{\mathcal{E}}$, we sum the pairwise distance between the clusters of the individual lineage trees that match with a leaf of the invariant lineage. $\mathcal{E}(c)$ is given by

$$\mathcal{E}(c) = \sum_{i=1}^{I-1} \sum_{k=i+1}^{I} e(l_i^c, l_k^c) \tag{4.21}$$

where l_i^c and l_k^c denote clusters in individual lineages that match with leaf c in candidate invariant lineage. $e(.)$ denotes the Euclidean distance between the gene expression value of two clusters.

4.8 Search algorithm for inferring invariant lineage

We use a two-step heuristic search algorithm for inferring the invariant lineage and the corresponding cluster matching.

1. The first step employs an iterative search. In each iteration, we first find a better cluster matching (see Appendix C Supplementary Methods for details) than the current matching $\mathcal{M}^*$, and then keeping this matching fixed, we improve the topology of the invariant tree. It is important to note that, a new cluster matching modifies both $\mathcal{D}_{\mathcal{E}}$ and $\mathcal{D}_{\mathcal{S}}$, whereas a new tree topology modifies only $\mathcal{D}_{\mathcal{S}}$. This iterative search goes on until cluster matching can not be improved further. Let us assume, $\mathcal{D}_{\mathcal{E}}^{best}$ is the distance corresponding to the best cluster matching achieved. We define a threshold value for the cluster matching distance
$$\mathcal{D}_{\mathcal{E}}^{thr} = \mathcal{D}_{\mathcal{E}}^{best} + thr \times \mathcal{D}_{\mathcal{E}}^{best} \tag{4.22}$$
2. In the second step, we try to improve the invariant lineage by improving the objective function $\mathcal{D}_{\mathcal{S}} + \mathcal{D}_{\mathcal{E}}$ using a stochastic search. In the joint $(\mathcal{T}^*, \mathcal{M}^*)$ space, we consider two types of moves to propose a new configuration. In each iteration, from the current configuration $(\mathcal{T}^*, \mathcal{M}^*)$, we either propose a new matching (Appendix C Supplementary Methods) $\mathcal{M}_{new}^*$ or a new tree topology $\mathcal{T}_{new}^*$ using the tree search moves. When a new matching $\mathcal{M}_{new}^*$ is proposed, we first ensure that the cluster matching distance for the new matching does not lead to values above the threshold $\mathcal{D}_{\mathcal{E}}^{thr}$. If this condition is satisfied and the objective function is minimized then the new matching is accepted. If the proposed tree topology $\mathcal{T}_{new}^*$ achieves lower value for the objective function, it is accepted. The search procedure terminates when the objective function does not improve or the maximum number of iterations has been reached.

4.9 GO analysis on clusters identified by LinTIMaT

To perform GO (Gene Ontology) analysis on invariant lineage clusters, we first identify a set of differentially expressed (DE) genes based on t-test of 2 groups of cells. The first group consists of the cells in the invariant cluster and the second group includes all other cells in the dataset. From the set of DE genes, we further select the genes that have higher mean expression in the first group, with a p-value smaller than 0.05 (or top 500 if more than 500 genes achieve this p-value). Finally, we use gprofiler [157] to perform GO query for the genes selected for each cluster.

4.10 Analyzing the cell clustering performance of a lineage tree

For assessing the cell clustering performance of a lineage tree, we use 63 cell types obtained by [154] as ground truth and use Adjusted Rand Index (ARI) as the clustering metric following [116]. Basically, ARI is calculated based on the number of agreements and number of disagreements of two groupings, with randomness taken into account. ARI is defined as follows. Let $X = \{X_1, X_2, ..., X_r\}$, $Y = \{Y_1, Y_2, ..., Y_s\}$ be two groupings, where X has r clusters and Y has s clusters. We can set the overlap between X and Y using a table N with size $r * s$, where $N_{ij} = |X_i \bigcap Y_j|$ denotes the number of objects that are common to both X_i and Y_j. Let $a_i = \sum_j N_{ij}$, $b_j = \sum_i N_{ij}$, n be the total number of samples, then ARI is given by

$$ARI = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex} = \frac{\sum_{ij}\binom{N_{ij}}{2} - (\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2})/\binom{n}{2}}{\frac{1}{2}(\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}) - (\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2})/\binom{n}{2}} \tag{4.23}$$

4.11 Processing of the input data

LinTIMaT is designed for single-cell datasets in which both edited barcode and scRNA-seq data are available from the same cell. For scGESTALT datasets, each CRISPR-Cas9 mutation event (edit) has variable length and a single event could span across multiple adjacent sites. To construct a lineage tree from the mutation data we first count the number of unique synthetic markers (Cas9 edits) that occur in the 9 mutation sites. For each cell, the mutated barcode is represented by a binary vector of length equal to the number of unique synthetic markers, where each bit represents the state of a synthetic marker. For example, for ZF1 in the scGESTALT dataset there are 324 entries in this vector for each cell. Similarly, for ScarTrace dataset, we also use a binary vector with length equal to all unique mutations to represent the mutated barcode for each cell, and each bit of the binary vector represents whether or not the cell contains the mutation event in at least one of its target sites. We use the mutation data to construct a paired-event matrix, $\mathcal{E}_{B\times S}$ for B unique barcodes and S unique editing events (synthetic markers), and an imputed gene-expression matrix, $\mathcal{Y}_{N\times G}$ for N cells and G genes.

Each row of the paired-event matrix $\mathcal{E}$, corresponds to a mutated barcode (or allele) and each column corresponds to a unique editing event. An entry e_{bs} of $\mathcal{E}$ is a binary variable that denotes the presence or absence of marker s in barcode b (1 or 0). Each cell c is associated with one, and only one, of the B unique barcodes. As a result, each barcode represents a group of cells. For each cell $c = 1, \ldots, N$, z_c denotes the barcode b profiled for that the cell, $z_c = b$, where $b \in \{1, \ldots, B\}$. Thus, the matrix $\mathcal{E}$ can be transformed to an $N \times S$ matrix for N cells and S markers, where the row c will correspond to the barcode z_c associated with cell c.

The other type of data our method uses is scRNA-seq data. In general, the method can work with any such data. For the specific datasets used in this paper, we observed a high dropout rate (94% entries were 0). To address this issue we tested a number of imputation methods (see Appendix C Supplementary Methods, and Figure 4.26) and selected DrImpute [73] for imputation. DrImpute first clusters the data, and then each zero expression value is imputed with the mean gene expression of the cells in the cluster the cell belongs to. Next, we normalized the expression of each cell and log2-transformed the results (Appendix C Supplementary Methods).

4.12 Results

4.12.1 Testing LinTIMaT using a benchmark *Caenorhabditis elegans* dataset

We first tested whether the underlying assumptions LinTIMaT is based on, namely that gene-expression information can be used to reduce errors in mutation data for lineage reconstruction, actually hold. For this, we used a well resolved lineage from *Caenorhabditis elegans*. A key advantage of this data for testing our method is the fact that its exact lineage has been fully reconstructed [192]. To benchmark LinTIMaT, we combined experimental *C. elegans* scRNA-seq data with simulated CRISPR-Cas9 mutation data. scRNA-seq data was obtained from Tintori *et al.* [203] who profiled the 16 cell embryos of *C. elegans*. Since we know the lineage for these cells (Figure 4.3a), we could use it to simulate CRISPR-Cas9 mutations based on the method proposed by *Salvador-Martínez et al* [168]. Simulated datasets were used to emulate different potential errors introduced as part of a CRISPR-based lineage reconstruction experiment. These include issues related to variability in the mutation rate (μ) for each cell division, site specific variability in mutation rates for different target sites and 'dropouts' of CRISPR mutations which refer to erasing some of the earlier lineage mutations by later ones [153].

We simulate CRISPR mutations based on the 16 cell *C. elegans* lineage using a similar strategy outlined in [168]. For simulation of CRISPR lineage recorders, each cell is represented as a vector of $m = 5$ target sites. The 16 cell lineage corresponds to a series of 4 cell divisions. The nonleaf nodes of the lineage represent the cells that underwent cell division. The root of the lineage represent the fertilized egg for which each CRISPR target is in an unmutated state. The branches that connect a nonleaf node to its children represent the branches where each unmutated target can mutate with a given probability μ denoting the mutation rate. Each target site can mutate to one of several possible mutated states. For each target site, the possible number of mutational events is chosen to be 8 and the different mutational events are considered to be equiprobable. After a mutation occurs at a target, it can no longer change in the absence of dropout. The simulation of CRISPR mutations starts from the root and follows the nonleaf nodes

in the order of the cell division they represent (cell division at level 1 followed by division at level 2 and so on). For simulating CRISPR mutations with varying mutation rate for different target sites, we first decide the value of mean mutation rate and standard deviation. Based on these two, we define a Beta distribution from which the mutation rate for each target is sampled. To introduce mutation dropouts, we first define dropout rate as the ratio of the expected number of dropout events and the number of internal branches in the cell lineage. Dropouts are introduced in the lineage with probability equal to dropout rate and can only affect the target sites that have been already mutated. Whenever dropout happens at a target site, its previous lineage recording gets erased.

The lineage reconstruction performance of LinTIMaT on this *C. elegans* benchmark dataset was compared against that of the Camin-Sokal Maximum Parsimony (MP) method, which was used in the original paper for reconstructing lineage trees from CRISPR mutation data and the neighbor-joining (NJ) method for reconstructing phylogenetic trees [167]. The accuracy of lineage reconstruction was measured based on a metric used in [168] which calculates the fraction of the non-trivial bipartitions in the ground truth lineage tree that are precisely recovered in the inferred lineage tree. In addition, we also computed the Robinson-Foulds (RF) distance [160] between the known lineage tree and the inferred lineage tree for all methods. RF distance calculates the number of non-trivial bipartitions that differ between the inferred and true lineage trees (we normalize this using the total number of bipartitions in the two trees). For the binary lineage trees inferred by LinTIMaT, we computed RF distance (same as FP and FN distance). In contrast, since the lineage trees inferred by MP or NJ can potentially be nonbinary (when a complete lineage barcode is shared by more than 2 cells), we separately computed the FP and FN distances between the true lineage tree and the lineage inferred by MP or NJ.

Figure 4.3b compares LinTIMaT, MP and NJ for varying mutation rates. As can be seen, for all values of mutation rates, LinTIMaT achieved better accuracy in lineage reconstruction compared to that of MP and NJ. For lower mutation rates ($\mu \leq 0.15$), LinTIMaT achieved upto 41.64% improvement in mean lineage reconstruction accuracy over that of MP (41.64% improvement for $\mu = 0.05$, 14.44% improvement for $\mu = 0.1$ and 32.02% improvement for $\mu = 0.15$ respectively) and upto 29.45% improvement over that of NJ (29.45% improvement for $\mu = 0.05$, 15.19% improvement for $\mu = 0.1$ and 21.81% improvement for $\mu = 0.15$ respectively). For these values of mutation rates, LinTIMaT also achieved lower RF distance compared to the FP and FN distances for the trees inferred by MP and NJ (Appendix C Figure 4.7). This indicates that by utilizing the transcriptomic data, LinTIMaT was indeed able to recover some of the branchings of the reference lineage that did not harbor any CRISPR mutations (which were indeed not recovered by MP or NJ). Performance of MP and NJ improved with an increase in mutation rate but even for datasets with higher mutation rates ($\mu \geq 0.2$), LinTIMaT was able to achieve better solution compared to that of MP (12.9%, 9.3% and 11.89% improvement in mean lineage reconstruction accuracy for $\mu = 0.2$, $\mu = 0.25$ and $\mu = 0.3$ respectively) and NJ (16.57%, 9.7% and 9% improvement in mean lineage reconstruction accuracy for $\mu = 0.2$, $\mu = 0.25$ and $\mu = 0.3$ respectively). Next, we simulated datasets for which the mutation rate differed between sites. In such cases, sites with higher mutation rate could saturate early in contrast to sites with lower mutation rate that might not harbor any mutation at all. For such datasets, LinTIMaT achieved higher accuracy ($13.56\% - 30.37\%$ improvement, Figure 4.3c) and lower RF distance compared to that of MP and NJ (Appendix C Figure 4.8). This indicates that

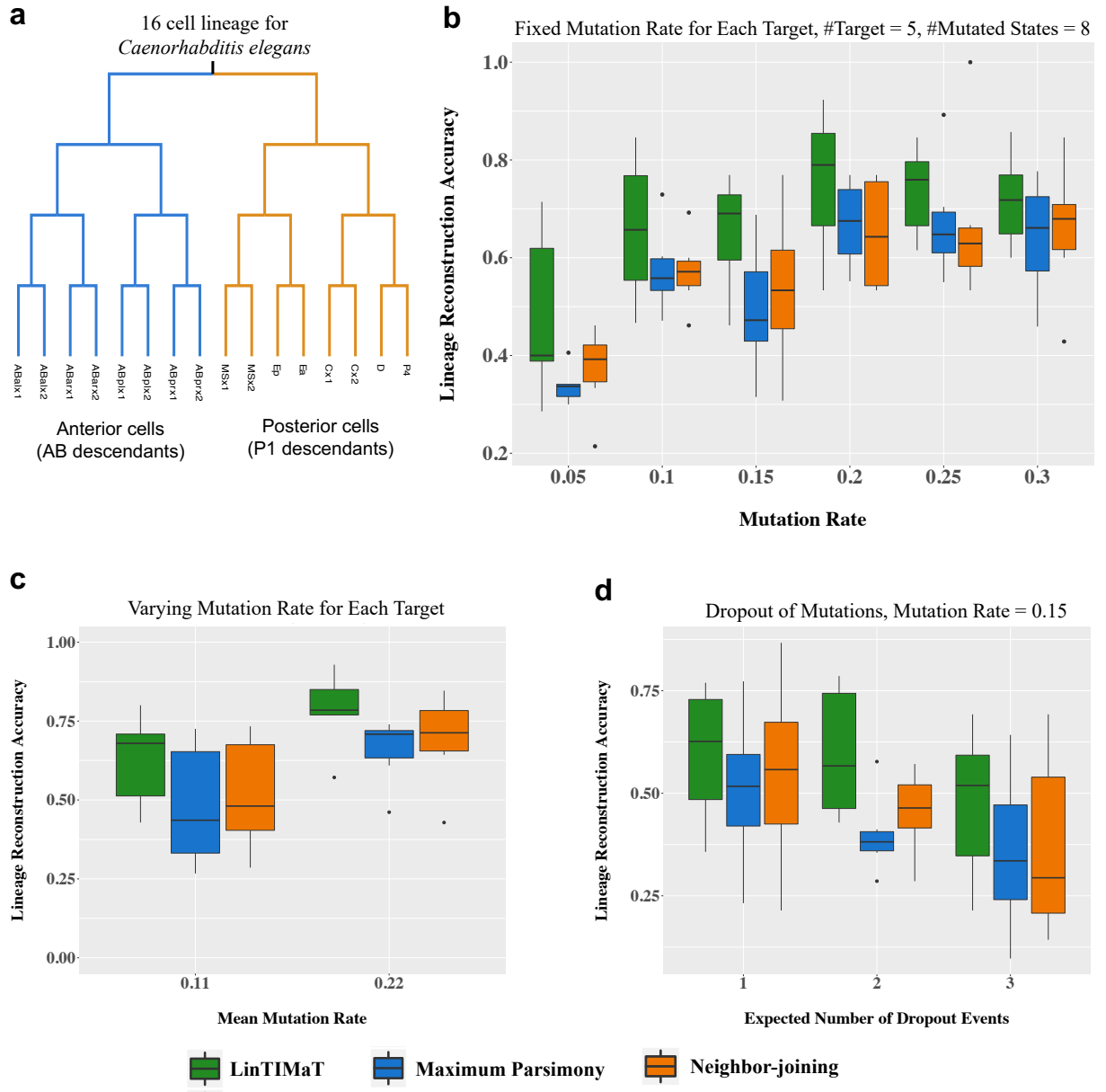

Figure 4.3: Benchmarking on *C. elegans* lineage. (a) 16-cell embryo lineage for *Caenorhabditis elegans*. scRNA-seq data for each leaf (cell) was obtained from [203] and included 6 replicates for each cell. (b) Comparison of LinTIMaT, Camin-Sokal Maximum Parsimony, and Neighbor-joining when varying the mutation rates. The number of possible mutational states was set to 8. Fixed mutation rate was used for each CRISPR target. Each box plot summarizes results for 6 replicates with varying simulated CRISPR mutation data and experimental scRNA-seq data. (c) Comparing lineage reconstruction methods when mutation rate varies between different target sites. Each box plot summarizes results for 6 replicates. (d) Comparison of accuracy of lineage reconstruction by LinTIMaT, Camin-Sokal Maximum Parsimony, and Neighbor-joining in the presence of mutation dropout. Fixed mutation rate, $\mu = 0.15$ was used for all targets. Each box plot summarizes results for 6 replicates.

LinTIMaT's performance is more robust to the increase in complexity in the CRISPR mutational history. CRISPR activity affecting multiple targets simultaneously can result in erasing some of the earlier lineage records [153]. Such 'dropouts' of CRISPR mutations have been shown to have significant impact on the lineage reconstruction accuracy [168]. In order to assess the performance of LinTIMaT in the presence of mutation dropouts, we simulated datasets with different dropout rates for a fixed mutation rate ($\epsilon_d = \{1, 2, 3\}$, $\mu = 0.15$) where ϵ_d denotes the expected number of dropout events in the cell lineage. As expected, lineage reconstruction accuracy of all methods decreased as the number of dropouts increased (Figure 4.3d). However, for all settings, LinTIMaT achieved better accuracy than MP (17.45%, 49.53% and 33.53% better mean accuracy for $\epsilon_d = 1$, $\epsilon_d = 2$ and $\epsilon_d = 3$ respectively) as well as NJ (8.8%, 31.37% and 28.3% better mean accuracy for $\epsilon_d = 1$, $\epsilon_d = 2$ and $\epsilon_d = 3$ respectively) indicating that in the presence of dropouts, LinTIMaT is able to recover more accurate branchings in the cell lineage compared to that of MP and NJ. This is further indicated by LinTIMaT's smaller RF distance for all settings compared to that of MP and NJ (Appendix C Figure 4.9). LinTIMaT was also able to consistently obtain higher accuracy compared to MP and NJ for $\epsilon_d = 2$ and mutation rate varying from $\mu = 0.05$ to $\mu = 0.3$ (Appendix C Figure 4.10). We also use this simulated dataset to test how close we can get to the optimal tree score for mutation likelihood. For this, we first calculate the optimal mutation likelihood (average -34.5) by using the known lineage tree of *C. elegans*. Then, we calculated the likelihood improvement obtained by LinTIMaT by calculating the difference between initial likelihood (average -325057.9) and final likelihood after optimization (average -49.2) and compared that to the best possible likelihood improvement (from -34.5 to minus -325057.9). This analysis allowed us to determine that the search performed by LinTIMat can improve mutation likelihood by roughly 99.98%, only 0.02% less than the best possible value (Appendix C Table 4.1).

4.12.2 LinTIMaT can recover convergent and divergent lineage relationships in the cell lineage

During development, cells belonging to some mature cell type can have distinct developmental trajectory (represents divergent lineage of same cell type) while cells from diverging cell types can display similar trajectories, at least up to a point. To assess LinTIMaT's ability to handle these scenarios, we combined zebrafish scRNA-seq data from scGESTALT [154] with synthetic CRISPR-Cas9 mutation data from simulated lineage of 100 cells containing divergent and convergent lineage relationships.

First, we evaluated LinTIMaT's ability to correctly infer the lineage relationship between two groups of cells that are transcriptionally very similar (same cell type) but diverged early. We selected forebrain neuron cells and divided into two groups (G1 and G2 containing 11 and 10 cells respectively). We simulated lineages on 100 cells placing these two groups of cells in two different subtrees. The other 79 cells were chosen from different neuron types (forebrain, midbrain and hindbrain), progenitor, blood and mixed cell types. We simulated lineages under a number of different settings for when the divergent occurs (Appendix C Figure 4.11a Appendix C Figure 4.12). For each we simulated CRISPR mutations with several different dropout settings. Results show that for all experimental settings LinTIMaT's lineage reconstruction error was

lower when compared to the average lineage reconstruction error resulting from placing the two groupsin the same subtree (Appendix C Figure 4.11b and 4.12b). Specifically, for all but 1 of the 10 settings, LinTIMaT was able to correctly place the two groups in different subtrees (Appendix C Figure 4.11c and 4.12c). These experiments illustrate LinTIMaT's ability to recover divergent lineage relationships.

Next, we assessed LinTIMaT's ability to correctly infer the convergent lineage relationship between two groups of cells that are transcriptionally distinct (different cell types) but share a common ancestry. Details of this analysis are provided in Appendix C Supporting Results. Again, we observed that LinTIMaT achieved lower lineage reconstruction when compared to the error resulting from placing the two groups in different subtrees (Appendix C Figure 4.13b) and in all cases correctly placed the two in the same subtree (Appendix C Figure 4.13c).

4.12.3 Integration of mutation and transcriptomic data improves the reconstruction of cell lineage trees

Next, we applied LinTIMaT on two experimental zebrafish datasets [4, 154], each using a different technology for inserting CRISPR-Cas9 mutations. The first dataset was generated using scGESTALT [154]. The second dataset was generated using ScarTrace [4].

For the scGESTALT dataset, we applied LinTIMaT on two zebrafish samples, ZF1 and ZF3, consisting of 750 and 376 cells respectively, from which both the transcriptome (20287 genes) and edited barcode (192 unique barcodes, 324 unique markers for ZF1 and 150 unique barcodes, 265 unique markers for ZF3) were recovered. For both fishes, our analysis shows that improving the likelihood function used by LinTIMaT increases the coherence of the resulting cell types for each subtree, without impacting the overall mutation likelihood (Figure 4.4a and Appendix C Figure 4.14) . For both fishes, LinTIMaT generated highly branched multiclade lineage trees (Figure 4.4b and Appendix C Figure 4.15). Blue nodes on the tree represent mutation events assigned while red nodes represent the clusters identified based on gene-expression data. It is important to note that cluster nodes do not necessarily represent common ancestors for the cells underneath, instead, cluster nodes are just a way of grouping nearby cells together based on expression information without affecting the mutational ancestor-descendant relationships. ZF1 lineage tree comprised 25 major clades (level 1 tree nodes) and 113 cluster nodes, 77 of which consisted of more than one cell. ZF3 lineage tree comprised 17 major clades and 42 cluster nodes, 33 of which consisted of more than one cell. We compared the lineage trees reconstructed by LinTIMaT to the trees reconstructed using maximum parsimony (MP) as used in the original study [154] by comparing the accuracy of cell clusters in the trees. In the original study, 63 transcriptionally distinct cell types were identified using an unsupervised, modularity-based clustering approach from 6 zebrafish samples. We used this clustering to compute the Adjusted Rand Index (ARI) for the cell clustering obtained from a lineage tree (Methods). For MP lineage trees, the unique barcodes represent cell clusters as mutation information was the only basis for reconstructing the tree. For each fish, the lineage tree reconstructed by LinTIMaT resulted in better cell clustering (37.5% and 36.4% improvement in ARI for ZF1 and ZF3 respectively) compared to MP results based on mutation data alone (see Appendix C Table 4.2 and Supplementary Results for details).

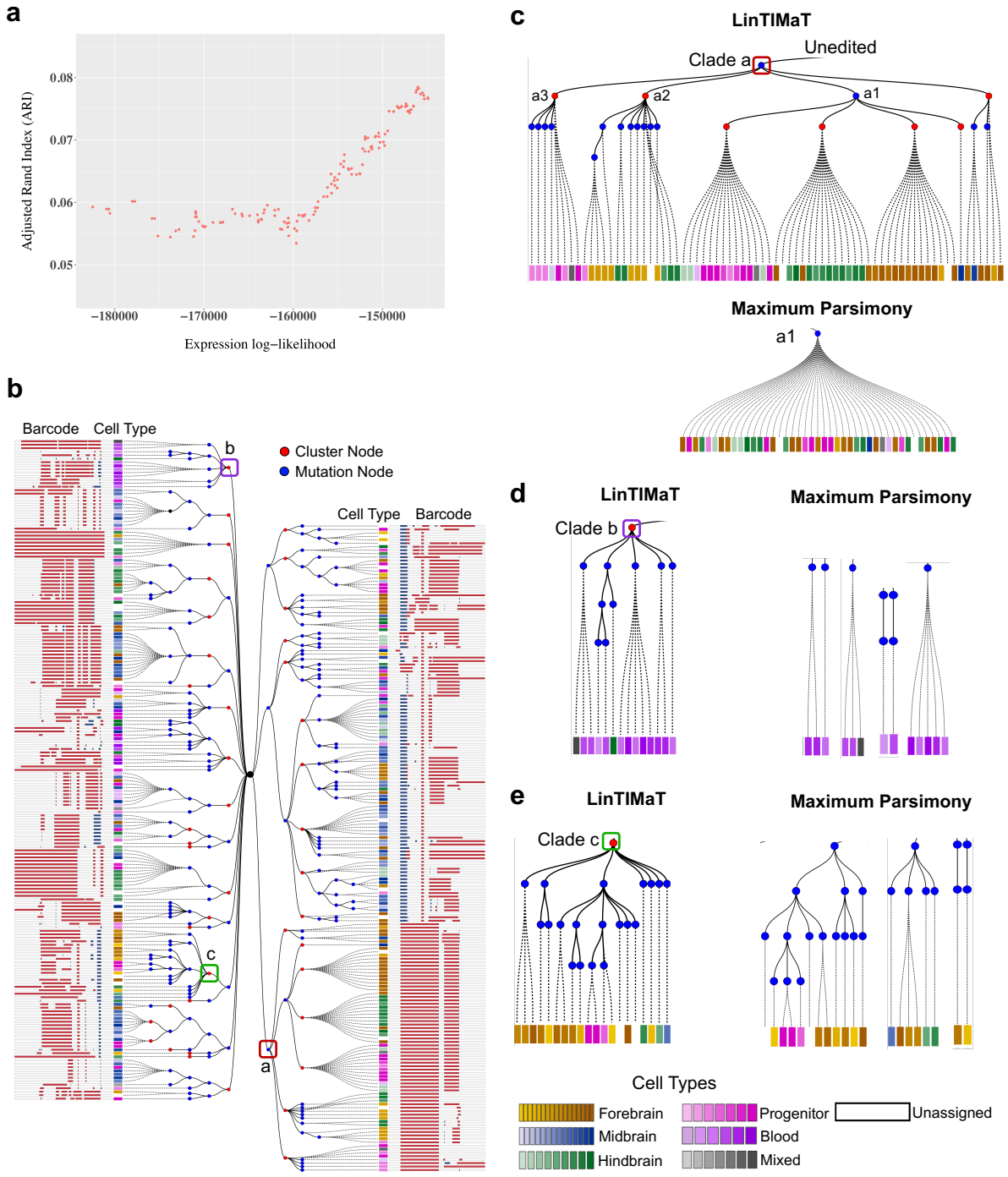

Figure 4.4: Reconstructed cell lineage for a single juvenile zebrafish brain (ZF3) from scGESTALT dataset. (a) Adjusted Rand Index (ARI) which measures the agreement between cell types in the tree clusters and cell types assigned by the original paper [154] as a function of the likelihood computed by LinTIMaT. The fact that as the likelihood increases the ARI increases as well indicates that the target function of LinTIMaT is capturing biologically relevant relationships between cells. (b) Reconstructed cell lineage tree for ZF3 built on 376 cells. Blue nodes represent Cas9-editing events (mutations) and red nodes represent clusters inferred from transcriptomic data. Each leaf node is a cell, represented by a square, and its color represents its assigned cell type as indicated in the legend. The mutated barcode for each cell is displayed as a white bar with insertions (blue) and deletions (red). (c) By using transcriptomics data LinTIMaT is able to further refine subtrees in which all cells share the same barcode which can help overcome saturation issues. (d-e) Example subtrees displaying LinTIMaT's ability to cluster cells with different barcodes together based on their cell types. In contrast, maximum parsimony puts these on distinct branches.

Lineage trees reconstructed using LinTIMaT showed successful integration of mutation and expression data. When using only mutation data, in several cases, cells belonging to very different cell types were clustered together. In contrast, in the trees reconstructed by LinTIMaT, these cells were correctly assigned to different subtrees corresponding to different cell types. Clade a1 in ZF3 lineage tree (Figure 4.4c) is one such example. In MP lineage tree for ZF3, neural progenitor cells, hindbrain granule cells, and neurons in ventral forebrain and hypothalamus (total 43 cells) were clustered together under clade a1 as they shared the same mutational barcode. The tree reconstructed by LinTIMaT corretly separated these cells into three major subtrees (progenitor, hindbrain, and forebrain) under the same mutational node. Similarly for ZF1, in the original MP lineage tree, clade a consisted of 198 cells including mostly forebrain and progenitor cells. LinTIMaT lineage tree successfully divided them into multiple subtrees, with the largest mainly containing forebrain neuron cells and the other subtrees mostly containing different types of progenitor cells (Appendix C Figure 4.16a). In addition, LinTIMaT trees also contain examples where cells belonging to similar cell types but carrying different mutational barcodes are identified as a cluster instead of being placed on distant branches as done by MP. Clades b and c in ZF3 lineage tree (Figure 4.4d-e) illustrate this scenario. In the LinTIMaT lineage tree for ZF3, clade b consists of mostly blood cells that carry different mutational barcodes. In MP lineage tree, these cells were placed in 4 distant branches which did not convey the fact that they belong to the same cell type. However, LinTIMaT successfully grouped them together in a cluster of blood cells while preserving their mutational differences as illustrated by the mutation nodes being descendants of the cluster node. Similarly, for clade c most of the cells were forebrain neurons that were placed in three distinct branches in the MP lineage tree owing to their mutational differences. LinTIMaT successfully identified these cells as a cluster consisting of mostly forebrain neuron cells. Similar examples can be seen in the tree reconstructed by LinTIMaT for ZF1 (Appendix C Figure 4.16b). We note that while the LinTIMaT reconstructed lineage trees displayed much better agreement with cell type coherence, this was not just a function of ignoring mutational data. In fact, the trees inferred by LinTIMaT had *higher* likelihoods based on mutation alone (Appendix C Table 4.3) when compared to the trees reconstructed by MP [154]. In fact, for each fish, the MP lineage tree violated the Camin-Sokal parsimony criterion for some mutations that resulted in a low mutation log-likelihood.

Following the analysis of [154], we also analyzed the trees for spatial enrichment of clusters. For this, groups of four or more cells were selected for both LinTIMaT and MP lineage trees. In both types of lineage trees, clusters were spatially enriched in hindbrain, forebrain and midbrain (Appendix C Figure 4.17). However, the trees reconstructed by LinTIMaT displayed better spatial enrichment. For example, for ZF3, more clusters in LinTIMaT lineage tree were enriched in forebrain and hindbrain compared to the barcode clusters in MP tree. Similarly, for ZF1, LinTIMaT lineage showed more enriched hindbrain clusters compared to the barcode clusters in MP tree. We also compared the lineages by assessing the functional significance of the clusters through Gene Ontology (GO) analysis. We observed that the clusters identified by LinTIMaT led to more significant enrichment of more GO functions compared to the barcode clusters in MP tree (Appendix C Figure 4.18 and Appendix C Table 4.8).

LinTIMaT lineage trees also revealed divergent lineage trajectories (Figure 4.4b). For example, for ZF3, LinTIMaT lineage tree displayed three major subtrees (Figure 4.4c) under clade a (a1, a2 and a3 respectively), with a1 being sub-divided into three major clusters. Clade a1

had three major clusters consisting mostly of progenitor cells, hindbrain and forebrain neurons respectively. The constructed tree indicates that the $her4.1^+$ and $atoh1c^+$ progenitor cells [52, 228] are closely related to $pax6b^+$ granule cells [198] in hindbrain, $gad2^+$ neurons in ventral forebrain [133], and $fezf1^+$ neurons [136] in hypothalamus region. On the other hand, $pitx2^+$ and $prdx1^+$ neurons [6] in forebrain (clade a2) were determined to be related to radial glia cells (clade a3). These results demonstrate LinTIMaT's ability to elucidate complex lineage relationships of cells.

To assess the ability of LinTIMaT to generalize to other types of CRISPR mutation data we further applied it on data generated by the ScarTrace [4] method. Similar to scGESTALT, ScarTrace also uses CRISPR-Cas9 technology for introducing heritable mutations, though it uses a different lineage recording system where genomic sites located on in-tandem copies of a transgene are targeted for inserting mutations (also called scars). For this dataset, we applied LinTIMaT on two zebrafish samples, R2 and R3, for which the cells were sampled from adult brain and eyes. For each of these fishes, we selected 750 cells based on their cell types. The mutational dataset for R2 consisted of 133 unique barcodes and 78 unique scars, whereas that for R3 consisted of 85 unique barcodes and 50 unique scars. Applying LinTIMaT to this data resulted in highly branched multiclade lineage trees (can be visualized at `https://jessica1338.github.io/LinTIMaT/`). For comparison, we also reconstructed lineage trees using MP for the two fishes. Similar to what we observed for the scGESTALT data, LinTIMaT was able to correctly separate different types of cells that were clustered together by MP. On the other hand, LinTIMaT was also able to cluster cells that belonged to similar cell types but carried different mutational barcodes. We present a number of examples for these results in Appendix C Figures 4.19, 4.20 and 4.21. In addition, the lineage trees reconstructed by LinTIMaT for the ScarTrace datasets had *higher* likelihoods based on mutation alone (Appendix C Table 4.4) when compared to the trees reconstructed by MP. The much lower likelihoods of the MP trees were caused by the violation of Camin-Sokal parsimony criterion by multiple mutations. See Appendix C Supplementary Results for more details.

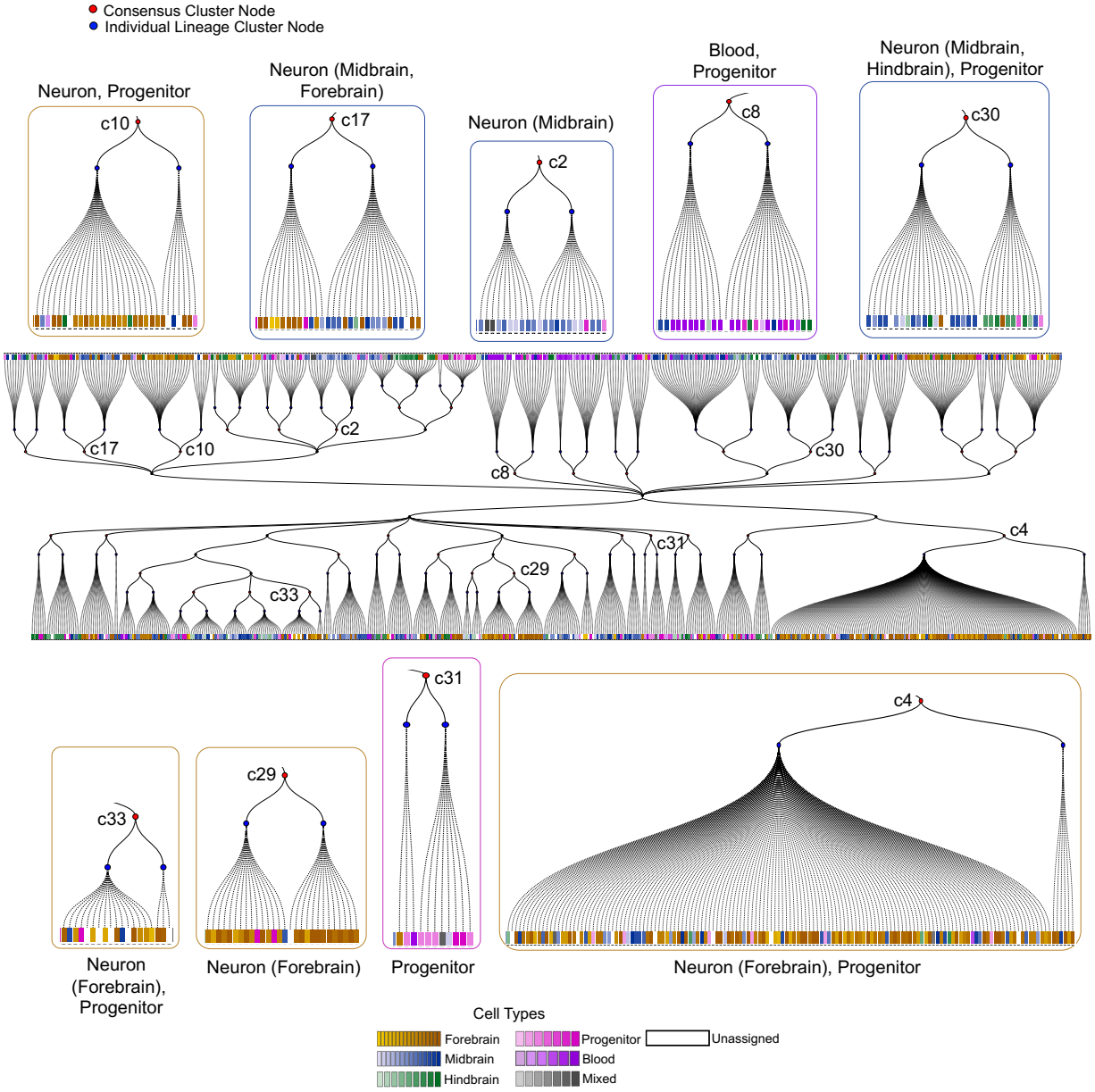

Figure 4.5: Invariant lineage tree for juvenile zebrafish brain for scGESTALT dataset. The two-sided tree in the middle represents the invariant lineage tree generated by LinTIMaT by combining the individual trees for ZF1 and ZF3. Blue nodes here represent the clusters from individual fishes (left node: ZF1, right node: ZF3), and red nodes represent the matched invariant clusters. Each leaf node is a cell, represented by a square, and its color represents its cell type as indicated in the legend. Subtrees illustrate examples of invariant clusters preserved in the individual lineage trees.

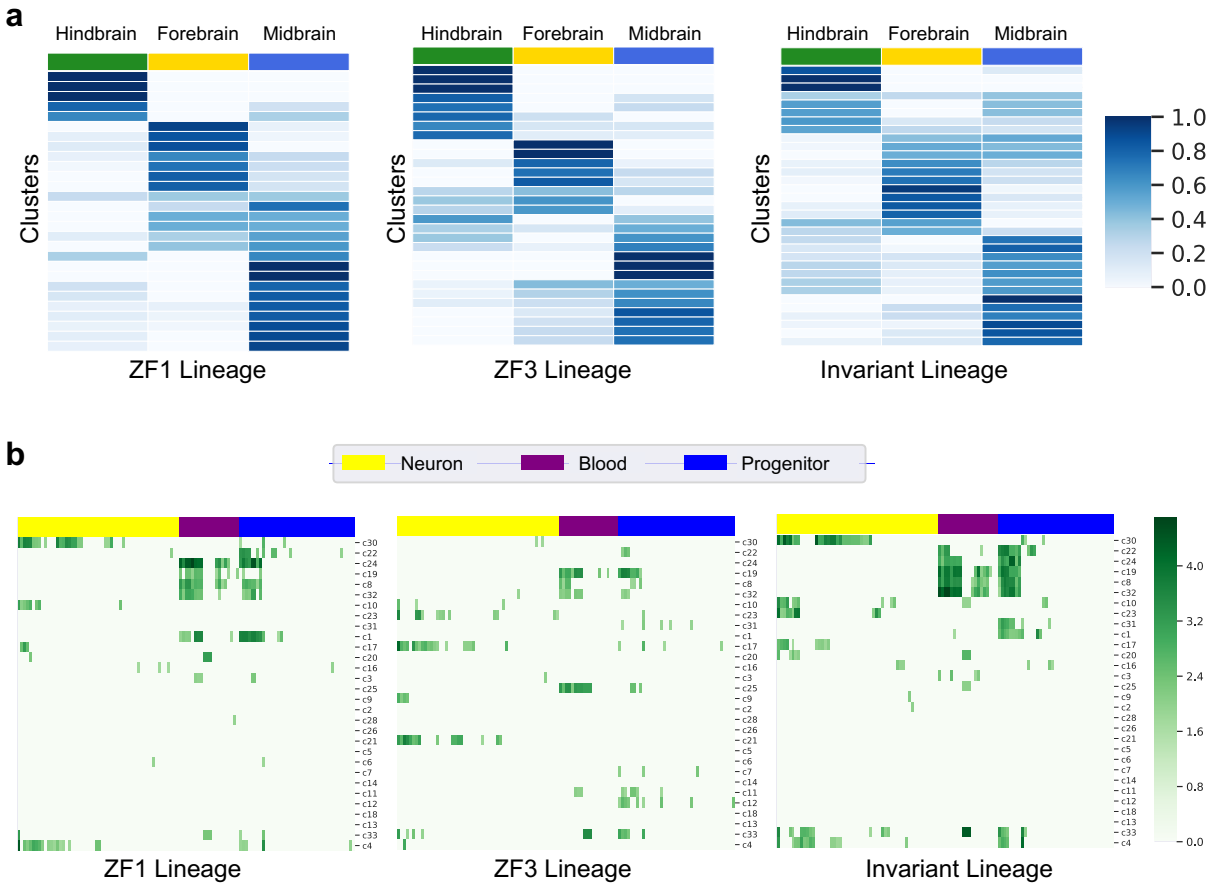

Figure 4.6: Functional analysis of cell clusters for scGESTALT datasets. (a) Heat map of the distribution of cell clusters for each region of the brain (columns). Cell types were classified as belonging to the forebrain, midbrain or hindbrain, and the proportions of cells within each region were calculated for each cluster. Each row sums to 1. Region proportions were colored as shown in key. The leftmost panel shows the heat map for the clusters in ZF1 lineage (subsampled), middle panel shows the heat map for ZF3 lineage and the rightmost panel shows the heat map for the invariant lineage. (b) Heat map of the p-values ($\sqrt{-\log(pvalue)}$, higher value means more significant) for GO terms for invariant clusters. Rows represent invariant clusters and columns represent different GO terms (Appendix C Table 4.9). Yellow, purple and blue columns correspond to GO terms related to neurons, blood and progenitors respectively. The leftmost panel shows the heat map for ZF1, middle panel for ZF3 and the rightmost panel for the invariant tree. As can be seen, the invariant tree correctly combines the unique terms identified for each tree. On one hand, it is able to identify neuron clusters, which are well represented in ZF3 but not in ZF1. On the other hand, it is able to identify progenitor clusters which are not well represented in ZF3.

4.12.4 invariant lineage tree successfully combines data from individual lineages

As mentioned above, combining CRISPR-Cas9-mutation-based individual lineage trees is challenging since mutations are random and so differ for the same cell types between experiments. To address this, we used LinTIMaT to combine data from the replicates generated by

scGESTALT and ScarTrace to infer invariant lineages for the development of juvenile zebrafish brain and the development of zebrafish brain and eyes respectively.

For the scGESTALT dataset, LinTIMaT inferred 113 clusters for ZF1 and 42 clusters for ZF3, out of which 33 clusters were found to be preserved in both lineages. Using these, LinTIMaT inferred an invariant lineage tree (Figure 4.5) with 33 leaves each of which represents a matched pair of clusters from the individual fishes. We first evaluated the invariant lineage by computing its Adjusted Rand Index (ARI) based on the 63 cell types obtained by [154]. Our analysis showed that, despite the individual fishes having different spatial distribution of cells (for example, ZF1 had more forebrain cells and ZF3 had more hindbrain cells), the ARI for the invariant lineage (0.079) was comparable to the individual LinTIMaT lineages (0.084 and 0.076 for ZF1 and ZF3 respectively) and higher than both individual MP lineages (0.061 and 0.056 for ZF1 and ZF3 respectively). While the invariant lineage preserved some of the ancestor-descendant relationship of the individual lineages (Appendix C Figure 4.22), it also placed similar cell clusters from different branches of the individual trees under the same subtree (Appendix C Figure 4.23). Thus, in addition to enabling the integration of data across experiments, by using more data, the invariant lineage tree method also improved the placement of the matched clusters on the individual trees themselves.

We further analyzed the matched invariant clusters for spatial enrichment. The clusters in the invariant lineage were enriched in all three regions of brain (hindbrain, forebrain and midbrain) as shown in Figure 4.6a. The invariant lineage showed more enriched hindbrain clusters compared to that of ZF1 and more enriched forebrain clusters compared to that of ZF3.

To determine the biological significance of the clusters identified by the invariant lineage, we performed Gene Ontology (GO) analysis (Methods) on matched clusters that contained more than 10 cells. We also filtered the matched clusters where the individual cluster contained less than 3 cells. We selected all GO terms related to the three major cell types (neuron, blood and progenitor) present in the data (see Appendix C Table 4.5 and Appendix C Table 4.9 for the keywords and list of GO terms). Figure 4.6b illustrates the enrichment of the GO terms in the clusters in terms of p-values. The invariant clusters show coherent enrichment of GO terms for all three major cell types. For example, clusters c23 (forebrain), c17 (midbrain and forebrain) and c2 (midbrain) had high p-value for the GO terms related to neuron but very low p-value for GO terms related to blood and progenitor. Clusters c4 and c10 consisting mostly of forebrain neurons and some progenitor cells showed enrichment of mostly neuron related GO terms and some progenitor GO terms. Similar GO enrichment was observed for cluster c30 that mostly consisted of midbrain and hindbrain neurons and some progenitor cells. The cluster c31 consisting of mostly progenitor cells displayed more enrichment of the progenitor GO terms. Clusters c8, and c32 that consisted mostly of blood and progenitor cells showed enrichment of GO terms related to these two cell types. The invariant clusters also uncovered additional GO functions that were not enriched in individual tree clusters (Appendix C Table 4.11). The coherence of enrichment can also be observed in the proportion of the GO terms related to the three major cell types (Appendix C Figure 4.24). Clusters in the individual lineage trees also showed enrichment of the three cell types. However, the invariant lineage clusters uncovered more GO terms with more significant p-values compared to the individual lineage clusters.

We further reconstructed an invariant lineage for the ScarTrace dataset. For this data LinTIMaT inferred 83 clusters for R2 and 90 clusters for R3; and method identified 52 matched clus-

ters which were used to reconstruct the invariant tree (visualized at `https://jessica1338.github.io/LinTIMaT/`) each of which represents a matched pair of clusters from the individual fishes. We next performed GO analysis to determine if the invariant clusters inferred by LinTIMaT indeed uncover functions coherent with the types of cells. For this, we selected all GO terms related to the three major cell types in the data (neuron, immune and eye, Appendix C Tables 4.6, 4.10). Appendix C Figure 4.25 displays the enrichment of the GO terms in the clusters in terms of p-values. As can be seen, the invariant clusters showed better enrichment of GO terms for all three major cell types. For example, clusters c7, and c21 showed enrichment of GO terms related to neurons. Clusters c43, c19 and c9 showed enrichment of GO terms related to immune cell types. Clusters c47, c11 and c52 showed enrichment of GO terms related to eye cell types. The invariant clusters also uncovered additional GO terms that were not identified as significant when using the individual tree clusters (Appendix C Table 4.12). For example, two invariant clusters (c33 and c44) were found to be associated with erythrocyte and myeloid cell development (corrected p-value ≤ 0.027). Two other clusters (c7 and c31) were found to be associated with positive regulation of synaptic transmission and photoreceptor cell outer segment organization (p-values ≤ 0.004 and ≤ 0.0028, respectively). In both cases, cells related to these categories were not identified in the individual fish trees.

4.13 Discussion

Recent studies [4, 154, 187] combine two complementary technologies, CRISPR-Cas9 genome editing and scRNA-seq for elucidating developmental lineages at whole organism level. These experimental techniques rely on introducing random heritable mutations during cell division using CRISPR-Cas9 and lineage trees are reconstructed based on these mutations using traditional phylogenetic algorithms [63] on profiled cells.

While this exciting new direction to address a decades old problem in-vivo has already led to several interesting insights into organ development in multicellular organisms, it suffers from a number of challenges that make it difficult to accurately reconstruct lineages and to combine trees reconstructed from repeat experiments. First, the tree reconstruction is performed solely based on recovered mutation data, which might be noisy. In addition, the space for the mutations is limited resulting in saturation restricting the ability to further subdivide cells at later stages. Finally, due to the random nature of these mutations, it is impossible to utilize them to reconstruct a invariant lineage tree by combining data from repeated experiments of the same species, in contrast to most phylogenetic studies [26]. No computational method has been developed to address these challenges.

To address these issues, we developed a new statistical method, LinTIMaT, which directly incorporates expression data along with mutation information for reconstructing both, individual and invariant lineage trees. Our method defines a global likelihood function that combines both mutation agreement and expression coherence.

We first used data from *C. elegans* for which ground truth is known to validate the underlying assumption of our method: that expression coherence can indeed help in overcoming mutation data noise. As we show, for several possible noise factors that can appear in CRISPR-Cas9 lineage experiments, LinTIMaT was able to successfully improve the reconstruction of the lineage

tree by using the additional expression information. We next used LinTIMaT on more complex data. While the ground truth for these lineages is unknown, we have shown that the trees reconstructed by LinTIMaT are as good as the best mutation-only lineage trees while they greatly improve over mutation-only lineages in terms of expression coherence, clade homogeneity and functional annotations. In addition, by employing agreement based on expression data, we could further reconstruct a invariant lineage that retains most of the original tree branching for each individual while improving on the individual lineages by uncovering more biologically significant GO annotations corresponding to different major cell types.

Our analysis shows that gene expression data can be very useful for selecting between several lineages with equivalent explanation of the mutation data. Since traditional phylogenetic maximum parsimony algorithms [63] as used in current studies [154] end up selecting a solution that is only slightly better or equivalent compared to several competing ones (though can be very different), the ability to use additional information (in our case gene expression) to select between these equally likely lineage trees is a major advantage of LinTIMaT. LinTIMaT's Bayesian hierarchical model for gene expression data also provides a statistical method for inferring cell clusters with coherent cell types from the lineage tree. While it is not clear yet if all organisms follow the same detailed developmental plan as *C. elegans* [193], the ability to combine lineage trees studied in multiple individuals of the same species can lead to more general trees that capture the major branching events for the species. In addition, invariant trees can be used to improve branchings in the individual trees by combining information from multiple experiments. To the best of our knowledge, LinTIMaT's solution, which is based on iteratively matching cell clusters based on their expression, is the first to enable the reconstruction of such invariant lineage trees from experiments that simultaneously profile lineage recordings and single-cell transcriptomes.

While LinTIMaT worked well on the datasets it was tested on there are still several potential problems with our approach. It is currently unclear if cell trajectories inferred by transcriptional state and lineage should be concordant in all cases. As we showed, LinTIMaT can correctly identify linage relationships even if such differences exist, but it is still possible that in some cases the use of expression data may lead to less accurate reconstructions. Another potential problem arises from our selection of clusters for reconstructing the linage invariant tree. Since we only use clusters observed in all individual trees, the method may leave out several key clusters (or lineages) if their expression levels are not well conserved between different organisms from the same species.

The application of LinTIMaT to zebrafish brain development illustrates its potential in delineating lineage relationships in complex tissues. The method is general and, as we showed, can work with data for several different related technologies. While the joint profiling of lineage recordings and single-cell transcriptomes by experimental methods such as scGESTALT laid the foundation for generating data suitable for identifying cellular relationships during development and disease, LinTIMaT provides the seminal computational approach for utilizing such data for accurate lineage reconstruction. As the usage of the experimental methods expands from zebrafish to other model organisms and human organoid samples [100], LinTIMaT would serve as a powerful component in the biologists' toolbox in reconstructing more accurate and detailed lineages for investigating normal as well as pathological development.

4.14 Appendix C: Supplement to LinTIMaTF: Single-cell Lineage Tracing by Integrating CRISPR-Cas9 Mutations with Transcriptomic Data

4.14.1 Supplementary Methods for LinTIMaTF

Imputation of gene expression data

The scRNA-seq data in both scGESTALT and ScarTrace datasets displayed a high dropout rate (about 94% entries were 0). To address this, we decided to impute the scRNA-seq data. To ensure that LinTIMaT's likelihood function is not sensitive to the imputation method, we tested two different imputation methods named DrImpute [73] and SAVER [94]. We observed that LinTIMaT's results were not dependent on the imputation method used and the cell clustering performance from the same lineage tree were similar for the different imputation methods (Figure 4.26). We finally selected DrImpute as it has been shown to perform better in other experiments [240].

Normalization of scRNA-seq data

After imputing the scRNA-seq data using DrImpute, we normalized the data. For normalization, we summed the expression values for each cell and multiplied the expression value with a scaling factor 10000, which is a common scaling factor for UMI (unique molecular identifier) data.

Prior distribution for computing expression likelihood

LinTIMaT's expression likelihood function computes the probability of the expression data under a node based on two alternative hypotheses. The first hypothesis computes the marginal probability of the data being generated from a single cluster. For computing this marginal probability using equation 4.7, we choose a univariate Gaussian distribution and the Normal-inverse-chi-squared (NIX) prior for β. NIX prior is the univariate version of normal-inverse-Wishart (NIW) prior, which is the prior suggested by BHC. We adopt the univariate version to reduce time and space complexity of LinTIMaT. NIX prior has the following parameters $\beta = (\mu_0, \kappa_0, \sigma_0^2, \nu_0)$, where μ_0 and σ_0^2 are the priors on the mean and variance of the Gaussian distribution. κ_0 and ν_0 are the confidence on the prior of the mean and variance respectively. The posterior parameters $\{\mu_v, \sigma_v^2, \kappa_v, \nu_v\}$ and the marginal probability for the subtree $\mathcal{T}^v$ rooted at node v under the NIX

prior are derived according to [135] and shown below

$$\kappa_v = \kappa_0 + n_v \tag{4.24}$$

$$\nu_v = \nu_0 + n_v \tag{4.25}$$

$$\mu_v = \frac{\kappa_0 \mu_0 + n_v \bar{\mathbf{y}}_v}{\kappa_v} \tag{4.26}$$

$$\sigma_v^2 = \frac{1}{\nu_v}\left(\nu_0\sigma_0^2 + \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_g^v} (\mathbf{y}^{(i)} - \bar{\mathbf{y}}_v)^2 + \frac{n_v \kappa_0}{\kappa_v}(\mu_0 - \bar{\mathbf{y}}_v)^2\right), \tag{4.27}$$

where $\bar{\mathbf{y}}_v$ is the sample mean of $\mathcal{Y}_g^v$, the gene expression values for gene g under the node v. The marginal likelihood is given by

$$P(\mathcal{Y}_g^v | \mathcal{H}_1^v) = \frac{\Gamma(\nu_v/2)}{\Gamma(\nu_0/2)} \sqrt{\frac{\kappa_0}{\kappa_v}} \frac{(\nu_0\sigma_0^2)^{\nu_0/2}}{(\nu_v\sigma_k^2)^{\nu_v/2}} \frac{1}{\pi^{n_v/2}} \tag{4.28}$$

For the hyperparameters of β, we set μ_0 and σ_0^2 to sample mean and sample variance based on all single cells, $\mu_0 = \frac{1}{N}\sum_{c=1}^{N} \mathcal{Y}_{cg}, \sigma_0^2 = \sum_{c=1}^{N} (\mathcal{Y}_{cg} - \mu_0)^2$. The confidence parameters are set to 1, $\kappa_0 = \nu_0 = 1$.

Proposal for cluster matching

We use a two-step heuristic search algorithm for inferring the consensus lineage and the corresponding cluster matching. We employ two different proposals for proposing a new cluster matching for the two steps of the consensus lineage search algorithm respectively.

1. Let us assume $\mathcal{M}_{old}$ denotes the current matching from which we want to propose a new cluster matching. In $\mathcal{M}_{old}$, K clusters (leaves) in the consensus lineage are matched to K clusters in each individual lineage. Let $\{c_1, \ldots, c_K\}$ denotes the K clusters in the consensus lineage. In $\mathcal{M}_{old}$, for individual lineage $\mathcal{T}_i^c$ ($i \in \{1, \ldots, I\}$), let $l_i^{c_x}$ and $l_i^{c_y}$ denote the clusters that match with clusters c_x and c_y in the consensus respectively. We can propose a new cluster matching $\mathcal{M}_{new}^i$ by swapping the matchings of $l_i^{c_x}$ and $l_i^{c_y}$ with c_x and c_y. There are $\mathcal{O}(K^2)$ such possible swaps. For each such swap η, we compute a score $\Sigma_\eta = \Delta_{\mathcal{S}}^\eta + \Delta_{\mathcal{E}}^\eta$, where $\Delta_{\mathcal{S}}^\eta$ denotes the improvement in $\mathcal{D}_{\mathcal{S}}$ after the swap and $\Delta_{\mathcal{E}}^\eta$ denotes the improvement in $\mathcal{D}_{\mathcal{E}}$ after the swap. The swaps for which both $\Delta_{\mathcal{E}}^\eta$ and Σ_η are positive are considered to be good swaps. One such good swap is chosen randomly to propose a new matching $\mathcal{M}_{new}^i$. If no good swap is available, no swapping is performed. This is done sequentially for all I individual lineages to produce a new cluster matching $\mathcal{M}_{new}$.
2. In the second step of the search algorithm, we perform a random swap to propose a new matching. For consensus lineage, we randomly choose two clusters c_x and c_y, and in the individual lineage $\mathcal{T}_i^c$, we swap their matchings with $l_i^{c_x}$ and $l_i^{c_y}$.

Visualizing LinTIMaT trees

Following [154], individual cells (leaves) in the lineage trees were annotated by their corresponding cell types. LinTIMaT lineage trees were converted into JSON objects using custom

python scripts and annotated with cell type membership. Finally, the JSON objects were visualized using custom scripts of [154] using D3 software framework. The visualization web page also displays additional information on each tree node such as mutations and cell type proportions.

4.14.2 Supplementary Results for LinTIMaTF

Applying LinTIMaT on ScarTrace zebrafish dataset

ScarTrace [4] is a lineage tracing experimental method that combined CRISPR-Cas9 lineage recording with SORT-seq transcriptome profiling for detecting mutational scars and scRNA-seq from single cells. In the original study, ScarTrace was applied on different organs of zebrafish (adult brain, eyes, caudal fin, etc.) for different replicates. We selected 2 zebrafish replicates (R2 and R3) for which the adult brain and eyes were profiled. After preprocessing (selecting the cells that have both mutation and expression data), R2 and R3 consisted of 1320 and 749 cells respectively with around 16K genes and 60-80 unique mutational scars. To match the cell type distribution of the cells in both replicates, we further selected 750 cells (out of 1320) for R2 for applying LinTIMaT. The large lineage trees inferred by LinTIMaT for R2 and R3 can be visualized at `https://jessica1338.github.io/LinTIMaT/`. For comparison, we also reconstructed lineage trees using MP for these two fishes. LinTIMaT was able to separate cells based on their cell types that were all clustered together by MP due to their shared mutational barcode. Clade a in R2 lineage tree (Figure 4.19) is one such example, where LinTIMaT was able to separate left midbrain neurons, rod cells and immune cells into two subtrees in contrast to MP that clustered all these cells together. Similar examples (Figure 4.20) can also be seen in R3 lineage tree, where in MP lineage, right eye neurons, immune cells and cells with unknown cell types were clustered together but LinTIMaT successfully assigned them into different subtrees under the same mutational node. In addition, LinTIMaT lineage for R3 also displayed example (Figure 4.21) where cells belonging to similar cell types but carrying different mutational barcodes were identified as a cluster. In clade c, LinTIMaT identified right midbrain neurons as a cluster that were otherwise in different branches and mixed with neurons from left midbrain and immune cells.

Computing ARI for cell clustering based on Maximum Parsimony lineage trees for scGESTALT dataset

To compare LinTIMaT lineages against Maximum Parsimony (MP) lineages from [154] for ZF1 and ZF3, we compared the cell clustering performance for the lineage trees. The cell clustering performance was measured by computing ARI. While LinTIMaT trees allow for inferring cell clusters based on gene expression data, MP lineage trees do not provide such option. For MP lineage trees, the unique barcodes can be treated as cell clusters. However, to be more thorough, we also cut the MP trees at different levels to obtain different possible cell clusterings. The ARI values for the clusterings obtained by cutting the MP trees at level 1-6 and the barcode level for both ZF1 and ZF3 are shown in Table 4.2. For both fishes, the barcode level clustering achieved the highest ARI values. Consequently, these values were used for comparing MP trees against

LinTIMaT trees.

4.14.3 Supplementary Tables and Figures for LinTIMaTF

Figure 4.7: Comparison of lineage reconstruction performance by LinTIMaT, Camin-Sokal Maximum Parsimony and Neighbor-joining with a lineage recorder of 5 CRISPR targets based on 16 cell *C. elegans* lineage over a range of mutation rates. The number of possible mutational states was set to 8. Fixed mutation rate was used for each CRISPR target. As a measure of performance, RF distance between the true and inferred lineage was computed for LinTIMaT, FP and FN distances between the true and inferred lineages were computed for Camin-Sokal Maximum Parsimony and Neighbor-joining. Lower distance corresponds to better lineage reconstruction. Each box plot summarizes results for 6 replicates with varying simulated CRISPR mutation data and experimental scRNA-seq data.

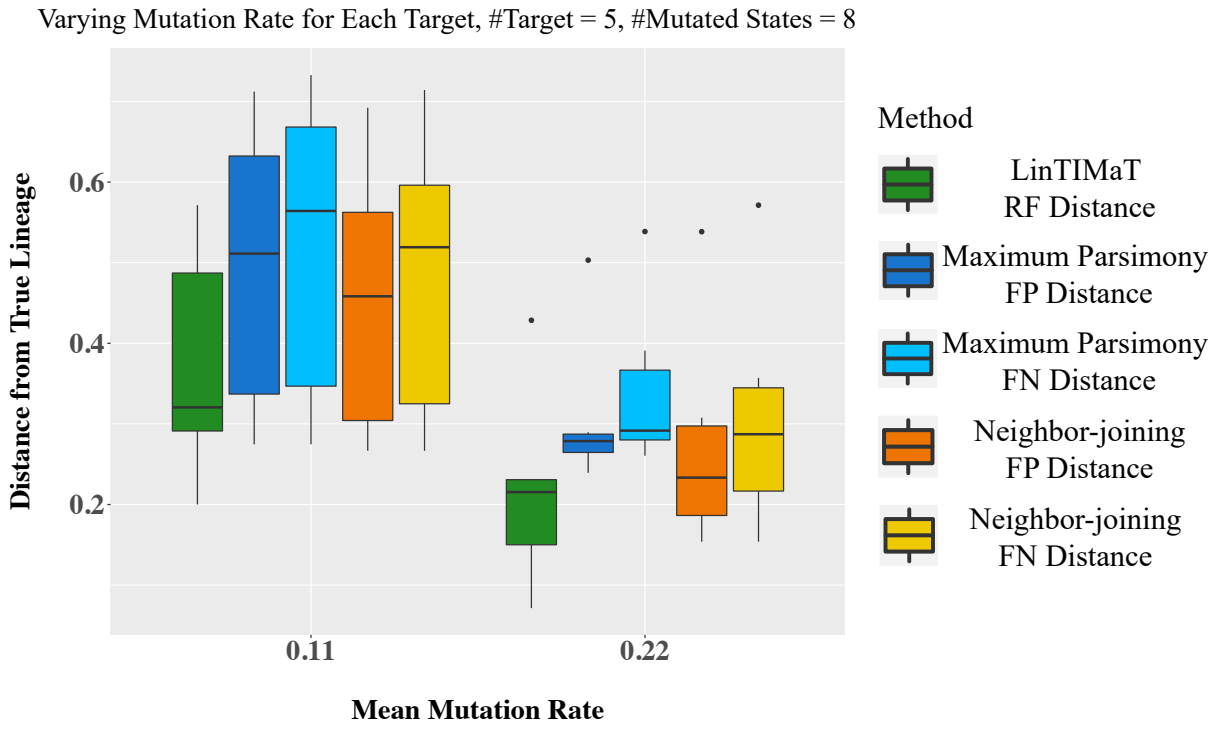

Figure 4.8: Comparison of lineage reconstruction performance by LinTIMaT, Camin-Sokal Maximum Parsimony and Neighbor-joining based on 16 cell *C. elegans* lineage when mutation rate was varied from one target to another. As a measure of performance, RF distance between the true and inferred lineage was computed for LinTIMaT, FP and FN distances between the true and inferred lineages were computed for Camin-Sokal Maximum Parsimony and Neighbor-joining. Lower distance corresponds to better lineage reconstruction. Each box plot summarizes results for 6 replicates with varying simulated CRISPR mutation data and experimental scRNA-seq data.

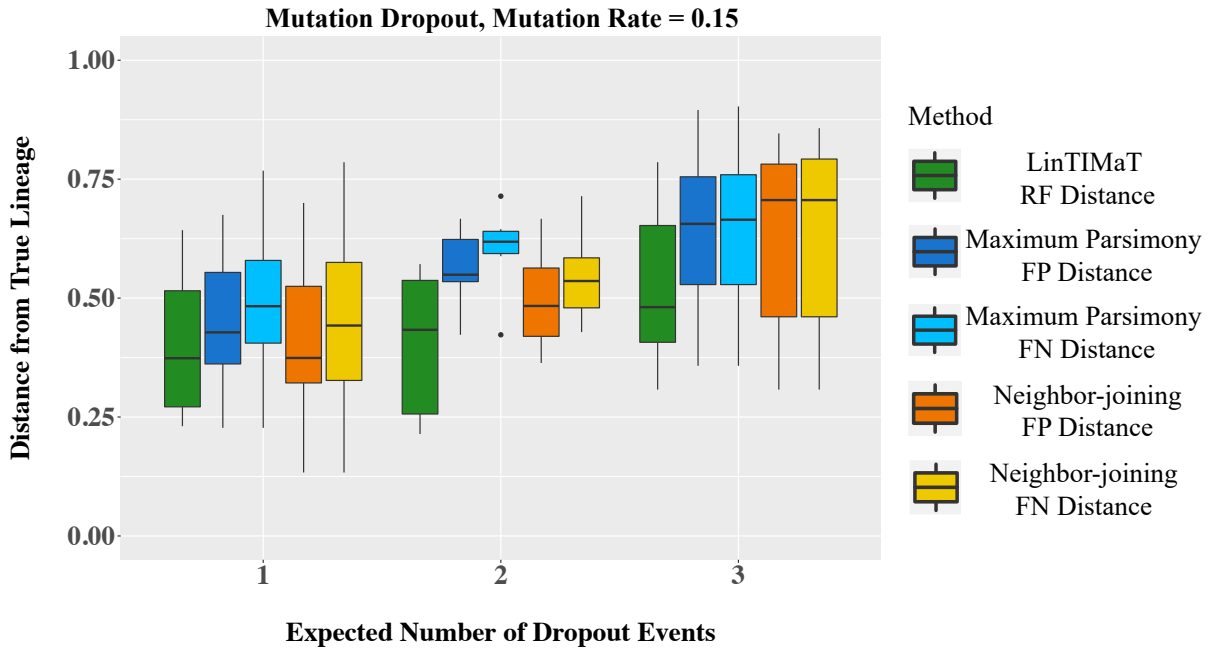

Figure 4.9: Comparison of lineage reconstruction performance by LinTIMaT, Camin-Sokal Maximum Parsimony and Neighbor-joining based on 16 cell *C. elegans* lineage in the presence of mutation dropout. Fixed mutation rate, $\mu = 0.15$ was used for each CRISPR target. As a measure of performance, RF distance between the true and inferred lineage was computed for LinTIMaT, FP and FN distances between the true and inferred lineages were computed for Camin-Sokal Maximum Parsimony and Neighbor-joining. Lower distance corresponds to better lineage reconstruction. Each box plot summarizes results for 6 replicates with varying simulated CRISPR mutation data and experimental scRNA-seq data.

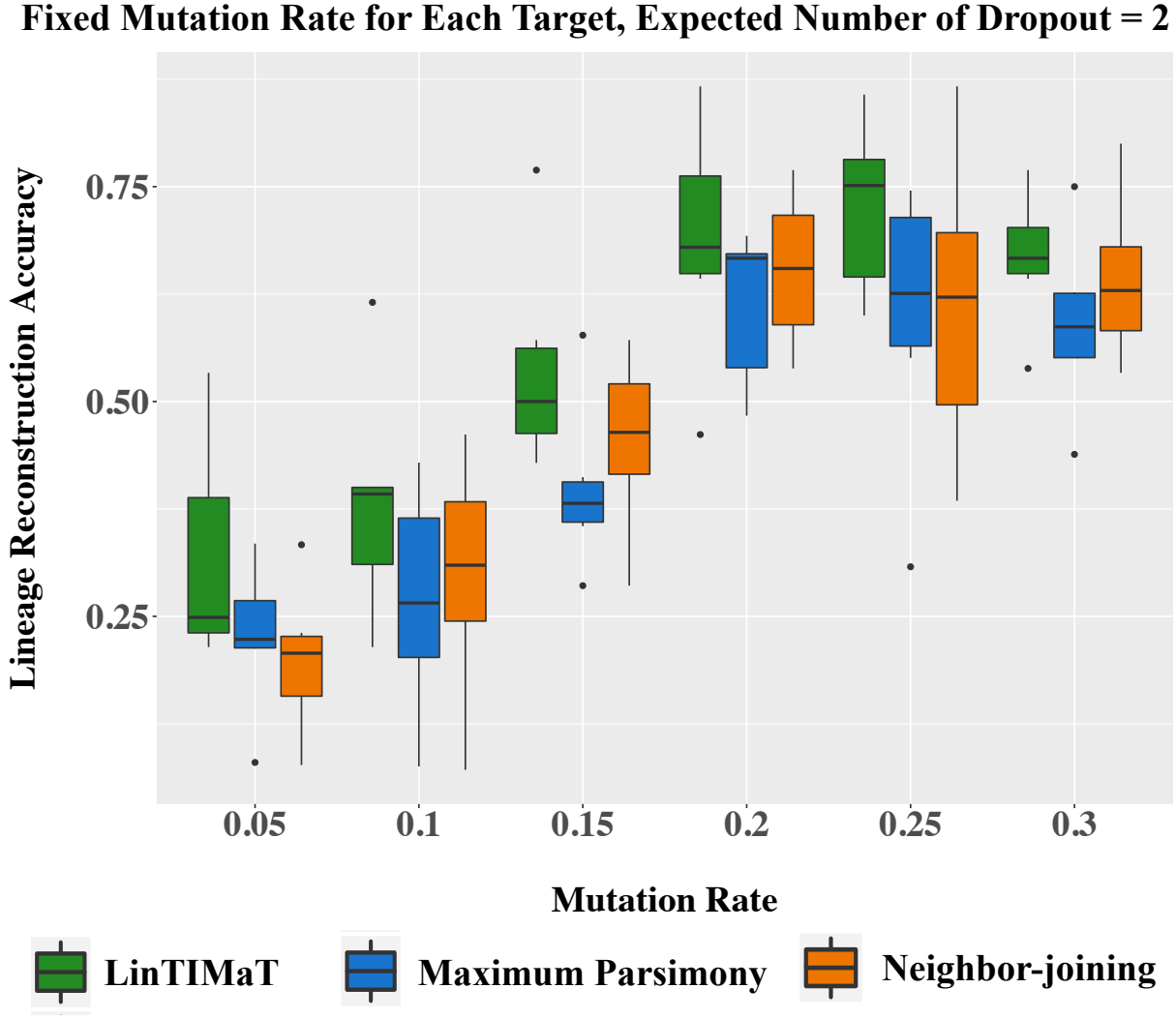

Figure 4.10: Comparison of lineage reconstruction performance by LinTIMaT, Camin-Sokal Maximum Parsimony and Neighbor-joining based on 16 cell *C. elegans* lineage in the presence of mutation dropout. Fixed mutation rate was used for each CRISPR target. For each setting, 2 dropouts were introduced. Mutation rate was varied from $\mu = 0.05$ to $\mu = 0.3$. As a measure of performance, RF distance between the true and inferred lineage was computed for LinTIMaT, FP and FN distances between the true and inferred lineages were computed for Camin-Sokal Maximum Parsimony and Neighbor-joining. Lower distance corresponds to better lineage reconstruction. Each box plot summarizes results for 6 replicates with varying simulated CRISPR mutation data and experimental scRNA-seq data.

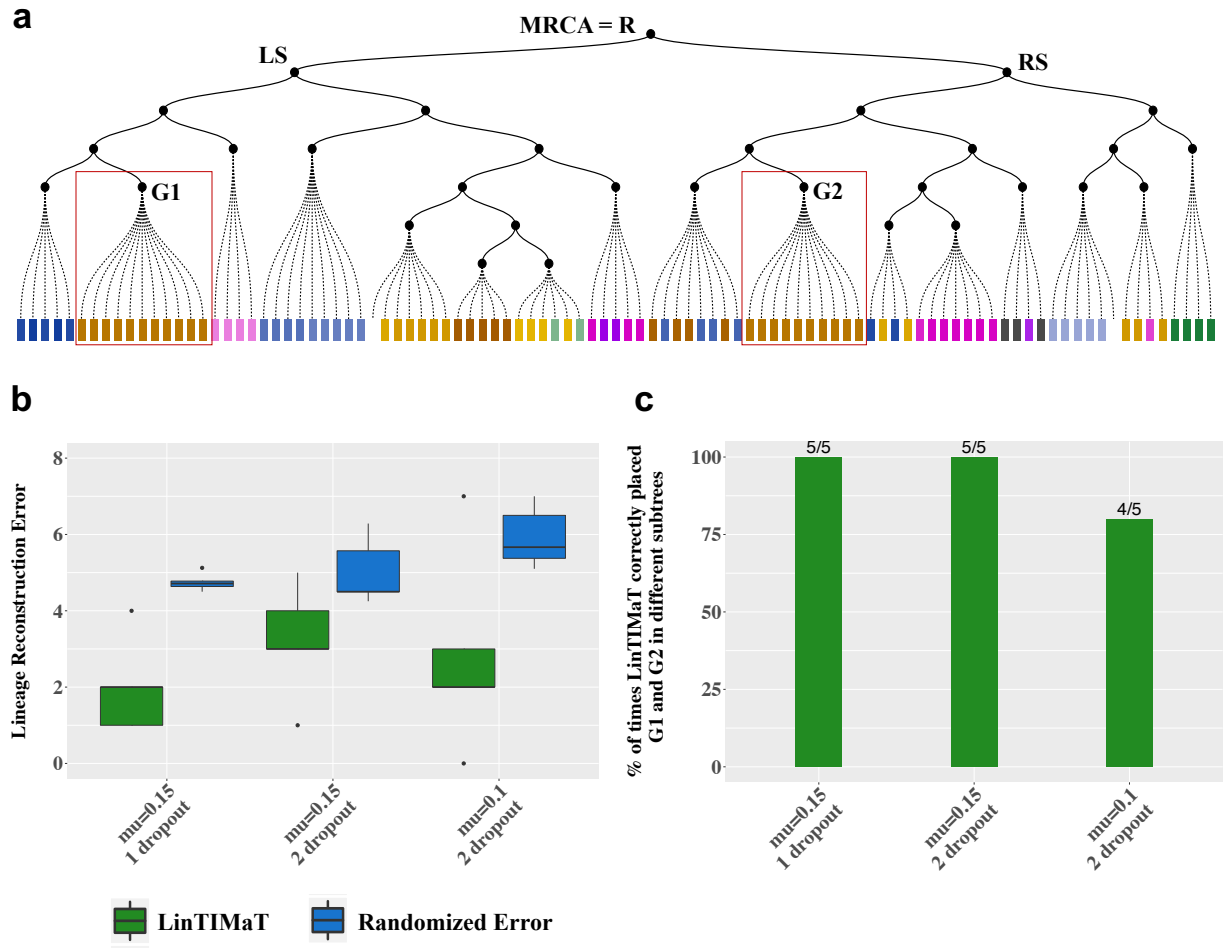

Figure 4.11: Performance of LinTIMaT in recovering divergent lineage relationship when no CRISPR mutations are shared between the groups of cells. (a) An example simulated lineage. G1 and G2 are the groups of cells that are from the same cell type but diverged from the root (their most recent common ancestor, MRCA) of the lineage. G1 is present in the left subtree (LS) and G2 is present in the right subtree (RS). (b) Performance of LinTIMaT in recovering the divergent lineage between G1 and G2. LinTIMaT's lineage reconstruction error is compared against a randomized error that represents the average lineage reconstruction error considering the case when G1 and G2 are placed in the same subtree. Each box plot summarizes results for 5 replicates. (c) Performance of LinTIMaT in placing G1 and G2 in two different subtrees under different experimental conditions.

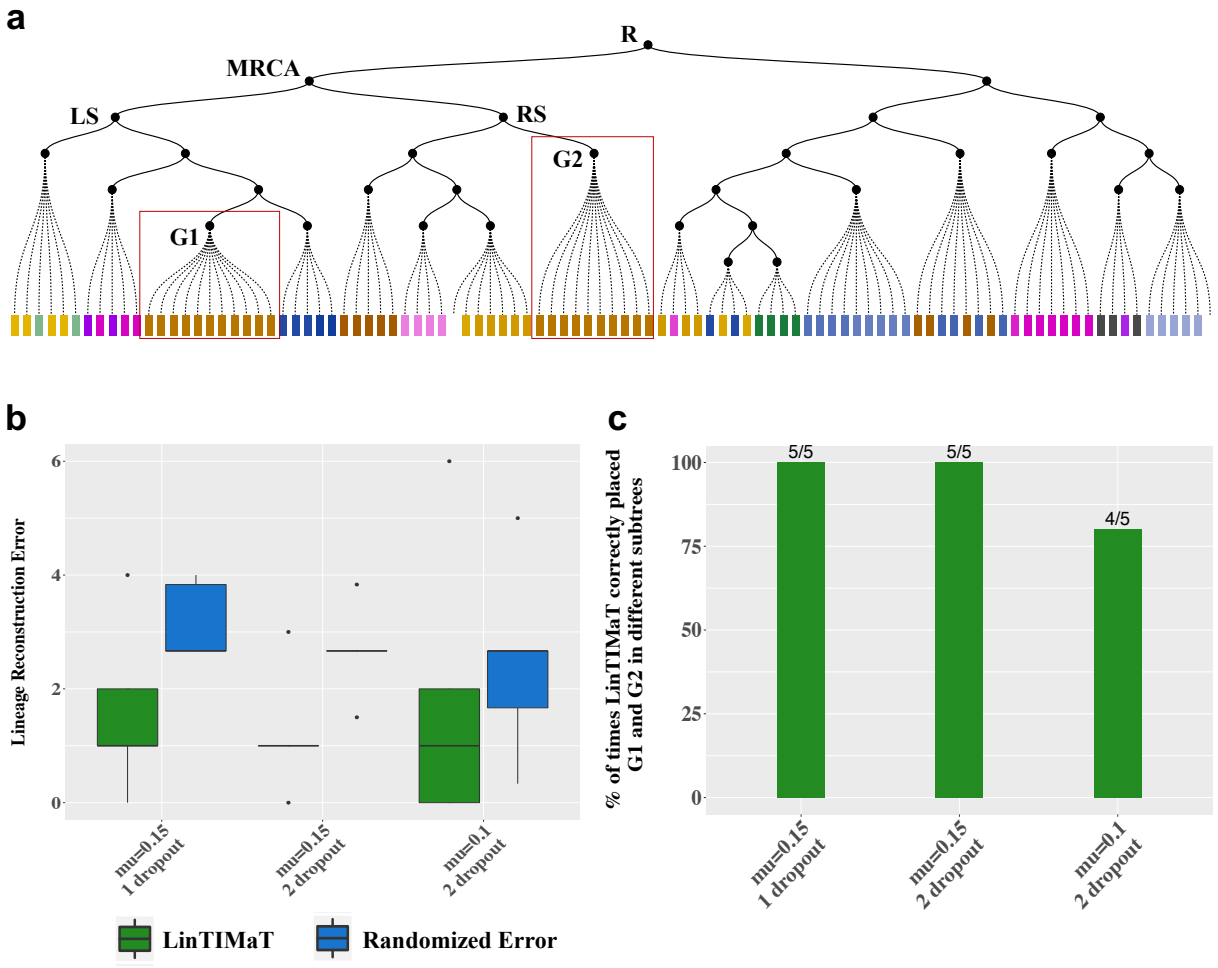

Figure 4.12: Performance of LinTIMaT in recovering divergent lineage relationship when some CRISPR mutations are possibly shared between the groups of cells. (a) An example simulated lineage. G1 and G2 are the groups of cells that are from the same cell type but diverged very early on in the lineage (their most recent common ancestor, MRCA is a child of root). G1 is present in the left subtree (LS) and G2 is present in the right subtree (RS). (b) Performance of LinTIMaT in recovering the divergent lineage between G1 and G2. LinTIMaT's lineage reconstruction error is compared against a randomized error that represents the average lineage reconstruction error considering the case when G1 and G2 are placed in the same subtree. Each box plot summarizes results for 5 replicates. (c) Performance of LinTIMaT in placing G1 and G2 in two different subtrees under different experimental conditions.

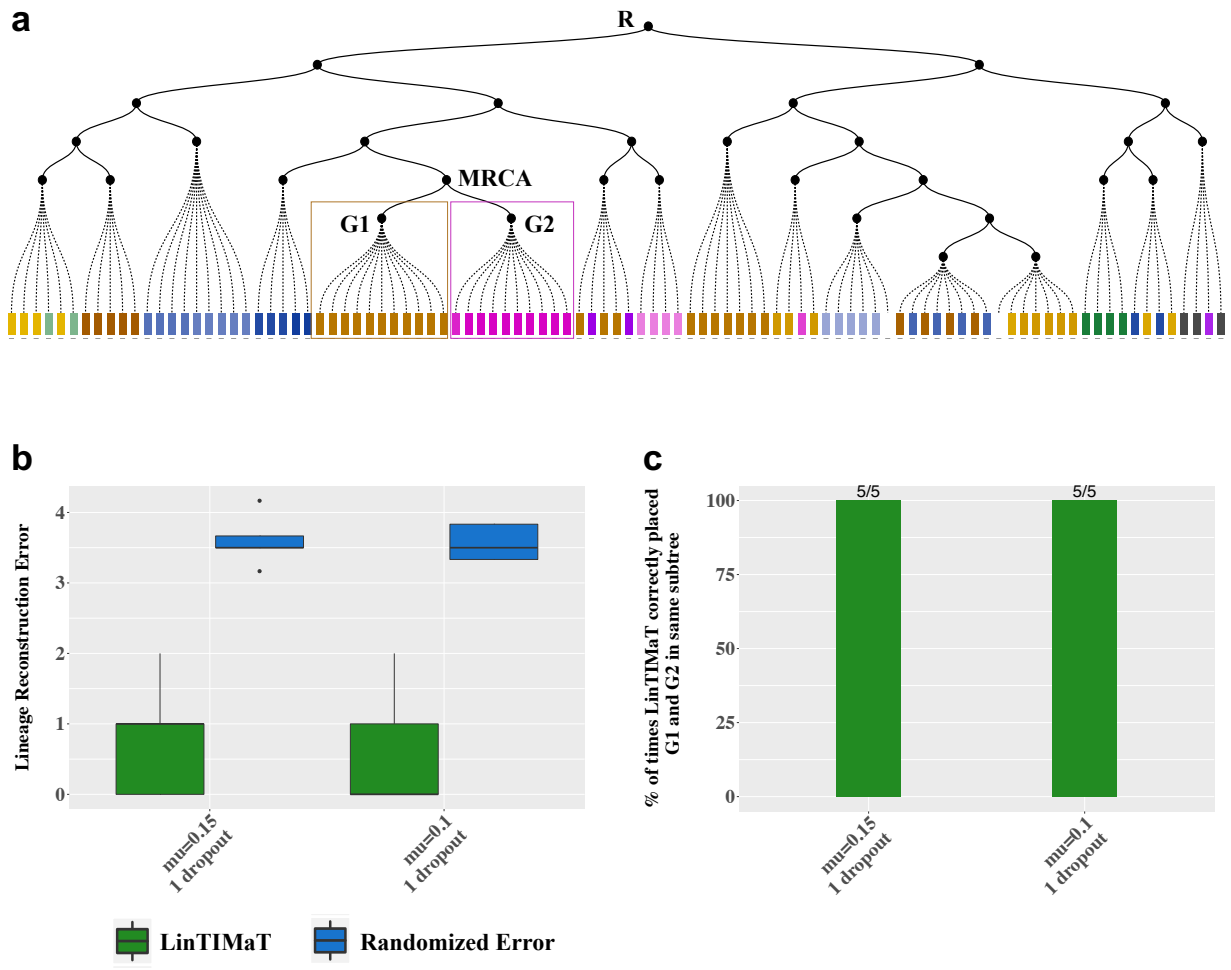

Figure 4.13: Performance of LinTIMaT in recovering convergent lineage relationship between two groups of cells that are transcriptionally distinct (different cell type) but have a common ancestry. (a) An example simulated lineage. G1 and G2 are the groups of cells that are from different cell types (neuron and progenitor) but they share the same lineage and are next to each other, parent of G1 and G2 is their most recent common ancestor (MRCA). (b) Performance of LinTIMaT in recovering the convergent lineage between G1 and G2. LinTIMaT's lineage reconstruction error is compared against a randomized error that represents the average lineage reconstruction error considering the case when G1 and G2 are placed in different subtrees. Each box plot summarizes results for 5 replicates. (c) Performance of LinTIMaT in placing G1 and G2 in the same subtree under different experimental conditions.

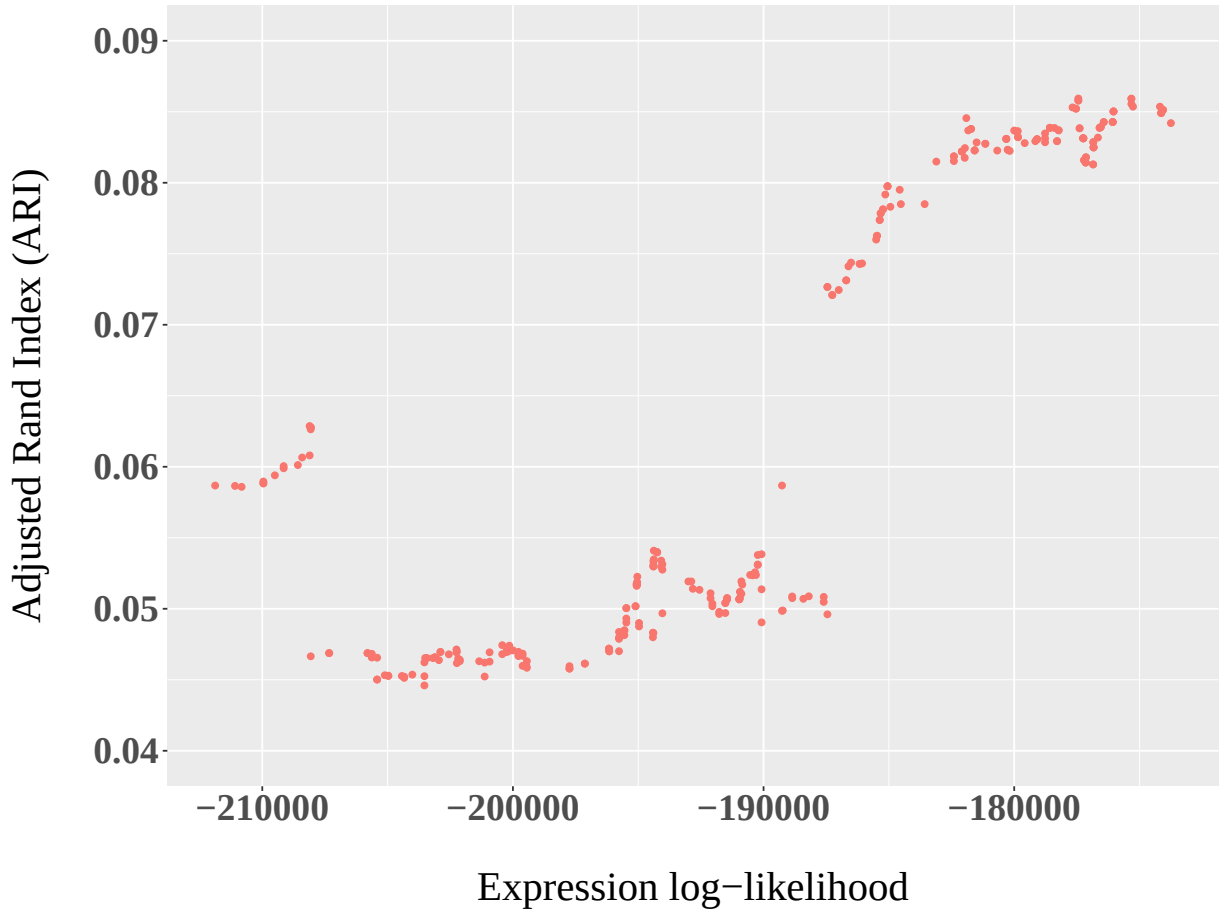

Figure 4.14: Adjusted Rand Index (ARI) which measures the agreement between cell types in the tree clusters and cell types assigned by the original paper [154] as a function of the likelihood computed by LinTIMaT for ZF1. The fact that as the likelihood increases the ARI increases as well indicates that the target function of LinTIMaT is capturing biologically relevant relationships between cells.

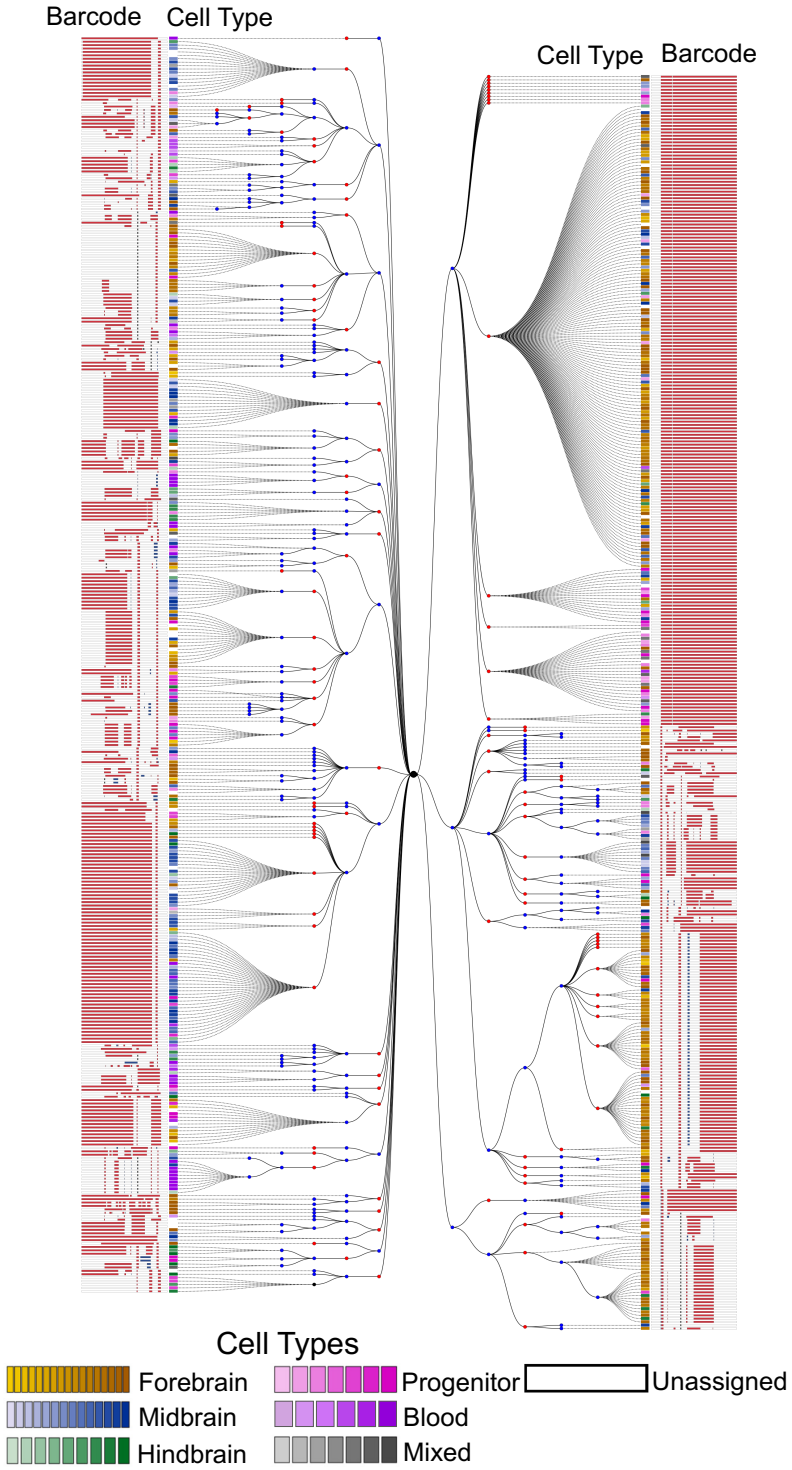

Figure 4.15: The lineage tree reconstructed by LinTIMaT from a single juvenile zebrafish brain (ZF1) dataset generated by scGESTALT. The lineage tree is built on 750 cells. Blue nodes represent Cas9-editing events (mutations) and red nodes represent clusters inferred by LinTIMaT from transcriptomic data. Each leaf node is a cell, represented by a square, and its color represents its cell type as indicated in the legend. The mutated barcode for each cell is displayed as a white bar with insertions (blue) and deletions (red).

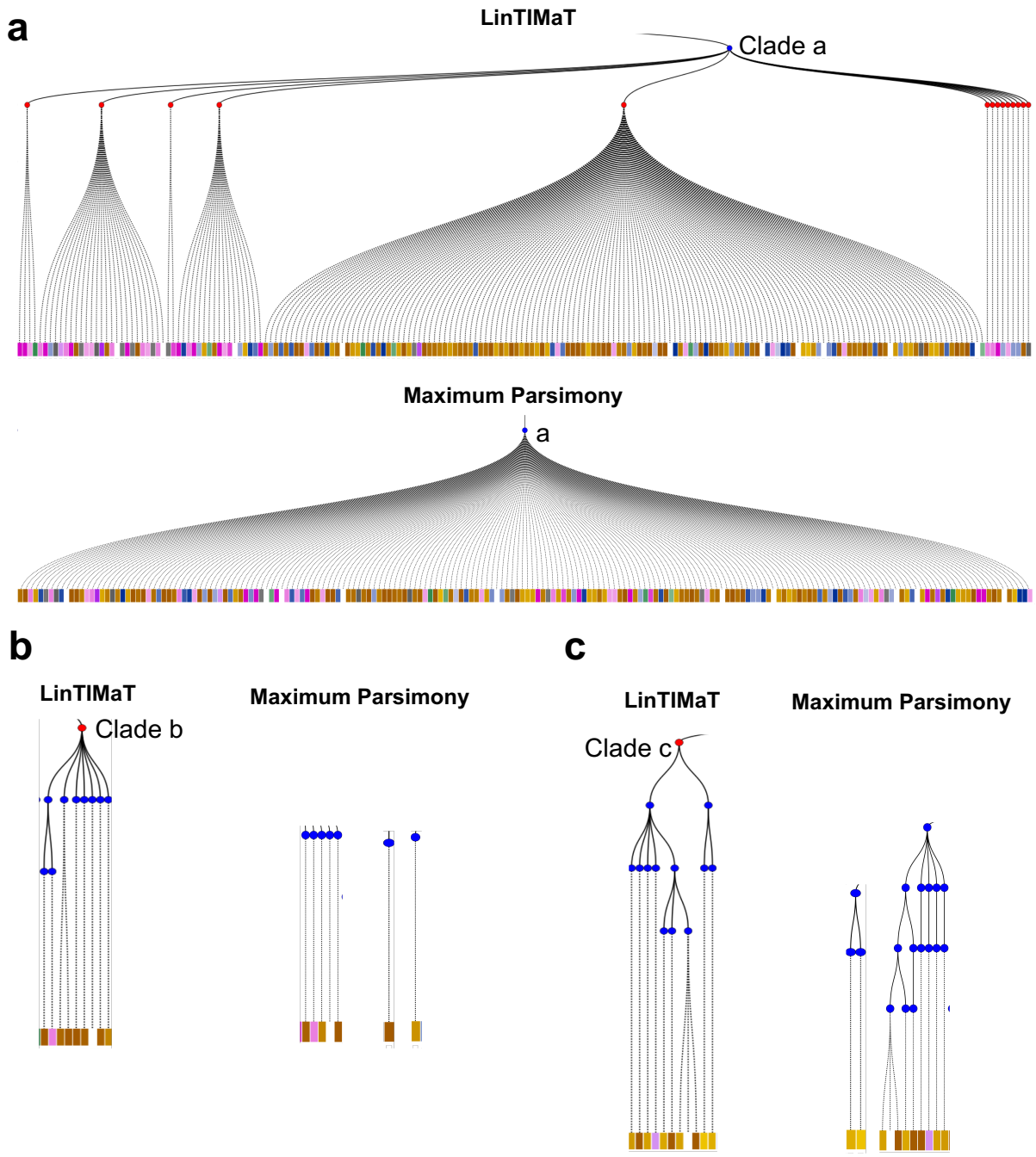

Figure 4.16: Example subtrees in the lineage tree reconstructed by LinTIMaT from a single juvenile zebrafish brain (ZF1) dataset generated by scGESTALT. (a) Example subtree showing ability of LinTIMaT in separating cells with exactly the same barcode to distinct clusters of cell types. (b-c) Example subtrees displaying LinTIMaT's ability to cluster cells with different barcodes together based on their cell types, maximum parsimony puts them on distinct branches.

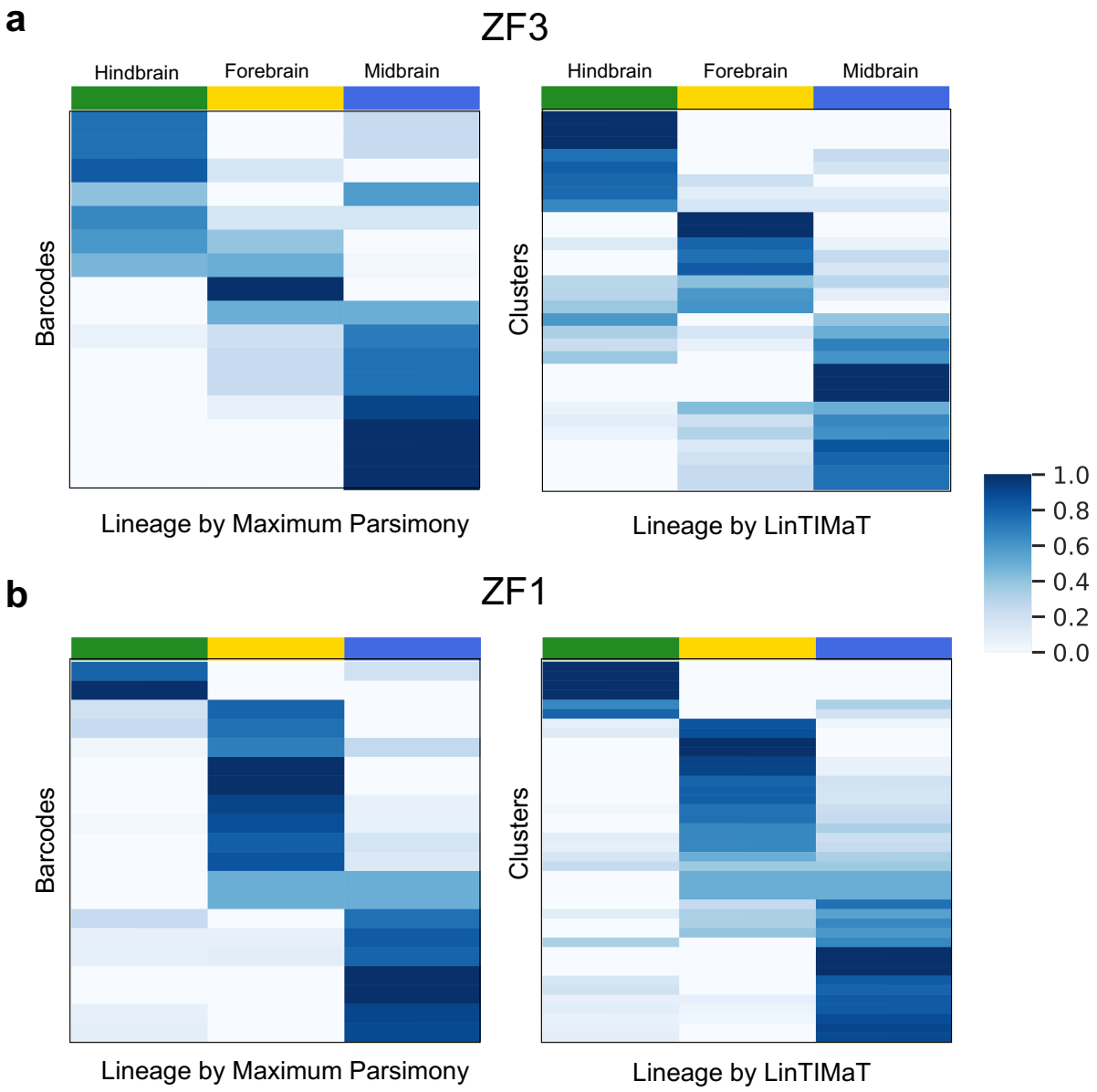

Figure 4.17: Distribution of cell types in the juvenile zebrafish brain for scGESTALT datasets. Heat map of the distribution of cell clusters for each region of the brain (columns). Cell types were classified as belonging to the forebrain, midbrain or hindbrain, and the proportions of cells within each region were calculated for each cluster. For MP lineage, the rows of the heat map represent barcodes, for LinTIMaT lineage, the rows represent clusters inferred from barcodes and expression data. (a) Comparison for ZF3. (b) Comparison for ZF1.

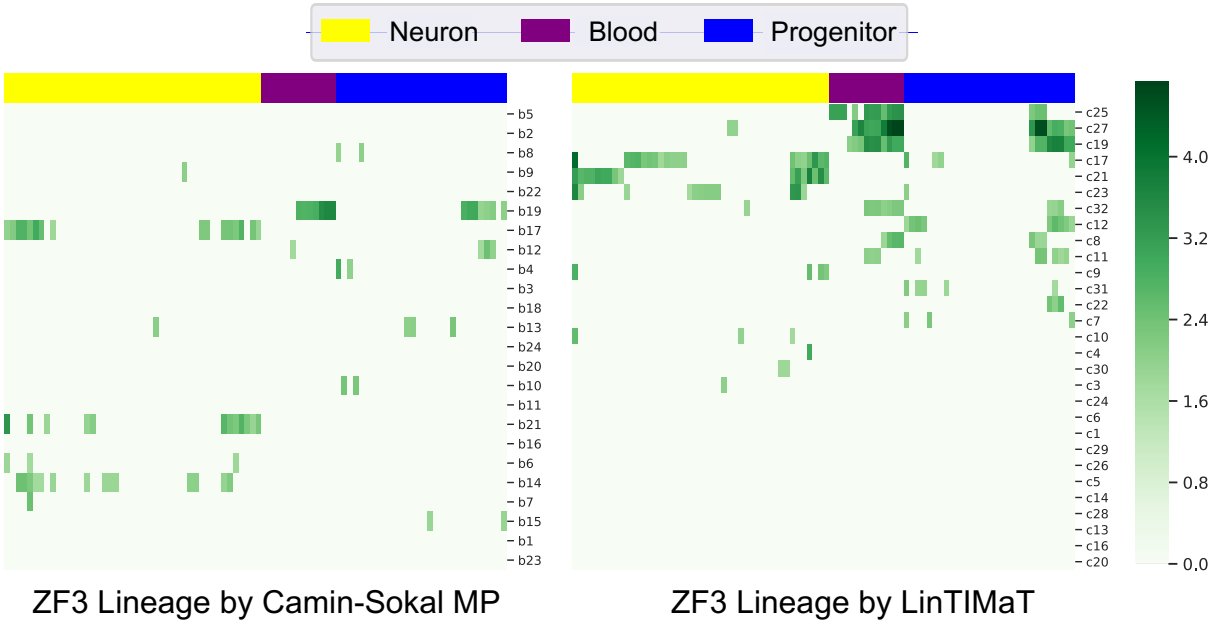

Figure 4.18: Comparison of GO analysis for lineage trees reconstructed by Camin-Sokal Maximum Parsimony and LinTIMaT for a single juvenile zebrafish brain (ZF3) dataset generated by scGESTALT. The figure shows heat map of the square rooted negative log p-values of all GO terms for the clusters in the reconstructed lineage. The rows represent clusters and the columns represent different GO terms as shown in Supplementary Tables. The values were colored as shown in the key. The yellow, purple and blue columns correspond to GO terms related to neurons, blood and progenitors respectively. The left panel shows the heat map for the barcode clusters in MP reconstructed lineage, and the right panel shows the heat map for the clusters in LinTIMaT reconstructed lineage.

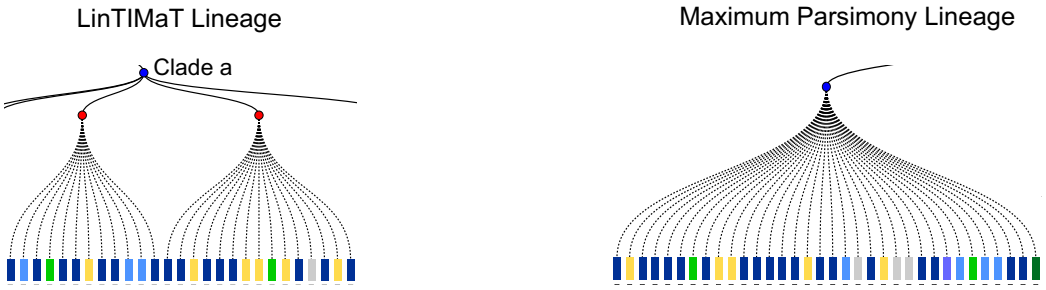

Figure 4.19: Example subtree in the lineage tree reconstructed by LinTIMaT for a zebrafish dataset (R2) generated using ScarTrace. This subtree shows the ability of LinTIMaT in separating cells with exactly the same barcode to distinct clusters of cell types. Figure 4.20 for cell type color legend.

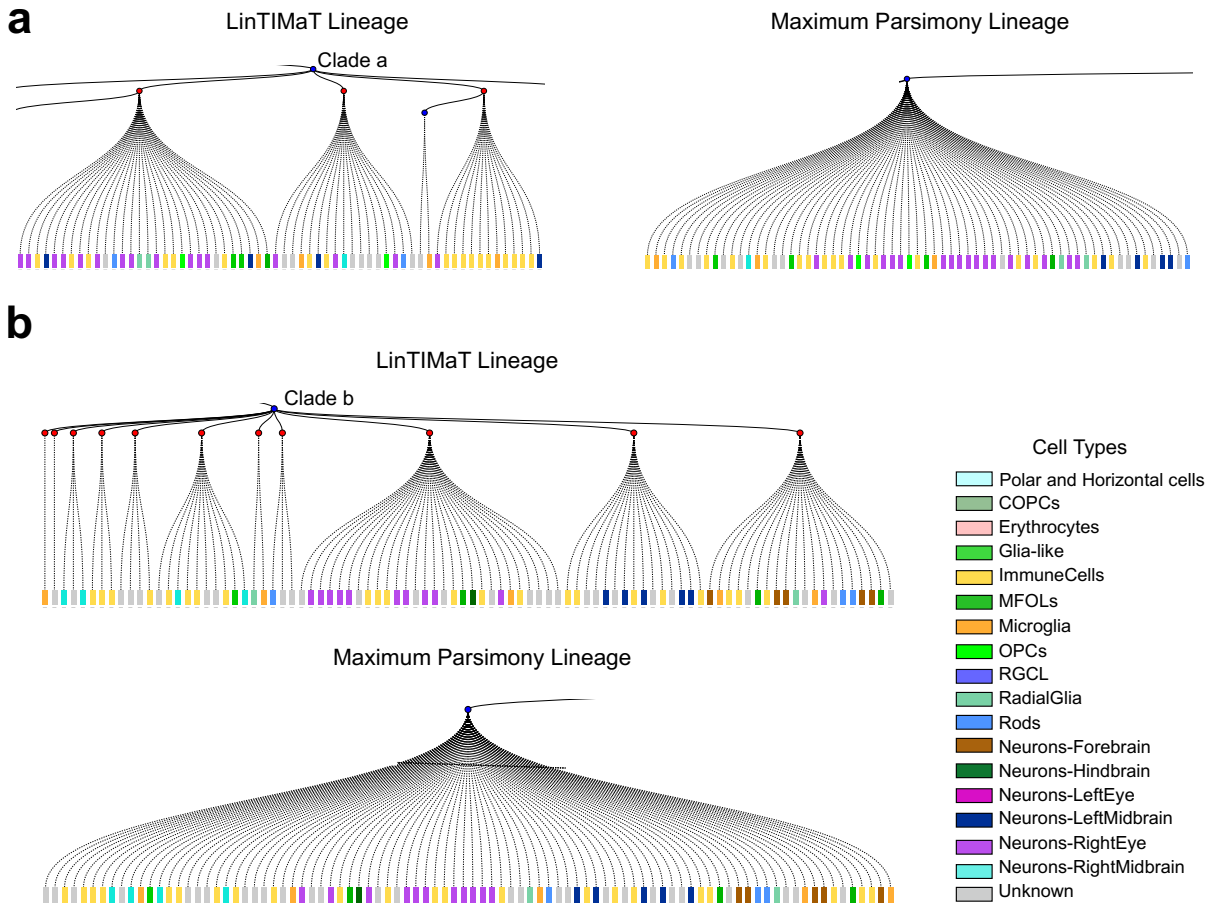

Figure 4.20: Example subtrees in the lineage tree reconstructed by LinTIMaT for a zebrafish dataset (R3) generated using ScarTrace. These subtrees illustrate the ability of LinTIMaT in separating cells with exactly the same barcode to distinct clusters of cell types.

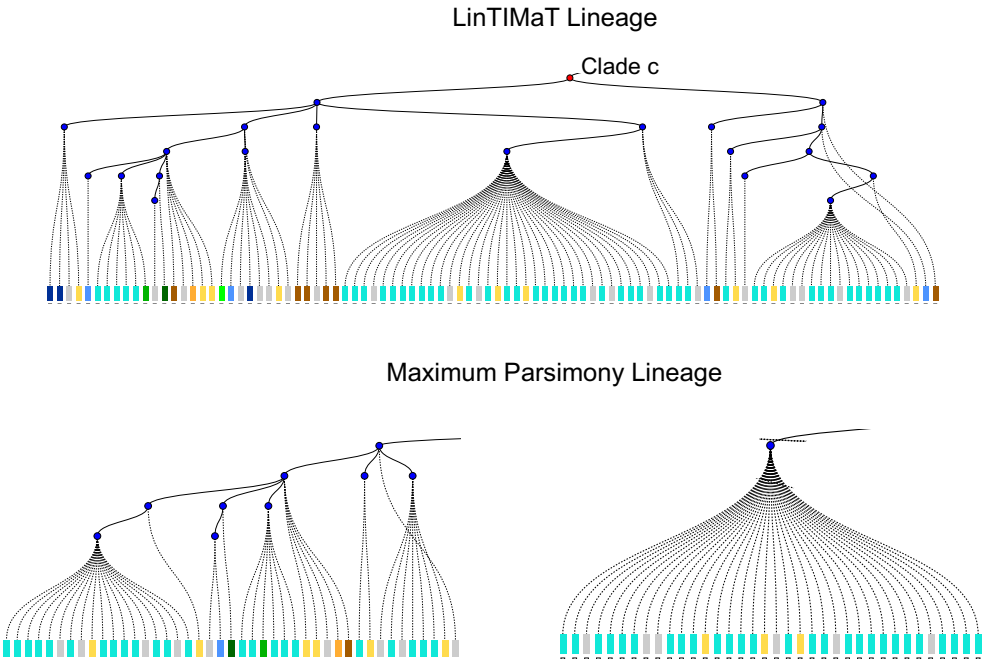

Figure 4.21: Example subtree in the lineage tree reconstructed by LinTIMaT for a zebrafish dataset (R3) generated using ScarTrace. This subtree displays LinTIMaT's ability to cluster cells with different barcodes together based on their cell types, maximum parsimony puts them on distinct branches. Figure 4.20 for cell type color legend.

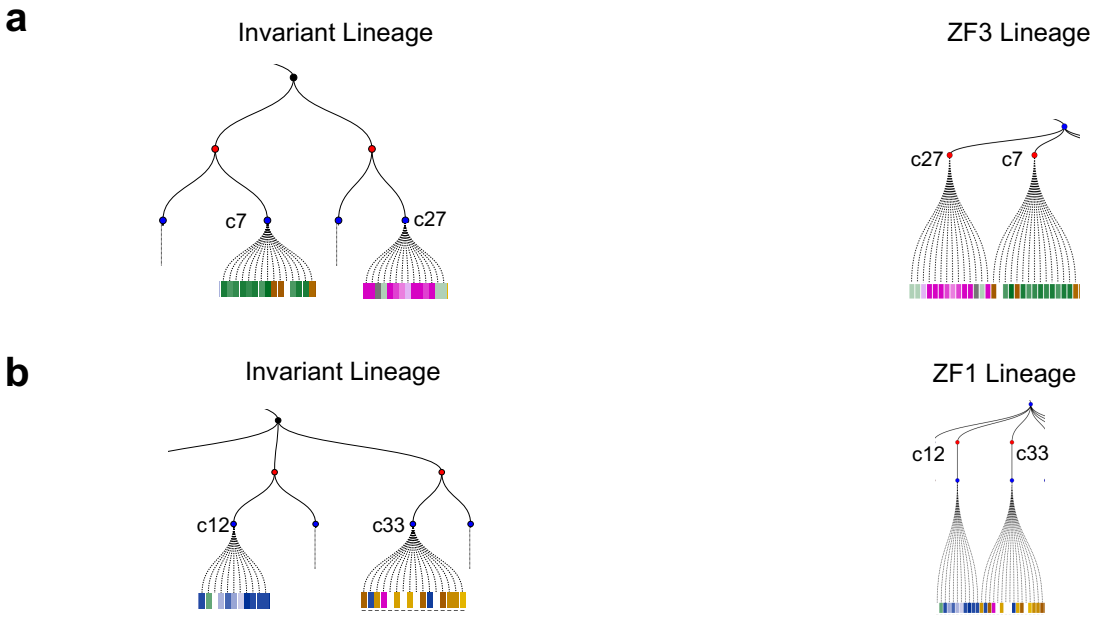

Figure 4.22: Invariant lineage preserves ancestor-descendant relationships in individual lineages reconstructed for scGESTALT datasets. (a) Clusters c7 and c27 are present in the same subtree in both the invariant lineage and ZF3 lineage. (b) Clusters c12 and c33 are present in the same subtree in both the invariant lineage and ZF1 lineage.

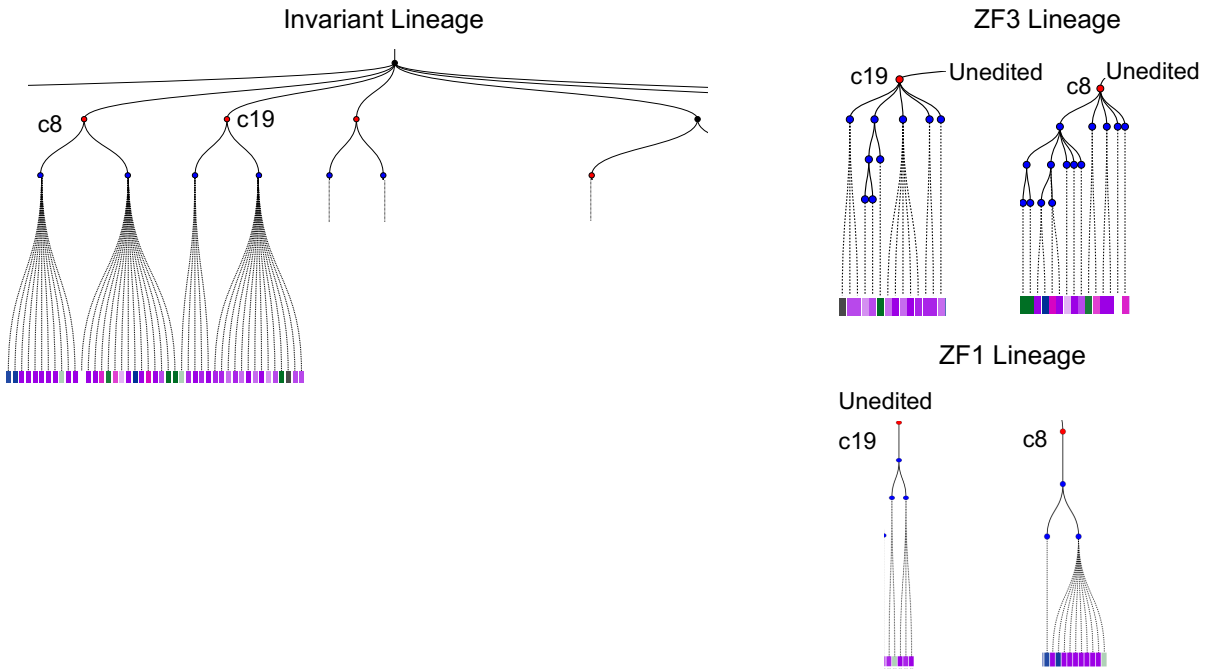

Figure 4.23: Invariant lineage places similar cell clusters together in the same subtree. In ZF3 (generated by scGESTALT) lineage, clusters c8 and c19 both contain cells belonging to blood cell type but these clusters are placed in different branches. In invariant lineage these clusters are placed in the same subtree. Similar examples are observed for ZF1 lineage.

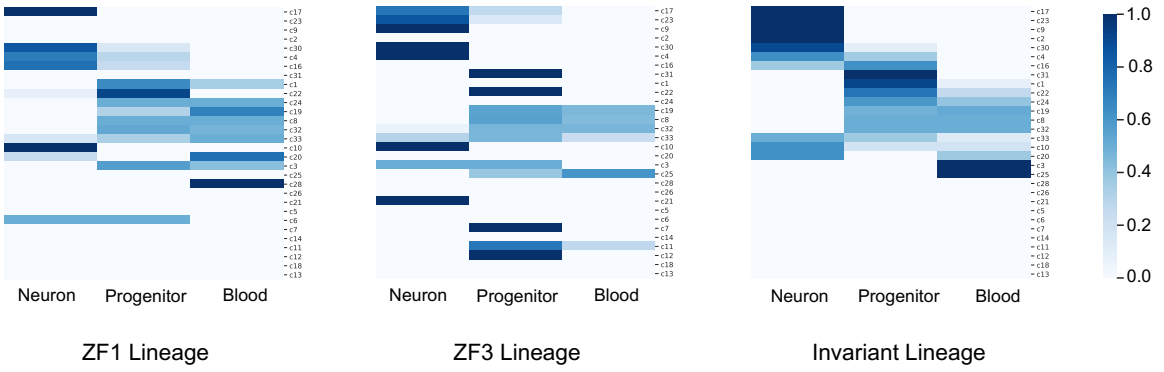

Figure 4.24: Proportions of each type of GO terms for the invariant clusters (for scGESTALT dataset). The rows represent invariant clusters and the columns represent different types of GO terms.

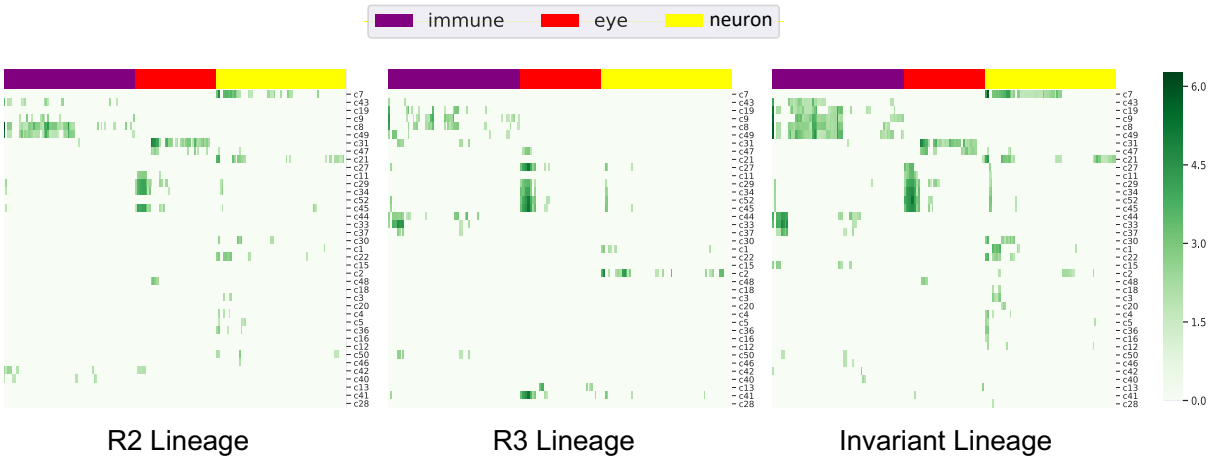

Figure 4.25: Heat map of the square rooted negative log p-values of all GO terms for the invariant clusters that contains 10 or more cells for the ScarTrace dataset. The rows represent selected invariant clusters and the columns represent different GO terms as shown in Table 4.10. The values were colored as shown in the key. The yellow, purple and red columns correspond to GO terms related to neurons, immune celltype and eye respectively. The leftmost panel shows the heat map for the clusters in R2 lineage, middle panel shows the heat map for R3 lineage and the righmost panel shows the heat map for the invariant lineage.

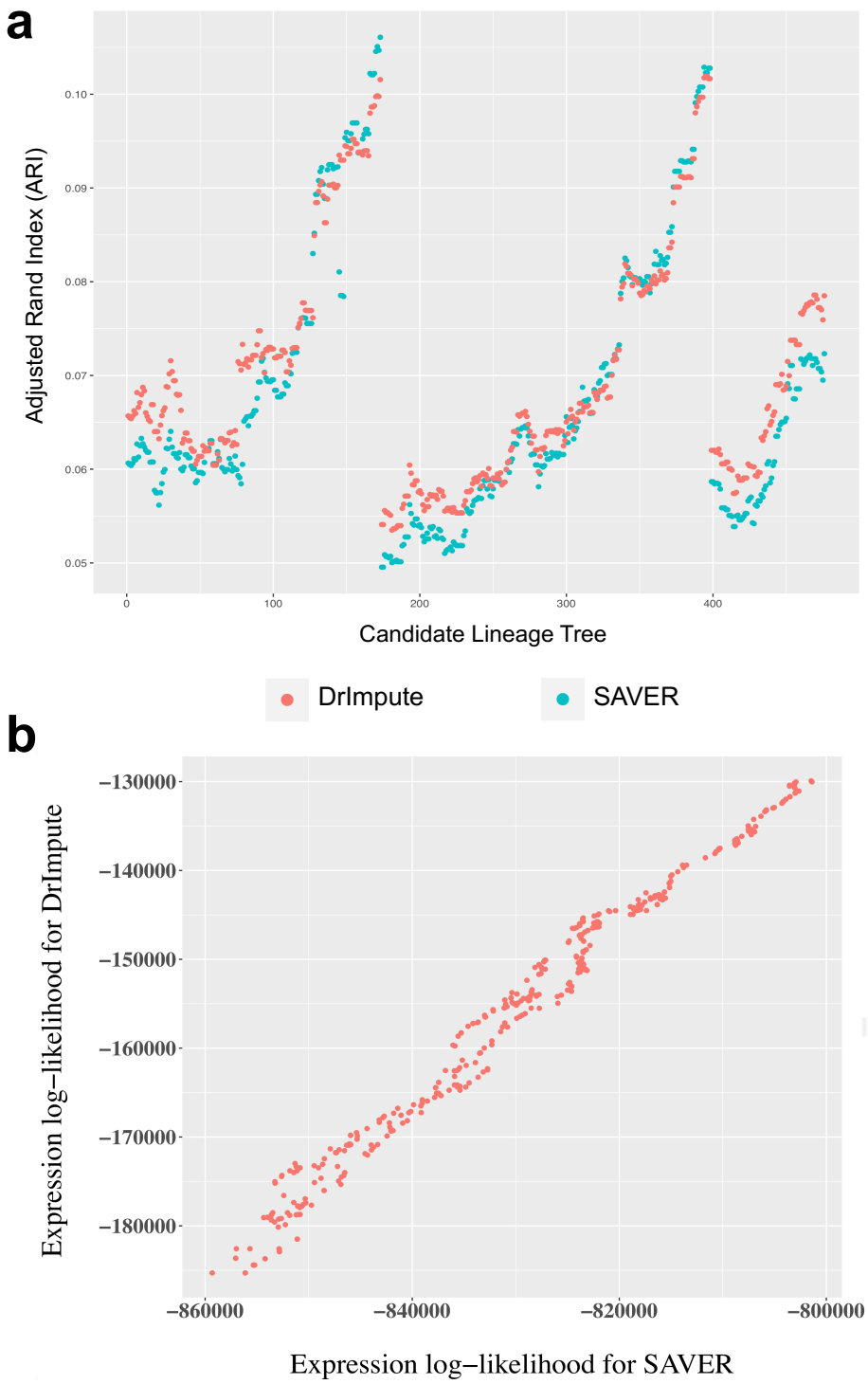

Figure 4.26: (a) Effect of the imputation method on LinTIMaT's expression likelihood function displayed through cell clustering performance. For a set of candidate lineage trees for ZF3, we compared the cell clustering based on expression likelihood for expression data imputed using two imputation methods: DrImpute and SAVER. The cell clustering performance is measured in terms of Adjusted Rand Index. (b) Plot comparing the expression log-likelihoods for a set of lineage trees for data imputed using DrImpute and SAVER. Correlation 0.9914.

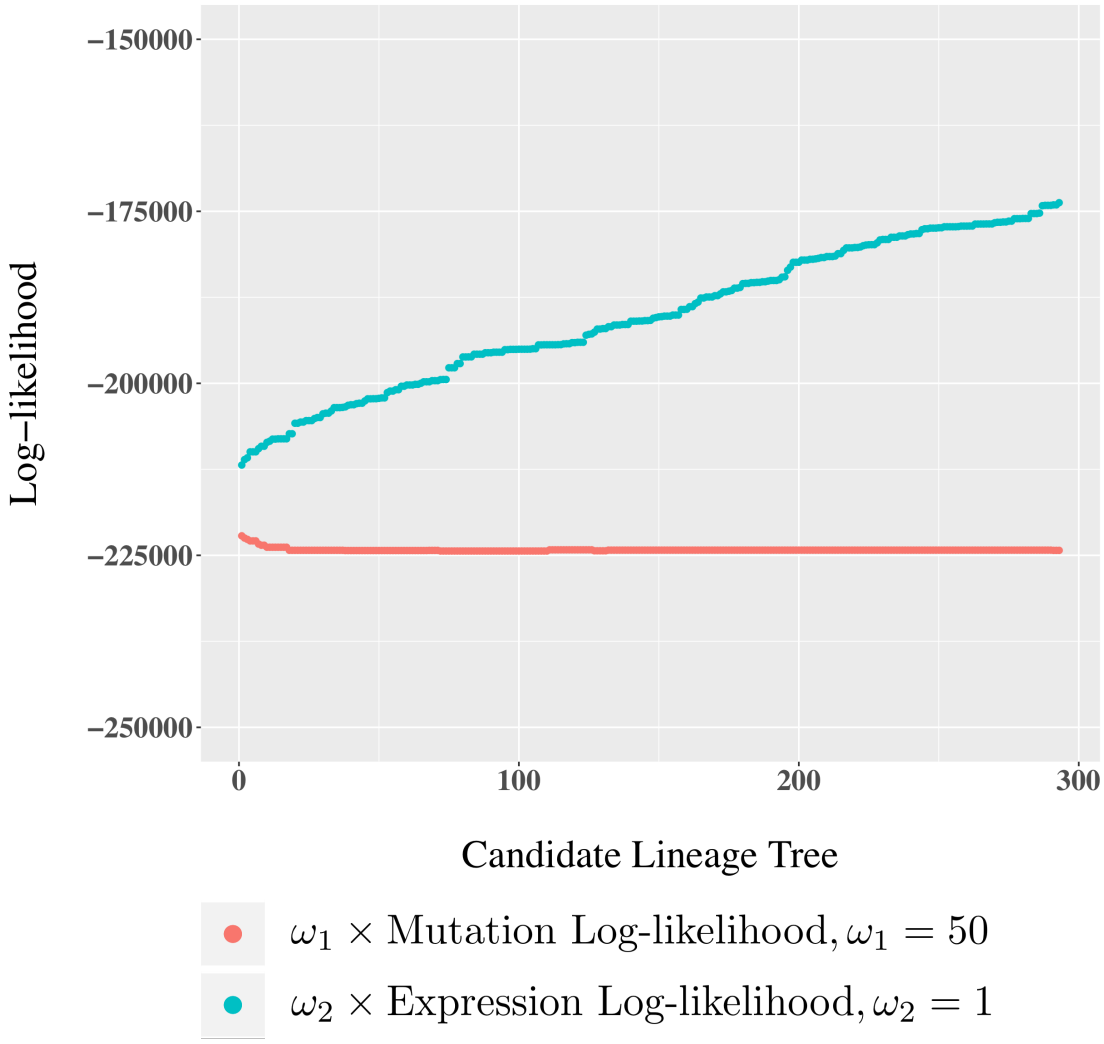

Figure 4.27: Comparison of weighted values of mutation log-likelihood and expression log-likelihood for specific weights, $\omega_1 = 50$ and $\omega_2 = 1$ for a set of candidate lineage trees for ZF1. For these values of weights, the weighted values of the two log-likelihoods remain in the same range.

Table 4.1: The average performance of mutation likelihood optimization for C. elegans simulated dataset

true likelihood	start likelihood	trained likelihood	improvement percentage
-34.5673477	-325057.8797	-49.24198487	0.9998492419

Table 4.2: The ARI for each scGESTALT tree calculated based on different levels

ZF1			ZF3		
Level	#cluster	ARI	Level	#cluster	ARI
1	25	0.035658001	1	23	0.01938034
2	75	0.047930182	2	67	0.037909748
3	363	0.041749244	3	153	0.045092761
4	531	0.06103636	4	279	0.027522376
5	676	0.04716886	5	368	0.002327659
6	750	0	6	376	0
Barcode	192	0.061237582	Barcode	150	0.056133354

Table 4.3: Comparison of log-likelihood score of lineage trees for scGESTALT datasets based on only mutation data

Method	ZF1	ZF3
LinTIMaT	-4485.564873	-2871.118661
MP	-303463.075241	-102474.127300

Table 4.4: Comparison of log-likelihood score of lineage trees for ScarTrace datasets based on only mutation data

Method	R2	R3
LinTIMaT	-2111.155680	-1816.091048
MP	-501741.626610	-301271.047508

Table 4.5: Strings for filtering the GO terms for each GO type for the scGESTALT dataset

GO type	**filter strings**
neuron	neuro nervous synap
blood	heme hema hemo erythrocyte myeloid hscs immune
progenitor	develop differentiat

Table 4.6: Strings for filtering the GO terms for each GO type for the ScarTrace dataset

GO type	**filter strings**
neuron	neuro nervous synap
immune	heme hema hemo erythrocyte myeloid hscs immune
eye	photoreceptor retina eye phototransduction optic visual light_stimulus

Table 4.7: Full list of GO terms and corresponding p-values for scGESTALT ZF3 appearing in LinTIMaT clusters but not in any individual clusters for MP tree.

GO term	cluster,p-value
Acetylcholine Neurotransmitter Release Cycle	(c23,1.49e-02)
Norepinephrine Neurotransmitter Release Cycle	(c23,1.11e-02)
neuron development	(c17,4.41e-04)
heme-copper terminal oxidase activity	(c25,5.87e-05)
neurotransmitter receptor complex	(c23,1.07e-02)
neurogenesis	(c17,1.26e-02)
oxidoreductase activity, acting on a heme group of donors, oxygen as acceptor	(c25,5.87e-05)
Serotonin Neurotransmitter Release Cycle	(c23,1.11e-02)
Optic neuropathy	(c10,1.98e-02)
presynaptic cytoskeleton	(c30,3.75e-02)
postsynaptic density	(c23,4.98e-02)
oxidoreductase activity, acting on a heme group of donors	(c25,5.87e-05)
animal organ development	(c27,5.08e-03),(c19,1.24e-04),(c11,2.45e-02),(c12,8.33e-03)
cytoskeleton of presynaptic active zone	(c30,3.75e-02)
peripheral nervous system neuron axonogenesis	(c32,2.66e-02)
peripheral nervous system neuron differentiation	(c27,1.90e-02)
Hematological neoplasm	(c27,5.56e-05),(c19,9.13e-03),(c25,9.13e-03)
peripheral nervous system neuron development	(c27,1.90e-02)
generation of neurons	(c17,3.43e-03)
postsynapse	(c23,1.21e-02)
regulation of cell differentiation	(c31,3.40e-02)
cell development	(c17,2.08e-02)
neuron differentiation	(c23,2.82e-02),(c17,9.02e-04)
Global developmental delay	(c17,4.05e-02)
neuron projection development	(c17,5.39e-03)
immune system process	(c19,1.47e-02)
Neurological speech impairment	(c17,5.80e-04)

Table 4.8: Full list of filtered GO terms used in GO p-value/proportion heat maps for the scGESTALT dataset individual tree of ZF3, see Table 4.5 for the keywords we used to filter these GO terms.

index	GO type	GO term	index	GO type	GO term
1	neuron	neuron_synapse part	45	neuron	neuron_anterograde trans-synaptic signaling
2	neuron	neuron_postsynaptic density	46	blood	blood_heme-copper terminal oxidase activity
3	neuron	neuron_neuronal ion channel clustering	47	blood	blood_erythrocyte homeostasis
4	neuron	neuron_chemical synaptic transmission	48	blood	blood_hemopoiesis
5	neuron	neuron_trans-synaptic signaling	49	blood	blood_Hematological neoplasm
6	neuron	neuron_neuron maturation	50	blood	blood_Abnormal erythrocyte morphology
7	neuron	neuron_vesicle-mediated transport in synapse	51	blood	blood_immune system development
8	neuron	neuron_postsynapse	52	blood	blood_oxidoreductase activity acting on a heme group of donors oxygen as acceptor
9	neuron	neuron_synaptic vesicle membrane	53	blood	blood_hematopoietic or lymphoid organ development
10	neuron	neuron_Acetylcholine Neurotransmitter Release Cycle	54	blood	blood_myeloid cell differentiation
11	neuron	neuron_Neurotransmitter release cycle	55	blood	blood_myeloid cell homeostasis
12	neuron	neuron_synaptic vesicle	56	blood	blood_oxidoreductase activity acting on a heme group of donors
13	neuron	neuron_neurotransmitter secretion	57	blood	blood_erythrocyte differentiation
14	neuron	neuron_neuron projection	58	blood	blood_immune system process
15	neuron	neuron_Norepinephrine Neurotransmitter Release Cycle	59	progenitor	progenitor_multicellular organism development
16	neuron	neuron_Glutamate Neurotransmitter Release Cycle	60	progenitor	progenitor_nervous system development
17	neuron	neuron_generation of neurons	61	progenitor	progenitor_cell development
18	neuron	neuron_neurotransmitter receptor complex	62	progenitor	progenitor_brain development
19	neuron	neuron_synaptic signaling	63	progenitor	progenitor_embryo development ending in birth or egg hatching
20	neuron	neuron_synapse	64	progenitor	progenitor_anatomical structure development
21	neuron	neuron_Neurological speech impairment	65	progenitor	progenitor_animal organ development
22	neuron	neuron_regulation of neurotransmitter levels	66	progenitor	progenitor_embryo development
23	neuron	neuron_neuron part	67	progenitor	progenitor_immune system development
24	neuron	neuron_synaptic vesicle cycle	68	progenitor	progenitor_posterior lateral line development
25	neuron	neuron_neurogenesis	69	progenitor	progenitor_peripheral nervous system neuron differentiation
26	neuron	neuron_peripheral nervous system neuron differentiation	70	progenitor	progenitor_hematopoietic or lymphoid organ development
27	neuron	neuron_presynapse	71	progenitor	progenitor_hindbrain development
28	neuron	neuron_peripheral nervous system neuron development	72	progenitor	progenitor_peripheral nervous system neuron development
29	neuron	neuron_synaptic vesicle exocytosis	73	progenitor	progenitor_central nervous system development
30	neuron	neuron_Dopamine Neurotransmitter Release Cycle	74	progenitor	progenitor_transdifferentiation
31	neuron	neuron_neuron development	75	progenitor	progenitor_myeloid cell differentiation
32	neuron	neuron_Neurotransmitter uptake and metabolism In glial cells	76	progenitor	progenitor_neuron development
33	neuron	neuron_neurotransmitter transport	77	progenitor	progenitor_head development
34	neuron	neuron_posterior lateral line neuromast development	78	progenitor	progenitor_posterior lateral line system development
35	neuron	neuron_Transmission across Chemical Synapses	79	progenitor	progenitor_mechanosensory lateral line system development
36	neuron	neuron_Optic neuropathy	80	progenitor	progenitor_posterior lateral line neuromast development
37	neuron	neuron_cytoskeleton of presynaptic active zone	81	progenitor	progenitor_Global developmental delay
38	neuron	neuron_peripheral nervous system neuron axonogenesis	82	progenitor	progenitor_developmental process
39	neuron	neuron_signal release from synapse	83	progenitor	progenitor_regulation of cell differentiation
40	neuron	neuron_neuron differentiation	84	progenitor	progenitor_system development
41	neuron	neuron_Neuronal System	85	progenitor	progenitor_neuron differentiation
42	neuron	neuron_neuron projection development	86	progenitor	progenitor_neuron projection development
43	neuron	neuron_Serotonin Neurotransmitter Release Cycle	87	progenitor	progenitor_erythrocyte differentiation
44	neuron	neuron_presynaptic cytoskeleton	88	progenitor	progenitor_chordate embryonic development

Table 4.9: Full list of filtered GO terms used in GO p-value/proportion heat maps for the scGESTALT dataset invariant tree, see Table 4.5 for the keywords we used to filter these GO terms.

index	GO type	GO term	index	GO type	GO term
1	neuron	neuron_synapse part	58	blood	blood_erythrocyte homeostasis
2	neuron	neuron_neuronal ion channel clustering	59	blood	blood_hemopoiesis
3	neuron	neuron_chemical synaptic transmission	60	blood	blood_Abnormality of the immune system
4	neuron	neuron_neuron maturation	61	blood	blood_myeloid cell development
5	neuron	neuron_vesicle-mediated transport in synapse	62	blood	blood_Abnormal cellular immune system morphology
6	neuron	neuron_postsynapse	63	blood	blood_Hematological neoplasm
7	neuron	neuron_neuron projection guidance	64	blood	blood_Abnormal erythrocyte morphology
8	neuron	neuron_postsynaptic density	65	blood	blood_immune system development
9	neuron	neuron_peripheral nervous system neuron development	66	blood	blood_oxidoreductase activity, acting on a heme group of donors, oxygen as acceptor
10	neuron	neuron_synaptic vesicle membrane	67	blood	blood_hematopoietic or lymphoid organ development
11	neuron	neuron_neuron projection cytoplasm	68	blood	blood_myeloid cell differentiation
12	neuron	neuron_Neurotransmitter release cycle	69	blood	blood_myeloid cell homeostasis
13	neuron	neuron_Acetylcholine Neurotransmitter Release Cycle	70	blood	blood_Cytokine Signaling in Immune system
14	neuron	neuron_synaptic signaling	71	blood	blood_definitive hemopoiesis
15	neuron	neuron_neurotransmitter secretion	72	blood	blood_oxidoreductase activity, acting on a heme group of donors
16	neuron	neuron_neuron projection	73	blood	blood_erythrocyte differentiation
17	neuron	neuron_Norepinephrine Neurotransmitter Release Cycle	74	blood	blood_immune system process
18	neuron	neuron_negative regulation of neuron differentiation	75	progenitor	progenitor_multicellular organism development
19	neuron	neuron_neuronal cell body	76	progenitor	progenitor_brain development
20	neuron	neuron_generation of neurons	77	progenitor	progenitor_cellular developmental process
21	neuron	neuron_presynaptic active zone	78	progenitor	progenitor_Delayed speech and language development
22	neuron	neuron_neurotransmitter receptor complex	79	progenitor	progenitor_RUNX1 regulates genes involved in megakaryocyte differentiation and platelet function
23	neuron	neuron_Neurodevelopmental delay	80	progenitor	progenitor_Developmental Biology
24	neuron	neuron_synapse	81	progenitor	progenitor_peripheral nervous system neuron development
25	neuron	neuron_Neurological speech impairment	82	progenitor	progenitor_erythrocyte development
26	neuron	neuron_regulation of neurotransmitter levels	83	progenitor	progenitor_cell development
27	neuron	neuron_neuron part	84	progenitor	progenitor_embryo development ending in birth or egg hatching
28	neuron	neuron_synaptic vesicle cycle	85	progenitor	progenitor_anatomical structure development
29	neuron	neuron_neurogenesis	86	progenitor	progenitor_myeloid cell development
30	neuron	neuron_peripheral nervous system neuron differentiation	87	progenitor	progenitor_nervous system development
31	neuron	neuron_presynapse	88	progenitor	progenitor_negative regulation of neuron differentiation
32	neuron	neuron_synaptic vesicle	89	progenitor	progenitor_positive regulation of developmental process
33	neuron	neuron_postsynaptic specialization membrane	90	progenitor	progenitor_animal organ development
34	neuron	neuron_synaptic vesicle exocytosis	91	progenitor	progenitor_tissue development
35	neuron	neuron_Dopamine Neurotransmitter Release Cycle	92	progenitor	progenitor_embryo development
36	neuron	neuron_neuron development	93	progenitor	progenitor_immune system development
37	neuron	neuron_Neurotransmitter uptake and metabolism In glial cells	94	progenitor	progenitor_Neurodevelopmental delay
38	neuron	neuron_neurotransmitter transport	95	progenitor	progenitor_Developmental regression
39	neuron	neuron_posterior lateral line neuromast development	96	progenitor	progenitor_peripheral nervous system neuron differentiation
40	neuron	neuron_Transmission across Chemical Synapses	97	progenitor	progenitor_hematopoietic or lymphoid organ development
41	neuron	neuron_Optic neuropathy	98	progenitor	progenitor_hindbrain development
42	neuron	neuron_cytoskeleton of presynaptic active zone	99	progenitor	progenitor_central nervous system development
43	neuron	neuron_peripheral nervous system neuron axonogenesis	100	progenitor	progenitor_regulation of developmental process
44	neuron	neuron_trans-synaptic signaling	101	progenitor	progenitor_myeloid cell differentiation
45	neuron	neuron_postsynaptic density membrane	102	progenitor	progenitor_head development
46	neuron	neuron_Glutamate Neurotransmitter Release Cycle	103	progenitor	progenitor_neuron development
47	neuron	neuron_signal release from synapse	104	progenitor	progenitor_Global developmental delay
48	neuron	neuron_neuron differentiation	105	progenitor	progenitor_posterior lateral line neuromast development
49	neuron	neuron_Neuronal System	106	progenitor	progenitor_developmental process
50	neuron	neuron_neuron projection development	107	progenitor	progenitor_regulation of cell differentiation
51	neuron	neuron_Polyneuropathy	108	progenitor	progenitor_neuron differentiation
52	neuron	neuron_Serotonin Neurotransmitter Release Cycle	109	progenitor	progenitor_neuron projection development
53	neuron	neuron_presynaptic cytoskeleton	110	progenitor	progenitor_cell differentiation
54	neuron	neuron_anterograde trans-synaptic signaling	111	progenitor	progenitor_system development
55	blood	blood_Abnormal immune system morphology	112	progenitor	progenitor_erythrocyte differentiation
56	blood	blood_heme-copper terminal oxidase activity	113	progenitor	progenitor_chordate embryonic development
57	blood	blood_erythrocyte development			

Table 4.10: Full list of filtered GO terms used in GO p-value/proportion heat maps for the ScarTrace dataset, see Table 4.6 for the keywords we used to filter these GO terms.

index	GO type	GO term	index	GO type	GO term
1	immune	immune_immune system process	100	eye	eye_sensory perception of light stimulus
2	immune	immune_Spontaneous hematomas	101	eye	eye_Aplasia/Hypoplasia of the optic nerve
3	immune	immune_hemopoiesis	102	eye	eye_Visual impairment
4	immune	immune_immune system development	103	eye	eye_photoreceptor cell development
5	immune	immune_hematopoietic or lymphoid organ development	104	eye	eye_Chorioretinal degeneration
6	immune	immune_erythrocyte homeostasis	105	eye	eye_photoreceptor cell differentiation
7	immune	immune_erythrocyte differentiation	106	eye	eye_Chorioretinal atrophy
8	immune	immune_myeloid cell differentiation	107	eye	eye_Attenuation of retinal blood vessels
9	immune	immune_myeloid cell homeostasis	108	eye	eye_Reduced visual acuity
10	immune	immune_Intestinal immune network for IgA production	109	eye	eye_photoreceptor cell outer segment organization
11	immune	immune_Abnormal lymphocyte morphology	110	eye	eye_Optic atrophy
12	immune	immune_Abnormal leukocyte count	111	eye	eye_Bone spicule pigmentation of the retina
13	immune	immune_Abnormal lymphocyte count	112	eye	eye_Abnormality of the optic disc
14	immune	immune_Abnormal lymphocyte physiology	113	eye	eye_Abnormal involuntary eye movements
15	immune	immune_leukocyte migration	114	eye	eye_Abnormal posterior eye segment morphology
16	immune	immune_myeloid leukocyte migration	115	eye	eye_Activation of the phototransduction cascade
17	immune	immune_leukocyte chemotaxis	116	eye	eye_Abnormality of the optic nerve
18	immune	immune_Innate Immune System	117	eye	eye_Optic disc hypoplasia
19	immune	immune_chemokine-mediated signaling pathway	118	eye	eye_Visual phototransduction
20	immune	immune_cellular response to chemokine	119	eye	eye_Retinal atrophy
21	immune	immune_response to chemokine	120	eye	eye_Cerebral visual impairment
22	immune	immune_Abnormality of the immune system	121	eye	eye_Optic disc pallor
23	immune	immune_macrophage chemotaxis	122	eye	eye_Abnormal conjugate eye movement
24	immune	immune_cell chemotaxis	123	eye	eye_Abnormality of eye movement
25	immune	immune_regulation of immune system process	124	neuron	neuron_neuron projection
26	immune	immune_Immune System	125	neuron	neuron_neuron part
27	immune	immune_G protein-coupled chemoattractant receptor activity	126	neuron	neuron_Demyelinating peripheral neuropathy
28	immune	immune_chemokine receptor activity	127	neuron	neuron_Polyneuropathy
29	immune	immune_C-C chemokine receptor activity	128	neuron	neuron_neurogenesis
30	immune	immune_chemokine binding	129	neuron	neuron_neuron development
31	immune	immune_C-C chemokine binding	130	neuron	neuron_neuron projection development
32	immune	immune_granulocyte chemotaxis	131	neuron	neuron_neuron differentiation
33	immune	immune_Abnormality of immune system physiology	132	neuron	neuron_generation of neurons
34	immune	immune_negative regulation of immune system process	133	neuron	neuron_Neurotransmitter release cycle
35	immune	immune_leukocyte differentiation	134	neuron	neuron_Transmission across Chemical Synapses
36	immune	immune_T cell activation	135	neuron	neuron_Neuronal System
37	immune	immune_T cell differentiation	136	neuron	neuron_synapse part
38	immune	immune_immune response	137	neuron	neuron_synapse
39	immune	immune_Abnormal immune system morphology	138	neuron	neuron_presynapse
40	immune	immune_Abnormal cellular immune system morphology	139	neuron	neuron_synaptic vesicle membrane
41	immune	immune_Abnormal leukocyte morphology	140	neuron	neuron_synaptic vesicle
42	immune	immune_Erythrocytes take up carbon dioxide and release oxygen	141	neuron	neuron_Sensorimotor neuropathy
43	immune	immune_O2/CO2 exchange in erythrocytes	142	neuron	neuron_neuron projection morphogenesis
44	immune	immune_mast cell activation	143	neuron	neuron_cell morphogenesis involved in neuron differentiation
45	immune	immune_Hematological neoplasm	144	neuron	neuron_neurotransmitter receptor complex
46	immune	immune_Abnormal erythrocyte morphology	145	neuron	neuron_synapse organization
47	immune	immune_embryonic hemopoiesis	146	neuron	neuron_postsynaptic specialization
48	immune	immune_myeloid cell development	147	neuron	neuron_neuron to neuron synapse
49	immune	immune_erythrocyte development	148	neuron	neuron_asymmetric synapse
50	immune	immune_Extramedullary hematopoiesis	149	neuron	neuron_Protein-protein interactions at synapses
51	immune	immune_Abnormality of T cell physiology	150	neuron	neuron_postsynaptic density
52	immune	immune_Hemolytic anemia	151	neuron	neuron_regulation of neurogenesis
53	immune	immune_heme biosynthetic process	152	neuron	neuron_postsynapse
54	immune	immune_heme metabolic process	153	neuron	neuron_postsynaptic density membrane
55	immune	immune_lymphocyte activation	154	neuron	neuron_postsynaptic specialization membrane
56	immune	immune_Autoimmune antibody positivity	155	neuron	neuron_regulation of trans-synaptic signaling
57	immune	immune_Cytokine Signaling in Immune system	156	neuron	neuron_modulation of chemical synaptic transmission
58	immune	immune_Subcutaneous hemorrhage	157	neuron	neuron_Dopamine Neurotransmitter Release Cycle
59	immune	immune_hematopoietic progenitor cell differentiation	158	neuron	neuron_neuronal cell body
60	immune	immune_hemoglobin complex	159	neuron	neuron_positive regulation of synaptic transmission
61	immune	immune_haptoglobin-hemoglobin complex	160	neuron	neuron_Glutamate Neurotransmitter Release Cycle
62	immune	immune_regulation of leukocyte migration	161	neuron	neuron_Upper motor neuron dysfunction
63	immune	immune_Abnormal macrophage morphology	162	neuron	neuron_postsynaptic membrane
64	immune	immune_positive regulation of immune system process	163	neuron	neuron_synaptic membrane
65	immune	immune_chemotaxis	164	neuron	neuron_synaptic signaling
66	immune	immune_regulation of leukocyte differentiation	165	neuron	neuron_Serotonin Neurotransmitter Release Cycle
67	immune	immune_Abnormal myeloid leukocyte morphology	166	neuron	neuron_Norepinephrine Neurotransmitter Release Cycle
68	immune	immune_regulation of hemopoiesis	167	neuron	neuron_Acetylcholine Neurotransmitter Release Cycle
69	immune	immune_Elevated erythrocyte sedimentation rate	168	neuron	neuron_protein localization to postsynaptic membrane
70	immune	immune_Abnormal erythrocyte sedimentation rate	169	neuron	neuron_neurotransmitter receptor transport, postsynaptic endosome to lysosome
71	immune	immune_lymphocyte differentiation	170	neuron	neuron_postsynaptic neurotransmitter receptor diffusion trapping
72	immune	immune_negative regulation of innate immune response	171	neuron	neuron_neurotransmitter receptor diffusion trapping
73	immune	immune_Adaptive Immune System	172	neuron	neuron_protein localization to postsynaptic specialization membrane
74	immune	immune_leukocyte activation	173	neuron	neuron_neurotransmitter receptor localization to postsynaptic specialization membrane
75	immune	immune_neutrophil chemotaxis	174	neuron	neuron_regulation of postsynaptic membrane neurotransmitter receptor levels
76	immune	immune_macrophage migration	175	neuron	neuron_receptor localization to synapse
77	eye	eye_Optic neuropathy	176	neuron	neuron_neuron projection guidance
78	eye	eye_Slow decrease in visual acuity	177	neuron	neuron_Neurodevelopmental abnormality
79	eye	eye_Retinal telangiectasia	178	neuron	neuron_regulation of neuronal synaptic plasticity
80	eye	eye_Leber optic atrophy	179	neuron	neuron_Optic neuropathy
81	eye	eye_Central retinal vessel vascular tortuosity	180	neuron	neuron_Sensory neuropathy
82	eye	eye_Retinal arterial tortuosity	181	neuron	neuron_Peripheral neuropathy
83	eye	eye_Retinal vascular tortuosity	182	neuron	neuron_synaptic transmission, glutamatergic
84	eye	eye_Progressive visual loss	183	neuron	neuron_Abnormality of neuronal migration
85	eye	eye_Abnormal retinal artery morphology	184	neuron	neuron_neuron projection organization
86	eye	eye_Abnormal visual electrophysiology	185	neuron	neuron_Na+/Cl- dependent neurotransmitter transporters
87	eye	eye_Retinal dystrophy	186	neuron	neuron_Neurodevelopmental delay
88	eye	eye_Phototransduction	187	neuron	neuron_regulation of neurotransmitter levels
89	eye	eye_Inactivation, recovery and regulation of the phototransduction cascade	188	neuron	neuron_Neurological speech impairment
90	eye	eye_The phototransduction cascade	189	neuron	neuron_vesicle-mediated transport in synapse
91	eye	eye_Visual field defect	190	neuron	neuron_synaptic vesicle cycle
92	eye	eye_Abnormality of visual evoked potentials	191	neuron	neuron_regulation of synaptic vesicle cycle
93	eye	eye_Visual loss	192	neuron	neuron_trans-synaptic signaling
94	eye	eye_Abnormal retinal vascular morphology	193	neuron	neuron_chemical synaptic transmission
95	eye	eye_Abnormality of retinal pigmentation	194	neuron	neuron_anterograde trans-synaptic signaling
96	eye	eye_Abnormality of the vasculature of the eye	195	neuron	neuron_signal release from synapse
97	eye	eye_Abnormal retinal morphology	196	neuron	neuron_neurotransmitter secretion
98	eye	eye_Constriction of peripheral visual field	197	neuron	neuron_neurotransmitter transport
99	eye	eye_visual perception	198	neuron	neuron_synaptic vesicle exocytosis

Table 4.11: Full list of GO terms and corresponding p-values appearing in invariant clusters but not in any individual clusters for scGESTALT dataset.

GO term	cluster,p-value
neuron projection	(c10,1.16e-02),(c23,1.05e-02),(c20,2.60e-03)
Developmental regression	(c33,2.09e-02),(c30,3.59e-02)
neuron projection development	(c33,8.45e-04)
Acetylcholine Neurotransmitter Release Cycle	(c17,1.39e-02)
erythrocyte homeostasis	(c22,2.02e-05)
Neurotransmitter release cycle	(c23,1.34e-02)
presynaptic active zone	(c33,3.88e-02)
neuron development	(c33,5.65e-04),(c4,6.96e-03)
regulation of neurotransmitter levels	(c4,3.83e-02)
animal organ development	(c32,6.68e-04)
heme-copper terminal oxidase activity	(c10,2.60e-02)
Global developmental delay	(c10,9.72e-03)
embryo development	(c22,6.09e-05)
Abnormal erythrocyte morphology	(c3,8.46e-03)
Polyneuropathy	(c33,9.37e-03)
synaptic vesicle cycle	(c30,1.94e-03)
trans-synaptic signaling	(c30,1.58e-03)
peripheral nervous system neuron development	(c16,2.28e-02)
synapse part	(c20,2.66e-02)
neuron projection cytoplasm	(c10,3.82e-03)
erythrocyte development	(c19,1.49e-02),(c22,4.11e-02),(c32,1.28e-02)
developmental process	(c31,2.49e-04),(c4,1.85e-03)
Serotonin Neurotransmitter Release Cycle	(c17,1.11e-02)
neuron projection guidance	(c4,1.11e-02)
Neurodevelopmental delay	(c10,2.95e-02)
myeloid cell homeostasis	(c3,5.89e-03),(c22,4.34e-05)
myeloid cell development	(c8,3.48e-02),(c32,5.34e-03)
synaptic signaling	(c30,1.75e-03)
Abnormal cellular immune system morphology	(c3,4.18e-02),(c32,1.65e-04)
peripheral nervous system neuron differentiation	(c16,2.28e-02)
Transmission across Chemical Synapses	(c23,4.75e-04)
neurotransmitter transport	(c4,3.61e-02)
synaptic vesicle	(c23,5.08e-04)
regulation of developmental process	(c16,1.72e-02),(c22,4.64e-03)
oxidoreductase activity, acting on a heme group of donors	(c10,2.60e-02)
anterograde trans-synaptic signaling	(c30,1.50e-03)
Optic neuropathy	(c33,8.66e-04)
chemical synaptic transmission	(c30,1.50e-03)
positive regulation of developmental process	(c16,4.12e-02)
Norepinephrine Neurotransmitter Release Cycle	(c17,1.11e-02)
neuronal cell body	(c9,1.14e-02)
myeloid cell differentiation	(c22,1.45e-03)
Abnormal immune system morphology	(c19,2.10e-03),(c32,2.33e-04)
generation of neurons	(c33,3.62e-03)
system development	(c31,1.03e-03),(c8,1.37e-02),(c32,3.78e-02)
signal release from synapse	(c4,1.85e-02)
Hematological neoplasm	(c3,2.01e-03)
neurogenesis	(c33,7.82e-03)
Neurotransmitter uptake and metabolism In glial cells	(c2,1.77e-02)
vesicle-mediated transport in synapse	(c30,1.94e-03)
synapse	(c20,1.49e-02)
embryo development ending in birth or egg hatching	(c31,1.60e-03),(c22,3.11e-04)
presynapse	(c33,4.28e-03),(c23,8.44e-04)
anatomical structure development	(c31,1.32e-04)
Neuronal System	(c23,1.85e-02)
Delayed speech and language development	(c10,3.85e-03)
erythrocyte differentiation	(c22,1.88e-05)
chordate embryonic development	(c31,1.52e-03),(c22,2.98e-04)
neurotransmitter secretion	(c4,1.85e-02)
Neurological speech impairment	(c33,2.01e-02)
oxidoreductase activity, acting on a heme group of donors, oxygen as acceptor	(c10,2.60e-02)
synaptic vesicle exocytosis	(c30,6.84e-03)
definitive hemopoiesis	(c19,3.68e-02)
neuron part	(c20,9.57e-04)
immune system process	(c8,4.81e-03),(c32,9.54e-04)
regulation of cell differentiation	(c16,6.97e-03)

Table 4.12: Full list of GO terms appearing in invariant clusters but not in any individual clusters for ScarTrace dataset.

GO term	cluster,p-value	GO term	cluster,p-value
granulocyte chemotaxis	(c9,9.00e-03)	Hematological neoplasm	(c15,1.22e-02)
Abnormality of the optic disc	(c47,1.02e-04)	postsynaptic membrane	(c7,7.61e-04)
presynapse	(c7,6.46e-05),(c4,2.74e-02),(c22,2.99e-03),(c47,4.78e-03)	neuron development	(c3,1.15e-02)
Neurotransmitter release cycle	(c30,1.19e-03),(c20,1.80e-02)	synaptic vesicle exocytosis	(c21,8.46e-04)
embryonic hemopoiesis	(c44,1.72e-02),(c37,1.03e-02),(c46,4.71e-02)	Subcutaneous hemorrhage	(c29,2.85e-02)
regulation of postsynaptic membrane neurotransmitter receptor levels	(c2,4.26e-02)	Activation of the phototransduction cascade	(c47,1.90e-03)
Abnormal lymphocyte physiology	(c43,8.95e-03)	Abnormal lymphocyte count	(c19,2.73e-02)
Abnormal retinal morphology	(c29,4.22e-02)	Acetylcholine Neurotransmitter Release Cycle	(c21,2.94e-02)
neuron part	(c12,5.05e-03),(c16,4.98e-03)	Sensorimotor neuropathy	(c52,2.34e-02)
response to chemokine	(c19,2.40e-02),(c9,2.88e-03),(c43,2.19e-02)	Optic disc hypoplasia	(c47,1.40e-03)
G protein-coupled chemoattractant receptor activity	(c9,3.05e-03),(c43,4.26e-02)	synapse part	(c12,1.18e-02)
erythrocyte development	(c33,2.67e-02),(c44,2.66e-02)	postsynaptic specialization	(c7,4.86e-02)
Abnormality of retinal pigmentation	(c47,1.19e-02)	cell chemotaxis	(c19,1.23e-03)
lymphocyte activation	(c19,1.48e-02)	Glutamate Neurotransmitter Release Cycle	(c7,5.28e-03)
neurogenesis	(c18,1.60e-02),(c28,3.24e-02)	synaptic signaling	(c7,3.99e-02)
Neurodevelopmental delay	(c12,3.92e-02),(c21,2.47e-02)	Innate Immune System	(c43,4.06e-02)
protein localization to postsynaptic specialization membrane	(c2,1.55e-02)	receptor localization to synapse	(c2,4.98e-02)
positive regulation of synaptic transmission	(c7,4.36e-03)	Autoimmune antibody positivity	(c19,3.98e-02)
synaptic vesicle	(c30,1.76e-04),(c7,5.00e-04),(c22,3.05e-02)	photoreceptor cell outer segment organization	(c31,2.79e-04)
neuron differentiation	(c18,1.22e-02),(c2,2.91e-02)	Protein-protein interactions at synapses	(c7,4.05e-02)
regulation of leukocyte differentiation	(c8,4.79e-02),(c49,4.10e-02)	regulation of immune system process	(c19,4.24e-05)
chemokine binding	(c9,1.67e-03),(c43,2.61e-02)	Abnormality of immune system physiology	(c49,9.30e-04)
neuron projection guidance	(c1,2.33e-02)	neuronal cell body	(c7,8.91e-03)
T cell differentiation	(c19,4.34e-03)	synaptic membrane	(c7,7.33e-04)
positive regulation of immune system process	(c49,2.15e-02)	Visual impairment	(c31,1.52e-02)
C-C chemokine receptor activity	(c9,1.95e-03),(c43,2.96e-02)	generation of neurons	(c18,8.21e-03)
neurotransmitter receptor localization to postsynaptic specialization membrane	(c2,1.55e-02)	visual perception	(c31,3.32e-02)
cell morphogenesis involved in neuron differentiation	(c1,3.50e-02)	Neurological speech impairment	(c21,5.69e-06)
hemopoiesis	(c19,5.14e-03),(c37,2.11e-02),(c15,4.10e-04)	Abnormality of the vasculature of the eye	(c29,3.09e-02)
immune system development	(c19,8.14e-03),(c37,3.18e-02),(c15,6.79e-04)	asymmetric synapse	(c7,4.86e-02)
myeloid leukocyte migration	(c19,2.38e-03),(c9,3.28e-05),(c43,2.74e-04)	Demyelinating peripheral neuropathy	(c11,3.08e-02)
chemokine-mediated signaling pathway	(c19,2.09e-02),(c9,2.46e-03),(c43,1.90e-02)	neutrophil chemotaxis	(c9,6.51e-03)
Visual field defect	(c52,1.62e-02)	regulation of neurogenesis	(c7,4.30e-02)
synaptic vesicle membrane	(c30,3.29e-03),(c7,6.71e-03),(c22,9.14e-03)	neuron to neuron synapse	(c7,4.86e-02)
C-C chemokine binding	(c9,1.67e-03),(c43,2.61e-02)	leukocyte differentiation	(c19,9.16e-03)
leukocyte chemotaxis	(c19,5.48e-03),(c43,6.19e-04)	Abnormal erythrocyte sedimentation rate	(c43,1.65e-02)
neuron projection development	(c3,3.22e-02),(c1,6.08e-06)	neuron projection morphogenesis	(c1,6.23e-03)
regulation of hemopoiesis	(c8,1.10e-02),(c49,8.16e-03)	Optic atrophy	(c47,2.81e-03)
neuron projection	(c30,3.14e-04),(c5,2.79e-03),(c4,3.94e-03),(c16,3.02e-02)	Abnormal erythrocyte morphology	(c15,7.05e-03)
negative regulation of immune system process	(c19,2.98e-02),(c49,2.32e-03)	Erythrocytes take up carbon dioxide and release oxygen	(c46,2.27e-02)
neurotransmitter receptor diffusion trapping	(c2,4.68e-03)	regulation of synaptic vesicle cycle	(c21,4.98e-02)
Abnormal conjugate eye movement	(c13,6.12e-04),(c21,3.33e-02)	Serotonin Neurotransmitter Release Cycle	(c21,2.19e-02)
regulation of neurotransmitter levels	(c5,1.38e-02)	T cell activation	(c19,4.29e-03)
Abnormal leukocyte morphology	(c9,7.62e-03),(c15,1.56e-02)	leukocyte activation	(c9,8.84e-03)
Abnormality of the immune system	(c19,2.22e-02),(c49,3.65e-03),(c43,3.32e-02)	Abnormality of T cell physiology	(c44,3.91e-03)
leukocyte migration	(c19,1.75e-03),(c43,1.52e-03)	synapse organization	(c7,4.83e-02)
postsynaptic neurotransmitter receptor diffusion trapping	(c2,4.68e-03)	Progressive visual loss	(c11,3.96e-02)
macrophage chemotaxis	(c19,2.94e-02),(c43,2.87e-02)	Abnormality of the optic nerve	(c47,3.62e-04)
hematopoietic or lymphoid organ development	(c19,6.58e-03),(c37,2.63e-02),(c15,5.38e-04)	Abnormal macrophage morphology	(c49,4.62e-03)
Neuronal System	(c30,2.28e-02),(c21,5.00e-05),(c20,2.19e-03)	Polyneuropathy	(c11,4.25e-02)
chemokine receptor activity	(c9,3.05e-03),(c43,4.26e-02)	myeloid cell differentiation	(c37,5.16e-04)
myeloid cell development	(c33,2.53e-02),(c44,2.53e-02)	O2/CO2 exchange in erythrocytes	(c46,2.27e-02)
Abnormal cellular immune system morphology	(c9,7.62e-03),(c15,1.56e-02)	Extramedullary hematopoiesis	(c44,1.31e-02)
Abnormal immune system morphology	(c9,1.13e-02),(c15,1.92e-03)	Optic neuropathy	(c11,3.96e-03)
Transmission across Chemical Synapses	(c30,5.90e-04),(c20,9.52e-04)	postsynapse	(c7,2.80e-02)
Abnormal posterior eye segment morphology	(c47,5.63e-03)	Abnormal retinal vascular morphology	(c47,8.80e-03)
neurotransmitter receptor transport, postsynaptic endosome to lysosome	(c2,4.68e-03)	postsynaptic density	(c7,4.34e-02)
cellular response to chemokine	(c19,2.40e-02),(c9,2.88e-03),(c43,2.19e-02)	chemotaxis	(c49,1.45e-02)
Elevated erythrocyte sedimentation rate	(c43,1.65e-02)	immune system process	(c15,8.35e-04)
lymphocyte differentiation	(c19,7.01e-03),(c9,6.54e-03)	protein localization to postsynaptic membrane	(c2,3.69e-03)
Abnormal myeloid leukocyte morphology	(c8,1.98e-02),(c49,1.39e-03)	Abnormality of eye movement	(c21,4.37e-03)

Chapter 5

Applying CSHMM to new biological data

This chapter focuses on the application of our methods to study a key biological problem in lung development. This chapter contains content extracted with changes from our paper [95] published in *Cell Stem Cell*: Hurley, Killian, et al. "Reconstructed Single-Cell Fate Trajectories Define Lineage Plasticity Windows during Differentiation of Human PSC-Derived Distal Lung Progenitors." Cell Stem Cell (2020).

5.1 Introduction

A central aim of developmental biology is to better understand the embryonic differentiation and maturation pathways that lead to functioning adult cells and tissues. Differentiation protocols applied to cultured human pluripotent stem cells (PSC) are designed to recapitulate these pathways in order to produce specific mature target cells. However, even the most optimized PSC differentiation protocols tend to yield a complex, heterogenous mix of cells of varying fates and maturation states, limiting the successful recapitulation of target cell identity or purity [177, 225]. This hurdle makes it challenging to understand the molecular mechanisms underlying human in vivo differentiation and consequently leads to limited clinical relevance and utility for several PSC-derived lineages. To address these issues here we present a general strategy for modelling developmental trajectories that can be used to better understand and improve differentiation protocols. We use a computational algorithm to interrogate the expression kinetics of a subset of genes profiled at high resolution in differentiating PSCs to select a set of optimal time points for global transcriptomic profiling. We apply CSHMM to construct developmental trajectories and to identify the regulators and pathways involved in controlling the process. We then use the computational model to predict both the type and timing of potential interventions which can be used to increase the fraction of cells branching to the desired fate. Finally, we combine lentiviral barcoding with scRNA-seq to validate the parent-progeny lineage relationships and fate bifurcations predicted by our model. The outcome of analyzing learnt CSHMM is a markedly improved understanding of the kinetics, fate trajectories, and cellular plasticity associated with in vitro human PSC directed differentiation, exemplified here by the derivation of lung alveolar epithelial cells from their developmental endodermal precursors.

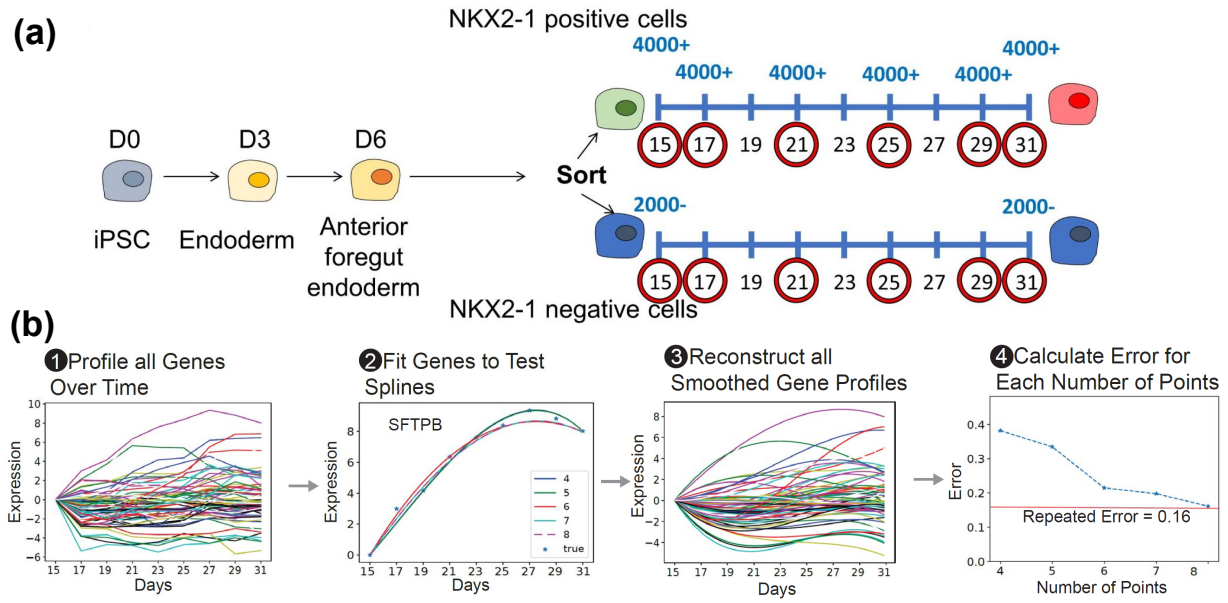

Figure 5.1: (a) The process of generating human lung dataset from BU group. Cells are sampled from day 15 to day 31 for every 2 days and 6 time point are selected based on spline fitting results. (b) Method for choosing the appropriate time points of the single-cell experiment. (1) 66 genes were profiled at high frequency using bulk cultured samples (2) regression splines are fitted in order to (3) model the expression of each gene and (4) iteratively evaluate the effect of removing time-points on the overall error until an optimal (elbow shape) is found.

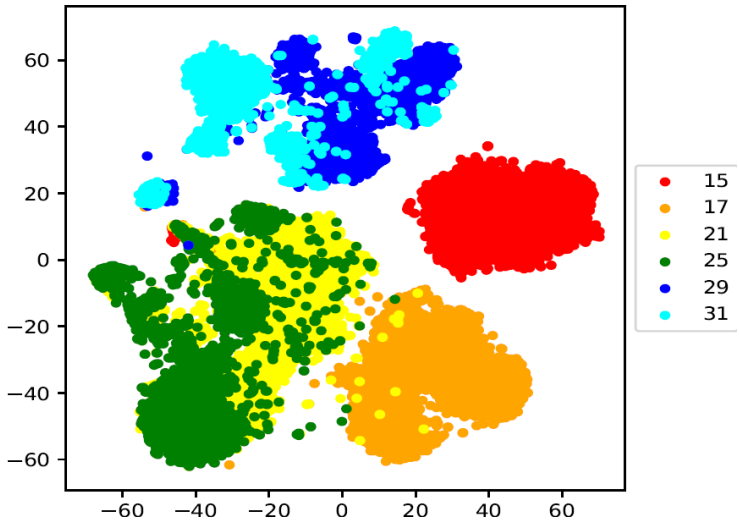

Figure 5.2: The T-SNE plot of the BU human lung dataset. Cells are colored based on measured time points.

5.2 Results

5.2.1 Time point selection for dataset generation

Figure 5.1 shows the process for generating lung dataset from BU group and how we perform time point selection (TPS) algorithm [102]. First, we profile 66 gene expressions for every 2 days between day 17 to day 33. From day 15, cells are separated based on NKX2-1, which is a lung marker, and sampled in triplicate. Based on spline curve fitting results, we select 6 time points for profiling all genes: 15, 17, 21, 25, 29, 31. Figure 5.1 (b) shows the analysis process for time point selection. We profile all the 66 genes and fit spline curve with subset of time points and we can see that an elbow shape happens at 6 time points. We process the data into log2 FPKM (Fragments Per Kilobase Million) format. Genes that expressed in less than 5% of cells are removed. After the preprocessing, we have 16596 cells and 6680 genes. Figure 5.2 shows the t-SNE (t-Distributed Stochastic Neighbor Embedding) plot of the dataset, we can observe that most of cells at 21 and 25 time point are overlapping, and so is 29 and 31. Therefore, we combined the (21, 25) and (29, 31) time points when running our CSHMM analysis.

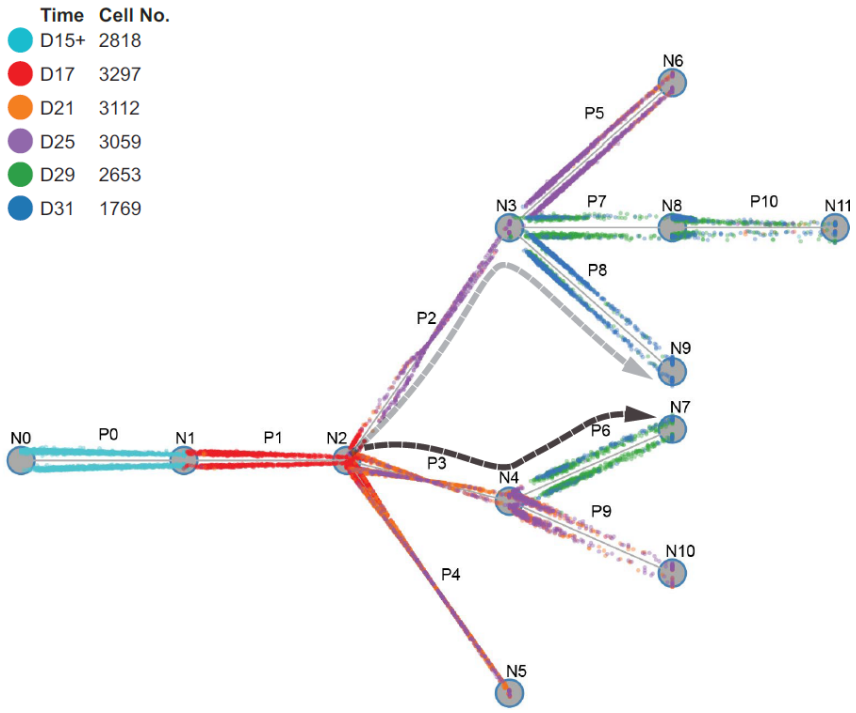

Figure 5.3: The resulting CSHMM model for lung directed differentiation based on scRNA-seq time series data. Each dot represents a cell, color denotes the time point in which the cell was sampled. Nodes are denoted by N0, N1 etc. while branches (paths) are denoted by P0, P1 etc. (note that several branches can share a node). As can be seen, this model predicts that cells remain homogeneous in terms of fate commitments until a point between D15 and D21. They then branch to two major paths, an "upper path" (grey) containing cells with non-lung endoderm and gut markers, and lower paths (black, especially P6) that are associated with cells expressing lung markers.

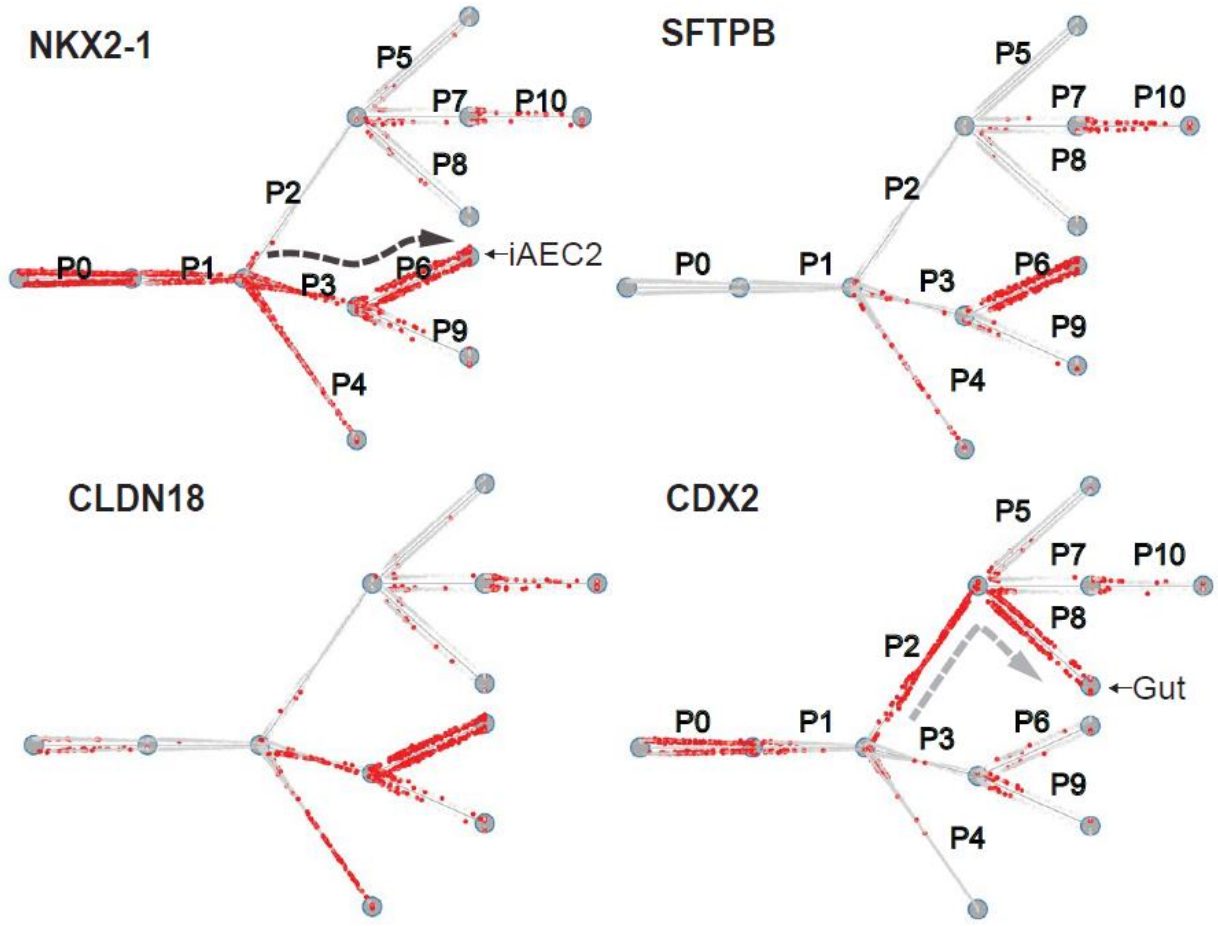

Figure 5.4: The relative expression levels of lung and intestinal markers on CSHMM. Cells are colored red if their expression is greater than a threshold.

5.2.2 CSHMM reconstructs the differentiation path of lung and intestinal cells

Figure 5.3 shows the cell developmental paths reconstructed by CSHMM for BU human lung dataset. Figure 5.4 shows the lung (NKX2-1, CLDN18, SFTPB) and intestinal (CDX2) marker expression on CSHMM. Cells are colored red if their expression pass a threshold. Based on the marker expressions, we think the upper path (P2-P8) corresponds to intestinal path and the lower path (P3-P6) corresponds to the lung path. Calculating the differentially expressed (DE) genes between P3 and P2 based on log2 fold change, we got all of the top 3 DE genes related to WNT signaling: THBS1[86], WIF1[139], and HIPK2[182]. We thus think that WNT signaling pathway might be related to the upper/lower path divergence.

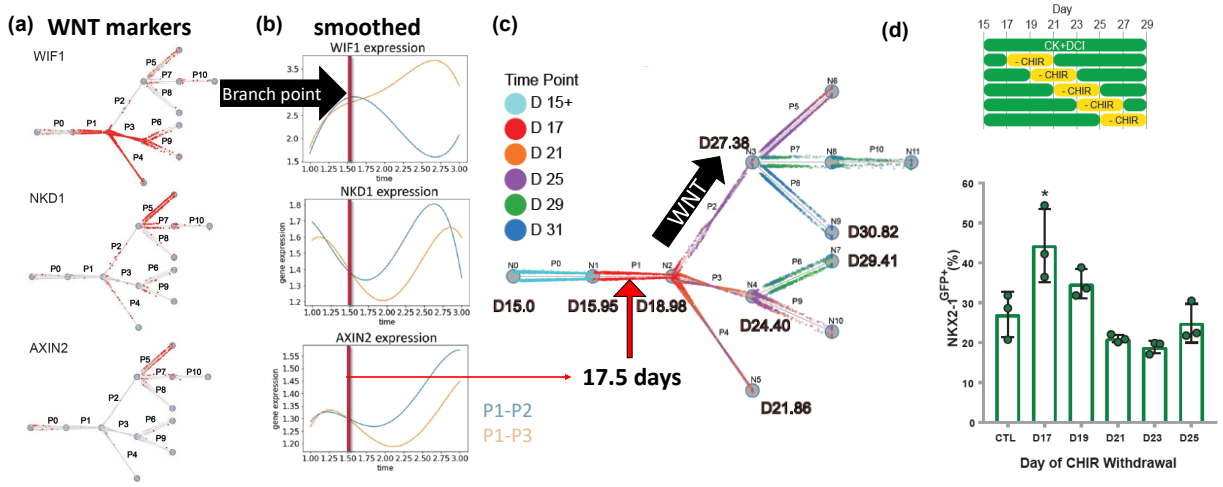

Figure 5.5: The process of CSHMM to predict time for WNT signaling. (a) Expression of key Wnt target genes enriched in upper paths (especially P1-P2), whereas Wnt inhibitory factor, WIF1, is enriched in lower paths (especially P1-P3). (b) To determine the exact time of Wnt pathway activation the continuous expression of these markers is reconstructed using splines to plot the reconstructed expression profiles for the three markers for cells assigned to the top paths (blue curve) vs. bottom paths (orange curve). For all three there is a split in expression values at the halfway point between nodes N1 and N2 (middle of P1). (c) To determine the real time denoted by this point a time is assigned for each node in the CSHMM tree by averaging the profiled times for cells assigned right before and right after this node. Since the two nodes that define P1 are assigned times D15.95 and D18.98 respectively, the middle point between them is D17.5, the predicted split time. (d) Testing the effect of time-dependent downregulation of canonical Wnt signalling by CHIR withdrawal. Retention of distal lung epithelial fate on day 29 of the experiment, measured by the frequency of cells expressing the NKX2-1GFP. Day 17 has the highest retention rate of lung cells in the Chir withdrawal experiments. *: significant difference from control (CTL)

5.2.3 CSHMM predicts the precise timing of Wnt modulation as a determinant of cell fate

Figure 5.5 shows the process of how CSHMM predicts the critical time for WNT signaling. First we plot the smoothed WNT marker expression for the upper and lower paths P1-P2 and P1-P3 (Figure 5.5 (a)(b)) and found that the expression start to diverge around the mid point of N1 and N2 (Figure 5.5 (c)). To find out the real time corresponds to this pseudo time point, we assign each node in the CSHMM model with a real time calculated by nearby cells. For example, the time of N1 is obtained by averaging all the cells assigned to the right half of P0 and the left half of P1, the range is about half of the path near the node. The predicted time of the pseudo time point (marked by a red bar) is 17.5. Since the divergence happens in P1, we assume that 16$\sim$19 days is a reasonable range for critical time for WNT signaling. This assumption is further validated by the WNT activator (Chir) withdrawing experiment shown in Figure 5.5 (d). Specifically, for control (CTL), the Chir is kept in whole time, while in other conditions the Chir is withdrawn for 4 days, starting at day 17, 19, 21, 23, 25. After 4 days the Chir is add back to ensure the the recovery of cell proliferation. The experiment was repeated three times and the value in Figure 5.5 (d) is the averaged value measured at day 29. As we can see, day 17 has the

best retention rate of lung cells (based on lung marker NKX2-1). Day 19 also has the retention rate better than control which is also consistent with our assumption (16∼19 days). This also support our assumption on the single-cell RNA-Seq data that cells are unsynchronized.

5.2.4 lentibarcode data projection further validates the branching time prediction of CSHMM

To further validate that cell fate is not decided at day 17, we perform another experiment that infects cells at day 17 with lentivirus and see if different cell fates could carry the same lentibarcode at day 27. If this is true, then the cell fates is not decided at day 17, then our assumption that the branching happens around day 17.5 and the range is 16∼19 holds. Then, we project the lentibarcode cells back to the CSHMM cell trajectory based on expression to see if the infected cells are assigned on both upper and lower paths. However, most of the cells are assigned to a single path. We thought that this might be the problem of experiment bias. Figure 5.6 shows the tSNE of both BU human lung and the top 5 infected lentibarcode.

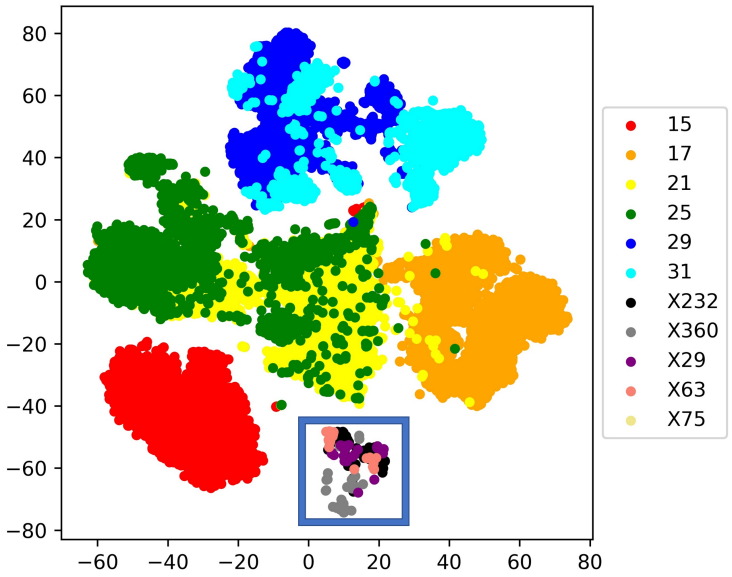

Figure 5.6: tSNE of for both BU human lung dataset and the lentibarcode data. The blue box shows that the lentibarcode forms a separate group and is not similar to any of the BU human lung dataset. The labels starts with "X" are the top lentibarcode with most cells infected.

We can see that the infected cells are distant to the uninfected cells if we use all the genes. To deal with this problem, we project the cells only use a smaller set of genes based on the learnt model. We select the DE genes between P2 and P3, combined with several lung and intestinal markers and redo the cell projection. The result is shown in Figure 5.7 and 5.8. We can see that about 8.1% of cells are assigned to P0 and P1 but we think that this is not a big problem because the dataset is imperfect anyway. Other than that, we found that the infected cells are distributed nearly equally to upper and lower paths. We can also found that a lot of lenti clusters have cells assigned to both upper and lower paths, which validates our assumption that at the

day of infection (17), the cell fate is still undecided. We also calculated the p-value of t-test and ranksum test against random cells to make sure that the cell project result is not random. For generating the random cells, we sample each gene from the distribution of the infected cells to make sure that the random cells have exactly the same gene distribution, just the combination of the genes is different. Then we perform tests on the absolute difference between the maximum probability assigned on the upper path and lower path. See Table 5.1 for the very strong result, this result indicate that our cell projection by CSHMM is not just random assignment. We also perform clonality analysis, calculate the proportion of lung cells for top lenti clusters and show the comparison in Table 5.2. We can see the assignment of CSHMM projection and clonality analysis mostly agrees for top lenti clusters. Specifically, for the lenti cluster with high proportion of lung cells, such as X360, X8, X345, we can observe that the clonality analysis also have high values. For other clusters that CSHMM have low proportion, the values from clonality analysis also agrees with CSHMM.

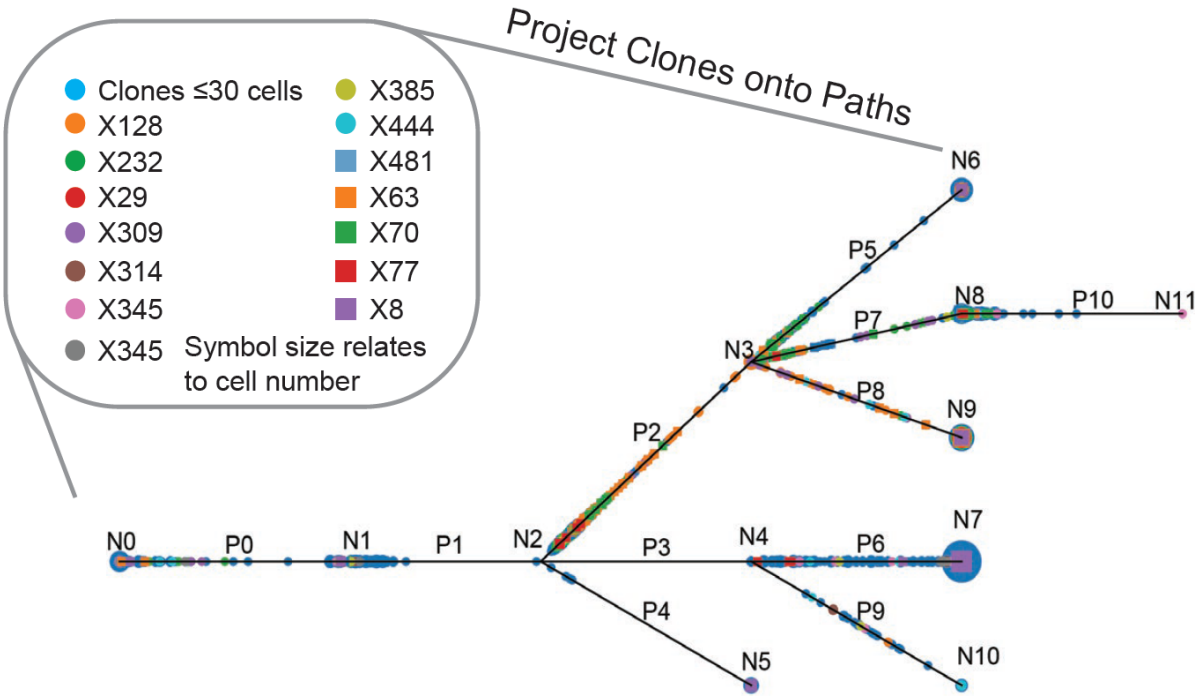

Figure 5.7: lentivirus infected cells projection to CSHMM. Cells are colored/shaped based on individual lentibarcodes indicating clones arising from distinctly tagged individual ancestors. Several large clusters are assigned to both top and bottom paths, validating the bifurcating trajectories predicted by the CSHMM and indicating that cell fate is not fully determined by Day 17.

Table 5.1: t-test and ranksum test of cell projection against different number of random cells

Method—#random cells	**1000**	**2566(#infected cells)**	**10000**
t-test	5.28e-71	6.19e-105	1.30e-124
ranksum test	3.64e-71	1.85e-130	5.05e-204

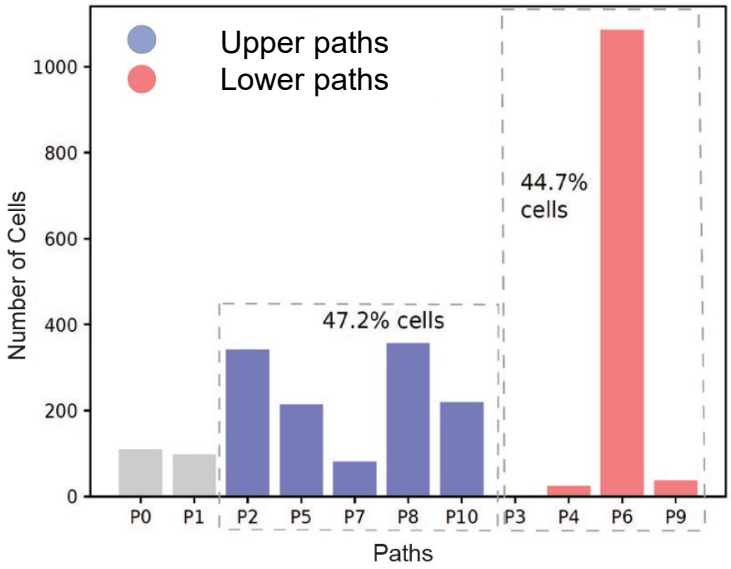

Figure 5.8: Percentage of lentibarcoded cells assigned to top and bottom paths. Similar proportions of cells are assigned to the paths as were seen in the original dataset (without lentiviral infection) indicating that the insertion of the virus did not appreciably impact or bias the differentiation of cells.

Table 5.2: the proportion of lung cells for each lenti cluster comparison between CSHMM projection and clonality analysis

size	lenti cluster	CSHMM projection	clonality analysis
212	X232	0.065326633	0.019323671
79	X360	0.710526316	0.835443038
77	X29	0.142857143	0.053333333
75	X63	0.02739726	0
69	X8	0.742424242	0.835820896
65	X128	0.046153846	0
61	X385	0.052631579	0.016666667
57	X309	0.054545455	0.035714286
40	X70	0.025	0
37	X444	0.142857143	0
34	X481	0.03030303	0.029411765
33	X345	0.793103448	0.696969697
32	X314	1	1
32	X77	0.28125	0.129032258

5.3 Discussion

To improve our understanding of PSC differentiation protocols we developed a new framework that combines experimental design, computational modelling, lentiviral barcoding, and

scRNA-seq profiling. The reconstructed continuous branching models generated by CSHMM provide details about developmental paths for cell fates. Each point in the model can be mapped back to real time enabling the prediction of time specific interventions. We validated several predictions of the model including the multipotency it implies and the timing of a predicted intervention that leads to better retention of the desired cell fate. Our approach can be immediately applied for the analysis of most current or future scRNA-Seq time series datasets, particularly those focused on differentiating stem cells. Unlike most prior methods for reconstruction of trajectories from scRNA-seq data, CSHMM uses a probabilistic model which utilizes all genes to infer cell assignments and branching. Using such model allows the method to overcome noise and internal stochasticity, both hallmarks of stem cell data [53]. Further, the branching type model assumed by CSHMM fits nicely with several stem cell differentiation experiments which attempt to induce one or more specific cell types. The additional ability to assign exact times to changes in expression (using the continuous assignments to reconstruct expression trajectories) and predict factors controlling the branching further enhances the usefulness of CSHMM. The framework we developed which combines predictive computational approaches with cell fate tracing is generalizable. It can be used to further understand and model several other directed differentiation strategies and disease pathogenesis, potentially leading to future cell therapies.

Chapter 6

Conclusion and Future Work

Single-cell lineage tracing is a long-standing open problem in biology. Recent technologies for single-cell RNA sequencing (scRNA-Seq) and inserting artificial markers have been introduced. This greatly increases the resolution of single-cell studies and allows researchers to develop new computational models for reconstructing single-cell lineage to study cell-fate decision.

6.1 Summary of contributions

In this thesis, we have talked about new technologies of profiling single-cell datasets and the challenges of developing models when using the outcomes of such studies. First, for time-series scRNA-Seq dataset, we introduced a new probabilistic model based on the Continuous-State Hidden Markov Model (CSHMM) for inferring continuous cell trajectories for single-cell lineage trees. Then, by extending CSHMM, we developed CSHMM-TF for adding continuous regulatory information to study how TFs interact with each other and affect the cell trajectories. In addition to methods for studying single-cell lineage using only gene expression data, we proposed a new method, LinTIMaT, to utilize both mutation and expression data for better cell lineage trees and for building an invariant cell lineage tree. We also applied CSHMM to a newly generated dataset to improve the protocol for differentiating human PSCs to lung cells.

6.1.1 CSHMM

Previous strategies for modeling single-cell developmental trajectories for time-series scRNA-Seq dataset suffer from disadvantages relying on dimensionality reductions or ordering cells based on limited number of discrete biological states. We developed a probabilistic model based on CSHMM that not only utilizes full expression profiles of single-cells but also provide continuous cell assignments to developmental paths for different cell fates. We formally defined the model as CSHMM and discussed the learning and inference. We applied CSHMM to both simulated and real data. Analysis shows that CSHMM can accurately reconstruct the branching model for cell differentiation process, correctly assign cells with different fates to different paths. Result of marker gene expression profiles of the cell orderings and Spearman correlations

between pseudo time and cell sampled time further support the accuracy of CSHMM.

6.1.2 CSHMM-TF

While TF-gene interactions are important in understanding gene expression, very few models utilize this information to learn parameters and assign precise TF activation time. We present CSHMM-TF which extends from CSHMM to allow continuous TF assignments based on target gene expressions. Analysis shows that CSHMM-TF identifies several key regulating TFs and biological studies support our finding since many of the TFs are known to play important roles in the developmental processes. Ohter identified TFs represent novel predictions about the regulation of specific branching events. Analysis of TF expressions also shows potential combinatorial and causal relationships between TFs assigned to the same developmental path. The TF regulation identified by CSHMM-TF can serve as complimentary information to current analysis methods based on differential expression. Joint analysis can further improve the confidence in the identified TFs.

6.1.3 LinTIMaT

Recent studies are able to insert artificial markers (mutations) to single-cells and at the same time profiles the expression of cells. However, studies still build the cell lineage trees only based on mutations without using expression information. These cell lineage trees suffer from challenges such as noise/saturation in artificial mutations, hundreds of possibly very different candidate trees with similar mutation scores, and different trees for repeated experiments of the same species. We introduce a new statical model, LinTIMaT, which use a global likelihood function that directly combines both expression and mutation information for reconstructing individual and invariant lineage trees. LinTIMaT's also provides a statistical method for inferring cell clusters with coherent cell types based on expressions from the lineage tree. We have tested LinTIMaT on *C. elegans* dataset for which the ground truth is known to validate the underlying assumption of our method: the expression coherence can help overcome mutation data noise. Analysis showing that trees reconstructed by LinTIMaT are not only as good as the best mutation-only lineage trees but also improve the expression coherence, clade homogeneity and functional annotations. With expression information, LinTIMaT can further reconstruct an invariant lineage tree that retains most of the original tree branching for each individual and uncovering more significant biological functions corresponding to different major cell types. To the best of our knowledge, LinTIMaT is the first method to enable the reconstruction of such invariant lineage trees from experiments that simultaneously profile lineage recordings and single-cell transcriptomes. We apply LinTIMaT to zebrafish brain development dataset and illustrates its potential in delineating lineage relationships in complex tissues. LinTIMaT is a seminal computational approach for utilizing both mutation and expression data for reconstructing more accurate and detailed lineages and is compatible with several different related technologies.

6.1.4 CSHMM application to human pluripotent stem cells (PSCs) for improving the protocol for differentiating human PSCs to lung cells

To improve the understanding of PCS differentiation protocols, we utilize CSHMM to help develop a new framework that include experimental design, computational modelling, scRNA-Seq profiling, and lentiviral barcoding. The reconstructed continuous branching models generated by CSHMM provide details about developmental paths for cell dates. Expression analysis of markers based on continuous cell assignments around CSHMM branchings shows pseudo time for the multipotency stage. Each pseudo time point in the model can be mapped back to real time which enables the prediction for time-specific interventions. Results of intervention time predicted by CSHMM leads to better retention of desired cell fate. Projection of lentiviral barcoded single cells onto CSHMM further supports the accuracy for CSHMM predicted time. This framework is generalizable, and it can be used to further understand and model several other directed differentiation strategies and disease pathogenesis, potentially leading to future cell therapies.

6.2 Potential applications to other biological processes

Although this thesis focused on using time series single-cell data to study development, the methods we presented can also be applied to study other biological processes including disease progression and response to stimuli. However, this would require a few changes to some of the underlying assumptions of our model. For example, in some cases we do not expect to see divergence in cell types for disease progression and response but rather a temporal change to the state of the same cell. In such cases, we can restrict CSHMM/CSHMM-TF to only assign one cluster (path) to each time point and generate a single continuous trajectory for the disease. Node and cell assignment can still be used for identifying key regulating TFs and for ordering cells based on their state. Another issue that should be addressed in such models is convergence, for example return to pre-treatment state which can be modeled by allowing cycles in the resulting network as we discuss below.

6.3 Future work

6.3.1 Convergent developmental process

For CSHMM/CSHMM-TF we assumed paths are only able to diverge and so our current formulation is unable to recover convergent developmental processes. In future work we would like to relax the assumption of CSHMM/CSHMM-TF that each path only has one parent. This will allow CSHMM/CSHMM-TF to assign multiple parents for each path leading to convergence. We can introduce a cutoff value of cluster distance during the initialization stage and keep the edges that pass the cutoff. For the cells assigned to the cluster with multiple parents, we can randomly assign them to the additional paths created during initialization and iterate as before for the final assignment.

6.3.2 Utilizing spatial information

The methods discussed in this thesis integrated time-series scRNA-Seq data with (mostly static) TF-gene interaction information and scRNA-Seq data with artificial markers. Recently, a number of new techniques have been developed to obtain spatial scRNA-Seq data [41]. Some of these have also been combined with time series scRNA-SEq to allow the profiling of spatio-temporal datasets [8]. Extending current models to utilize spatial information is an exciting future direction. For this, we will first need to determine a proper data representation for the data being modeled. Possible data representations for single cells could be any of the following: treating individual cells as points with location, expression and time information, creating spatial snapshots for cells frozen in time, or profiling expression at pre-defined fixed locations for every time points [9]. We can select the most appropriate data representation based on the application and the technologies for profiling the dataset. Then, as we mentioned in Chapter 1, measuring the expressions will consume the cells so how to align the spatial information of different time points will become a challange. Methods for this would require the use of specific distance functions for performing the alignment [165]. Clustering will also be essential since it would be hard to align individual cells between time points. Any space-time statistic for finding hotspots [57], methods for finding spatial-temporal points [16], or methods for doing image/video clustering and classification could be applied. Another issues to consider is the scaling and thresholds used for different types of data (as we have done when combining mutation and expression in Chapter 4). To extend CSHMM and CSHMM-TF with spatial information we would need to apply regularization as part of the initialization (only nearby cells can be clustered together), and define a proper likelihood function for spatial information so that we can use the combined likelihood to assign cells and learn parameters by optimizing the new combined likelihood. Moreover, by using similarity/distance functions that properly scaling spatio-temporal information to combine with expression, non-spatio-temporal clustering methods or outlier detection methods can also be applied.

6.3.3 Integrating additional types of data

In the near future, we would have a larger variety of data types available ***at the same time*** to study cell lineage tracing in addition to the datasets we already used in this thesis (bulk-Seq, scRNA-Seq, TF-gene interactions, artificial markers, and spatial information, etc.). For example, recent innovations in single-cell Assay for Transposase Accessible Chromatin sequencing (scATAC-seq) enable the profiling of genome-wide chromatin accessibility for tens of thousands of individual cells. scATAC-seq experiments profile DNA, which leads to more dropouts (1–10% of peaks detected per cell) compared to scRNA-Seq data (10–45% of expressed genes detected per cell). scATAC-seq matrix can be very large (hundreds of thousands of regions) and is extremely sparse, i.e. less than 3% of entries are non-zero [113]. Therefore, imputation of count matrix is another crucial step and a likelihood function that can account for the dropout events is also important. Possible methods for integrating scATAC-Seq and scRNA-Seq are described in [27, 56]. These methods computationally pair scATAC-Seq and scRNA-Seq data by utilizing bulk data as reference, or define coupling matrix for both types of dataset and perform coupling clustering. Currently, CSHMM-TF relies on the expression of targets to identify active TFs. We

can further refine this process based on the information from scATAC-Seq. For example, we can keep the target genes for each time point only if their promoter is open at that time. Also, when using scATAC-Seq dataset we can actively search for new targets for TF using motif and open chromatin information rather than relying on the static interaction input file.

All these emerging data types for single-cells makes multimodal methods more important in single-cell studies. Just as we have described how to include spatial information in the previous section, the first challenge is to represent and summarize the multimodal dataset that can combine both complimentary and redundant information. The heterogeneity in data types and application domains for single-cell makes it challenging to construct a universal multimodal representation. For example, mutations are discrete while expressions are continuous, and embryogenesis studies need to model cell differentiation while disease progression studies usually do not involve differentiation. Common multimodel representation can be joint representations that project unimodal representations together into a reduced dimension space (embedding), which is usually applied when using a neural network [10]. These neural networks can be trained end-to-end, that is, learning both to represent data and perform a particular task simultaneously. The pre-trained representation can be extracted from the hidden layer of the neural network [116] for other task. However, most neural networks cannot handle missing values. Also, due to model complexity, deep neural networks are often hard to train [71] and very data-hungry. Thus, to use deep learning to single-cell studies, larger number of cells will be required.

6.3.4 Larger scale of datasets

New sequencing technologies now allow researchers to profile many more cells and the number is growing exponentially [196]. In some studies the number of cells can reach 10 million [5]. Computational methods to integrate datasets in reduced dimension space are also emerging [190]. This makes it possible to use more complex methods like deep neural network to model cell lineage tracing without overfitting given the size of the training dataset. However, 10 million may still not be enough since neural networks requires samples that scale linearly with the number of free parameters (usually, at least 10 times of the free parameters). Based on this principle, for a 10M dataset with 20K input features (genes), if we want to use a feed-forward neural network with 1 hidden layer for binary classification and avoid overfitting, we can only use at most 50 hidden units (the parameters in the first layer will already be 20K*50 = 1M). To reduce number of free parameters, possible ways are simplifying model structure (in our case, reduce input feature size) or applying regularization to the neural network. Prior works have integrated TF information into neural network to reduce the number of parameters [116]. For reducing input feature size, we can perform feature selections to select most important features (like most variable genes) or project the cells onto a reduced dimension first (like PCA) then use the reduced dimension as the input for neural network [49]. Other than simplifying the neural network structure, there are also techniques that can help mitigate the issue of overfitting such as adding dropout layers, early stopping, weight decay (L2 regularization), sparsity (L1 regularization), and augmenting the dataset. However, while deep learning models are getting more complex over time and the number of features for single-cell datasets is huge (around 20K genes), more cells can certainly help the training of deep neural networks if we want to keep all the genes. In Chapter 2 we have discussed the time complexity aspects for CSHMM. the time complexity for

CSHMM is $O(N * P * G * S)$, where N is the number of cells, P is the number of paths (edges), G is the number of genes, S is the number of sampled points for cell assignments and for learning K. In our current model we have tested 15K cells with 20K genes which takes several hours for each iteration. To improve runtme for larger datasets, we can reduce the number of genes used, reduce the number of sampled points for cell assignments, or sample subset of cells for training and later project all the cells to the model. Also, parallelization for each cell can also help reduce the total running time if we have computing clusters.

These are exciting times for studies related to single cell analysis ans we are hopeful that the methods discussed in this thesis will enable researchers to extract more meaningful information from these large scale datasets.

Bibliography

[1] Phillip L Ainsleigh. Theory of continuous-state hidden markov models and hidden gauss-markov models. 2001. 3.8.1

[2] Phillip L Ainsleigh. Theory of continuous-state hidden markov models and hidden gauss-markov models. Technical report, NAVAL UNDERSEA WARFARE CENTER DIV NEWPORT RI, 2001. 2

[3] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, 1998. 2.10.1

[4] Anna Alemany, Maria Florescu, Chloé S Baron, Josi Peterson-Maduro, and Alexander Van Oudenaarden. Whole-organism clone tracing using single-cell sequencing. *Nature*, 556(7699):108, 2018. 1.1.3, 4, 4.12.3, 4.12.3, 4.13, 4.14.2

[5] Matthew Amodio, David Van Dijk, Krishnan Srinivasan, William S Chen, Hussein Mohsen, Kevin R Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy, et al. Exploring single-cell data with deep multitasking neural networks. *Nature methods*, pages 1–7, 2019. 6.3.4

[6] Olivier Armant, Martin März, Rebecca Schmidt, Marco Ferg, Nicolas Diotel, Raymond Ertzer, Jan Christian Bryne, Lixin Yang, Isabelle Baader, Markus Reischl, et al. Genome-wide, whole mount in situ analysis of transcriptional regulators in zebrafish embryos. *Developmental biology*, 380(2):351–362, 2013. 4.12.3

[7] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25, 2000. (document), 1.3.2, 1.2

[8] Michaela Asp, Stefania Giacomello, Ludvig Larsson, Chenglin Wu, Daniel Fürth, Xiaoyan Qian, Eva Wärdell, Joaquin Custodio, Johan Reimegård, Fredrik Salmén, et al. A spatiotemporal organ-wide gene expression and cell atlas of the developing human heart. *Cell*, 179(7):1647–1660, 2019. 6.3.2

[9] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys (CSUR)*, 51(4):1–41, 2018. 6.3.2

[10] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018. 6.3.3

[11] Ziv Bar-Joseph, Anthony Gitter, and Itamar Simon. Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13 (8):552–564, 2012. 1.1, 3.6.3

[12] Nick Barker, Johan H Van Es, Jeroen Kuipers, Pekka Kujala, Maaike Van Den Born, Miranda Cozijnsen, Andrea Haegebarth, Jeroen Korving, Harry Begthel, Peter J Peters, et al. Identification of stem cells in small intestine and colon by marker gene Lgr5. *Nature*, 449(7165):1003, 2007. 1.1.3

[13] Sean C Bendall, Kara L Davis, El-ad David Amir, Michelle D Tadmor, Erin F Simonds, Tiffany J Chen, Daniel K Shenfeld, Garry P Nolan, and Dana Pe'er. Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development. *Cell*, 157(3):714–725, 2014. 1.1.1

[14] Eva Beuling, Boaz E Aronson, Luc MD Tran, Kelly A Stapleton, Ellis N ter Horst, Laurens ATM Vissers, Michael P Verzi, and Stephen D Krasinski. Gata6 is required for proliferation, migration, secretory cell maturation, and gene expression in the mature mouse colon. *Molecular and cellular biology*, pages MCB–00070, 2012. 3.8.2

[15] Agnieszka B Bialkowska, Vincent W Yang, and Sandeep K Mallipattu. Krüppel-like factors in mammalian stem cells and development. *Development*, 144(5):737–754, 2017. 3.8.2

[16] Derya Birant and Alp Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007. 6.3.2

[17] KU Birkenkamp and PJ Coffer. Regulation of cell survival and proliferation by the foxo (forkhead box, class o) subfamily of forkhead transcription factors, 2003. 3.8.2

[18] Judith A Blake, Carol J Bult, James A Kadin, Joel E Richardson, Janan T Eppig, and Mouse Genome Database Group. The mouse genome database (mgd): premier model organism resource for mammalian genomics and genetics. *Nucleic acids research*, 39 (suppl_1):D842–D848, 2010. 1.3.3, 3.8.1

[19] Maria Boije, Tijana Krajisnik, Yiwen Jiang, Marianne Kastemar, and Lene Uhrbom. Upregulation of sox5 perturbs human glioma cell proliferation and is associated with proneural glioblastoma, 2012. 3.8.2

[20] Olivier Boucherat, Kim Landry-Truchon, Félix-Antoine Bérubé-Simard, Nicolas Houde, Laurent Beuret, Guillaume Lezmi, William D Foulkes, Christophe Delacourt, Jean Charron, and Lucie Jeannotte. Epithelial inactivation of yy1 abrogates lung branching morphogenesis. *Development*, 142(17):2981–2995, 2015. 3.6.2

[21] Stephen T Bradford, Dagmar Wilhelm, Roberto Bandiera, Valerie Vidal, Andreas Schedl, and Peter Koopman. A cell-autonomous role for wt1 in regulating sry in vivo. *Human molecular genetics*, 18(18):3429–3438, 2009. 3.8.2

[22] Sarah Bradshaw, W Jim Zheng, Lam C Tsoi, Gary Gilkeson, and Xian K Zhang. A role for fli-1 in b cell proliferation: implications for sle pathogenesis. *Clinical Immunology*, 129(1):19–30, 2008. 3.8.2

[23] Martin Breuss, Julian Ik-Tsen Heng, Karine Poirier, Guoling Tian, Xavier Hubert Jaglin,

Zhengdong Qu, Andreas Braun, Thomas Gstrein, Linh Ngo, Matilda Haas, et al. Mutations in the β-tubulin gene tubb5 cause microcephaly with structural brain abnormalities. *Cell reports*, 2(6):1554–1562, 2012. 2.10.2

[24] James P Bridges, Angelica Schehr, Yanhua Wang, Liya Huo, Valérie Besnard, Machiko Ikegami, Jeffrey A Whitsett, and Yan Xu. Epithelial scap/insig/srebp signaling regulates multiple biological processes during perinatal lung maturation. *PloS one*, 9(5):e91376, 2014. 3.6.2

[25] Heather A Bruns and Mark H Kaplan. The role of constitutively active stat6 in leukemia and lymphoma. *Critical reviews in oncology/hematology*, 57(3):245–253, 2006. 3.8.2

[26] David Bryant. A classification of consensus methods for phylogenetics. *DIMACS series in discrete mathematics and theoretical computer science*, 61:163–184, 2003. 4.13

[27] Jason D Buenrostro, M Ryan Corces, Caleb A Lareau, Beijing Wu, Alicia N Schep, Martin J Aryee, Ravindra Majeti, Howard Y Chang, and William J Greenleaf. Integrated single-cell analysis maps the continuous regulatory landscape of human hematopoietic differentiation. *Cell*, 173(6):1535–1548, 2018. 6.3.3

[28] Zoë D Burke, Karen R Reed, Sheng-Wen Yeh, Valerie Meniel, Owen J Sansom, Alan R Clarke, and David Tosh. Spatiotemporal regulation of liver development by the wnt/β-catenin pathway. *Scientific reports*, 8(1):2735, 2018. 3.6.5, 3.8.2

[29] Laura Calvo, Begona Anta, Saray López-Benito, Carlos Martín-Rodriguez, Francis S Lee, Pilar Pérez, Dionisio Martín-Zanca, and Juan C Arévalo. Bex3 dimerization regulates ngf-dependent neuronal survival and differentiation by enhancing trka gene transcription. *Journal of Neuroscience*, 35(18):7190–7202, 2015. 2.8.2

[30] Joseph H. Camin and Robert R. Sokal. A Method for Deducing Branching Sequences in Phylogeny. 19(3):311–326, 1965. doi: 10.2307/2406441. URL `http://www.jstor.org/stable/2406441`. 4.2.1

[31] J Gray Camp, Keisuke Sekine, Tobias Gerber, Henry Loeffler-Wirth, Hans Binder, Malgorzata Gac, Sabina Kanton, Jorge Kageyama, Georg Damm, Daniel Seehofer, et al. Multilineage communication regulates human liver bud development from pluripotency. *Nature*, 546(7659):533, 2017. 3.6.1, 3.6.6, 3.8.1

[32] Kieran R Campbell and Christopher Yau. Order under uncertainty: robust differential expression analysis using probabilistic models for pseudotime inference. *PLoS computational biology*, 12(11):e1005212, 2016. 2.8.1, 3.6.5

[33] Zhaodan Cao, Robert M Umek, and Steven L McKnight. Regulated expression of three c/ebp isoforms during adipose conversion of 3t3-l1 cells. *Genes & development*, 5(9): 1538–1552, 1991. 3.6.3

[34] Dong Chen, Chuanzhen Hu, Gen Wen, Qingcheng Yang, Changqing Zhang, and Huilin Yang. Downregulated sox4 expression suppresses cell proliferation, migration, and induces apoptosis in osteosarcoma in vitro and in vivo. *Calcified tissue international*, 102 (1):117–127, 2018. 3.8.2

[35] Fei Chen, Hao Yao, Minjun Wang, Bing Yu, Qinggui Liu, Jianxiu Li, Zhiying He, and Yi-

Ping Hu. Suppressing pitx2 inhibits proliferation and promotes differentiation of ihepscs. *The international journal of biochemistry & cell biology*, 80:154–162, 2016. 3.6.2, 3.7

[36] Nai-Ming Chen, Garima Singh, Alexander Koenig, Geou-Yarh Liou, Peter Storz, Jin-San Zhang, Lisanne Regul, Sankari Nagarajan, Benjamin Kühnemuth, Steven A Johnsen, et al. Nfatc1 links egfr signaling to induction of sox9 transcription and acinar–ductal transdifferentiation in the pancreas. *Gastroenterology*, 148(5):1024–1034, 2015. 3.6.3

[37] Yi-Hsien Chen, Scott M Gianino, and David H Gutmann. Neurofibromatosis-1 regulation of neural stem cell proliferation and multilineage differentiation operates through distinct ras effector pathways. *Genes & development*, 2015. 3.8.2, 3.8.2

[38] Zhou-Feng Chen, Alice J Paquette, and David J Anderson. Nrsf/rest is required in vivo for repression of multiple neuronal target genes during embryogenesis. *Nature genetics*, 20(2):136, 1998. 3.8.2

[39] Yurii Chinenov and Tom K Kerppola. Close encounters of many kinds: Fos-jun interactions that mediate transcription regulatory specificity. *Oncogene*, 20(19):2438, 2001. 3.6.3

[40] ENCODE Project Consortium et al. Identification and analysis of functional elements in 1% of the human genome by the encode pilot project. *Nature*, 447(7146):799, 2007. 1.3.3, 3.8.1

[41] Nicola Crosetto, Magda Bienko, and Alexander Van Oudenaarden. Spatially resolved transcriptomics and beyond. *Nature Reviews Genetics*, 16(1):57–66, 2015. 6.3.2

[42] Edroaldo Lummertz da Rocha, R Grant Rowe, Vanessa Lundin, Mohan Malleshaiah, Deepak Kumar Jha, Carlos R Rambo, Hu Li, Trista E North, James J Collins, and George Q Daley. Reconstruction of complex single-cell trajectories using cellrouter. *Nature communications*, 9(1):892, 2018. 1.1.2, 3.6.5

[43] Yayun Dai, Marie-Pierre Cros, Clément Pontoizeau, Bénédicte Elena-Hermann, Günther K Bonn, and Pierre Hainaut. Downregulation of transcription factor e4f1 in hepatocarcinoma cells: Hbv-dependent effects on autophagy, proliferation and metabolism. *Carcinogenesis*, 35(3):635–650, 2013. 3.8.2

[44] Sara D'Annibale, Jihoon Kim, Roberto Magliozzi, Teck Yew Low, Shabaz Mohammed, Albert JR Heck, and Daniele Guardavaccaro. Proteasome-dependent degradation of transcription factor ap4 (tfap4) controls mitotic division. *Journal of Biological Chemistry*, pages jbc–M114, 2014. 3.8.2

[45] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979. 2.10.1

[46] Maria Agnese Della Fazia, Giuseppe Servillo, and Paolo Sassone-Corsi. Cyclic amp signalling and cellular proliferation: regulation of creb and crem. *FEBS letters*, 410(1): 22–24, 1997. 3.8.2

[47] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016. 2.10.1, 3.8.1

[48] Carmen Díaz-Ruiz, Rosanna Parlato, Fernando Aguado, Jesús M Ureña, Ferran Burgaya, Albert Martínez, Maria A Carmona, Grzegorz Kreiner, Susanne Bleckmann, A Jose, et al. Regulation of neural migration by the creb/crem transcription factors and altered dab1 levels in creb/crem mutants. *Molecular and Cellular Neuroscience*, 39(4):519–528, 2008. 3.8.2

[49] Jiarui Ding, Anne Condon, and Sohrab P Shah. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature communications*, 9(1): 1–13, 2018. 6.3.4

[50] Jun Ding, Bruce Aronow, Naftali Kaminski, Joseph Kitzmiller, Jeffrey Whitsett, and Ziv Bar-Joseph. Reconstructing differentiation networks and their regulation from time series single cell expression data. *Genome Research*, 2018. 2.10.1

[51] Jun Ding, Bruce Aronow, Naftali Kaminski, Joseph Kitzmiller, Jeffrey Whitsett, and Ziv Bar-Joseph. Reconstructing differentiation networks and their regulation from time series single cell expression data. *Genome research*, pages gr–225979, 2018. 1.1.1, 1.1.2, 1.4.2, 2.4, 2.8.3, 3.2, 3.5, 3.6.1, 3.6.5, 3.8.1

[52] Nicolas Diotel, Rebecca Rodriguez Viales, Olivier Armant, Martin März, Marco Ferg, Sepand Rastegar, and Uwe Strähle. Comprehensive expression map of transcription regulators in the adult zebrafish telencephalon reveals distinct neurogenic niches. *Journal of Comparative Neurology*, 523(8):1202–1221, 2015. 4.12.3

[53] Peng Dong and Zhe Liu. Shaping development by stochasticity and dynamics in gene regulation. *Open biology*, 7(5):170030, 2017. 5.3

[54] Jennifer A Doudna and Emmanuelle Charpentier. The new frontier of genome engineering with crispr-cas9. *Science*, 346(6213):1258096, 2014. 1.3.4

[55] Yina Du, Minzhe Guo, Jeffrey A Whitsett, and Yan Xu. 'lunggens': a web-based tool for mapping single-cell gene expression in the developing lung. *Thorax*, 70(11):1092–1094, 2015. 2.8.1

[56] Zhana Duren, Xi Chen, Mahdi Zamanighomi, Wanwen Zeng, Ansuman T Satpathy, Howard Y Chang, Yong Wang, and Wing Hung Wong. Integrative analysis of single-cell genomics data by coupled nonnegative matrix factorizations. *Proceedings of the National Academy of Sciences*, 115(30):7723–7728, 2018. 6.3.3

[57] Emre Eftelioglu, Shashi Shekhar, Dev Oliver, Xun Zhou, Michael R Evans, Yiqun Xie, James M Kang, Renee Laubscher, and Christopher Farah. Ring-shaped hotspot detection: A summary of results. In *2014 IEEE International Conference on Data Mining*, pages 815–820. IEEE, 2014. 6.3.2

[58] Manal A Eid, M Vijay Kumar, Kenneth A Iczkowski, David G Bostwick, and Donald J Tindall. Expression of early growth response genes in human prostate cancer. *Cancer research*, 58(11):2461–2468, 1998. 3.8.2

[59] Jason Ernst, Oded Vainas, Christopher T Harbison, Itamar Simon, and Ziv Bar-Joseph. Reconstructing dynamic regulatory maps. *Molecular systems biology*, 3(1):74, 2007. 1.3.3, 3.2, 3.8.1

[60] Jason Ernst, Heather L Plasterer, Itamar Simon, and Ziv Bar-Joseph. Integrating multiple evidence sources to predict transcription factor binding in the human genome. *Genome research*, 2010. 1.3.3, 3.8.1

[61] Jeffrey A Farrell, Yiqun Wang, Samantha J Riesenfeld, Karthik Shekhar, Aviv Regev, and Alexander F Schier. Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. *Science*, 360(6392):eaar3131, 2018. 2.8

[62] Pietro Fazzari, An Snellinx, Victor Sabanov, Tariq Ahmed, Lutgarde Serneels, Annette Gartner, S Ali M Shariati, Detlef Balschun, and Bart De Strooper. Cell autonomous regulation of hippocampal circuitry via aph1b-γ-secretase/neuregulin 1 signalling. *Elife*, 3: e02196, 2014. 2.10.2

[63] Joseph Felsenstein. *PHYLIP (phylogeny inference package), version 3.5 c*. Joseph Felsenstein., 1993. 1.1.3, 4.13

[64] Joseph Felsenstein and Joseph Felenstein. *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004. 4.3

[65] Walter M Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416, 1971. 4.2.1

[66] Per Flodby, Changgong Li, Yixin Liu, Hongjun Wang, Megan E Rieger, Parviz Minoo, Edward D Crandall, David K Ann, Zea Borok, and Beiyun Zhou. Cell-specific expression of aquaporin-5 (aqp5) in alveolar epithelium is directed by gata6/sp1 via histone acetylation. *Scientific reports*, 7(1):3473, 2017. 3.6.2

[67] Victoria C Foletta, David H Segal, and Donna R Cohen. Transcriptional regulation in the immune system: all roads lead to ap-1. *Journal of Leukocyte Biology*, 63(2):139–152, 1998. 3.8.2

[68] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008. 3.8.1

[69] Dan Frumkin, Adam Wasserstrom, Shai Kaplan, Uriel Feige, and Ehud Shapiro. Genomic variability within an organism exposes its cell lineage tree. *PLoS computational biology*, 1(5):e50, 2005. 1.1.3

[70] Veronica Garcia-Carpizo, Sergio Ruiz-Llorente, Jacinto Sarmentero, Osvaldo Graña-Castro, David G Pisano, and Maria J Barrero. Crebbp/ep300 bromodomains are critical to sustain the gata1/myc regulatory axis in proliferation. *Epigenetics & chromatin*, 11(1): 30, 2018. 3.8.2

[71] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 6.3.3

[72] Orit Goldman, Idan Cohen, and Valerie Gouon-Evans. Functional blood progenitor markers in developing human liver progenitors. *Stem cell reports*, 7(2):158–166, 2016. 3.6.2

[73] Wuming Gong, Il-Youp Kwak, Pruthvi Pota, Naoko Koyano-Nakagawa, and Daniel J Garry. DrImpute: imputing dropout events in single cell RNA sequencing data. *BMC bioinformatics*, 19(1):220, 2018. 4.11, 4.14.1

[74] Miriam Gordillo, Todd Evans, and Valerie Gouon-Evans. Orchestrating liver development. *Development*, 142(12):2094–2108, 2015. 3.8.2

[75] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`. 2.10.1, 3.8.1

[76] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014. 2.10.1, 3.8.1

[77] Lloyd A Greene, Hae Young Lee, and James M Angelastro. The transcription factor atf5: role in neurodevelopment and neural tumors. *Journal of neurochemistry*, 108(1):11–22, 2009. 3.8.2

[78] Alex Greenfield, Christoph Hafemeister, and Richard Bonneau. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics*, 29(8):1060–1067, 2013. 3.6.1

[79] Dominic Grün, Anna Lyubimova, Lennart Kester, Kay Wiebrands, Onur Basak, Nobuo Sasaki, Hans Clevers, and Alexander van Oudenaarden. Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*, 525(7568):251, 2015. 1.1

[80] Francesco Gualdrini, Cyril Esnault, Stuart Horswell, Aengus Stewart, Nik Matthews, and Richard Treisman. Srf co-factors control the balance between cell proliferation and contractility. *Molecular cell*, 64(6):1048–1061, 2016. 3.8.2

[81] Jing Guo and Jie Zheng. Hopland: single-cell pseudotime recovery using continuous hopfield network-based modeling of waddington's epigenetic landscape. *Bioinformatics*, 33(14):i102–i109, 2017. 1.1.2, 3.6.5

[82] Lucas Gutierrez, Alexander Yoon, Kym Francis Faull, and Edith Porter. Induction of innate immune factor expression in a549 lung alveolar type ii cells by cytokines interferon-g and tumor necrosis factor-α, 2016. 2.8.1

[83] Adam L Haber, Moshe Biton, Noga Rogel, Rebecca H Herbst, Karthik Shekhar, Christopher Smillie, Grace Burgin, Toni M Delorey, Michael R Howitt, Yarden Katz, et al. A single-cell survey of the small intestinal epithelium. *Nature*, 551(7680):333, 2017. 1.1

[84] Laleh Haghverdi, Maren Buettner, F Alexander Wolf, Florian Buettner, and Fabian J Theis. Diffusion pseudotime robustly reconstructs lineage branching. *Nature methods*, 13(10):845, 2016. 1.1.1

[85] Keren Bahar Halpern, Rom Shenhav, Orit Matcovitch-Natan, Beáta Tóth, Doron Lemze, Matan Golan, Efi E Massasa, Shaked Baydatch, Shanie Landen, Andreas E Moor, et al. Single-cell spatial reconstruction reveals global division of labour in the mammalian liver. *Nature*, 542(7641):352, 2017. 1.1

[86] Bo Han, Szu-Yu Chen, Ying-Ting Zhu, and Scheffer CG Tseng. Integration of bmp/wnt signaling to control clonal growth of limbal epithelial progenitor cells by niche cells. *Stem cell research*, 12(2):562–573, 2014. 5.2.2

[87] Xiaofeng Han, Peng Zhang, Rong Jiang, Fei Xia, Meiling Li, and Feng-Jin Guo. Explore on the effect of atf6 on cell growth and apoptosis in cartilage development. *Histochemistry and cell biology*, 142(5):497–509, 2014. 3.8.2

[88] Qianyun Hao, Xuesong Zhao, Yi Zhang, Ziming Dong, Tao Hu, and Ping Chen. Targeting overexpressed activating transcription factor 1 (atf1) inhibits proliferation and migration and enhances sensitivity to paclitaxel in esophageal cancer cells. *Medical science monitor basic research*, 23:304, 2017. 3.8.2

[89] Daisuke Hasegawa, Veronica Calvo, Alvaro Avivar-Valderas, Abigale Lade, Hsin-I Chou, Youngmin A Lee, Eduardo F Farias, Julio A Aguirre-Ghiso, and Scott L Friedman. Epithelial xbp1 is required for cellular proliferation and differentiation during mammary gland development. *Molecular and cellular biology*, pages MCB–00136, 2015. 3.8.2

[90] Kim E Haworth, Surendra Kotecha, Timothy J Mohun, and Branko V Latinkic. Gata4 and gata5 are essential for heart and liver development in xenopus embryos. *BMC developmental biology*, 8(1):74, 2008. 3.8.2

[91] Martin Hedegaard, Christian Matthäus, Søren Hassing, Christoph Krafft, Max Diem, and Jürgen Popp. Spectral unmixing and clustering algorithms for assessment of single cells by raman microscopic imaging. *Theoretical Chemistry Accounts*, 130(4-6):1249–1260, 2011. 2.10.1

[92] Kristian Helin. Regulation of cell proliferation by the e2f transcription factors. *Current opinion in genetics & development*, 8(1):28–35, 1998. 3.8.2

[93] Katherine A Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM, 2005. 1.4.3, 4, 4.1, 4.2.2, 4.2.2

[94] Mo Huang, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang. SAVER: gene expression recovery for single-cell RNA sequencing. *Nature methods*, 15(7):539, 2018. 4.14.1

[95] Killian Hurley, Jun Ding, Carlos Villacorta-Martin, Michael J Herriges, Anjali Jacob, Marall Vedaie, Konstantinos D Alysandratos, Yuliang L Sun, Chieh Lin, Rhiannon B Werder, et al. Reconstructed single-cell fate trajectories define lineage plasticity windows during differentiation of human psc-derived distal lung progenitors. *Cell Stem Cell*, 2020. 5

[96] Paola Indovina, Francesca Pentimalli, Nadia Casini, Immacolata Vocca, and Antonio Giordano. Rb1 dual role in proliferation and apoptosis: cell fate control and implications for cancer therapy. *Oncotarget*, 6(20):17873, 2015. 3.8.2

[97] Agnieszka Jezierska-Drutel, Steven A Rosenzweig, and Carola A Neumann. Role of oxidative stress and the microenvironment in breast cancer development and progression. In *Advances in cancer research*, volume 119, pages 107–125. Elsevier, 2013. 3.8.2

[98] Peter F Johnson. Molecular stop signs: regulation of cell-cycle arrest by c/ebp transcription factors. *Journal of cell science*, 118(12):2545–2555, 2005. 3.8.2

[99] Young Seok Ju, Inigo Martincorena, Moritz Gerstung, Mia Petljak, Ludmil B Alexandrov,

Raheleh Rahbari, David C Wedge, Helen R Davies, Manasa Ramakrishna, Anthony Fullam, et al. Somatic mutations reveal asymmetric cellular dynamics in the early human embryo. *Nature*, 543(7647):714, 2017. 1.1.3

[100] Lennart Kester and Alexander van Oudenaarden. Single-cell transcriptomics meets lineage tracing. *Cell Stem Cell*, 2018. 1.1, 1.1.3, 4.13

[101] Peter V Kharchenko, Lev Silberstein, and David T Scadden. Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11(7):740–742, 2014. 2.8.3, 2.10.2

[102] Michael Kleyman, Emre Sefer, Teodora Nicola, Celia Espinoza, Divya Chhabra, James S Hagood, Naftali Kaminski, Namasivayam Ambalavanan, and Ziv Bar-Joseph. Selecting the most appropriate time points to profile in high-throughput studies. *Elife*, 6:e18541, 2017. 5.2.1

[103] Jay D Kormish, Débora Sinner, and Aaron M Zorn. Interactions between sox factors and wnt/β-catenin signaling in development and disease. *Developmental Dynamics*, 239(1): 56–68, 2010. 3.6.3

[104] Chia-Chen Ku, Hitomi Hasegawa, Chang-Shen Lin, Ming-Ho Tsai, Kenly Wuputra, Richard Eckner, Naoto Yamaguchi, and Kazunari K Yokoyama. Control of the cell cycle and mitosis by phosphorylated activating transcription factor 2 and its homologue 7. *Journal of Nature and Science*, 1(4):e74, 2015. 3.8.2

[105] Sophie Kusy, Bastien Gerby, Nicolas Goardon, Nathalie Gault, Federica Ferri, Delphine Gérard, Florence Armstrong, Paola Ballerini, Jean-Michel Cayuela, André Baruchel, et al. Nkx3. 1 is a direct tal1 target gene that mediates proliferation of tal1-expressing human t cell acute lymphoblastic leukemia. *Journal of Experimental Medicine*, pages jem–20100745, 2010. 3.8.2

[106] Atsushi Kuwahara, Hiroshi Sakai, Yuanjiang Xu, Yasuhiro Itoh, Yusuke Hirabayashi, and Yukiko Gotoh. Tcf3 represses wnt–β-catenin signaling and maintains neural stem cell population during neocortical development. *PloS one*, 9(5):e94408, 2014. 3.8.2

[107] Clemens Lakner, Paul Van Der Mark, John P Huelsenbeck, Bret Larget, and Fredrik Ronquist. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Systematic biology*, 57(1):86–103, 2008. 4.4

[108] John Le Lay and Klaus H Kaestner. The fox genes in the liver: from organogenesis to functional integration. *Physiological reviews*, 90(1):1–22, 2010. 3.6.2

[109] Janice S Lee, William O Ward, Geremy Knapp, Hongzu Ren, Beena Vallanat, Barbara Abbott, Karen Ho, Seth J Karp, and J Christopher Corton. Transcriptional ontogeny of the developing liver. *BMC genomics*, 13(1):33, 2012. 3.6.2

[110] Xiao Li, Sergio Florez, Jianbo Wang, Huojun Cao, and Brad A Amendt. Dact2 represses pitx2 transcriptional activation and cell proliferation through wnt/beta-catenin signaling during odontogenesis. *PloS one*, 8(1):e54868, 2013. 3.8.2

[111] Xu Li, Wenqi Wang, Yuanxin Xi, Min Gao, MyKim Tran, Kathryn E Aziz, Jun Qin, Wei Li, and Junjie Chen. Foxr2 interacts with myc to promote its transcriptional activities and tumorigenesis. *Cell reports*, 16(2):487–497, 2016. 3.8.2

[112] Yongzhe Li, Jianjiao Wang, Yongri Zheng, Yan Zhao, Mian Guo, Yang Li, Qiuli Bao, Yu Zhang, Lizhuang Yang, and Qingsong Li. Sox11 modulates neocortical development by regulating the proliferation and neuronal differentiation of cortical intermediate precursors. *Acta Biochim Biophys Sin*, 44(8):660–668, 2012. 3.8.2, 3.8.2

[113] Zhijian Li, Christoph Kuppe, Mingbo Cheng, Sylvia Menzel, Martin Zenke, Rafael Kramann, and Ivan G Costa. scopen: chromatin-accessibility estimation of single-cell atac data. *bioRxiv*, page 865931, 2019. 6.3.3

[114] Zhongyou Li, Paul F Szurek, Chuantao Jiang, Annie Pao, Brian Bundy, Wei-dong Le, Allan Bradley, and Y Eugene Yu. Neuronal differentiation of nte-deficient embryonic stem cells. *Biochemical and biophysical research communications*, 330(4):1103–1109, 2005. 2.10.2

[115] Chieh Lin and Ziv Bar-Joseph. Continuous State HMMs for Modeling Time Series Single Cell RNA-Seq Data. *Bioinformatics*, 2019. doi: 10.1093/bioinformatics/btz296. URL `https://doi.org/10.1093/bioinformatics/btz296`. 2, 3.5, 3.8.1, 3.8.1, 4.2.2

[116] Chieh Lin, Siddhartha Jain, Hannah Kim, and Ziv Bar-Joseph. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic acids research*, 45(17): e156–e156, 2017. 4.10, 6.3.3, 6.3.4

[117] Chieh Lin, Jun Ding, and Ziv Bar-Joseph. Inferring tf activation order in time series scrna-seq studies. *PLOS Computational Biology*, 16(2):e1007644, 2020. 3

[118] Kimberly C Lin, Hyun Woo Park, and Kun-Liang Guan. Regulation of the hippo pathway transcription factor tead. *Trends in biochemical sciences*, 2017. 3.8.2

[119] Tien-ho Lin, Naftali Kaminski, and Ziv Bar-Joseph. Alignment and classification of time series gene expression in clinical studies. *Bioinformatics*, 24(13):i147–i155, 2008. 2.1

[120] Tapio Lönnberg, Valentine Svensson, Kylie R James, Daniel Fernandez-Ruiz, Ismail Sebina, Ruddy Montandon, Megan SF Soon, Lily G Fogg, Arya Sheela Nair, Urijah Liligeto, et al. Single-cell rna-seq and computational analysis using temporal mixture modelling resolves th1/tfh fate bifurcation in malaria. *Science immunology*, 2(9), 2017. 1.1.1

[121] Paul PY Lu and Narendrakumar Ramanan. Serum response factor is required for cortical axon growth but is dispensable for neurogenesis and neocortical lamination. *Journal of Neuroscience*, 31(46):16651–16664, 2011. 3.8.2

[122] Tongyi Lu, Binhua Wu, Yunfei Yu, Wenhui Zhu, Simin Zhang, Yinmei Zhang, Jiaying Guo, and Ning Deng. Blockade of onecut2 expression in ovarian cancer inhibited tumor cell proliferation, migration, invasion and angiogenesis. *Cancer science*, 109(7):2221, 2018. 3.8.2

[123] Yihui Luan and Hongzhe Li. Clustering of time-course gene expression data using a mixed-effects model with B-splines. *Bioinformatics*, 19(4):474–482, 2003. 4.2.2

[124] Eugenio Marco, Robert L Karp, Guoji Guo, Paul Robson, Adam H Hart, Lorenzo Trippa, and Guo-Cheng Yuan. Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proceedings of the National Academy of Sciences*, 111(52):E5643–

E5650, 2014. 1.1.1

[125] Sabrina Margagliotti, Frédéric Clotman, Christophe E Pierreux, Jean-Bernard Beaudry, Patrick Jacquemin, Guy G Rousseau, and Frédéric P Lemaigre. The onecut transcription factors hnf-6/oc-1 and oc-2 regulate early liver expansion by controlling hepatoblast migration. *Developmental biology*, 311(2):579–589, 2007. 3.6.2, 3.6.5

[126] Jeffrey L Mason, James M Angelastro, Tatyana N Ignatova, Valery G Kukekov, Grace Lin, Lloyd A Greene, and James E Goldman. Atf5 regulates the proliferation and differentiation of oligodendrocytes. *Molecular and Cellular Neuroscience*, 29(3):372–380, 2005. 3.8.2

[127] Volker Matys, Olga V Kel-Margoulis, Ellen Fricke, Ines Liebich, Sigrid Land, A Barre-Dirrie, Ingmar Reuter, D Chekmenev, Mathias Krull, Klaus Hornischer, et al. Transfac® and its module transcompel®: transcriptional gene regulation in eukaryotes. *Nucleic acids research*, 34(suppl_1):D108–D110, 2006. 1.3.3, 3.8.1

[128] Christian Mayer, Christoph Hafemeister, Rachel C Bandler, Robert Machold, Renata Batista Brito, Xavier Jaglin, Kathryn Allaway, Andrew Butler, Gord Fishell, and Rahul Satija. Developmental diversification of cortical inhibitory interneurons. *Nature*, 555(7697):457, 2018. 3.6.1, 3.6.6, 3.8.1

[129] Aaron McKenna, Gregory M Findlay, James A Gagnon, Marshall S Horwitz, Alexander F Schier, and Jay Shendure. Whole-organism lineage tracing by combinatorial and cumulative genome editing. *Science*, 353(6298):aaf7907, 2016. 1.1.3

[130] Yuichi Miura, Mizuho Morooka, Nicolas Sax, Rahul Roychoudhuri, Ari Itoh-Nakadai, Andrey Brydun, Ryo Funayama, Keiko Nakayama, Susumu Satomi, Mitsuyo Matsumoto, et al. Bach2 promotes b cell receptor–induced proliferation of b lymphocytes and represses cyclin-dependent kinase inhibitors. *The Journal of Immunology*, page ji1601863, 2018. 3.8.2

[131] Yasuhiro Mochizuki and Philip W Majerus. Characterization of myotubularin-related protein 7 and its binding partner, myotubularin-related protein 9. *Proceedings of the National Academy of Sciences*, 100(17):9768–9773, 2003. 2.8.2

[132] Dylan Mooijman, Siddharth S Dey, Jean-Charles Boisset, Nicola Crosetto, and Alexander Van Oudenaarden. Single-cell 5hmC sequencing reveals chromosome-wide cell-to-cell variability and enables lineage reconstruction. *Nature biotechnology*, 34(8):852, 2016. 1.1.3

[133] Thomas Mueller and Mario Wullimann. *Atlas of early zebrafish brain development: a tool for molecular neurogenetics*. Academic Press, 2015. 4.12.3

[134] Heiko Müller, Adrian P Bracken, Richard Vernell, M Cristina Moroni, Fred Christians, Emanuela Grassilli, Elena Prosperini, Elena Vigo, Jonathan D Oliner, and Kristian Helin. E2fs regulate the expression of genes involved in differentiation, development, proliferation, and apoptosis. *Genes & development*, 15(3):267–285, 2001. 3.6.3

[135] Kevin P Murphy. Conjugate Bayesian analysis of the Gaussian distribution. *def*, 1(2σ2): 16, 2007. 4.14.1

[136] Victor Muthu, Helen Eachus, Pam Ellis, Sarah Brown, and Marysia Placzek. Rx3 and Shh direct anisotropic growth and specification in the zebrafish tuberal/anterior hypothalamus. *Development*, 143(14):2651–2663, 2016. 4.12.3

[137] Shalin H Naik, Leïla Perié, Erwin Swart, Carmen Gerlach, Nienke van Rooij, Rob J de Boer, and Ton N Schumacher. Diverse and heritable lineage imprinting of early haematopoietic progenitors. *Nature*, 496(7444):229, 2013. 1.1.3

[138] Atsushi Nakamura, Risa Ebina-Shibuya, Ari Itoh-Nakadai, Akihiko Muto, Hiroki Shima, Daisuke Saigusa, Junken Aoki, Masahito Ebina, Toshihiro Nukiwa, and Kazuhiko Igarashi. Transcription repressor bach2 is required for pulmonary surfactant homeostasis and alveolar macrophage function. *Journal of Experimental Medicine*, pages jem–20130028, 2013. 3.6.2

[139] R CL Ng, D Matsumaru, A SH Ho, MM Garcia-Barceló, ZW Yuan, D Smith, L Kodjabachian, P KH Tam, G Yamada, and V CH Lui. Dysregulation of wnt inhibitory factor 1 (wif1) expression resulted in aberrant wnt-β-catenin signaling and cell death of the cloaca endoderm, and anorectal malformations. *Cell death and differentiation*, 21(6):978, 2014. 5.2.2

[140] Linh Ngo, Matilda Haas, Zhengdong Qu, Shan Shan Li, Jennifer Zenker, Kathleen Sue Lyn Teng, Jenny Margaret Gunnersen, Martin Breuss, Mark Habgood, David Anthony Keays, et al. Tubb5 and its disease-associated mutations influence the terminal differentiation and dendritic spine densities of cerebral cortical neurons. *Human molecular genetics*, 23(19):5147–5158, 2014. 2.10.2

[141] Andrea Ocone, Laleh Haghverdi, Nikola S Mueller, and Fabian J Theis. Reconstructing gene regulatory dynamics from high-dimensional single-cell snapshot data. *Bioinformatics*, 31(12):i89–i96, 2015. 2.10.1

[142] Colin E Olsen, Brant E Isakson, Gregory J Seedorf, Richard L Lubman, and Scott Boitano. Extracellular matrix-driven alveolar epithelial cell differentiation in vitro. *Experimental lung research*, 31(5):461–482, 2005. 2.8.1

[143] Massimiliano Paganelli, Omar Nyabi, Brice Sid, Jonathan Evraerts, Imane El Malmi, Yves Heremans, Laurent Dollé, Carley Benton, Pedro-Buc Calderon, Leo van Grunsven, et al. Downregulation of sox9 expression associates with hepatogenic differentiation of human liver mesenchymal stem/progenitor cells. *Stem cells and development*, 23(12): 1377–1391, 2014. 3.6.2, 3.6.5

[144] Alice Parisi, Floriane Lacour, Lorenzo Giordani, Sabine Colnot, Pascal Maire, and Fabien Le Grand. Apc is required for muscle stem cell proliferation and skeletal muscle tissue repair. *J Cell Biol*, pages jcb–201501053, 2015. 3.8.2

[145] Divya Patel and Jaideep Chaudhary. Increased expression of bhlh transcription factor e2a (tcf3) in prostate cancer promotes proliferation and confers resistance to doxorubicin induced apoptosis. *Biochemical and biophysical research communications*, 422(1):146–151, 2012. 3.8.2

[146] Weike Pei, Thorsten B Feyerabend, Jens Rössler, Xi Wang, Daniel Postrach, Katrin Busch, Immanuel Rode, Kay Klapproth, Nikolaus Dietlein, Claudia Quedenau, et al. Polylox

barcoding reveals haematopoietic stem cell fates realized in vivo. *Nature*, 548(7668):456, 2017. 1.1.3

[147] Paulina Piairo, Rute S Moura, Maria João Baptista, Jorge Correia-Pinto, and Cristina Nogueira-Silva. Stats in lung development: Distinct early and late expression, growth modulation and signaling dysregulation in congenital diaphragmatic hernia. *Cellular Physiology and Biochemistry*, 45(1):1–14, 2018. 3.6.2

[148] Alexis Poncy, Aline Antoniou, Sabine Cordi, Christophe E Pierreux, Patrick Jacquemin, and Frédéric P Lemaigre. Transcription factors sox4 and sox9 cooperatively control development of bile ducts. *Developmental biology*, 404(2):136–148, 2015. 3.6.2

[149] Jean-Francois Poulin, Bosiljka Tasic, Jens Hjerling-Leffler, Jeffrey M Trimarchi, and Rajeshwar Awatramani. Disentangling neural cell diversity using single-cell transcriptomics. *Nature neuroscience*, 19(9):1131, 2016. 1.1

[150] Raewyn C Poulsen, Andrew J Carr, and Philippa A Hulley. Cell proliferation is a key determinant of the outcome of foxo3a activation. *Biochemical and biophysical research communications*, 462(1):78–84, 2015. 3.8.2

[151] Xiaojie Qiu, Andrew Hill, Jonathan Packer, Dejun Lin, Yi-An Ma, and Cole Trapnell. Single-cell mrna quantification and differential analysis with census. *Nature methods*, 14 (3):309, 2017. 1.1.1, 2.8.1, 3.6.5

[152] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. 1.4.1, 1.4.1

[153] Bushra Raj, James A Gagnon, and Alexander F Schier. Large-scale reconstruction of cell lineages using single-cell readout of transcriptomes and crispr–cas9 barcodes by scgestalt. *Nature protocols*, 13(11):2685, 2018. 4.12.1, 4.12.1

[154] Bushra Raj, Daniel E Wagner, Aaron McKenna, Shristi Pandey, Allon M Klein, Jay Shendure, James A Gagnon, and Alexander F Schier. Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. *Nature biotechnology*, 2018. (document), 1.1.3, 1.3, 4, 4.10, 4.12.2, 4.12.3, 4.12.3, 4.4, 4.12.4, 4.13, 4.14.1, 4.14.2, 4.14

[155] Sabrina Rashid, Darrell N Kotton, and Ziv Bar-Joseph. Tasic: determining branching models from time series single cell data. *Bioinformatics*, page btx173, 2017. 1.1.1, 1.4.2, 2.1, 2.8.1, 3.4, 3.8.1

[156] John E Reid and Lorenz Wernisch. Pseudotime estimation: deconfounding single cell time series. *Bioinformatics*, 32(19):2973–2980, 2016. 1.1.1

[157] Jüri Reimand, Tambet Arak, Priit Adler, Liis Kolberg, Sulev Reisberg, Hedi Peterson, and Jaak Vilo. g: Profiler—a web server for functional interpretation of gene lists (2016 update). *Nucleic acids research*, 44(W1):W83–W89, 2016. 2.7, 4.9

[158] Andreas M Reimold, Amit Etkin, Isabelle Clauss, Andrew Perkins, Daniel S Friend, John Zhang, Heidi F Horton, Andrew Scott, Stuart H Orkin, Michael C Byrne, et al. An essential role in liver development for transcription factor xbp-1. *Genes & development*, 14(2): 152–157, 2000. 3.8.2

[159] Abbas H Rizvi, Pablo G Camara, Elena K Kandror, Thomas J Roberts, Ira Schieren, Tom

Maniatis, and Raul Rabadan. Single-cell topological RNA-seq analysis reveals insights into cellular differentiation and development. *Nature biotechnology*, 35(6):551, 2017. 1.1.1, 2.8.1

[160] D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1):131 – 147, 1981. ISSN 0025-5564. 4.12.1

[161] Briana E Rockich, Steven M Hrycaj, Hung Ping Shih, Melinda S Nagy, Michael AH Ferguson, Janel L Kopp, Maike Sander, Deneen M Wellik, and Jason R Spence. Sox9 plays multiple roles in the lung epithelium during branching morphogenesis. *Proceedings of the National Academy of Sciences*, 110(47):E4456–E4464, 2013. 3.6.2

[162] Abraham B Roos, Tove Berg, Jenny L Barton, Lukas Didon, and Magnus Nord. Airway epithelial cell differentiation during lung organogenesis requires c/ebpα and c/ebpβ. *Developmental Dynamics*, 241(5):911–923, 2012. 3.6.2

[163] Edith M Ross and Florian Markowetz. OncoNEM: inferring tumor evolution from single-cell sequencing data. *Genome biology*, 17(1):69, 2016. 4.7

[164] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 2.10.1

[165] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000. 6.3.2

[166] Lisa Russell and Lee Ann Garrett-Sinha. Transcription factor ets-1 in cytokine and chemokine gene regulation. *Cytokine*, 51(3):217–226, 2010. 3.8.2

[167] N Saitou and M Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 07 1987. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040454. URL `https://doi.org/10.1093/oxfordjournals.molbev.a040454`. 4.12.1

[168] Irepan Salvador-Martínez, Marco Grillo, Michalis Averof, and Maximilian J Telford. Is it possible to reconstruct an accurate cell lineage using CRISPR recorders? *Elife*, 8:e40292, 2019. 4.12.1, 4.12.1

[169] Nathan Sandbo, Steven Kregel, Sebastien Taurin, Sangeeta Bhorade, and Nickolai O Dulin. Critical role of serum response factor in pulmonary myofibroblast differentiation induced by tgf-β. *American journal of respiratory cell and molecular biology*, 41(3): 332–338, 2009. 3.6.2

[170] Jeffry D Sander and J Keith Joung. Crispr-cas systems for editing, regulating and targeting genomes. *Nature biotechnology*, 32(4):347, 2014. 1.3.4

[171] Yumeko Satou, Kohei Minami, Erina Hosono, Hajime Okada, Yuuri Yasuoka, Takashi Shibano, Toshiaki Tanaka, and Masanori Taira. Phosphorylation states change otx2 activity for cell proliferation and patterning in the xenopus embryo. *Development*, pages dev–159640, 2018. 3.8.2

[172] Theresa Schacht, Marcus Oswald, Roland Eils, Stefan B Eichmüller, and Rainer König.

Estimating the activity of transcription factors by the effect on their target genes. *Bioinformatics*, 30(17):i401–i407, 2014. 3.1

[173] Susanne Schlisio, Terri Halperin, Miguel Vidal, and Joseph R Nevins. Interaction of yy1 with e2fs, mediated by rybp, provides a mechanism for specificity of e2f function. *The EMBO journal*, 21(21):5775–5786, 2002. 3.6.3

[174] Sebastian Schmeier, Tanvir Alam, Magbubah Essack, and Vladimir B Bajic. Tcof-db v2: update of the database of human and mouse transcription co-factors and transcription factor interactions. *Nucleic acids research*, page gkw1007, 2016. 3.6.4

[175] Marcel H Schulz, William E Devanny, Anthony Gitter, Shan Zhong, Jason Ernst, and Ziv Bar-Joseph. Drem 2.0: Improved reconstruction of dynamic regulatory networks from time-series expression data. *BMC systems biology*, 6(1):104, 2012. 1.3.3, 3.2, 3.6.1, 3.8.1

[176] Marcel H Schulz, Kusum V Pandit, Christian L Lino Cardenas, Namasivayam Ambalavanan, Naftali Kaminski, and Ziv Bar-Joseph. Reconstructing dynamic microrna-regulated interaction networks. *Proceedings of the National Academy of Sciences*, 110 (39):15686–15691, 2013. 3.1

[177] Jeremy Schwartzentruber, Stefanie Foskolou, Helena Kilpinen, Julia Rodrigues, Kaur Alasoo, Andrew J Knights, Minal Patel, Angela Goncalves, Rita Ferreira, Caroline Louise Benn, et al. Molecular and functional variation in ipsc-derived sensory neurons. *Nature genetics*, 50(1):54–61, 2018. 5.1

[178] Ruth L Seal, Susan M Gordon, Michael J Lush, Mathew W Wright, and Elspeth A Bruford. genenames. org: the hgnc resources in 2011. *Nucleic acids research*, 39(suppl_1): D514–D519, 2010. 1.3.3, 3.8.1

[179] Manu Setty, Michelle D Tadmor, Shlomit Reich-Zeliger, Omer Angel, Tomer Meir Salame, Pooja Kathail, Kristy Choi, Sean Bendall, Nir Friedman, and Dana Pe'er. Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nature biotechnology*, 34(6):637, 2016. 1.1.1

[180] Alex K Shalek, Rahul Satija, Xian Adiconis, Rona S Gertner, Jellert T Gaublomme, Raktima Raychowdhury, Schragi Schwartz, Nir Yosef, Christine Malboeuf, Diana Lu, et al. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 498(7453):236, 2013. 1.1

[181] Ying Shan, Tao Chang, Si Shi, Mingming Tang, Lili Bao, Li Li, Bo You, and Yiwen You. Foxj2 overexpression is associated with poor prognosis, progression, and metastasis in nasopharyngeal carcinoma. *OncoTargets and therapy*, 10:3733, 2017. 3.8.2

[182] Nobuyuki Shimizu, Shizuka Ishitani, Atsushi Sato, Hiroshi Shibuya, and Tohru Ishitani. Hipk2 and pp1c cooperate to maintain dvl protein levels required for wnt signal transduction. *Cell reports*, 8(5):1391–1404, 2014. 5.2.2

[183] L Shu, K Zien, G Gutjahr, J Oberholzer, F Pattou, J Kerr-Conte, and K Maedler. Tcf7l2 promotes beta cell regeneration in human and mouse pancreas. *Diabetologia*, 55(12): 3296–3307, 2012. 3.8.2

[184] Steven O Simmons and Jonathan M Horowitz. Nkx3. 1 binds and negatively regulates

the transcriptional activity of sp-family members in prostate-derived cells. *Biochemical Journal*, 393(1):397–409, 2006. 3.6.3

[185] Daniel A Skelly, Galen T Squiers, Micheal A McLellan, Mohan T Bolisetty, Paul Robson, Nadia A Rosenthal, and Alexander R Pinto. Single-cell transcriptional profiling reveals cellular diversity and intercommunication in the mouse heart. *Cell reports*, 22(3):600–610, 2018. 1.1

[186] Bastiaan Spanjaard and Jan Philipp Junker. Methods for lineage tracing on the organism-wide level. *Current opinion in cell biology*, 49:16–21, 2017. 1.1

[187] Bastiaan Spanjaard, Bo Hu, Nina Mitic, Pedro Olivares-Chauvet, Sharan Janjuha, Nikolay Ninov, and Jan Philipp Junker. Simultaneous lineage tracing and cell-type identification using CRISPR–Cas9-induced genetic scars. *Nature biotechnology*, 36(5):469, 2018. 1.1.3, 4.13

[188] Jyoti Srivastava, Chadia L Robertson, Devaraja Rajasekaran, Rachel Gredler, Ayesha Siddiq, Luni Emdad, Nitai D Mukhopadhyay, Shobha Ghosh, Phillip B Hylemon, Gregorio Gil, et al. Aeg-1 regulates retinoid x receptor and inhibits retinoid signaling. *Cancer research*, 74(16):4364–4377, 2014. 3.8.2

[189] Kelly Street, Davide Risso, Russell B Fletcher, Diya Das, John Ngai, Nir Yosef, Elizabeth Purdom, and Sandrine Dudoit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC genomics*, 19(1):477, 2018. 1.1.1, 3.6.5

[190] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019. 6.3.4

[191] LJ Su, CF Chang, HP Han, H Ma, and CS Xu. Analysis of changes about hsbp1, hsf1, hsf2 and hsp70's expression levels in rat's regenerating liver. *Fen zi xi bao sheng wu xue bao= Journal of molecular cell biology*, 39(3):258–264, 2006. 3.8.2

[192] J.E. Sulston, E. Schierenberg, J.G. White, and J.N. Thomson. The embryonic cell lineage of the nematode Caenorhabditis elegans. *Developmental Biology*, 100(1): 64 – 119, 1983. ISSN 0012-1606. doi: https://doi.org/10.1016/0012-1606(83)90201-4. URL `http://www.sciencedirect.com/science/article/pii/0012160683902014`. 4.12.1

[193] John E Sulston, E Schierenberg, John G White, and JN Thomson. The embryonic cell lineage of the nematode caenorhabditis elegans. *Developmental biology*, 100(1):64–119, 1983. 4.13

[194] Jianlong Sun, Azucena Ramos, Brad Chapman, Jonathan B Johnnidis, Linda Le, Yu-Jui Ho, Allon Klein, Oliver Hofmann, and Fernando D Camargo. Clonal dynamics of native haematopoiesis. *Nature*, 514(7522):322, 2014. 1.1.3

[195] Kai Sun, Michele A Battle, Ravi P Misra, and Stephen A Duncan. Hepatocyte expression of serum response factor is essential for liver function, hepatocyte proliferation and survival, and postnatal body growth in mice. *Hepatology*, 49(5):1645–1654, 2009. 3.6.2

[196] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of

single-cell rna-seq in the past decade. *Nature protocols*, 13(4):599–604, 2018. 6.3.4

[197] Yasuo Takashima, Kenichi Horisawa, Miyako Udono, Yasuyuki Ohkawa, and Atsushi Suzuki. Prolonged inhibition of hepatocellular carcinoma cell proliferation by combinatorial expression of defined transcription factors. *Cancer Science*, 109(11):3543, 2018. 3.8.2

[198] Miki Takeuchi, Shingo Yamaguchi, Yoshimasa Sakakibara, Takuto Hayashi, Koji Matsuda, Yuichiro Hara, Chiharu Tanegashima, Takashi Shimizu, Shigehiro Kuraku, and Masahiko Hibi. Gene expression profiling of granule cells and Purkinje cells in the zebrafish cerebellum. *Journal of Comparative Neurology*, 525(7):1558–1585, 2017. 4.12.3

[199] Li Tan, Xiaoping Wei, Lixia Zheng, Jincai Zeng, Haibo Liu, Shaojiang Yang, and Huo Tan. Amplified hmga2 promotes cell growth by regulating akt pathway in aml. *Journal of cancer research and clinical oncology*, 142(2):389–399, 2016. 3.8.2

[200] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, et al. mrna-seq whole-transcriptome analysis of a single cell. *Nature methods*, 6(5):377, 2009. 1.3.1

[201] Andrew E Teschendorff and Tariq Enver. Single-cell entropy for accurate estimation of differentiation potency from a cell's transcriptome. *Nature communications*, 8:15599, 2017. 2.1, 3.4

[202] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 2.3

[203] Sophia C Tintori, Erin Osborne Nishimura, Patrick Golden, Jason D Lieb, and Bob Goldstein. A transcriptional lineage of the early c. elegans embryo. *Developmental cell*, 38(4): 430–444, 2016. (document), 4.12.1, 4.3

[204] Ankana Tiwari, Shivananda Swamy, Kodaganur S Gopinath, and Arun Kumar. Genomic amplification upregulates estrogen-related receptor alpha and its depletion inhibits oral squamous cell carcinoma tumors in vivo. *Scientific reports*, 5:17621, 2015. 3.8.2

[205] A Kemal Topaloglu, Alejandro Lomniczi, Doris Kretzschmar, Gregory A Dissen, L Damla Kotan, Craig A McArdle, A Filiz Koc, Ben C Hamel, Metin Guclu, Esra D Papatya, et al. Loss-of-function mutations in pnpla6 encoding neuropathy target esterase underlie pubertal failure and neurological deficits in gordon holmes syndrome. *The Journal of Clinical Endocrinology & Metabolism*, 99(10):E2067–E2075, 2014. 2.10.2

[206] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381–386, 2014. 1.1, 1.1.1, 2.8.1, 3.6.1, 3.6.5, 3.8.1

[207] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. Pseudo-temporal ordering of individual cells reveals dynamics and regulators of cell fate decisions. *Nature biotechnology*, 32(4):381, 2014. 1.1.2, 3.6.5

[208] Barbara Treutlein, Doug G Brownfield, Angela R Wu, Norma F Neff, Gary L Mantalas, F Hernan Espinoza, Tushar J Desai, Mark A Krasnow, and Stephen R Quake. Reconstructing lineage hierarchies of the distal lung epithelium using single cell rna-seq. *Nature*, 509 (7500):371, 2014. 1.1, 1.1.1, 2.8, 2.8.1, 3.6.1, 3.6.5, 3.8.1

[209] Barbara Treutlein, Qian Yi Lee, J Gray Camp, Moritz Mall, Winston Koh, Seyed Ali Mohammad Shariati, Sopheak Sim, Norma F Neff, Jan M Skotheim, Marius Wernig, et al. Dissecting direct reprogramming from fibroblast to neuron using single-cell rna-seq. *Nature*, 534(7607):391, 2016. 1.1, 2.8, 2.8.2, 3.6.1, 3.8.1

[210] Fong-Ying Tsai and Stuart H Orkin. Transcription factor gata-2 is required for proliferation/survival of early hematopoietic cells and mast cell formation, but not for erythroid and myeloid terminal differentiation. *Blood*, 89(10):3636–3643, 1997. 3.8.2

[211] Gianluca Turcatel, Nicole Rubin, Douglas B Menke, Gary Martin, Wei Shi, and David Warburton. Lung mesenchymal expression of sox9 plays a critical role in tracheal development. *BMC biology*, 11(1):117, 2013. 3.6.2

[212] Paul R Van Ginkel, Kuang-Ming Hsiao, Hilde Schjerven, and Peggy J Farnham. E2f-mediated growth regulation requires transcription factor cooperation. *Journal of Biological Chemistry*, 272(29):18367–18374, 1997. 3.6.3

[213] Alexandra-Chloé Villani, Rahul Satija, Gary Reynolds, Siranush Sarkizova, Karthik Shekhar, James Fletcher, Morgane Griesbeck, Andrew Butler, Shiwei Zheng, Suzan Lazo, et al. Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science*, 356(6335):eaah4573, 2017. 1.1

[214] Alejandro F Villaverde, John Ross, Federico Morán, and Julio R Banga. Mider: network inference with mutual information distance and entropy reduction. *PloS one*, 9(5):e96732, 2014. 3.6.1

[215] Dominique Vlieghe, Albin Sandelin, Pieter J De Bleser, Kris Vleminckx, Wyeth W Wasserman, Frans Van Roy, and Boris Lenhard. A new generation of jaspar, the open-access repository for transcription factor binding site profiles. *Nucleic acids research*, 34 (suppl_1):D95–D97, 2006. 1.3.3, 3.8.1

[216] Chunyan Wan, Guohua Yuan, Daoshu Luo, Lu Zhang, Heng Lin, Huan Liu, Lei Chen, Guobin Yang, Shuo Chen, and Zhi Chen. The dentin sialoprotein (dsp) domain regulates dental mesenchymal cell differentiation through a novel surface receptor. *Scientific reports*, 6:29666, 2016. 3.8.2

[217] Tao Wang, Hui Zhao, Hua Gao, Changming Zhu, Yao Xu, Liping Bai, Junbo Liu, and Feng Yan. Expression and phosphorylation of foxo1 influences cell proliferation and apoptosis in the gastrointestinal stromal tumor cell line gist-t1. *Experimental and therapeutic medicine*, 15(4):3197–3202, 2018. 3.8.2

[218] Xiaoying Wang, Ying Ju, MI Zhou, Xiaoli Liu, and Chengjun Zhou. Upregulation of sox9 promotes cell proliferation, migration and invasion in lung adenocarcinoma. *Oncology letters*, 10(2):990–994, 2015. 3.8.2

[219] Yu Wang, Hong Chang, Di Gao, Lei Wang, Nan Jiang, and Bin Yu. Cdc5l contributes to

malignant cell proliferation in human osteosarcoma via cell cycle regulation. *International Journal of Clinical & Experimental Medicine*, 9(10), 2016. 3.8.2

[220] Yang-An Wen, Xiaopeng Xiong, Yekaterina Y Zaytseva, Dana L Napier, Emma Vallee, Austin T Li, Chi Wang, Heidi L Weiss, B Mark Evers, and Tianyan Gao. Downregulation of srebp inhibits tumor growth and initiation by altering cellular metabolism in colon cancer. *Cell death & disease*, 9(3):265, 2018. 3.8.2

[221] Adam Westmacott, Zoe D Burke, Guillermo Oliver, Jonathan MW Slack, and David Tosh. C/ebpα and c/ebpβ are markers of early liver development. *The International journal of developmental biology*, 50(7):653, 2006. 3.6.2

[222] F Alexander Wolf, Fiona K Hamey, Mireya Plass, Jordi Solana, Joakim S Dahlin, Berthold Göttgens, Nikolaus Rajewsky, Lukas Simon, and Fabian J Theis. Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome biology*, 20(1):59, 2019. 1.1.1, 3.6.5

[223] Mollie B Woodworth, Kelly M Girskis, and Christopher A Walsh. Building a lineage from single cells: genetic techniques for cell lineage tracking. *Nature Reviews Genetics*, 18(4): 230, 2017. 1.1, 1.1.3

[224] Bing Wu, Yunqi Wang, Chaojun Wang, Gang Greg Wang, Jie Wu, and Yisong Y Wan. Bptf is essential for t cell homeostasis and function. *The Journal of Immunology*, page 1600642, 2016. 3.8.2

[225] Haojia Wu, Kohei Uchimura, Erinn L Donnelly, Yuhei Kirita, Samantha A Morris, and Benjamin D Humphreys. Comparative analysis and refinement of human psc-derived kidney organoid differentiation with single-cell transcriptomics. *Cell Stem Cell*, 23(6): 869–881, 2018. 5.1

[226] Sen Wu, Xiaodong Feng, and Wenjun Zhou. Spectral clustering of high-dimensional data exploiting sparse representation vectors. *Neurocomputing*, 135:229–239, 2014. 2.10.1

[227] Shourong Wu, Huimin Wang, Yanjun Li, Yudan Xie, Can Huang, Hezhao Zhao, Makoto Miyagishi, and Vivi Kasim. Transcription factor yy1 promotes cell proliferation by directly activating the pentose phosphate pathway. *Cancer Research*, pages canres–4047, 2018. 3.8.2

[228] Mario F Wullimann, Thomas Mueller, Martin Distel, Andreas Babaryka, Benedikt Grothe, and Reinhard W Köster. The long adventurous journey of rhombic lip cells in jawed vertebrates: a comparative developmental analysis. *Frontiers in neuroanatomy*, 5:27, 2011. 4.12.3

[229] Lei Xia, Yan Gong, Aiqun Zhang, Shouwang Cai, and Qiang Zeng. Loss of gata5 expression due to gene promoter methylation induces growth and colony formation of hepatocellular carcinoma cells. *Oncology letters*, 11(1):861–869, 2016. 3.8.2

[230] Kai-Min Xiang and Xiao-Rong Li. Mir-133b acts as a tumor suppressor and negatively regulates tbpl1 in colorectal cancer cells. *Asian Pac J Cancer Prev*, 15(8):3767–72, 2014. 3.8.2

[231] Chao Xie, Yunwei Han, Yi Liu, Lei Han, and Jie Liu. mirna-124 down-regulates sox8

expression and suppresses cell proliferation in non-small cell lung cancer. *International journal of clinical and experimental pathology*, 7(11):7518, 2014. 3.8.2

[232] Wei Xu, Rita G Domingues, Diogo Fonseca-Pereira, Manuela Ferreira, Helder Ribeiro, Silvia Lopez-Lastra, Yasutaka Motomura, Lara Moreira-Santos, Franck Bihl, Veronique Braud, et al. Nfil3 orchestrates the emergence of common helper innate lymphoid cell precursors. *Cell reports*, 10(12):2043–2054, 2015. 3.8.2

[233] Honghua Yang, Min Min Lu, Lili Zhang, Jeffrey A Whitsett, and Edward E Morrisey. Gata6 regulates differentiation of distal lung epithelium. *Development*, 129(9):2233–2246, 2002. 3.6.2

[234] Zhong-Fa Yang, Karen Drumea, James Cormier, Junling Wang, Xuejun Zhu, and Alan G Rosmarin. Gabp transcription factor is required for myeloid differentiation, in part, through its control of gfi-1 expression. *Blood*, pages blood–2010, 2011. 3.8.2

[235] Guo-Cheng Yuan, Long Cai, Michael Elowitz, Tariq Enver, Guoping Fan, Guoji Guo, Rafael Irizarry, Peter Kharchenko, Junhyong Kim, Stuart Orkin, et al. Challenges and emerging directions in single-cell analysis. *Genome biology*, 18(1):84, 2017. 1.3.1

[236] Hamim Zafar, Anthony Tzen, Nicholas Navin, Ken Chen, and Luay Nakhleh. SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models. *Genome biology*, 18(1):178, 2017. 1.1.3

[237] Hamim Zafar, Nicholas Navin, Ken Chen, and Luay Nakhleh. SiCloneFit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data. *bioRxiv*, page 394262, 2018. 4.7

[238] Habil Zare, Parisa Shooshtari, Arvind Gupta, and Ryan R Brinkman. Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC bioinformatics*, 11(1):1, 2010. 2.10.1

[239] Dianbao Zhang, Ying Li, Rui Wang, Yunna Li, Ping Shi, Zhoumi Kan, and Xining Pang. Inhibition of rest suppresses proliferation and migration in glioblastoma cells. *International journal of molecular sciences*, 17(5):664, 2016. 3.8.2

[240] Lihua Zhang and Shihua Zhang. Comparison of computational methods for imputing single-cell RNA-sequencing data. *IEEE/ACM transactions on computational biology and bioinformatics*, 2018. 4.14.1

[241] Xinyue Zhang, Jieyu Guo, Xiangxiang Wei, Cong Niu, Mengping Jia, Qinhan Li, and Dan Meng. Bach1: Function, regulation, and involvement in disease. *Oxidative medicine and cellular longevity*, 2018, 2018. 3.8.2

[242] Zhen Zhang, Kim Newton, Sarah K Kummerfeld, Joshua Webster, Donald S Kirkpatrick, Lilian Phu, Jeffrey Eastham-Anderson, Jinfeng Liu, Wyne P Lee, Jiansheng Wu, et al. Transcription factor etv5 is essential for the maintenance of alveolar type ii cells. *Proceedings of the National Academy of Sciences*, 114(15):3903–3908, 2017. 2.8.1

[243] Zhongbao Zhang, Guangju Meng, Liang Wang, Yingying Ma, and Zhongzheng Guan. The prognostic role and reduced expression of foxj2 in human hepatocellular carcinoma. *Molecular medicine reports*, 14(1):254–262, 2016. 3.8.2

[244] Jie Zheng, Shuna Yu, Zhengchen Jiang, Caixing Shi, Jin Li, Xiaodong Du, Hailiang Wang, and Jiying Jiang. Microarray comparison of the gene expression profiles in the adult vs. embryonic day 14 rat liver. *Biomedical reports*, 2(5):664–670, 2014. 3.6.2

[245] Shuping Zhong, Jody Fromm, and Deborah L Johnson. Tbp is differentially regulated by c-jun n-terminal kinase 1 (jnk1) and jnk2 through elk-1, controlling c-jun expression and cell proliferation. *Molecular and cellular biology*, 27(1):54–64, 2007. 3.8.2

[246] Yonghao Zhong, Hongyang Huang, Min Chen, Jinzhou Huang, Qingxia Wu, Guang-Rong Yan, and De Chen. Pou2f1 over-expression correlates with poor prognoses and promotes cell growth and epithelial-to-mesenchymal transition in hepatocellular carcinoma. *Oncotarget*, 8(27):44082, 2017. 3.8.2

[247] Qingjun Zhou, Richard W Gedrich, and Daniel A Engel. Transcriptional repression of the c-fos gene by yy1 is mediated by a direct interaction with atf/creb. *Journal of virology*, 69 (7):4323–4330, 1995. 3.6.3

[248] Zhenhua Zhou, Yan Li, Qi Jia, Zhiwei Wang, Xudong Wang, Jingjing Hu, and Jianru Xiao. Heat shock transcription factor 1 promotes the proliferation, migration and invasion of osteosarcoma cells. *Cell proliferation*, 50(4):e12346, 2017. 3.8.2

[249] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012. 2.10.1