Inferring Developmental Trajectories and Causal Regulations with Single-cell Genomics

Xiaojie Qiu

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:

Bruce C Trapnell, Chair

Sui Huang

Ilya Shmulevich

Program Authorized to Offer Degree:

Molecular and Cellular Biology

University of Washington

Abstract

Inferring Developmental Trajectories and Causal Regulations with Single-cell Genomics

Xiaojie Qiu

Chair of the Supervisory Committee:

Bruce C Trapnell

Genome Sciences

Development is commonly regarded as a hierarchical branching process which is governed by underlying gene regulatory networks. Single-cell genomics, single-cell RNA-seq (scRNA-seq) in particular, holds the promise to resolve the dynamics of this process. However, learning the structure of complex single-cell trajectories with multiple branches remains a challenging computational problem. In this thesis, I will present the toolkit, Monocle 2, which uses reversed graph embedding to reconstruct single-cell trajectories in a fully unsupervised manner. Monocle 2 learns an explicit "principal graph" that passes through the middle of the data as opposed to other *ad hoc* methods, greatly improving the robustness and accuracy of its trajectories. I will demonstrate that Monocle 2 is able to accurately reconstruct developmental trajectories for complicated systems, including hematopoiesis involving multiple different cell fates. When coupled

with another statistical framework, BEAM (branch expression analysis modeling), Monocle 2 is able to detect genes specific to different developmental lineages. The unprecedented high resolution of the reconstructed developmental trajectories not only enables us to determine which genes are playing important roles at the critical time point of cell fate transition but also to directly infer causal gene regulatory networks. To this end, I have been developing a new toolkit, Scribe, which applies novel information theory techniques to detect causal interactions responsible for fate transitions. Scribe provides intuitive visualizations of causal interactions and can additionally incorporate information from "RNA-velocity" for causality detection. Scribe accurately reconstructs core networks specifying myelocytic or chromaffin cells. Finally, I will show a compendium of the inferred causal regulatory network for *C elegans*' early embryogenesis based on lineage resolved live imaging data, demonstrating Scribe's generalizability.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

Time flies. In a sudden, It already comes close to the end of my Ph.D. career. For this important period in my life, I received various supports from mentors, labmates, friends and family members, among many others. Firstly, I would like to thank Profs. Ping Ao (Shanghai University), Sui Huang (Institute for Systems Biology) and Mrs. MaryEllin Robinson (who had retired from MCB program) who helped me to get recruited by the MCB program. Without their support, it won't be possible for me to pursue my research at the University of Washington. Secondly, I would like to especially thank my advisor, Prof. Cole Trapnell, for his incredible supervision in the past four years. I still vividly remember the first day I met him to talk about single-cell RNA-seq. The energy, enthusiasm, and sharpness projected from him impressed me and solidified my interests to join his lab as the first graduate student. It turns out to be a blessing four years in his lab. At the early stages of the lab when there is only he and me, I got the privilege to discuss and work with him from time to time, which quickly trained me to be an excellent scientist. Thanks to this unique experience, I am able to be very productive and publish two first-author papers in Nature Methods in 2017. As I expressed in one of the very first emails to Cole, I mentioned that "Definitively I will be an enthusiastic and motivated student". I hope my diligence, hard work, and contributions to the lab realized me what I had promised.

During my Ph.D., I enjoyed wonderful collaboration with various people with a range of background. I would like to thank the great collaboration with Andrew Hill (BEAM project), Qi Mao and Li Wang (Monocle 2), Prof. Sreeram Kannan and Arman Rahimzamani (Scribe) and many others, including Yian Ma, Darren Cusanovich, Junyue Cao, etc. I would also like to thank

my very supportive committee members, Profs. Robert Waterson, Bill Noble, Robert Bradley, Ilya Shmulevich and Sui Huang on their insightful suggestions on the research directions I planned for.

Every Ph.D. career has times of hardship. For example, my first manuscript got rejected twice and took more than one year and a half to be reviewed. To overcome hardships like this, my labmates, especially Sanjay Srivatsan, Raghav Chawla, among others, generously invested time and energy to encourage, help me to get through. I really appreciate their friendship and support. We would also like to thank my friends outside the lab, including Tim Durham, Ken Chen, Aaron Seo, my roommates for their various supports during my Ph.D. career.

Finally, I would like to express my great gratitude to my family members, especially my amazing parents. Without their understanding, devotion, and sacrifice in supporting my education, it won't be possible for me to be the first person in our large family to study abroad. I have not visited my family for more than 4 years for reasons of Visa. Although I didn't go home, I hope they know that I love them and will be proud to see my academic achievements.

I would like to thank many others who I may have forgotten to mention!

DEDICATION

I would like to dedicate this thesis to all previous Chinese scientist who came to the USA or other foreign countries to study and devoted their entire life in facilitating the advancement of China's and the global academic community. I am proud to be a citizen of a country with more than 5, 000 years' history and it is my responsibility to restore this country which has suffered too much in the past 300 years.

Introduction

Almost every organism on the planet is produced from a single cell. To understand how does a multicellular organism develop thus lies at the heart of biology. A natural way to address this question is to directly follow each cell division during the developmental process. At the 70s, John Sulston manually recorded every cell division of *c. elegans*' entire embryogenesis, literally using pen and paper with observation under a microscope (Sulston et al. 1983). This epic work produced the first and also the only comprehensive lineage map for a multicellular organism. Various alternative approaches are later developed, including cell marking and tracing with dyes or enzyme (Weisblat, Sawyer, and Stent 1978)(Keller 1975), recombinase-mediated (for example, Cre-Loxp) activated report genes (Zinyk et al. 1998) and more recently somatic mutation or repurposed CRISPR-Cas9 based molecular recorders (McKenna et al. 2016)(Frieda et al. 2017).

Although lineage maps provide a great resource to study development, they don't provide direct mechanistic understanding on when and how the cells are specified and what are the genes driving the cell fate specifications. 50 years ago, Conard Waddington proposed an elegant explanation in his book *The Strategy of the Genes* to explain the mechanism of cell fate bifurcation using a metaphor, the epigenetic landscape (Waddington 2014). In this metaphor, Waddington represented the zygote as ball located on the top of a hill. The downward movement along the valley (cherod or the developmental path on the landscape) on the landscape of the ball or zygote represents cell differentiation with a decrease of the developmental potential (or the capability to differentiate into different terminal cell types). The developmental process on the

landscape is further represented as a hierarchical tree structure where the stem cells can choose different valleys to bifurcate into different progenitor cells and eventually terminal cell types. How is the topography of the epigenetic landscape shaped? In the same book, Waddington also proposed the guy-rope model. In this model, the pegs or genes in the ground will interact with each other and pull the surface of the landscape to form distinct valleys and thus determine the shape of the epigenetic landscape; that is the underlying gene regulatory network will determine the cell lineage specification.

Although the epigenetic landscape was first proposed as a metaphor to explain the developmental process, it is actually closely related to many key concepts in other fields, for example, the potential landscape in physics, the adaptive landscape in population genetics and protein folding funnel landscape in biochemistry, *etc (Ao 2009)*. Moreover, the epigenetic landscape has a theoretical foundation and can be rigorously quantified. Over the past decade or so, there is intensive efforts to mathematically reconstruct the epigenetic landscape from small network motifs (J. Wang et al. 2011). When I was Master student, I also contributed in this direction. I took a systems biology approach and modeled the cell types (stem cell or terminal cell types) as the attractor (a steady state to which a dynamical system evolves after a long enough time) with a set of stochastic differential equations. The cell state transitions, including differentiation, reprogramming as well as transdifferentiation, were modeled as the least action path between the attractor states (Qiu, Ding, and Shi 2012a).

As early as 2009 (F. Tang et al. 2009), single-cell RNA sequencing (scRNA-Seq) began to reveal its unprecedented power in studying cell fate specification as it offers a novel high-resolution vista on cell states and thus can connect the theory of cell state transitions with

the actual transcriptional dynamics. Recognizing its promise when I started my Ph.D in 2013, I joined Dr. Cole Trapnell's lab as his first graduate student, just when he started his own lab, to specifically work on computational method development for single-cell genomics. These computational methods, as I will discuss later, are inspired by some of the above theories but relies on techniques from statistics, machine learning, and computer sciences to provide new insights of developmental process.

Traditionally, when we study the developmental process, for example, the hematopoiesis, we may collect bulk samples at different time points and measure their transcriptome with microarray or bulk RNA-seq. Since we only have bulk measurements, we cannot observe the heterogeneity across individual cells. With single-cell genomics approach, for example, scRNA-seq, each individual cell's the transcriptomic state is profiled. Since different cells may differentiate from a pool of different stem cells starting from different time points and different cells can have different developmental timing, even we collected cells from the same time point, they may be distributed in different stages of the developmental process. In order to obtain a high-resolution view of the developmental process, we can take advantage of the heterogeneity and asynchrony from single-cell data to computationally reconstruct the branching developmental trajectories. The reconstructed developmental trajectory provides a description of cell progression during development. Since we only capture snap-shot data from single-cell RNA-seq, although we study the time with the reconstructed trajectory they are not real time, so we call it *pseudotime*. We define the pseudotime as the *geodesic distance* (in a loose metaphoric manner in computer science) of a cell to the root cell determined by prior biological knowledge

on the tree structure; large pseudotime means later or longer in the developmental progression under study (Trapnell et al. 2014b).

The recent emerging single-cell genomics technologies are revolutionary technologies which enables us to measure various aspects of the genome for thousands of cells. For example, single-cell RNA-seq or scRNA-seq measures the RNA abundance for all the transcripts in the cell while scATAC-seq (Buenrostro et al. 2015), scHi-C (Ramani et al. 2017) or scProteomics (Stoeckius et al. 2017) measures the transposase accessible chromatin, chromatin 3D structures and surface or cellular protein expression levels for thousands of cells. A typical scRNA-seq experiment works as following, we start with some tissue samples, then we isolate the sample into individual cells, then the RNA inside the cell is extracted and reverse transcribed into cDNA. The cDNA is then amplified into a library and subjected to sequencing using a next-generation sequencer. In the end, we use computational algorithms to quantify the abundance of RNA in each individual cell.

The number of cells a single scRNA-seq experiment can measure is expected to increase exponentially over time, likely to follow the biological equivalent of Moore's Law as has been shown for other omics technologies (Svensson, Vento-Tormo, and Teichmann 2018). It started with a few dozens of cells, processed by manual handling, then hundreds of cells with the streamlined microfluidics, to reach now a throughput of tens of thousands of cells with the droplet-based approaches. A recent breakthrough, single-cell combinatorial indexing RNA-seq (sci-RNA-seq), relies on combinatorial indexing without physical separation of cells, which can in principle be scaled to tens of millions of cells (Cao et al. 2017b). The increase in scale for scRNA-seq experiments has already launched several initiatives for building a comprehensive

cell atlas of all tissues, organs as well as model organisms, including *C. elegans*, mouse, and human. These cell atlas projects have already generated an enormous amount of data and will generate thousand times more datasets, thus they pose great computational challenges for data manipulation, visualization as well as analyzation.

There are a few key properties we need to keep in mind if we analyze the scRNA-seq data. First of all, the scRNA-seq data has very high dimension. With the current technology, we can easily sequence thousands of cells (as mentioned above) for the entire transcriptome, for example about 30, 000 genes in the human genome. Secondly, scRNA-seq data is very noisy. This is because this technology is pretty new and still not perfect. Finally, scRNA-seq only offers snapshot data. This is because the cells are killed when we perform the experiment.

My thesis tries to address three major questions. The first one is: How can we map complex developmental trajectories from scRNA-seq data? The second question is: What are the genes associated with the bifurcating developmental process? Finally, how can we recover the underlying developmental regulatory networks? Thus this thesis will be organized into four chapters. The first chapter discusses a generative model of a single-cell RNA-seq experiment. The second chapter discusses a general machine learning framework, Reversed Graph Embedding or RGE, to reconstruct the complex developmental trajectory from scRNA-seq data. The third chapter discusses a statistical approach, BEAM (Branch Expression Analysis Modeling), to identify the genes associated with each lineage or branch. The last chapter discusses a toolkit, Scribe, which uses a novel information metric Restricted Direction Information or RDI, to infer the causal regulatory network from single-cell RNA-seq data.

Chapter 1 A generative model of single-cell RNA-seq

A version of this chapter has been previously published as part of the following paper:

Single-cell mRNA quantification and differential analysis with Census. **X Qiu**, A Hill, J Packer, D Lin, YA Ma, C Trapnell ***Nature methods*** 14 (3), 309

Recently I developed Census (Qiu, Hill, et al. 2017b), an algorithm that converts conventional measures of relative expression such as transcript per million (TPM) in single cells to normalized transcript counts without the need for spike-in standards or UMIs. However, all recent single-cell RNA-seq protocol, including drop-seq, usually uses UMIs. In addition, the linear assumption that TPM is proportional to the true relative abundance in the cell lysis limits the accuracy of Census. These facts make Census become increasingly irrelevant. So I will not discuss extensively Census, instead mainly talk about Census' underlying generative model of scRNA-seq experiment with spike-in ladder, which mimics natural eukaryotic mRNAs and controls the various source of variance during the scRNA-seq experiment. The spike-in ladder we use are usually the 92 exogenous RNA standards developed by the External RNA Controls Consortium (ERCC), each has 250 to 2,000 nt in length.

1.1 A generative model for single-cell RNA-seq experiments with a spike-in ladder

Census is motivated by a generative model of single-cell (sc) RNA-Seq similar to the one developed by Kim *et al (Kim et al. 2015)*. When performing sc-RNA-seq, each individual cell is lysed to recover its endogenous RNA molecules, some fraction of which may be degraded or lost. Lysis thus involves an RNA recovery rate α. Spike-in transcripts are then added into the

cell lysate. Note that spike-in transcripts are added to the lysate as naked RNA, and thus may be degraded at different rates from the endogenous RNA. We denote the ladder recovery rate as β. The RNA counts in the lysate can be written:

$$Cell\ lysate: \begin{cases} Y_{ij}^{i} \approx \alpha_i Y_{ij}^{c} \\ S_{ij}^{i} \approx \beta_i S_{\cdot j}^{c} \end{cases}$$

where Y^{l}, S^{l}, S are the transcript counts of endogenous RNA in cell lysate, spike-in transcript counts in cell lysate and the spike-in transcript counts added into the cell lysate. The first subscript in all variables (here and below) corresponds to cell while the second subscript corresponds to gene index. Note that we are not able to directly observe Y_{ij}^{c}, the true transcript counts for gene j in cell i and thus α is an unknown variable.

The RNA molecules and spike-in transcripts will then be subjected to reverse transcription and amplified to make a cDNA library. The expected number of cDNA molecules generated from each RNA molecules is denoted by θ. The cDNA counts can be written:

$$cDNA: \begin{cases} Y_{ij}^{d} = Y_{ij}^{l} \cdot \theta_i \\ S_{ij}^{d} = \beta_i S_{ij}^{l} \cdot \theta_i \end{cases}$$

where Y^{d}, S^{d} are the cDNA counts of endogenous RNA, spike-in cDNA counts successfully converted from the corresponding transcript counts Y^{t}, s^{l} in cell lysate under a uniform capture rate θ, which for current protocols is less than 1.

Our model generates sequencing reads from the cDNA. The relative cDNA abundances are calculated as $\frac{Y_{ij}^d}{\sum_{j=1}^{n}(Y_{ij}^d + S_{ij}^d)}$ for endogenous RNA, or $\frac{S_{ij}^d}{\sum_{j=1}^{n}(Y_{ij}^d + S_{ij}^d)}$ for spike-in RNA.

The model then generates γ reads per cDNA molecule on average; with sufficient sequencing, γ will be larger than 1; we expect each cDNA molecule to generate at least one sequencing read. This process can be regarded as a multinomial sampling of R reads $(R_i = \gamma \sum_{j=1}^{n}(Y_{ij}^d + S_{ij}^d))$ from the distribution of relative cDNA abundances mentioned above which can be represented as:

$$Readcounts: \begin{cases} Y_{ij}^r \tilde{} multinomial(\frac{Y_{ij}^d}{\sum_{j=1}^{n}(Y_{ij}^d + S_{ij}^d)}, R_i^e) \\ S_{\cdot j}^r \tilde{} multinomial(\frac{S_{ij}^d}{\sum_{j=1}^{n}(Y_{ij}^d + S_{ij}^d)}, R_i^s) \end{cases}$$

where R_i^e, R_i^s denotes the reads sampled for cDNA from the endogenous RNA or spike-in RNA in cell i, $Y_{ij}^\gamma, S_{\cdot j}^\gamma$ denotes the reads sampled for cDNA from the endogenous RNA j or spike-in RNA j in cell i.

The model described here is essentially a special case of the model in Kim *et al.*, and differs mainly in that their model describes transcript-level capture rates and sequencing rates with beta and gamma distributions, respectively. In contrast, we simply use global constants for these rates.

1.2 A simulator for the sc-RNA-seq process

To generate an *in silico* library for a single cell, we built a simulator that first selects *G* genes at random from a relative expression profile (P_{bulk}) derived from a bulk RNA-Seq experiment to represent the hypothetical relative abundance of a single-cell in cell lysate. These values are rescaled to proportions (i.e. summing to 1), or ρ_{scale}

$$\rho_{scale} \tilde{} scale(uniform(P_{bulk}(1, 2, ...), G)$$

These proportions are then used to parameterize a multinomial distribution from which *T* transcripts are drawn to obtain the transcripts in the library space where we also consider there is percentage of the RNA is degraded. Therefore, we have:

$$\text{library: } y_{ij}^{t} \tilde{} multinomial(\rho_{scale}, (1 - \alpha_i)T_i)$$

To this pool of transcripts, a fixed number of spike-in transcripts are added, forming a mixture of simulated "endogenous" and "spike-in" mRNAs where the degradation of spike-in transcripts is represented by β_i. Of these, θ_i percent are selected uniformly at random to simulate incomplete mRNA capture by the reverse transcription process. Finally, the abundances of these cDNAs relative to one another were used to parameterize another multinomial, from which R_i reads are sampled. The read counts are then used to calculate the relative abundance for the spike-in and the endogenous RNA.

In this study, we systematically simulated the sc RNA-seq process obtained from bulk RNA-Seq measurements made in Trapnell and Cacchiarelli *et al(Trapnell et al. 2014b)* by

varying the gene number G, capture rate θ, endogenous RNA degradation α, spike-in degradation β, total endogenous transcript count T and total number of reads R.

Chapter 2 Reversed graph embedding resolves complex developmental trajectories

A version of this chapter has been previously published as part of the following paper:

Reversed graph embedding resolves complex single-cell trajectories. **X Qiu**, Q Mao, Y Tang, L Wang, R Chawla, HA Pliner, C Trapnell ***Nature methods*** 14 (10), 979

2.1 Introduction

Most cell state transitions, whether in development, reprogramming, or disease, are characterized by cascades of gene expression changes. We recently introduced a bioinformatics technique called "pseudotemporal ordering", which applies machine learning to single-cell transcriptome sequencing (RNA-Seq) data to order cells by progression and reconstruct their "trajectory" as they differentiate or undergo some other type of biological transition(Trapnell et al. 2014b). Despite intense efforts to develop scalable, accurate pseudotime reconstruction algorithms (recently reviewed in (Kumar, Tan, and Cahan 2017)), state-of-the-art tools have several major limitations. Most pseudotime methods can only reconstruct linear trajectories, while others such as Wishbone(Setty et al. 2016) or DPT(Haghverdi et al. 2016a) support branch identification with heuristic procedures, but either are unable to identify more than one branch point in the trajectory or require that the user specify the number of branches and cell fates as an input parameter.

Here, we describe Monocle 2 (https://github.com/cole-trapnell-lab/monocle-release), which applies reversed graph embedding (RGE)(Mao, Wang, et al. 2015; Mao et al. 2016a), a

recently developed machine learning strategy, to accurately reconstruct complex single-cell trajectories. Monocle 2 requires no *a priori* information about the genes that characterize the biological process, the number of cell fates or branch points in the trajectory, or the design of the experiment. Monocle 2 outperforms not only its previous version but also more recently developed methods, producing more accurate, robust trajectories.

2.2 dpFeature: a general unsupervised feature selection approach for trajectory inference

Monocle 2 begins by identifying genes that define biological process using an unsupervised procedure that we term "dpFeature". The procedure works by selecting the genes differentially expressed between clusters of cells identified with tSNE dimension reduction followed by density peak clustering . When applied to four different datasets(Trapnell et al. 2014b; Treutlein et al. 2014; Olsson et al. 2016; Paul et al. 2015) most of the genes returned by dpFeature were also recovered by a semi-supervised selection method guided by aspects of the experimental design and were highly enriched for Gene Ontology relevant to myogenesis, confirming that dpFeature is a powerful and general unsupervised feature selection approach. (**Supplementary Figures 1**)

2.3 Reversed graph embedding (RGE): a general framework to reconstruct complex developmental trajectories

We next sought to develop a pseudotime trajectory reconstruction algorithm that does not require the number of cell fates or branches as input parameter. To do so, we employed reversed graph

embedding(Mao, Wang, et al. 2015; Mao et al. 2016a), a machine learning technique to learn a parsimonious *principal graph*. Informally, a principal graph is like a principal curve(Hastie and Stuetzle 1989) that passes through the "middle" of a dataset but is allowed to have branches(Gorban and Zinovyev 2009). However, learning a principal graph that describes a population of single-cell RNA-Seq profiles is very challenging because each expressed gene adds an additional dimension to the space. In general, learning geometry is dramatically harder in high-dimensional spaces(Bellman 1954). Reversed graph embedding (RGE) solves this problem by finding a mapping between the high dimensional gene expression space and a much lower dimensional one while simultaneously learning the structure of the graph in this reduced space. In the following, I will provide mathematical details about RGE and how can we integer RGE to assign pseudotime and branch for each cell .

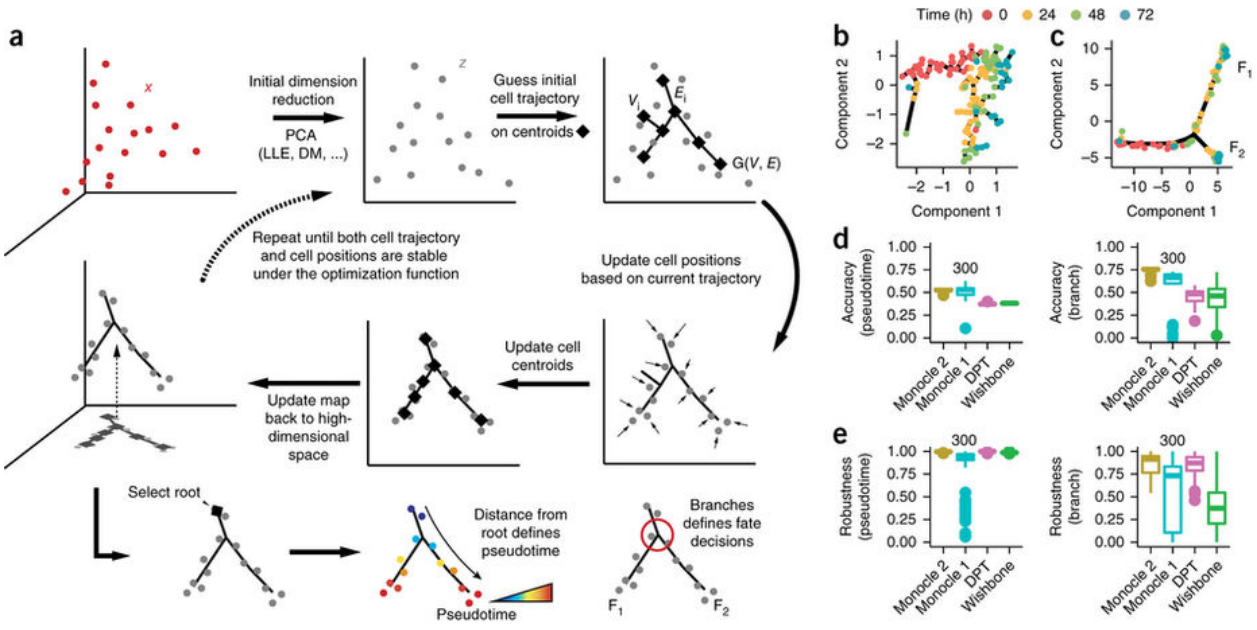


Figure 2.1. Monocle 2 discovers a cryptic alternative outcome in myoblast differentiation. (**A**) Monocle 2 learns single-cell trajectories by reversed graph embedding. Each cell can be

represented as a point in a high-dimensional space where each dimension corresponds to the expression level of an ordering gene. The high dimensional data are first projected to a lower dimensional space (Z) by any of several dimension reduction methods such as PCA (default), diffusion maps, etc. Monocle 2 then constructs a spanning tree on an automatically selected set of centroids of the data. The number of centroids (black diamonds) is determined using a formula that scales sublinearly in the number of cells. These centroids are chosen automatically using k-medoids clustering in the initialized low-dimensional space. The algorithm then moves the cells towards their nearest vertex of the tree, updates the positions of the vertices to "fit" the cells, learns a new spanning tree, and iteratively continues this process until the tree and the positions of the cells have converged (see **Equation 3** in **Methods**). Throughout this process, Monocle 2 maintains an invertible map between the high-dimensional space and the low-dimensional one, thus both learning the trajectory and reducing the dimensionality of the data. In effect, the algorithm acts as soft K-means clustering on points Z that maps them to the centroids, and jointly learns a graph on the centroids. Once Monocle 2 learns the tree, the user selects a tip as the "root". Each cell's pseudotime is calculated as its geodesic distance along the tree to the root, and its branch is automatically assigned based on the principal graph. (**B**) Monocle 1 reconstructs a linear trajectory for differentiating human skeletal myoblasts (HSMM)(Trapnell et al. 2014b). (**C**) Monocle 2 automatically learns the underlying trajectory and detects that cells from 24-72 hours are divided into two branches. The same genes selected with dpFeature (**Supplementary Figure 1; Methods**) were used for ordering for both of Monocle 1 and Monocle 2. (**D**) Accuracy of pseudotime calculation or branch assignments from each algorithm under repeated subsamples of 80% of the cells on the Paul dataset (Paul et al.

2015). A marker based ordering (see **Methods**) is used as ground truth for results from each software in all downsamplings to compare with. (**E**) Consistency of pseudotime calculation or branch assignments from each algorithm under repeated subsamples of 80% of the cells on the Paul dataset(Paul et al. 2015). All pairwise downsamplings are used to calculate the Pearson's Rho and adjusted Rand index (ARI). Monocle 2, DPT, and Wishbone all use the full dataset for benchmark while Monocle 1 only uses a random downsampled 300 cells as for benchmarking.

2.3 Reversed graph embedding

Monocle 2 uses a technique called reversed graph embedding(Mao et al. 2016a; Mao, Wang, et al. 2015; Mao et al., n.d.) (RGE) to learn a graph structure that describes a single-cell experiment. RGE simultaneously learns a principal graph that represents the cell trajectory, as well as a function that maps points on the trajectory (which is embedded in low dimensions) back to the original high dimensional space. RGE aims to learn both a set of latent points $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ where N is the number of the set (or cell numbers) and an undirected graph $\mathcal{G}$ that connects these latent points. The latent points in the low-dimensional space corresponds to the input data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ in the high-dimensional space. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a set of vertexes $\mathcal{V} = \{V_1, \ldots, V_N\}$ and a set of weighted, undirected edges $\mathcal{E}$, where each V_i corresponds to latent point $\mathbf{z}_i$, so the graph also resides in the latent, low-dimensional space.

In the context of the single-cell trajectory construction problem, $\mathbf{x}_i$ is typically a vector of the feature genes' expression values (for example, based on dpFeature selection, see **Supplementary Method**) of the i-th cell in a single-cell RNA-Seq experiment, $\mathcal{G}$ is the learned

trajectory (for example, a tree) along which the cells transit, and z_i is the principal point on $\mathcal{G}$ corresponding to the cell $\mathbf{x}_i$.

RGE learns the graph $\mathcal{G}$ as well as a function that maps back to the input data space. Let $b_{i,j}$ denote the weight of edge (V_i, V_j), which represents the connectivity between $\mathbf{z}_i$ and $\mathbf{z}_j$. In other words, $b_{i,j} > 0$ means that edge exists in $\mathcal{G}$, and 0 otherwise. Define $f_{\mathcal{G}}$ as the projection function from $\mathbf{z}_i$ to some point in the high-dimensional space. To learn $\mathcal{G}$, $\mathcal{Z}$ and $f_{\mathcal{G}}$, we need to optimize

$$\min_{\mathcal{G} \in G_b} \min_{f_{\mathcal{G}} \in \mathcal{F}} \min_{\mathcal{Z}} \sum_{(V_i, V_j) \in \mathcal{E}} b_{i,j} ||f_{\mathcal{G}}(\mathbf{z}_i) - f_{\mathcal{G}}(\mathbf{z}_j)||^2$$

where G_b is a set of feasible graph structures parameterized by $i, j, \forall i, j$, and $\mathcal{F}$ is a set of functions mapping a latent, low-dimensional point to a point in the original, high-dimensional space.

As shown in (Mao et al., n.d.), the above optimization will learn graph structures in the latent space, but it does not measure the deviations of latent points to the observed data. That is, no effort is made to ensure that the graph nodes are embedded in a way relevant to the cloud of observed data points. To ensure the graph describes the overall structure of the observed data, RGE aims to position the latent points such that their image under the function (that is, their corresponding positions in the high-dimensional space) will be close to the input data while also ensures neighbor points on low dimensional principal graph be ”neighbors” in the input dimension. The optimization problem is formulated as

$$\min_{\mathcal{G}\in G_b}\min_{f_{\mathcal{G}}\in\mathcal{F}}\min_{\mathcal{Z}}\sum_{i=1}^{N}||\mathbf{x}_i-f_{\mathcal{G}}(\mathbf{z}_i)||^2+\frac{\lambda}{2}\sum_{(V_i,V_j)\in\mathcal{E}}b_{i,j}||f_{\mathcal{G}}(\mathbf{z}_i)-f_{\mathcal{G}}(\mathbf{z}_j)||^2$$

where λ is a parameter that adjusts the relative strength of these two summations. In practice, implementing reversed graph embedding requires that we place some constraints on G_b and $f_{\mathcal{G}}$, as summarized briefly in the following sections.

2.3.1 SimplePPT: A simple principal tree algorithm

SimplePPT is the first RGE technique proposed by Mao et al for learning a tree structure to describe a set of observed data points. The tree can be learned in the original space or in some lower dimension retrieved by dimensionality reduction methods such as PCA(Mao et al., n.d.) . SimplePPT makes some choices that simplify the optimization problem. Notably, $f_G z_i$ is optimized as one single variable instead of two separate sets of variables. Moreover, the loss function in the reversed graph embedding is replaced by the empirical quantization error, which serves as the measurement between the $f_G(z_i)$ and its corresponding observed points x_i. The joint optimization of $f_G(z_i)$ is efficient from the perspective of optimization with respect to $\{b_{ij}\}$, which is solved by simply finding the minimum spanning tree.

2.3.2 The principal graph $\mathcal{L}_1$ algorithm

Mao et al (Mao et al. 2016a) later proposed an extension of SimplePPT that can learn arbitrary graphs, rather than just trees, which describes large datasets embedded in the same space as the input. An $\mathcal{L}_1$ graph is a sparse graph which is based on the assumption that each data point (or cell) has a small number of neighborhoods in which the minimum number of points that span a

low-dimensional affine subspace[21] passing through that point. In addition, there may exist noise in certain elements of and a natural idea is to estimate the edge weights by tolerating these errors. In general, a sparse solution is more robust and facilitates the consequent identification of test sample (or sequenced single-cell samples). Unlike SimplePPT, this method learns the graph by formulating the optimization as a linear programming problem.

In the same work[6], they also proposed a generalization of SimplePPT, which we term as SGL-tree (Principal Graph and Structure Learning for tree), to learn tree structure for large dataset by similarly considering clustering of data points as in DDRTree. Principal $\mathcal{L}_1$ graph and SGL-tree are all treated as SGL in this study.

2.3.3 DDRTree: Discriminative dimensionality reduction via learning a tree

DDRTree[5], the default RGE technique used by Monocle 2, provides two key features not offered by SimplePPT learning framework. First, DDRTree does not assume that the graph resides in the input space, and can reduce its dimensionality while learning the trajectory. Second, it also does not require that there be one node in the graph per data point, which greatly accelerates the algorithm and reduces its memory footprint.

Like SimplePPT, DDRTree learns a latent point for each cell, along with a linear projection function $f_{\mathcal{G}}(\mathbf{z}_i) = \mathbf{W}\mathbf{z}_i$, where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_d] \in R^{D \times d}$ is a matrix with columns that form an orthogonal basis $\{\mathbf{w}_1, \ldots, \mathbf{w}_d\}$ (D is the dimension of feature genes, d is the dimension of latent space). DDRTree simultaneously learns a graph on a second set of latent points $\mathcal{Y} = \{\mathbf{y}_k\}_{k=1}^{K}$. These points are treated as the centroids of $\{\mathbf{z}_i\}_{i=1}^{N}$ where $K \leq N$ and the

principal graph is the spanning tree of those centroids. The DDRTree scheme works by optimizing

$$\min_{\mathbf{W},\mathbf{B},\mathbf{R},\mathcal{Y},\mathcal{Z}} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{W}\mathbf{z}_i||^2 + \frac{\lambda}{2} \sum_{k,k'} b_{k,k'} ||\mathbf{W}\mathbf{y}_k - \mathbf{W}\mathbf{y}_{k'}||^2 + \gamma[\sum_{k=1}^{K} \sum_{i=1}^{N} r_{i,k}(||\mathbf{z}_i - \mathbf{y_k}||^2 + \sigma \log r_{i,k})]$$

$$\text{s.t.} \{b_{k,k'}\} \text{ is a spanning tree}$$

$$\mathbf{W}^T\mathbf{W} = \mathbf{I}$$

$$\sum_{k=1}^{K} r_{i,k} = 1, r_{i,k} \geq 0, \forall i, k$$

In effect, the algorithm acts as soft k-means clustering on points $\mathcal{Z}$, and jointly learns a graph on the K cluster centers. The matrix $\mathbf{R}$ with the (i,k)-th element as transforms the hard assignments used in k-means into soft assignments with $\sigma > 0$ as a regularization parameter. The above problem contains a number of analytical steps, and can be solved by alternating optimization until convergence. Moreover, because some of the more expensive numerical operations involve matrices that are K dimensional (instead of N dimensional), they have complexity that is invariant of the size of the input data for a small fixed K. In Monocle 2, we provide a procedure to automatically chooses a value of that should work well for a wide range of datasets based on the number of cells in the experiment:

$$K = \begin{cases} N, & \text{if} N < 100 \\ \dfrac{2 \cdot 100 \cdot log(N)}{log(N) + log(100)}, & \text{otherwise} \end{cases}$$

During the first optimization iteration, these K centroids are initialized by using k-mediods clustering in the low-dimensional space.

2.3.4 Pseudotime calculation and branch assignment

By default, Monocle 2 calls DDRTree to learn the principal tree describing a single-cell experiment, and then projects each cell onto its nearest location on the tree. Monocle 2 allows users to conveniently select a tip of the tree as the root and then transverses the tree from the root, computing the geodesic distance of each cell to the root cell, which is taken as its pseudotime, and assign branch or segment simultaneously.

DDRTree returns a principal tree of the centroids of cell clusters in low dimension. To calculate pseudotimes, Monocle 2 projects the cell's latent points , to the principal graph formed by principal points, . For latent points not near tip principal points (end nodes of the principal tree), Monocle 2 finds the nearest line segment on the principal tree and then project them to the nearest point on that segment. More formally, we can define a vector between a cell $c = (c_1, c_2, ...)$, where $c = (c_1, c_2, ...)$ denotes the coordinates of the cell in the latent space, to the nearest principal point by . The line segment formed by the two nearest principal points ($A = (A_1, A_2, ...), B = (B_1, B_2, ...)$ is $\overrightarrow{AB}$. Then we can calculate t as $t = \frac{\overrightarrow{Ac} \cdot \overrightarrow{AB}}{\|\overrightarrow{AB}\|}$. The projection can be calculated as:

$$p = \begin{cases} A, & \text{if } t < 0 \\ B, & \text{if } t > 1 \\ A + t \cdot \overrightarrow{AB} & \text{if } 0 \leq t \leq 1 \end{cases}$$

For latent points near the tip principal points, we will orthogonally project the latent point to the line segment formed by extending the tip principal point and its nearest neighbor principal point in the graph to obtain the projection point, that is, $A + t \cdot \overrightarrow{AB}$.

We then calculate the distance between all the projection points and construct a minimal spanning tree (MST) on the projection points. To avoid zero values of distance between cells projected to the same principal points, which prevents the calculation of a MST, the smallest positive distance between all cell pairs is added to all distance values. This MST is used to assign pseudotime for each cell (See below).

To encode the position of each cell within the branching structure of the trajectory, Monocle 2 performs a depth-first traversal of the principal tree learned during RGE. Without loss of generality, we assume one principal point corresponds to one latent point (for example, in the case we set or each cell corresponds to its own cluster). Following the definition introduced in (Trapnell et al. 2014b), an ordering of cells (principal points) is obtained through a depth first search (DFS) of the learned principal tree starting from the root cell. We can then assign each cell to a trajectory segment, $b_x(G, \pi, i)$ which specifies the segment b_x by where the cell i is located based on the ordering list, π , and the graph structure, G. We set $b_x = 1$ at the root cell and increase a segment counter b_x every time we reach a new branch point. More formally, we can write the formula of segment assignment as:

$$b_x(G, \pi, i) = \begin{cases} 1, & \text{if } i = 0 \\ max(b_x(G, \pi, j)),\ j \preceq i, & \text{if } |E(G_i)| \leq 2, \\ max(b_x(G, \pi, j)) + 1\ j \preceq i, & \text{if } |E(G_i)| \geq 3 \end{cases}$$

where $j \preceq i$ represents all precedents j of i in the ordering π, $|E(G_i)|$ represents the degree of cell i. T. For the general cases where the principal points is less than the cell numbers, cells will inherit the segment assignment of their nearest principal point.

Similar to our previous definition of pseudotime (Trapnell et al. 2014b), Monocle 2 calculates pseudotime based on the geodesic distance of each cell to the root cells on the MST of the projection points. Define pseudotime of cell i from a branching biological process s with branches given by b_x as $\varphi_t(b_x, s_i)$, we can calculate its pseudotime recursively by adding the pseudotime of its parent cell on the MST of the projection points (closest cell on the same branch) with the Euclidean distance, $\|\overrightarrow{p}(b_x, s_i), \overrightarrow{p}(\text{Parent}(b_x, s_i))\|_2$, between current and the parent on the MST, by setting the root cell as pseudotime 0. That is,

$$\varphi_t(b_x, s_i) = \begin{cases} 0, & \text{if } b_x = 1, i = 0 \\ \varphi_t(\text{Parent}(b_x, s_i)) + \|\overrightarrow{p}(b_x, s_i), \overrightarrow{p}(\text{Parent}(b_x, s_i))\|_2, & \text{if } i > 0 \end{cases}$$

where $\overrightarrow{p}(b_x, s_i)$ defines the projection coordinates of cell i on the embedded tree.

2.4 RGE is not limited to learning tree structure

Monocle 2 uses DDRTree (Mao, Wang, et al. 2015; Mao et al. 2016a), a scalable RGE algorithm, to learn a principal tree on a population of single cells by default, asserting that it describes the sequence of changes to global gene expression levels as a cell progresses through the biological process under study (**Figure 1A**). In contrast to other methods (Trapnell et al. 2014b; Setty et al. 2016; Haghverdi et al. 2016a; Welch, Hartemink, and Prins 2016), Monocle 2 identifies branch points that describe significant divergences in cellular state automatically. Monocle is also

equipped with alternative RGE methods(Mao, Wang, et al. 2015; Mao et al. 2016a) including one that in principle can learn cyclical or disjoint trajectories, though doing so requires some degree of parameter optimization on behalf of the user.

2.5 Benchmark Monocle 2 with other alternative algorithms

To assess the Monocle 2's accuracy, we first applied it to myoblasts, which we previously reported to differentiate along a linear trajectory(Trapnell et al. 2014b) (**Figure 1B**). Surprisingly, Monocle 2 reconstructed a trajectory with a single branch point leading to two outcomes (**Figure 1C**). Some genes associated with mitogen withdrawal, such as *CCNB2* showed similar kinetics on both branches, but a number of genes required for muscle contraction were strongly activated only on one of the two branches of the Monocle 2 trajectory (**Supplementary Figure 4**). A global search for genes with significant branch-dependent expression using Branch Expression Analysis Modeling (BEAM)(Qiu, Hill, et al. 2017a) revealed that cells along these two outcomes, F_1 and F_2, differed in the expression of 887 genes (FDR < 10%), including numerous components of the contractile muscle program. The BEAM analysis suggested that only outcome F_1 represented successful progression to fused myotubes (**Supplementary Figure 4**), consistent with immunofluorescence measurements of *MYH2*, which show a substantial fraction of isolated nuclei lacking MYH2 and that are not incorporated into myotubes (ref. **Figures 1** and **4** of(Trapnell et al. 2014b)).

A simulation of differentiation driven by a set of stochastic differential equations controlled by a hypothetical gene regulatory network(Qiu, Ding, and Shi 2012b) demonstrated that Monocle 2 robustly and accurately reconstructed trajectories with up to three fates

(**Supplementary Figure 5-8, Supplementary Data 1, 2**)(Y. Tang et al. 2016). In contrast to other methods, Monocle 2 also accurately learned a complex tree with five branches in a fully automatic fashion (**Supplementary Figure 6B, Supplementary Data 3**).

We next sought to compare Monocle 2 to state-of-the art algorithms for inferring single-cell trajectories, including Monocle 1(Trapnell et al. 2014b), Wishbone(Setty et al. 2016), Diffusion Pseudotime (DPT)(Haghverdi et al. 2016a), and SLICER(Welch, Hartemink, and Prins 2016). Unlike Monocle 2, these methods do not construct an explicit tree. Instead they order cells based on pairwise geodesic distances between them as approximated by a nearest-neighbor graph (Wishbone and SLICER) or minimum spanning tree (Monocle 1) or calculated analytically (DPT). Wishbone, SLICER, and DPT identify branches implicitly by analyzing patterns in the pseudotime orderings that are inconsistent with a linear trajectory. Furthermore, Wishbone assumes the trajectory has exactly one branch point, while DPT can detect more than one, but provides no means of automatically determining how many genuine branches exist in the data. We hypothesized that Monocle 2's explicit trajectory structure would yield more robust pseudotimes and branch assignments than alternative algorithms.

We tested each algorithm using data from Paul et al, who analyzed transcriptomes of several thousand differentiating blood cells(Paul et al. 2015). Monocle 2, DPT, and Wishbone produced qualitatively similar trajectories, with CMP cells residing upstream of a branch at which GMP and erythroid cells diverge (**Supplementary Figure 9-11**). SLICER generated a branched trajectory in which the branch occurs within the erythroid population to bifurcate into either CMPs or GMPs. Monocle 2, Wishbone, and DPT produced orderings that were highly correlated with a "reference ordering", constructed using a panel of markers similar to the

approach introduced by Tirosh et al(Tirosh et al. 2016a), while SLICER and Monocle 1 were less so. Monocle 2 assigned cells to branches as or more accurately than other methods (**Figure 1D, Supplementary Figure 10**), but Monocle 2's assignments were far more consistent when provided with subsampled fractions of the cells (**Figure 1E, Supplementary Figure 9F,G**). When run on the myoblast data, DPT positioned most fully differentiated cells along a major branch, with incompletely differentiated cells split along a minor branch or not assigned to either, while Wishbone failed to discriminate correctly between the two outcomes (**Supplementary Figure 12**). Although Monocle 2 can be tuned for several user-specified parameters, its results were similar to the defaults over widely varying values (**Supplementary Figures 13-14**). Monocle 2's running time scaled linearly in the number of input cells, consistent with its linear algorithmic complexity, processing 8365 cells in 9 minutes (**Supplemental Figures 13C**) These benchmarks demonstrate that Monocle 2 produces trajectories that are as accurate and more robust than state-of-art methods and yet makes fewer assumptions regarding the number of cell fates generated by the trajectory.

We also assessed Monocle 2's alternative algorithms for dimensionality reduction and graph learning. DDRTree, SimplePPT and SGL-tree, which implement RGE to learn principal trees reported highly concordant trajectories when the data was initially reduced with PCA, ICA, and diffusion maps (**Supplementary Figure 15**. LLE, a reduction technique known to be highly sensitive to tuning parameters, sometimes led to incorrect reconstructions with SimplePPT. L1-graph, an RGE algorithm that can learn graphs with multiple components or cycles, often reported less refined graphs with numerous minor branches, but captured the overall trajectory structure accurately.

2.6 Monocle 2 learns complex developmental trajectories

Because Monocle 2 can in principle learn complex trajectories with many branches, we reanalyzed the data from Paul et al. in 10 dimensions (selected based on variance explained by PCs) rather than the default of two. This higher-dimensional trajectory contained five branching events leading to six different outcomes, with cells classified by Paul et al. as fully differentiated monocytes, neutrophils, eosinophil, basophils, dendritic cells, megakaryocytes, and erythrocytes confined to distinct outcomes (**Supplemental Figure 16**). Thus, Monocle 2 can resolve complex branching processes.

Although Monocle 2's trajectories for differentiating myoblasts and common myeloid progenitors were broadly consistent with the known sequence of regulatory events governing those processes, we sought further experimental means of validating the structure of the algorithm's trajectories. Recently, Olsson *et al* profiled several hundred FACS-sorted cells during various stages of murine myelopoeisis, *i.e.* LSK, CMP, GMP and LKCD34+ cells. We analyzed these cells with Monocle 2 and reconstructed a trajectory with two major branches and three distinct fates (**Figure 2, Supplementary Figure 17, 18**). Lin-Sca1+c-Kit+ (LSK) cells were concentrated at one tip of the tree, which we designated the root, with CMP, GMP, and LKCD34+ cells distributed over the remainder of the tree (**Figure 2A, Supplementary Figure 17A**).

Monocle 2 placed cells classified as erythrocytes or megakaryocytes on a path to outcome F_E, while granulocytes and monocytes by Olsson *et al* were confined to outcomes F_G and F_M respectively. Genes associated with the granulocytic and monocytic programs became

progressively more differentially expressed following the second branch (**Supplementary Figure 17B, C**). Many of the genes with significantly branch-dependent expression (BEAM test(Qiu, Hill, et al. 2017a), FDR < 1%), were bound at their promoters by Irf8 or Gfi1, key activators of the monocytic and granulocytic expression programs, respectively (**Supplementary Figure 17D, E**).

2.7 Mutation of myelopoiesis regulators diverts cells to alternative branches

Providing cells from mice lacking *Gfi1* or *Irf8* to Monocle 2 did not substantially alter the structure of the myeloid differentiation trajectory (**Figure 2B**). However, cells from *Gfi1*-- mice were largely excluded from the branch occupied by wild-type granulocytes, and Irf8-- cells were depleted from the wild-type monocyte branch. That is, the loss of a gene known to activate a fate-specific expression program appeared to divert cells to the opposite fate. Cells from double knockout mice (*Gfi1*-- *Irf8*--) were present on both monocyte and granulocyte branches, but concentrated closer to the branch point and away from the tips of the tree, suggesting that they did not fully differentiate (**Supplementary Figure 19A**).

Testing whether *Gfi1*-- or *Irf8*--had fully adopted the monocyte and granulocyte expression programs, revealed that *Gfi1*-- cells on the branch to F_M express higher levels of genes from normally associated with granulocytes than wild-type monocytes (**Supplementary Figure 19, Methods)**. Likewise, cells from *Irf8*-- mice on the branch to F_G showed aberrantly high levels of monocytic genes. Analysis of genetic perturbations from the large-scale transcriptomic study of hematopoiesis reported by Paul *et al* also revealed diversions of cells

onto specific branches of the trajectory, suggesting that diversion of cells from one fate to another may be a consequence of losing a key fate regulator (**Supplementary Figure 19G, H)**.

In addition to known differentiated cell types, Olsson *et al.* detected cells that express a mix of genes specific to different terminal cell fates. They also reported rare, transient cell states that in which hematopoietic multipotent markers coexist with differentiated markers. They concluded that both types of "mixed lineage" cells reside in the developmental hierarchy downstream of long-term and short-term HSCs but upstream of cells that have committed to a lineage. Consistent with this interpretation, Monocle 2 positioned mixed-lineage cells and rare transient cells (**Supplementary Figure 20**) upstream of the granulocyte-monocyte branch.

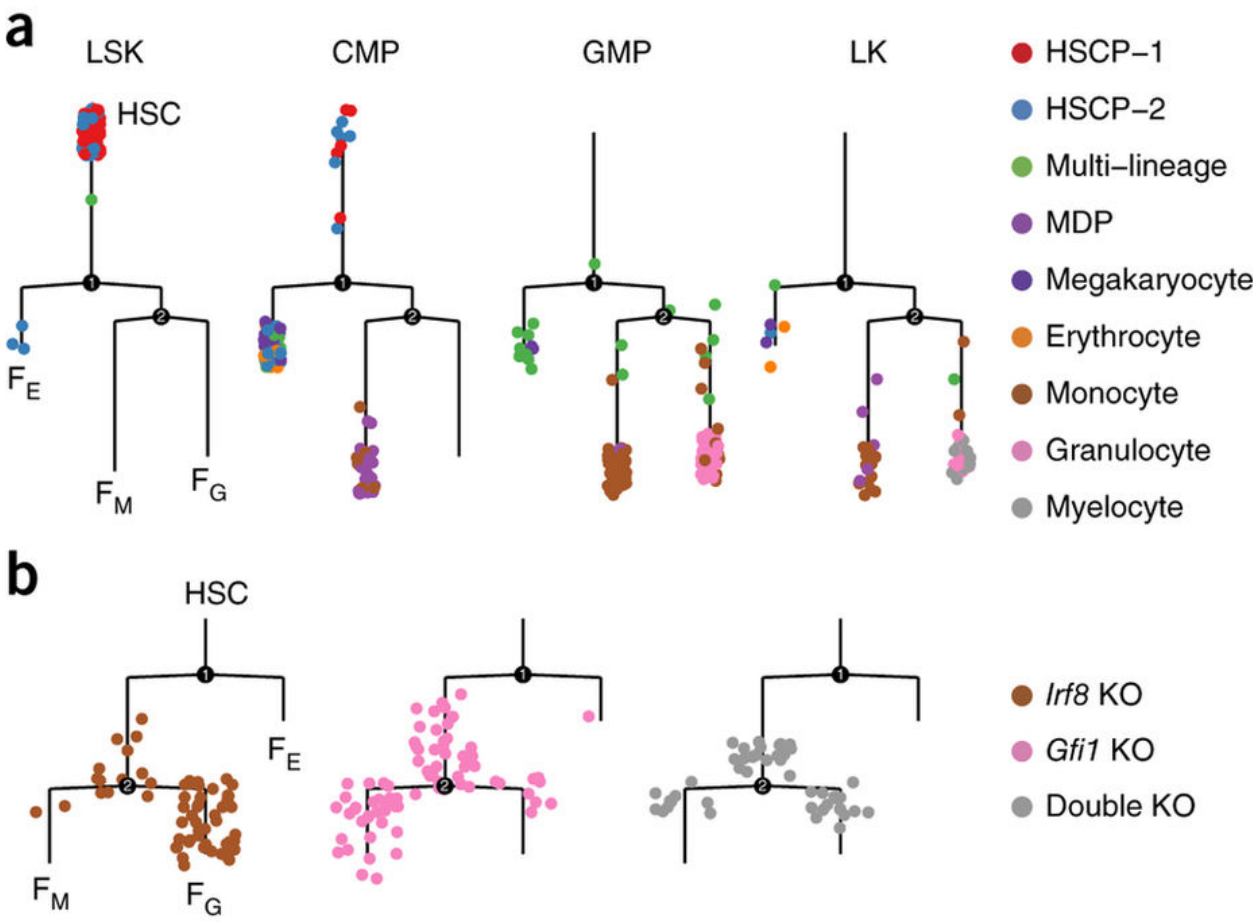


Figure 2.2. Genetic perturbations divert cells to alternative outcomes in Monocle 2 trajectories. (**A**) Monocle 2 trajectory of differentiating blood cells collected by Olsson et al(Olsson et al. 2016). Each subpanel corresponds to cells collected from a particular FACS gate in the experiment. Cells are colored according to their classification by the authors of the original study. (**B**) Cells with a single knockout of Irf8 or Gfi1 are diverted into the alternative granulocyte or monocyte branch, respectively. Double knockout cells are localized to both granulocyte and monocyte branches but concentrated near the branch point. Two branch points are identified, one that divides the erythroid or megakaryocyte outcome (F_E) from the granulocytemonocyte progenitors (GMP), which then branches to the monocyte (F_M) and

granulocyte (F_G) outcomes. All trajectories are reconstructed in four dimensions selected based variance explained by each PCA but rendered in two dimensions using layout_as_tree() from the igraph package.

2.8 Discussion

Single-cell RNA-Seq has spurred an explosion of computational methods to infer the precise sequence of gene regulatory events that drive transitions from one cellular state to another. However, most current methods rely on strong assumptions about the structure of a biological trajectory. Many also require the user to supervise trajectory inference, inject large amounts of *a priori* biological knowledge, or both.

Monocle 2 learns complex cellular trajectories with multiple branches in a fully data-driven, unsupervised fashion with only limited assumptions regarding its structure. It employs a class of manifold learning algorithms that aim to embed a principal graph amongst the high-dimensional single-cell RNA-seq data. In contrast to previous methods that infer branch structure using heuristic analyses of pairwise distances between cells, Monocle 2 can use this graph to directly identify developmental fate decisions. We have demonstrated through extensive benchmarking that Monocle 2 compares favorably with other tools such as Wishbone without requiring the user to specify the structure of the trajectory.

Analysis of multiple real and synthetic datasets demonstrated that Monocle 2 reconstructs trajectories that faithfully characterize cellular differentiation. Previously, we showed that loss of interferon signaling can create a new branch in an otherwise linear trajectory that reflects the response of dendritic cells to antigen(Qiu, Hill, et al. 2017a). Here, we show that cells from mice

that lack transcription factors required for establishing specific myeloid fates were diverted onto alternative fates of the same trajectory without altering its structure. Why some loss of function mutations create branches while others divert cells along existing ones is unclear, but this question underscores the increasing power of analyzing single-cell trajectories. We also anticipate that Monocle 2 will be useful not just for expression data, but for single-cell chromatin accessibility(Cusanovich et al. 2015) or 3D structure(Ramani et al. 2017) analysis as well. We are confident that Monocle 2 will help reveal how various layers of gene regulation coordinate developmental decision making within individual cells.

Supplementary Figures

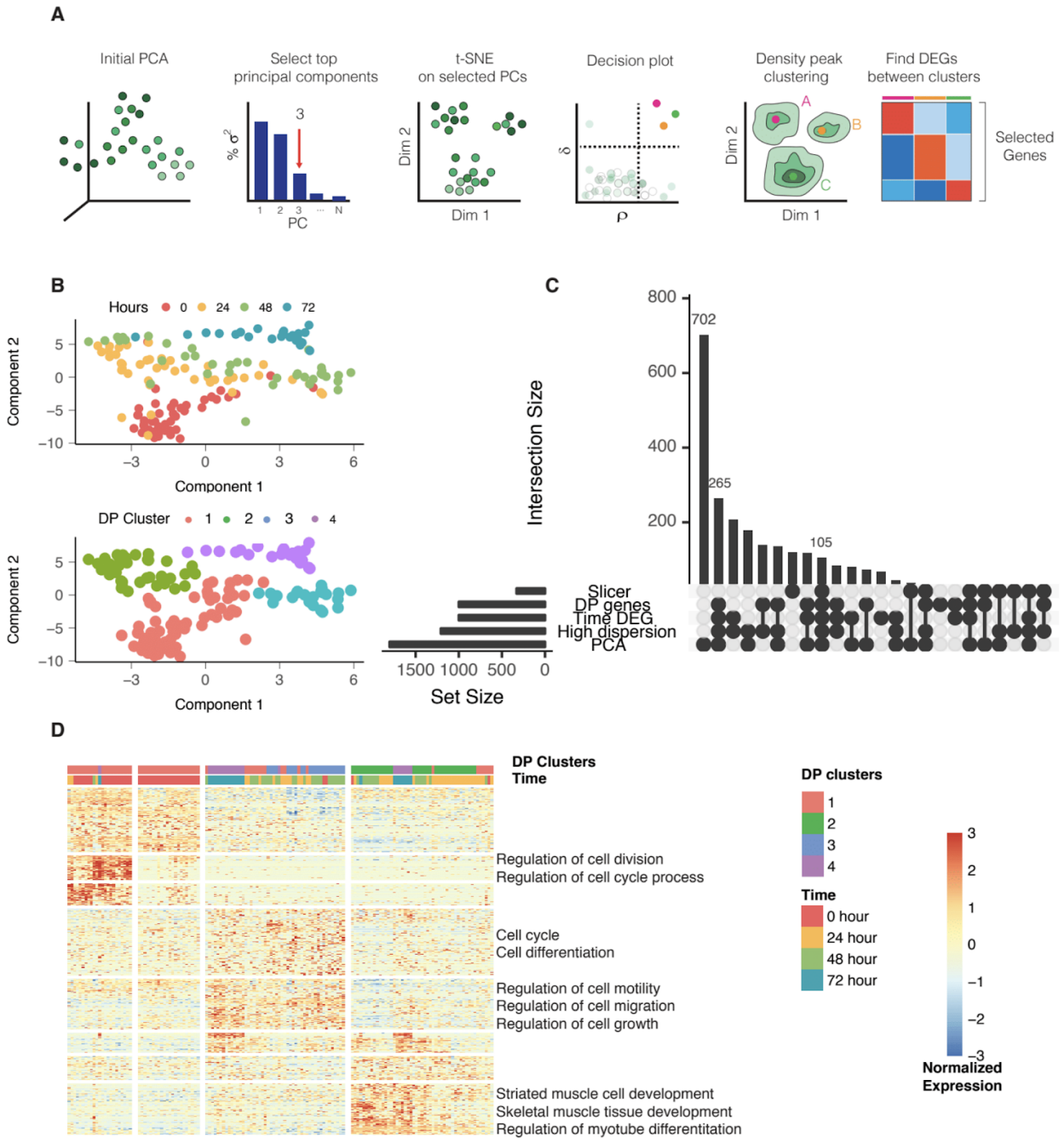
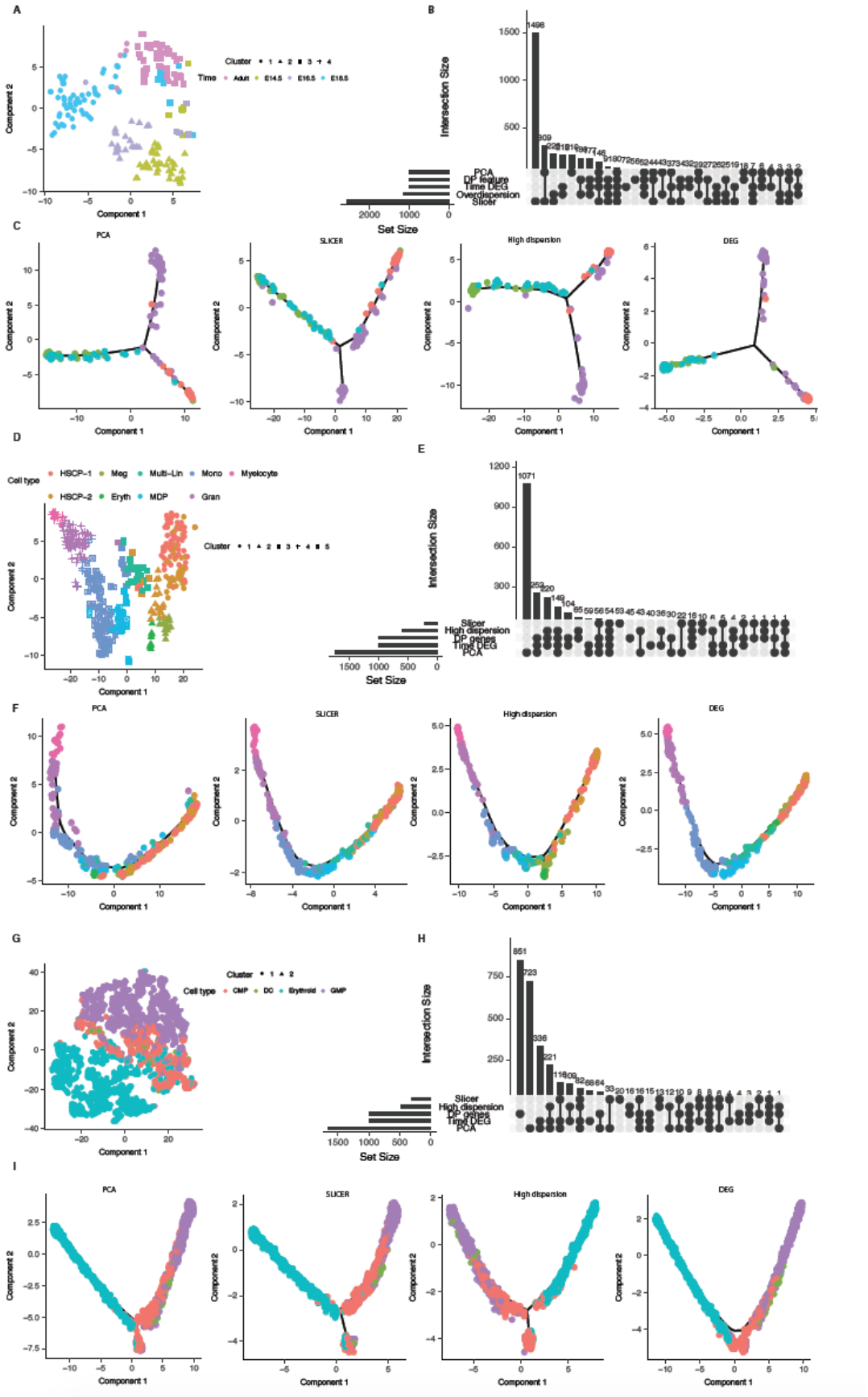
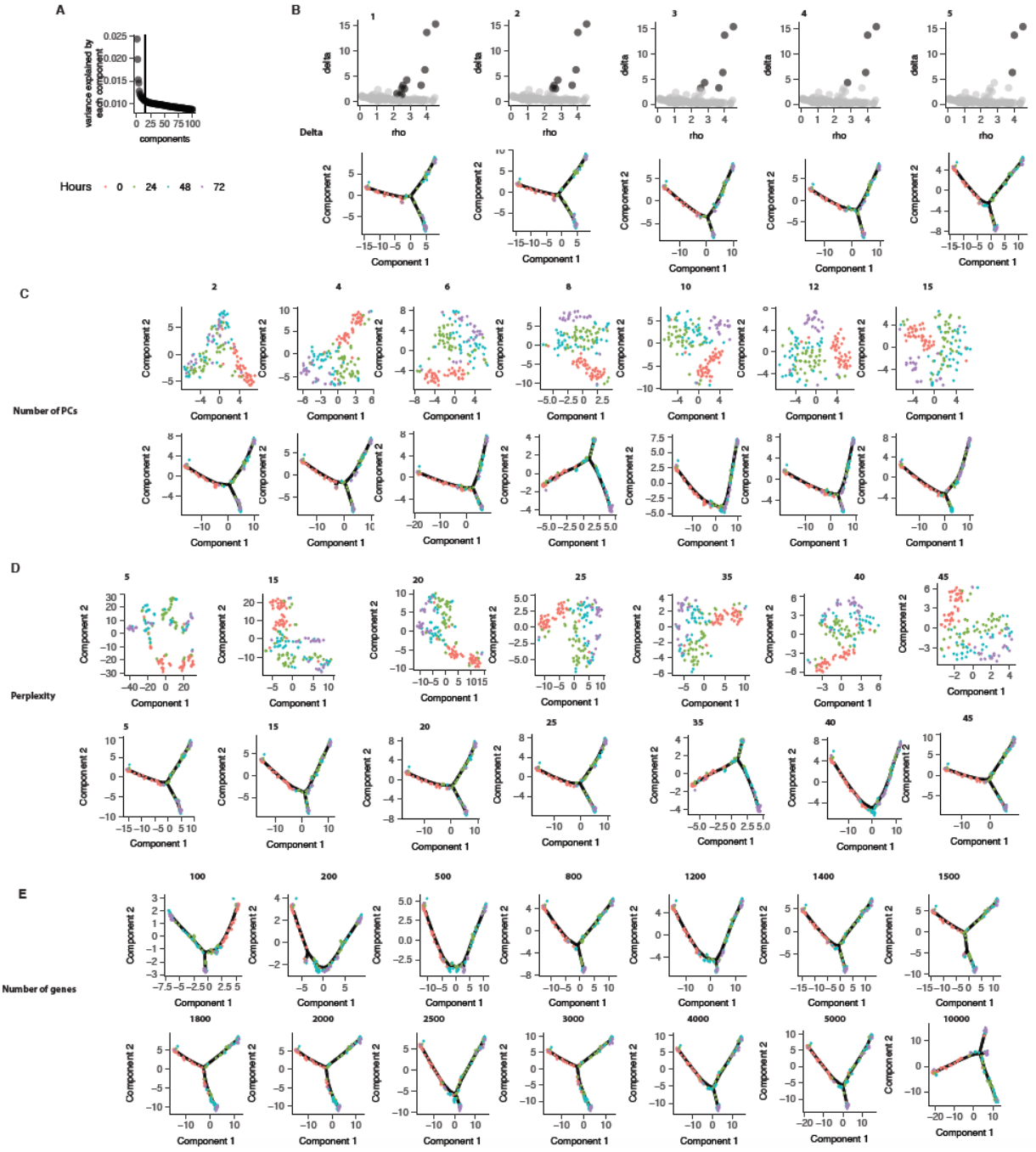


Figure 1: dpFeature detects important genes associated with biological processes. (A) Flowchart of unsupervised feature selection based on density peak clustering (dpFeature). The density peak algorithm 9 is used to cluster cells in a two-dimensional representation generated by t-SNE. Genes that are significantly differentially expressed between clusters are then selected for downstream trajectory inference in Monocle 2. (B) tSNE dimension reduction based on top principal components (PCs) and density peak clustering for the HSMM data. (C) Five sets of ordering genes are shown: genes that are highly loaded on the first 2 principal components ("PCA"), have high dispersion relative to the mean ("High dispersion"), significantly differ between time points ("Time DEG"), selected via the procedure in SLICER, or identified by dpFeature ("DP genes"). The UpSet plot 11 plot shows the number of genes returned by each method (bottom left), along with the size of the possible gene set intersections (up right). (D) Clustered heatmap of genes from dpFeature along with selected enriched Gene Ontology terms. Relative transcript counts for each gene (rows) are scaled across cells (columns) and thresholded between the range -3 and 3.

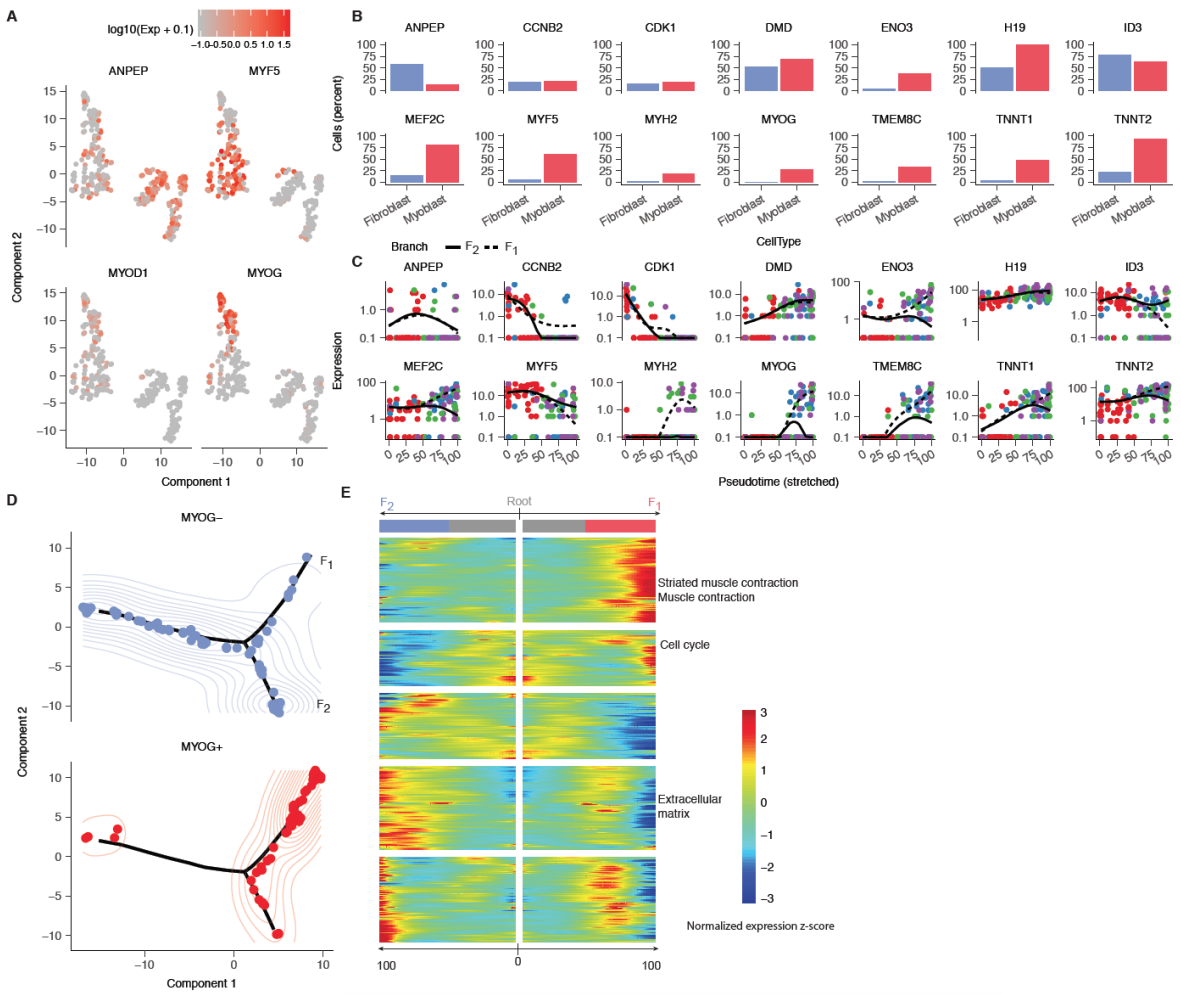

A
Component 1
Component 2
Cluster
Time
Adult
E14.5
E16.5
E18.5
B
Intersection Size
Set Size
PCA
DP feature
Time DEG
Overdispersion
Slicer
C
PCA
SLICER
High dispersion
DEG
D
Cell type
HSCP-1
HSCP-2
Meg
Eryth
Multi-Lin
MDP
Mono
Gran
Myelocyte
E
F
G
CMP
DC
Erytroid
GMP
H
I

Supplementary Figure 2. dPFeature shapes the reconstruction of developmental trajectories by selecting informative genes. (**A**) tSNE plot from dpFeature clusters cells from lung data(Treutlein et al. 2014) into four different clusters. Color corresponds to sample collection time points while shape corresponds to the cluster assignment. (**B**) UpSet plot of the ordering genes selected by various procedures, similar to **Supplementary Figure 1C.** (**C**) Differentiation trajectories learned with each set of ordering genes. (**D-I**) Similar analysis for the hematopoietic data reported by Olsson et al. (panels **D-F**) and Paul et al. (panels **G-I**).



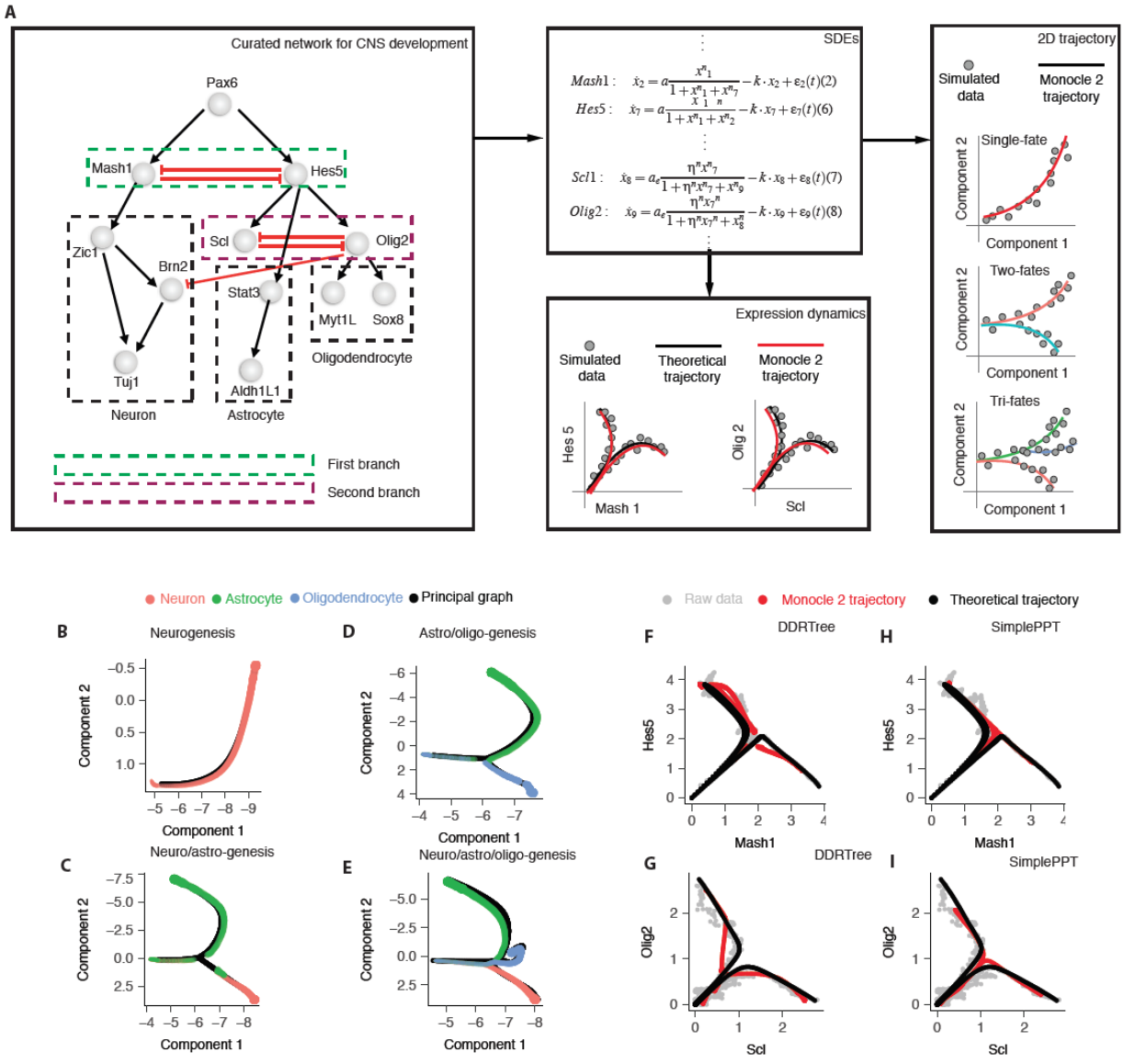
Supplementary Figure 3. Monocle 2 robustly reconstructs the developmental trajectory for HSMM data over a large range of values for parameters used in dpFeature. (**A**) Variance explained by each PCA component for the HSMM dataset. The vertical line corresponds to the

15th PC. The legend for hours is used across all the other panels. (**B-E**) Each panel shows trajectories produced by Monocle 2 while varying one parameter and holding the others fixed at the values specified in the HSMM analysis section (**Methods**). **B:** Trajectory reconstructed under different values for parameter *delta* used in the density peak clustering step of dpFeature. **C**: Trajectory reconstructed under different value for parameter *number of PC components* used in tSNE dimension reduction step of dpFeature. **D**: Trajectory reconstructed under different value for parameter *perplexity* used for tSNE dimension reduction step. **E**: Trajectory reconstructed under different value for parameter *number of genes* used for trajectory reconstruction.



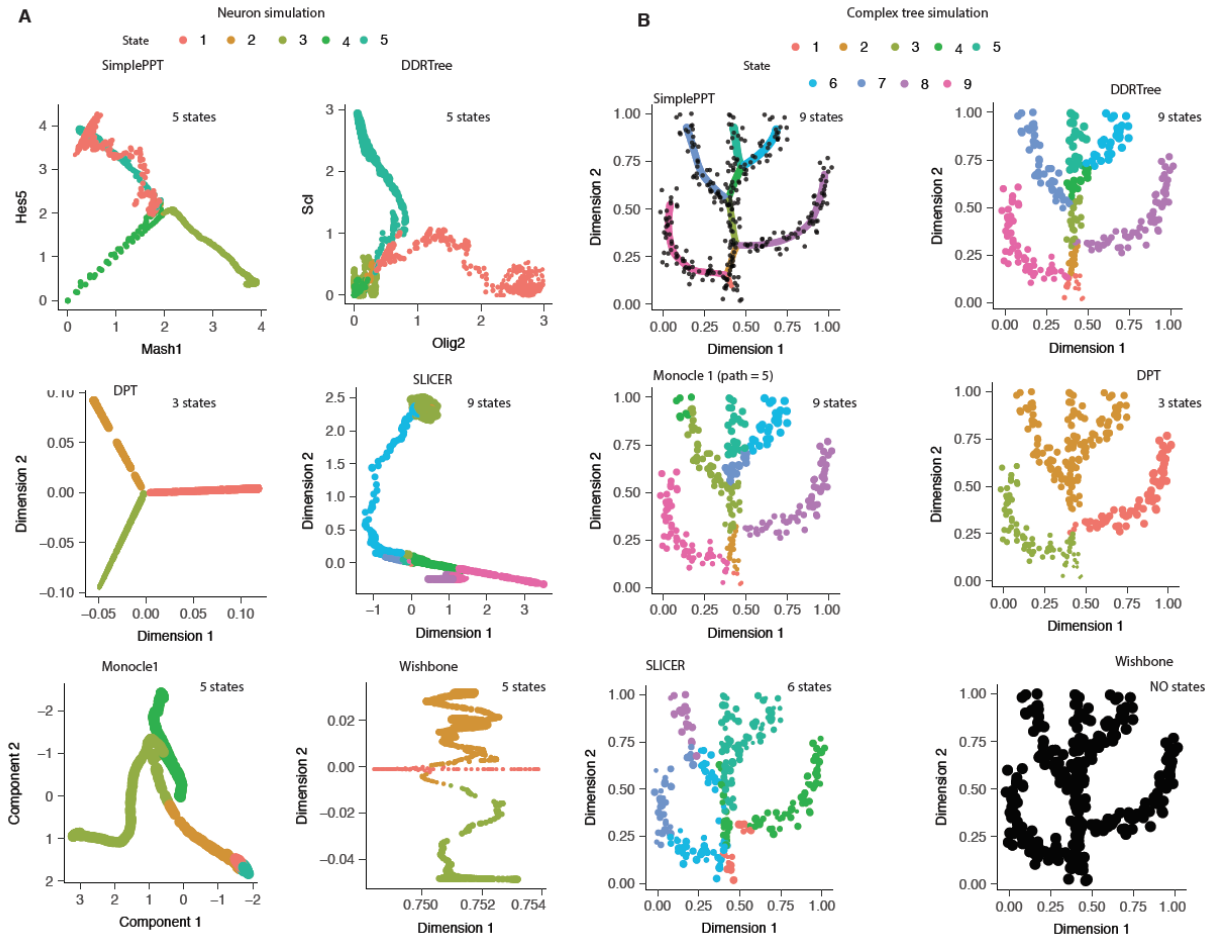
Supplementary Figure 4. Myoblasts differentiate along a branched trajectory. (**A**) tSNE plot from dpFeature clusters cells from HSMM data into two major clusters. Cells in the rightupper express a fibroblast-associated gene, ANPEP, while the leftlower cluster has numerous cells expressing muscle-specific genes (MYF5, MYOD1, MYOG). (**B**) Percentage of cells expressing selected muscle or fibroblast-associated genes.(**C**) Branch expression curves for genes in panel **A**. Dashed line indicates the spline fit for cells on the path from the root of the tree in **Figure 1C** to outcome F_1, while the solid line indicates the curve for the path to F_2. (**D**) Distribution of cells detectably expressing MYOG along the trajectory. (**E**) Branch kinetic heatmap of significantly

branch-dependent genes (BEAM test; FDR < 10%) and selected gene ontology categories enriched in groups of genes with similar kinetics.

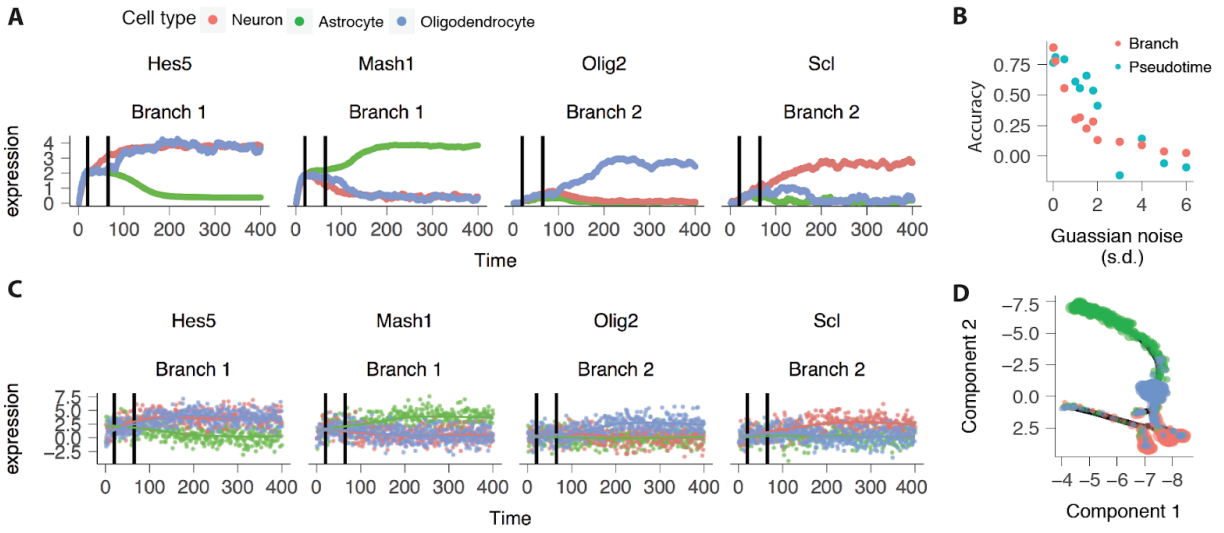


Supplementary Figure 5: Monocle 2 correctly recovers trajectories driven by simulated gene regulatory networks. (**A**) A hypothetical gene regulatory network and system of stochastic differential equations to drive three-way cell fate specification. The transcriptional regulatory

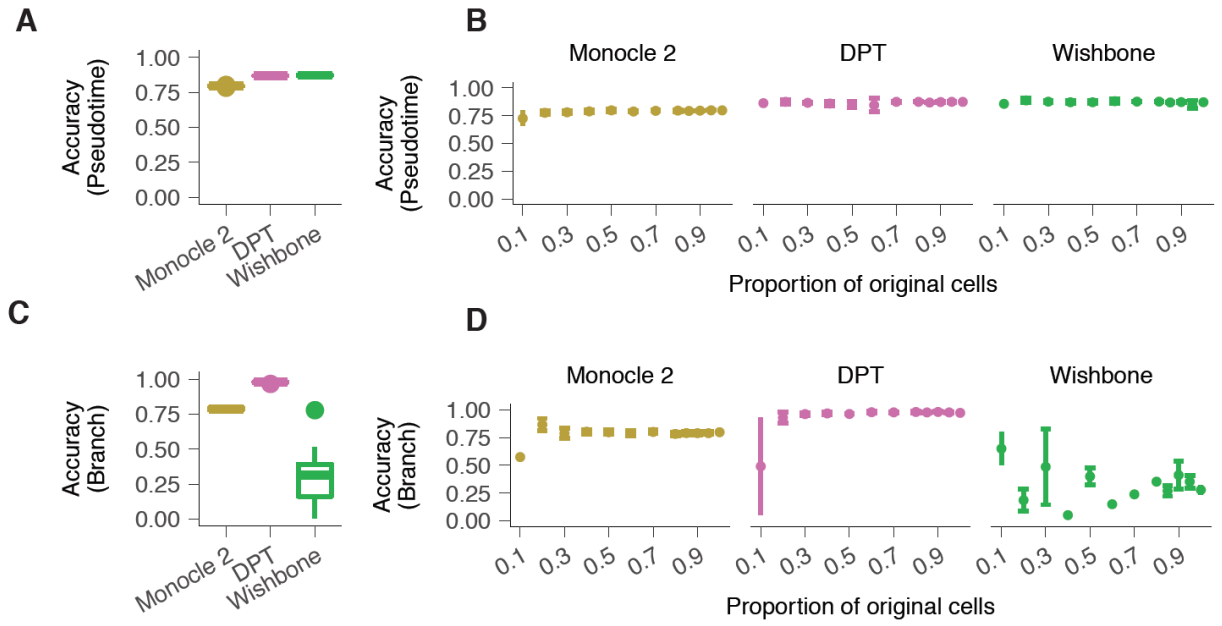
network, modified from Qiu et al(Qiu, Ding, and Shi 2012c), specifies neural progenitor cells to either neurons, astrocytes or oligodendrocytes through a pair of mutual inhibition interactions(Qiu, Ding, and Shi 2012b). The network summarizes a set of stochastic differential equations that describe gene expression dynamics over time. Initializing this network with small amounts of stochastic noise and following expression kinetics over time simulates the trajectory followed by a single cell, which can be compared to the ideal theoretical trajectory (Y. Tang, Yuan, and Ao 2014). (**B-E**) Monocle 2 trajectories learned on four ensembles of simulated data points. (**F-I**) Reverse embed (see **Methods**) the lower dimensional principal graph learned by DDRTree back the original gene expression space (**F, G**) or using principal graph from SimplePPT in the same dimension (**H, I**) along with the theoretical trajectory visualizes branching kinetics of individual regulators in the network.



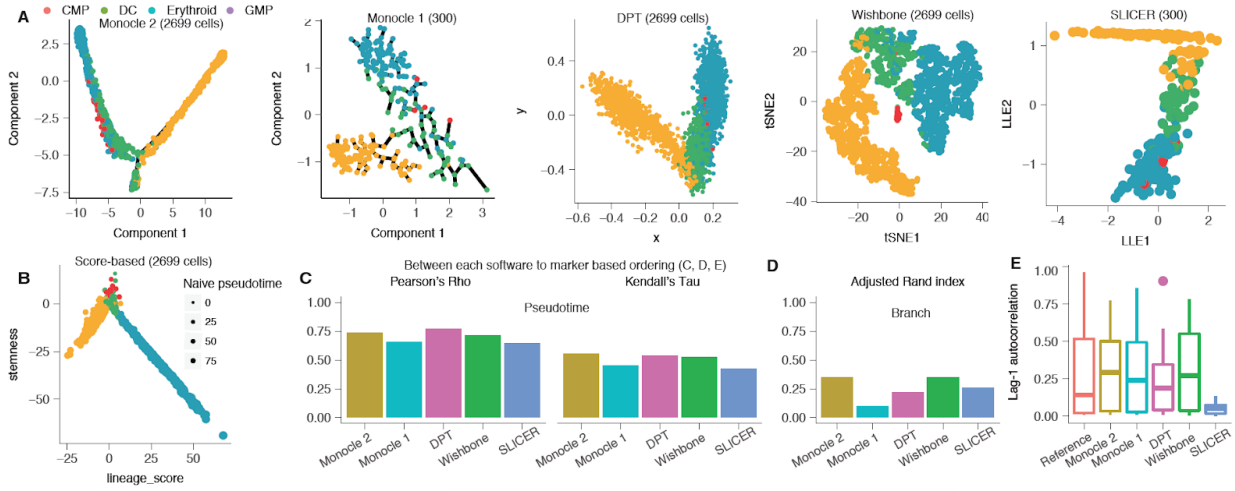
Supplementary Figure 6. Reversed graph embedding accurately learns complex trajectories in simulation datasets. (**A**) Comparison of the trajectory and branch assignment learned by several programs (Monocle 2, either with DDRTree or SimplePPT, DPT, SLICER, Monocle 1, and Wishbone) for the simulated two-branch neuro/astro-oligo-genesis process. (**B**) Comparison of each program for learning a complex tree structure from (Mao et al. 2016b).



Supplementary Figure 7. Robustness of Monocle 2 over increasing levels of measurement noise in expression data. (A) The black vertical lines indicate the branch point pseudotimes in the simulation from **Supplementary Figure 5**. Color represents the cell types. The same colors are used in panels C and D. **(B)** Accuracy, measured by the pseudotime Pearson correlation and the branch assignment ARI under different levels of Gaussian noise. The x-axis represents the standard deviation of the added Gaussian noise. The real simulation time and branch assignments determined by manual inspection of bifurcation point of master regulator pairs (Mash1-Hes5, Scl-Olig2) in the simulation as shown in panel **A** is used as reference (also used for **Supplementary Figure 8**). (**C**) Expression dynamics of master regulators under noise level of S.D. of 2. (**D**) Trajectory learned from Monocle 2 under noise level of S.D. of 2.

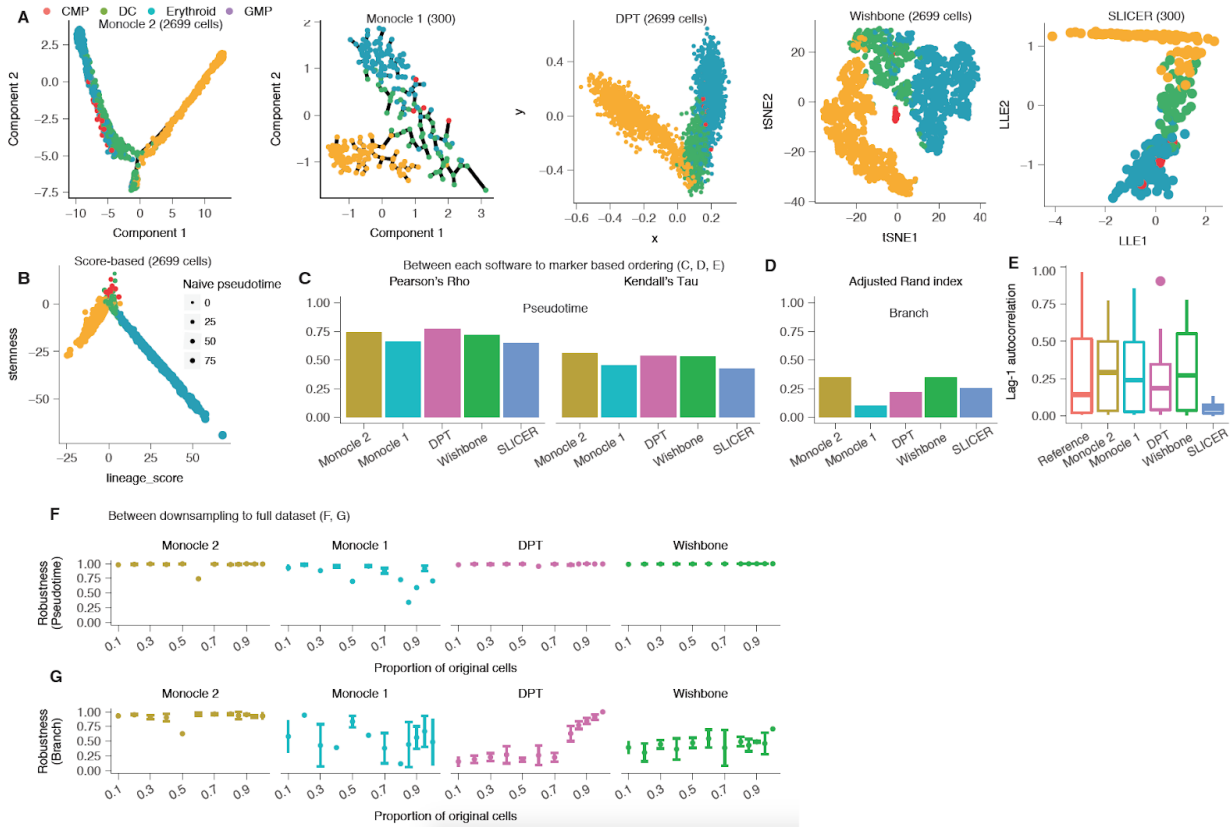


Supplementary Figure 8. Accuracy of trajectory reconstruction algorithms on the simulation dataset. (**A, B**) Accuracy of pseudotime assignment of Monocle 2, DPT, Wishbone over repeated downsampling or progressively smaller fractions of cells from the simulation in **Supplementary Figure 5**. Pearson correlation between each algorithm's pseudotime assignments and the true simulation times at a depth of 80% of original dataset (**A**) or progressive downsampling (**B**) from 10% to 100% of the full dataset. (**C, D**) Accuracy of branch assignment under repetitive downsampling (**C**) or progressive downsampling (**D**).



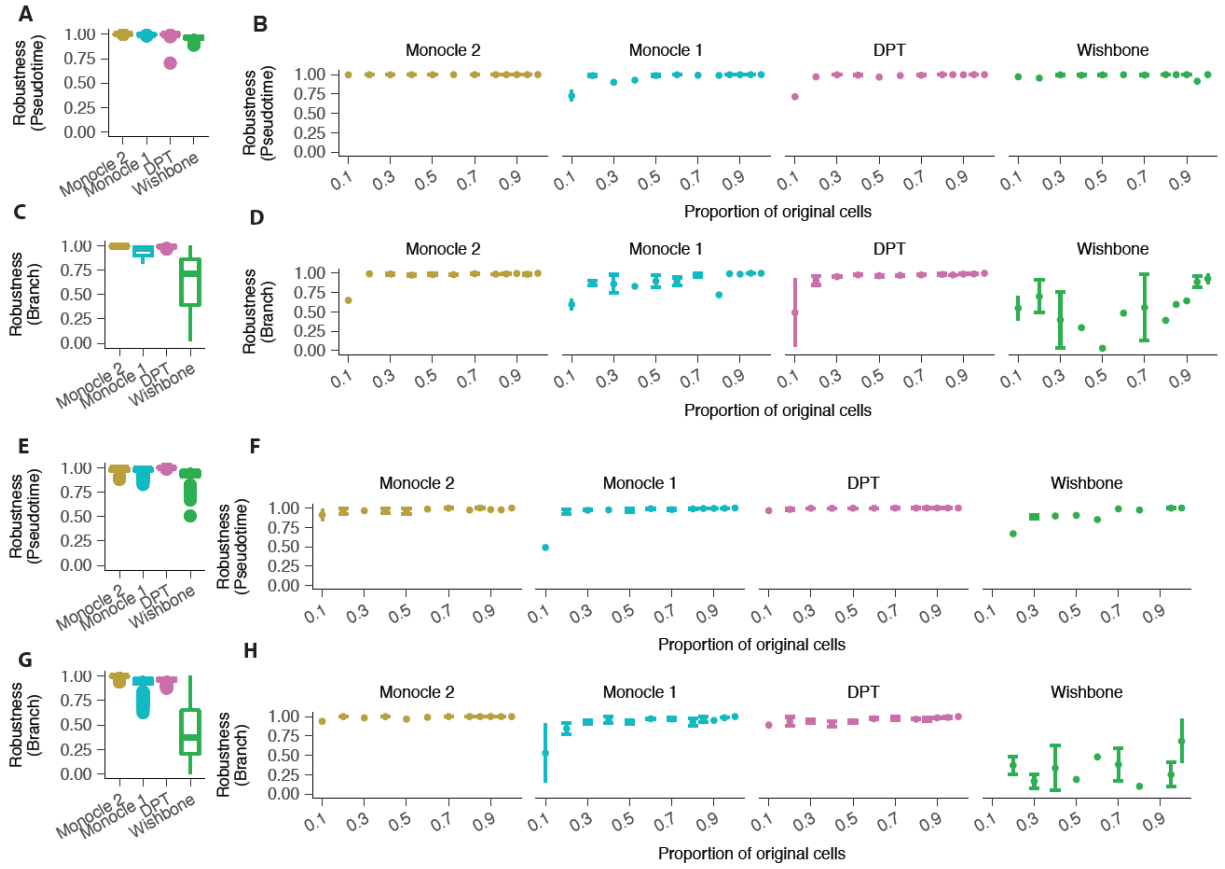
Supplementary Figure 9. Monocle 2 accurately and robustly reconstructs the major erythroid and myeloid bifurcation in hematopoiesis. The data from Paul et al. was provided to each of several programs for trajectory analysis. (**A**) Representation of bifurcation of common myeloid progenitors (CMP) into either erythroid cells or granulocyte-macrophage progenitors (GMP), with trajectories inferred by Monocle 2 and other algorithms. (**B**) Representation of bifurcation of CMP into either erythroid or GMP branch with the score-based "reference trajectory" approach of Tirosh et al (Tirosh et al. 2016b). (**C**) Correlation between the reference trajectory and each tool. (**D**) Adjusted Rand index between the branch assignments from the reference trajectory and branch assignments from each algorithm. (**E**) Lag-1 autocorrelation for the fitted spline curve based on pseudotime from each method for all marker genes used in the score-based ordering. (**F, G**) Robustness analysis of pseudotime calculation (**F**) or branch assignment (**G**) for each algorithm run with progressively smaller fractions of the cells. For panels **C-, E**, the same 300 random sampled cells are used for all software. Because Monocle 1 often fails on datasets with more than 1,000 cells, in **F, G**, Monocle 2, DPT, and Wishbone all use the full dataset for benchmark while Monocle 1 only uses the same random downsampled

300 cells as above for benchmarking. Only common cells between downsampling and the full dataset (**F, G**) is used for pseudotime's robustness (*Pearson's Rho*) or branch's robustness (*Adjusted Rand index calculations*) assessment.



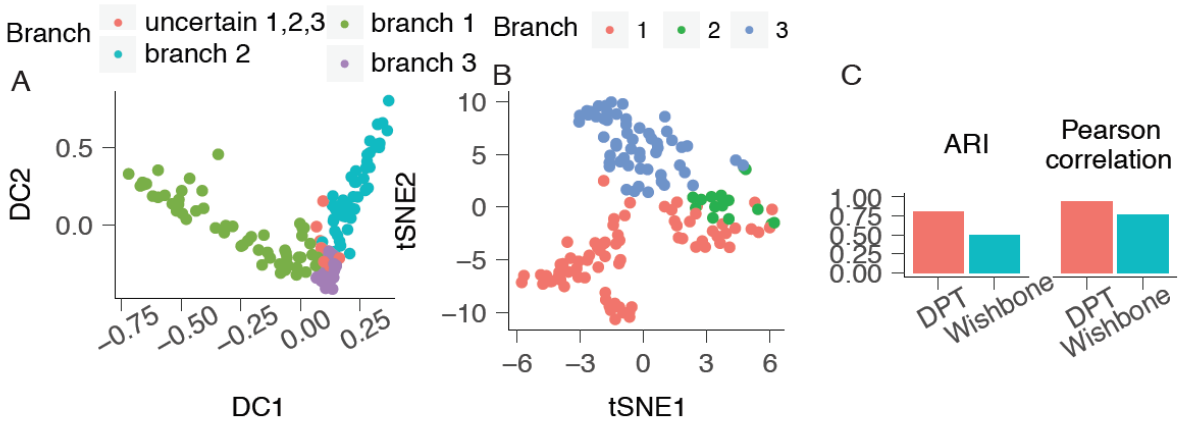
Supplementary Figure 10. Accuracy of trajectory reconstruction algorithms on Paul et. al dataset. (**A**) Same analysis as in **Supplementary Figure 8B** on the Paul dataset with pseudotime from marker based ordering as the reference. (**B**) Same analysis as in **Supplementary Figure 8D** on the Paul dataset with cell type assignment (CMP, GMP or erythroid) suggested by the original study as the reference. For Monocle 1, we used the same random sampled 300 cells from

Supplementary Figure 9 as the universe for benchmark (marker based ordering is also applied on this set of sampled cells).



Supplementary Figure 11. Robustness of trajectory reconstruction algorithms on additional datasets. Robustness of Monocle 2 trajectories for the simulation dataset in **Supplemental Figure 5 (A-D)** and the lung data from Truetlein et al (**E-H**). Robustness of (**A, E**) pseudotime (Pearson's Rho) or adjusted Rand Index of branch assignments (**C, G**). Values shown are for pairwise comparisons of different downsamples of 80% of the full set of cells. (**B,D**): Robustness of pseudotime assignment (by Pearson correlation) relative to results with the full dataset for

Monocle 2, DPT and Wishbone under progressive downsampling from 10% to 100% of the cells. (**D, H**) Same analysis as in (**B, D**) but for robustness of branch assignment (by ARI).



Supplementary Figure 12. Trajectory analysis of myoblast differentiation with alternative approaches. (**A**) DPT's trajectory and branch assignment for the HSMM dataset. The first and second diffusion components are used for visualization. (**B**) Wishbone's trajectory and branch assignment for the HSMM dataset. First and second tSNE components are used for visualization. (**C**) Adjusted Rand index (ARI) for branch consistency and pseudotime's Pearson correlation between branch assignment and pseudotime from DPT, Wishbone to that from Monocle 2.

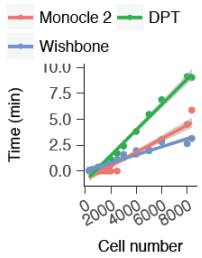
A

	Meaning	Effects of parameters
Dimension	Number of latent space's dimension	Principal tree can be constructed in more than two dimensions to capture more variance of the data
Lambda	Regularization parameter for inverse graph embedding	This parameter controls the length of tree structure. The larger the parameter is, the smaller the length of tree is. In other words, a large lambda will lead to a small tree where points prefer to move to the center of the tree
MaxIter	Maximal number of optimization iterations	The maximum number of iterations is used. The larger it is, the algorithm is more guaranteed to converge. The small MaxIter means early stop and leads to less smooth tree.
Gamma	Regularization parameter for k-means	This parameter controls the fitting of points to its own centers. The larger it is, the better the clusterings appear in the tree structure.
Sigma	Bandwidth parameter	This parameter is used to model the noise of data points. A large sigma is preferred for large noise. The large noise usually causes a large eclipse. A large sigma can enforce points to move to the skeleton of the point cloud that forms the eclipse. In other words, it makes the tree smooth but do not change the main skeleton of the data. A sigma with an overestimated value will shrink the tree structure and make it smoother.
Ncenter	Number of nodes allowed in the regularization graph	The number of clusters used to represent the main structure of the learned tree. It can be same size of the data points if the data size is small. However, if data size is large, it is more efficient to set a moderate number of centers so that the algorithm can run in a reasonable time.

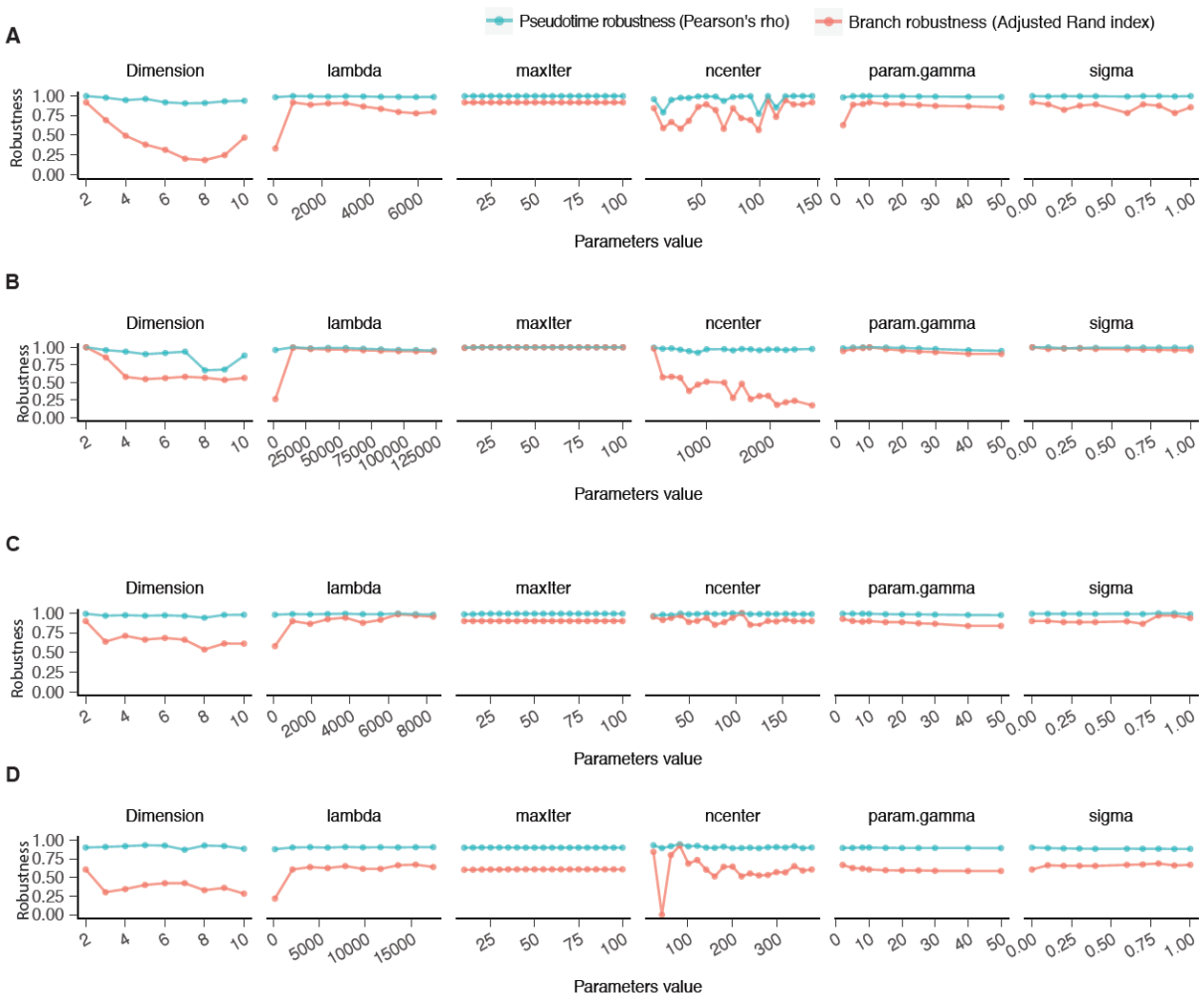
B

Algorithm	Methodology	Tuning parameters	Algorithm complexity (D: gene number; K: centroid number; N: cell number)
DDRTree / DRTree	Learn a set of low-dimensional principal points and a tree over these points	**See above**	$O(K^3+D^3+DK^2+ND^2+NDK)$
SimplePPT	Learning a set of principal points in original space and a tree over these points	**Lambda**: control the balance of the length of tree and the fitness to the input data **Sigma**: smoothness of the tree structure	$O(N^3 + DN^2)$
SGL-tree	Centroids and neighborhood graphs are incorporated into SimplePPT for large-scale problem	**Lambda**: control the balance of the length of tree and the fitness to the input data **Gamma**: fit of the centroids to data **Sigma**: smoothness of the tree structure	$O(K^3+DKN+DK^2)$
L1-graph	Learn a set of principal points in original space and a general sparse graph over these points.	**Lambda**: control the balance of the length of tree and the fitness to the input data **Gamma**: fit of the centroids to data **Sigma**: smoothness of the tree structure	$O(K^3+DKN+DK^2+L)$ L is the complexity of linear programming

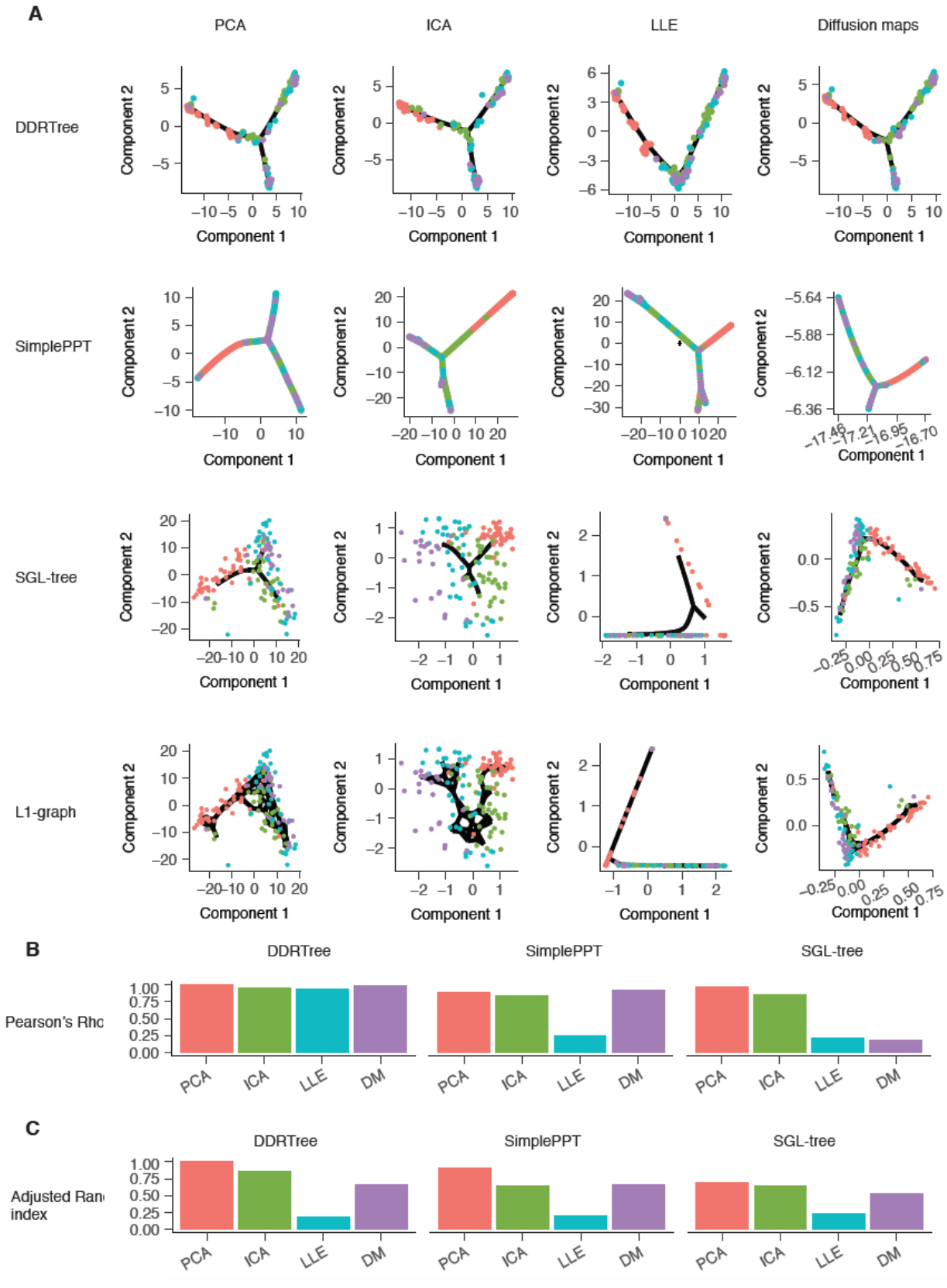
C



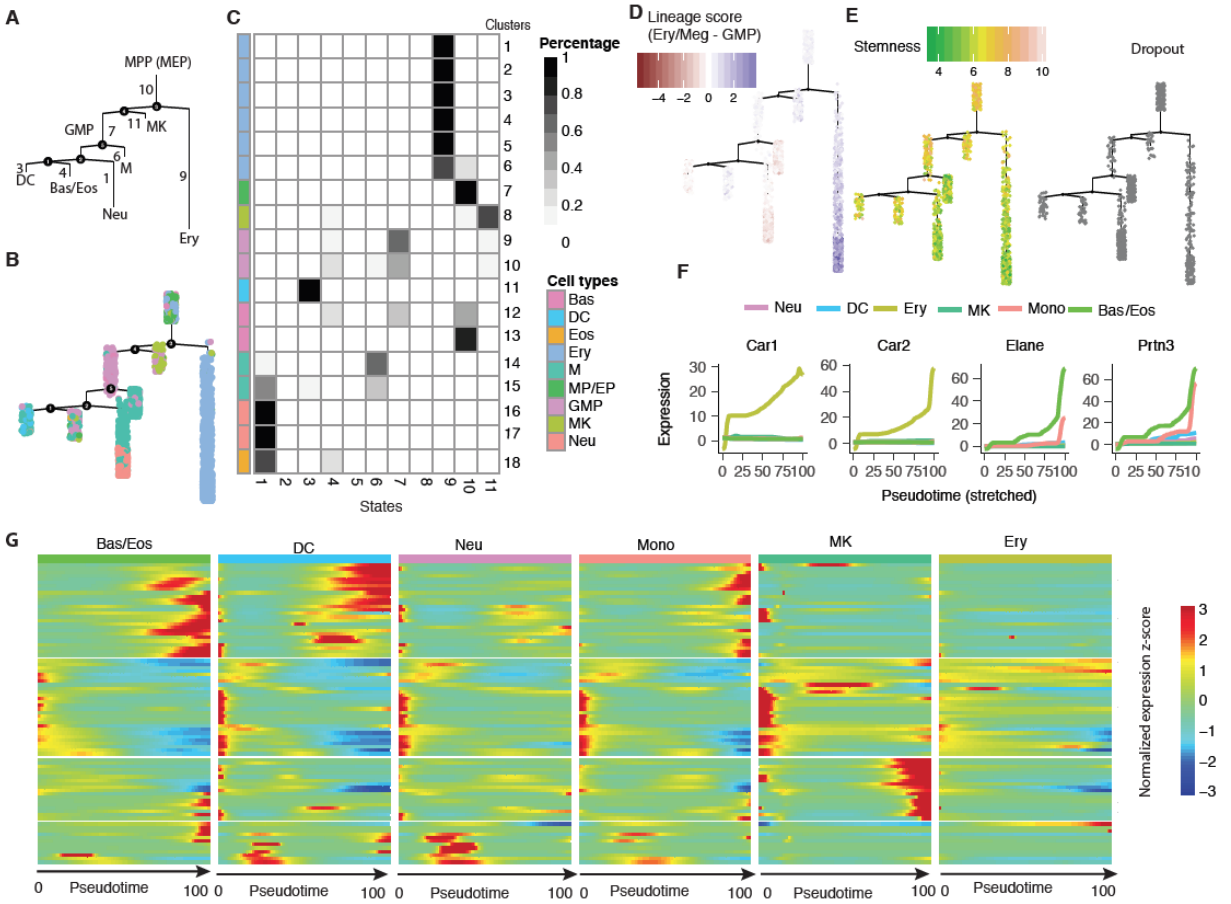
Supplementary Figure 13. Parameters of DDRTree and complexity of current available RGE implementations. (**A**) Parameters used in Monocle 2 (with DDRTree) for trajectory reconstruction. (**B**) Algorithmic complexity for DDRTree, SimplePPT, SGL-tree and L1Graph as a function of cells, ordering genes, and dimension of the embedding space. (**C**) Running time for Monocle 2, DPT and Wishbone on the full dataset (8365 cells) from Paul *et al*. over varying fractions of the cells.



Supplementary Figure 14. Robustness of Monocle 2 under a large range of parameters used in DDRTree. Each panel shows the Pearson correlation of pseudotimes and ARI of branch assignments with respect to the results obtained by Monocle 2 when run as described in the Methods section "**Details on analyzing datasets used in this study".** (**A**) HSMM dataset (**B**) Paul dataset (**C**) Lung dataset (**D**) Olsson dataset. All parameters accepted by DDRTree are included in this analysis.

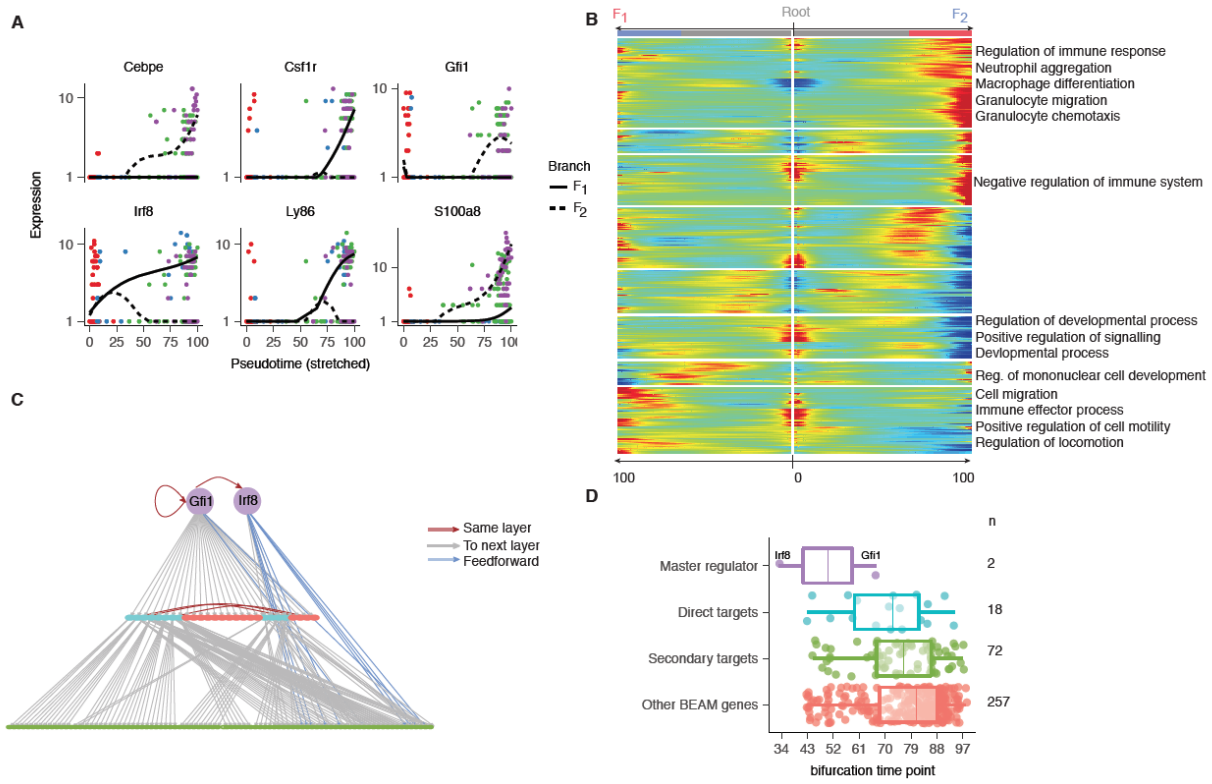

A
PCA
ICA
LLE
Diffusion maps
DDRTree
SimplePPT
SGL-tree
L1-graph
Component 1
Component 2
B
DDRTree
SimplePPT
SGL-tree
Pearson's Rho
PCA
ICA
LLE
DM
C
DDRTree
SimplePPT
SGL-tree
Adjusted Rand index
PCA
ICA
LLE
DM

Supplementary Figure 15. Different reversed graph embedding techniques reveal the same incomplete myogenesis branch. (**A**) Trajectory learned with the different RGE techniques, on the (DDRTree, SimplePPT, SGL-tree, L1-graph) or initialized with the (DDRTree) reduced two dimensional space obtained by running PCA, ICA, LLE or diffusion maps. (**B**) Pearson's *Rho* between pseudotime calculated with DDRTree initialized with PCA dimension reduction (reference) and pseudotime calculated with DDRTree, SimplePPT, L1 SGL-tree or L1-graph based on reduced dimension space obtained with different dimension reduction methods, using the dpFeature selected genes as used in **Figure 1B,C** (same as panel **C** below). (**C**) Same analysis as in panel **B** but for branch consistency by aAdjusted Rand index. Benchmark for L1-graph is not included in panel B, C because it often learns a general graph and awaits for further developments of new pseudotime and branch assignment techniques.



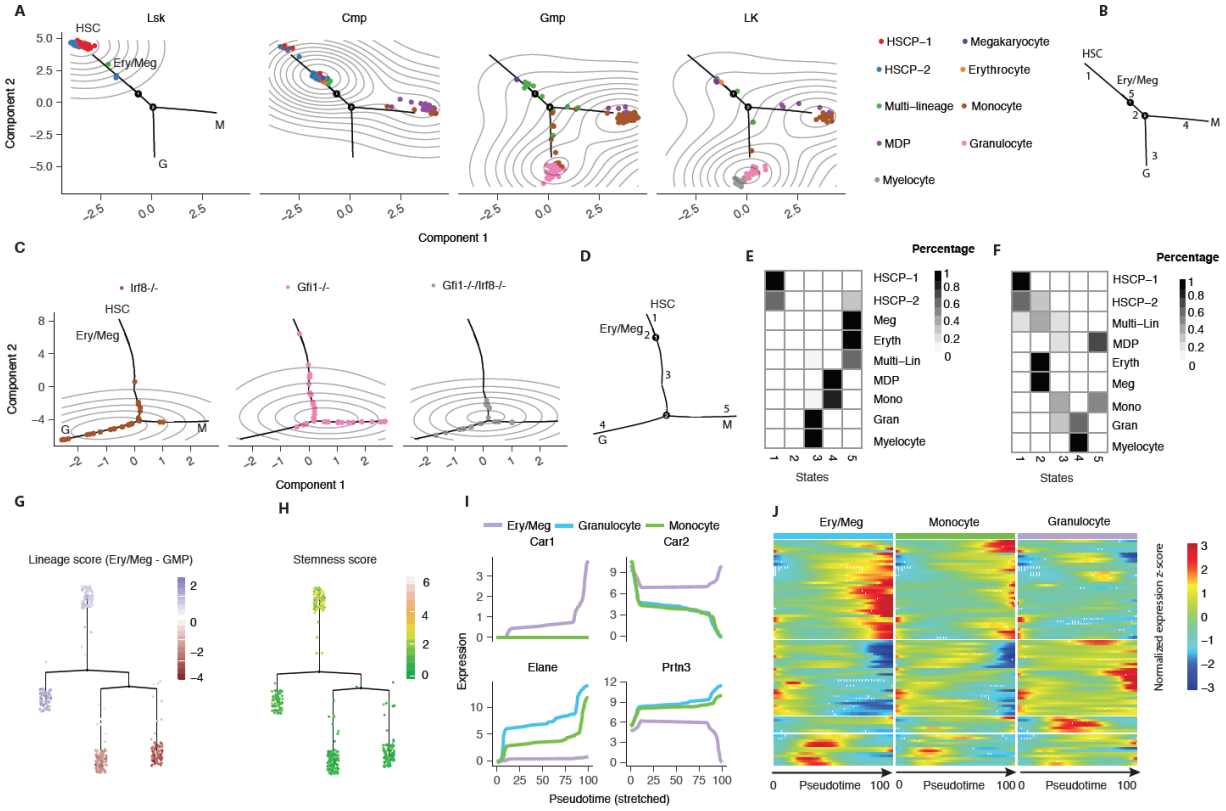
Supplementary Figure 16. Monocle 2 resolves a complex haemopoiesis hierarchy for the Paul dataset. (**A**) The skeleton of the trajectory learned by Monocle 2 describing the lineage relationships learned with Monocle 2. The numerical labels correspond to the "State" label of each segment of the tree. We have added labeled with our interpretation of corresponding cell type (inferred based on comparison with classifications made in the original study, see **B** and **C**). **MPP (MEP)**: multipotent pluripotent progenitor or myeloid and erythroid progenitors; **MK**: megakaryocyte; **GMP**: granulocyte and monocyte progenitor; **DC**: dendritic cell; **Neu/Eos**: neutrophil or eosinophil, **Bas**: basophil, **M**: monocyte; **Ery**: erythrocyte. (**B**) The trajectory learned with Monocle 2 as in panel **A**, where cells are colored by the cell types suggested by the

original study. The trajectory is reconstructed in 10 dimensions but visualized as tree layout in two dimensions using layout_as_tree() from the igraph package. (**C**) The distribution of clusters from the original study in each segment of the tree as shown in **A, B.** c.f. a similar **Figure N4B** of Haghvergdi et al (Haghverdi et al. 2016b) (**D, E**) Lineage or stemness score for cells on the tree (Same analysis as Olsson dataset, see **Supplementary Figure 18G, H**). Genes used for calculating lineage (**D**) / stemness score (**E**) are based on significant genes returned by differential expression test from the Olsson dataset. Cells show in the "Dropout" panel are those that don't express all genes used in calculating the score. (**F**) Kinetic curves for an example subset of genes used for calculating lineage score in panel **D**. Each curve corresponds to the dynamics of that gene in a particular lineage. (**G**) Multi-way heatmaps for all the genes used in panel **E**. Each sub-heatmap corresponds to a particular lineage where its pseudotime on the x-axis starts from 0 (i.e. the root cell). Similar to panel **F**, six lineages are shown.



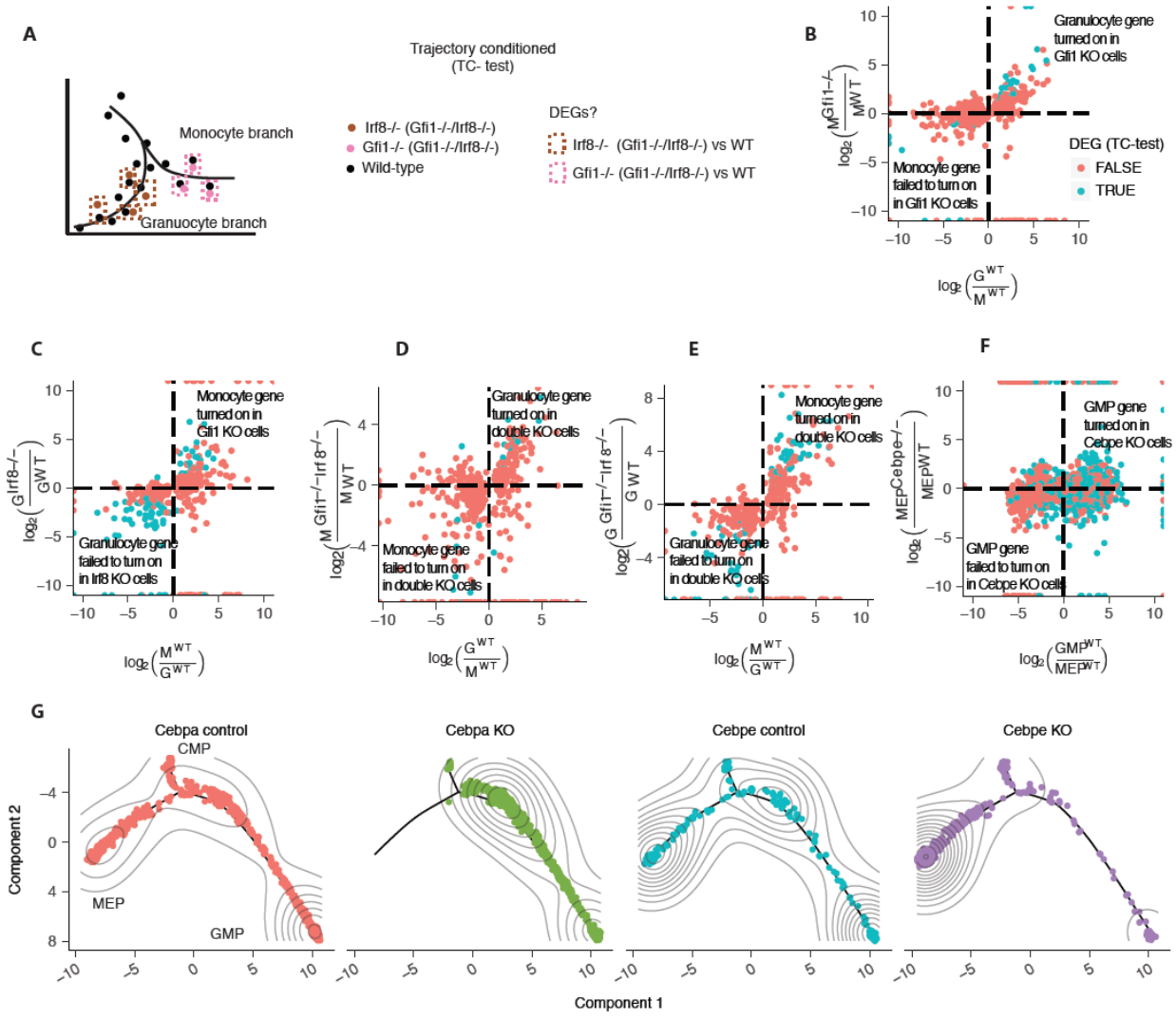
Supplementary Figure 17. Monocle 2 trajectory branches correspond to developmental fate decisions. (**A**) Monocle 2 trajectory of differentiating blood cells collected by Olsson et al (Olsson et al. 2016). Each subpanel corresponds to cells collected from a particular FACS gate in the experiment. Cells are colored according to their classification by the authors of the original study. The trajectory was reconstructed in four dimensions; only the first two are visualized here. Numbers in black squares indicate the erythroid/GMP branch (1) and the granulocyte/monocyte branch (2). This panel corresponds to panel **A** of **Figure 2**. (**B**) Branch kinetic curves of markers of the monocyte and granulocyte fates. (**C**) Branched heatmap for all the significant branch genes for the wild type data (1015 genes, BEAM test, FDR < 10%). (**D**) A network plot describing direct targets of Irf8 and Gfi1 (derived from ChIP-Seq) and secondary targets (derived

by motif analysis; see **Methods**). (**E**) Distribution of bifurcation time points for Irf8, Gfi1, and their direct and secondary targets as well as other genes in panel **C**.



Supplementary Figure 18. Monocle 2 resolves a complex haemopoiesis hierarchy for the Olsson dataset. (**A-D**) Trajectory for the Olsson WT (**A, B**) or full (**C, D**) dataset visualized with component 1 and 2. (**B, D**) are skeletons of the trajectories for the wildtype (WT) (**A**) or full (**C**) dataset and its corresponding states as well as lineages. (**E, F**) Distribution of cells from each cluster from the original study into different branches on the WT dataset (**E**) or full dataset (**F**). (**G, H**) Lineage (**G**) or stemness score (**H**) for cells on the tree for the WT dataset. Genes used for calculating lineage / stemness score are based on significant genes returned by differential

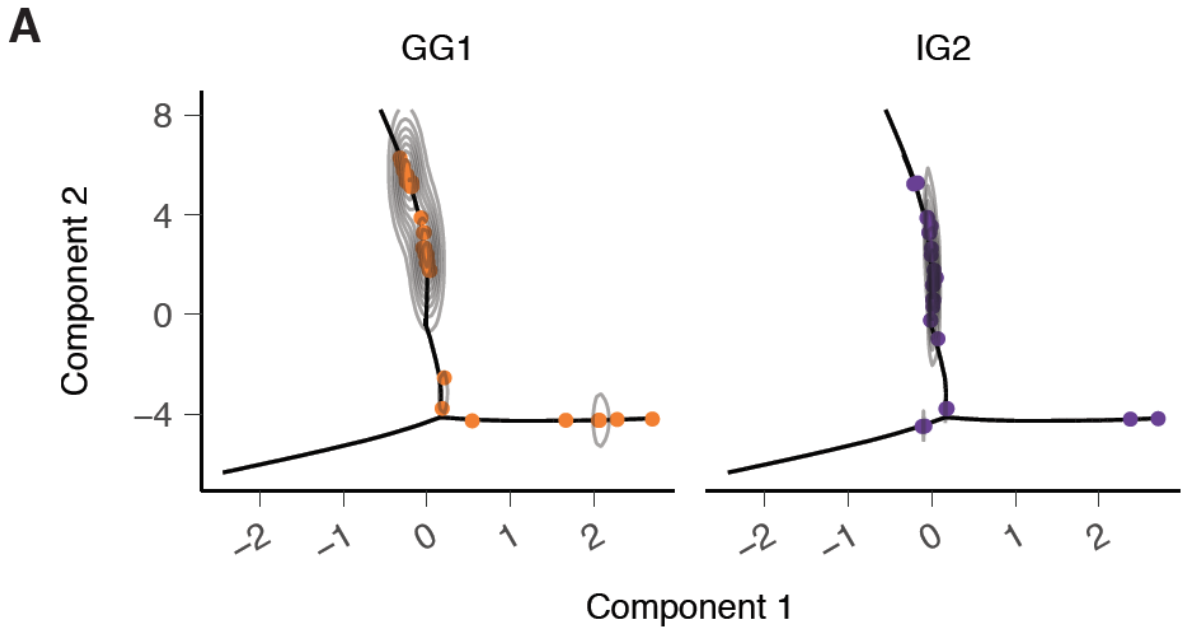
expression test from the WT dataset. **(I**) Kinetic curves for example genes used for calculating lineage score in panel **G**. Each curve corresponds to the dynamics of that gene in a particular lineage. (**J**) Multi-way heatmaps for all the genes used in panel **G**. Each sub-heatmap corresponds to a particular lineage where its pseudotime on the x-axis starts from 0 (or the root cell). Similar to panel **I**, three lineages for WT data are shown.



Supplementary Figure 19. Trajectories reveal how genetic perturbations divert cells to alternative fates. Panels (**A-F**) are based on the data from Olsson et al while panels (**G, H**) are based on the data from Paul et al. (**A**) Cells with a single knockout of Irf8 or Gfi1 are diverted into the alternative granulocyte or monocyte branch, respectively. Double knockout cells are localized to both granulocyte and monocyte branches but concentrated near the branch point. This panel corresponds to panel **B** of **Figure 2**. (**B**) A "trajectory-conditioned" test for

identifying genes that are differentially expressed between genotypes that controls for the cells' positions on the trajectory. (**C-D**) Expression changes for branch-dependent genes (shown in

Supplementary Figure 17C) between wild-type granulocytes and monocytes (horizontal axis) plotted against the effects of Gfi1 or Irf8 knockout. The vertical axis in panel **C** shows expression changes in Gfi1-/- monocytes relative to wild-type monocytes, while panel **D** shows changes in Irf8-/- granulocytes compared to wild-type. Genes with significant trajectory-conditioned differences in expression between knockout and wild-type are highlighted (FDR < 10%). (**E-F**) Expression changes in BEAM genes between double KO "monocytes" (**E**) or "granulocytes" (**F**) and wild-type. (**G**) Monocle 2 accurately positions cells from Cebpa-/- or Cebpe-/- collected by Paul *et al*. Loss of Cebpa fully blocked the MEP branch while Cebpe KO partially blocked the GMP branch. (**H**) The trajectory-condition test between Cebpe-/- cells and nearby wild-type cells on the MEP branch reveals aberrant expression of GMP-branch specific genes in MEP branch.



Supplementary Figure 20. Monocle 2 correctly positions transient wild-type cells. (**A**) Gated rare transient cell from Olsson et al are enriched upstream of the branch point separating monocytes and granulocytes.

Supplementary Methods

Assessing accuracy or robustness of pseudotime and branch assignments

We assessed the accuracy and robustness of each algorithm's pseudotime assignment against the reference ordering by two measures of correlation (Pearson's Rho (default) and Kendall's Tau) between their pseudotime values.

We used adjusted Rand index (ARI) (Rand 1971), a common metric used for measuring clustering accuracy, to measure the accuracy of tree segment assignment. Given the number of common cells (300 in this case), denoted as S, between the reference ordering and the ordering

based on an algorithm (Monocle 2, Monocle 1, DPT, Wishbone or SLICER), and corresponding trajectory segment assignments for reference ordering and ordering based on a different algorithm, $\mathcal{X}$ and $\mathcal{Y}$, namely, $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_r\}$ and $\mathcal{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_s\}$. The overlap between cells from segment i, $\mathcal{X}_i$ and cells from segment j, $\mathcal{Y}_j$ in each of the two orderings is represented by the number $n_{i,j}$ of cells in common, i.e., $n_{i,j} = |\mathcal{X}_i \cap \mathcal{Y}_j|$. Define the number of cells with segment i from reference ordering is $a_i = \sum_{j=1}^{s} n_{i,j}$, and the number of cells with segment j from ordering based on an algorithm is $b_j = \sum_{i=1}^{r} n_{i,j}$. The Adjusted Rand Index is then formulated as

$$ARI(\mathcal{X}, \mathcal{Y}) = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

which is a measure of the similarity between two data clusterings (or segment assignment in this case). When ARI is closer to 1, the segment assignment is more consistent between the two orderings.

For calculating the accuracy of pseudotime and branch assignment of the simulation (neurogenesis) and Paul dataset, the reference ordering corresponds to the real simulation and branches assignment based on manual assessment (see **Supplementary methods**) or the pseudotime and branch (or cell type suggested from the original study (Paul et al. 2015)) from the marker-based ordering (see next section).

For calculating the robustness, the reference ordering is defined in the context of downsampling. We apply two different downsampling strategies. First, we downsample the full dataset, including the simulation data for neuronastro-genesis, Paul dataset and the lung dataset, selecting of the cells from the full dataset 25 times without replacement. Then we run Monocle 2, Monocle 1, DPT, and Wishbone to construct branched trajectory. SLICER was excluded from the downsampling analysis on account of its long running times and instability on occasional down-sample runs. Then we compare all pairs of downsamples by the metrics discussed above. We also progressively downsample all the full dataset over a range of increasing fractions of cells from the full dataset. Sampling is performed without replacement and three different subsets are generated for each proportion to serve as replicates. Then we run each software, including Monocle 1, Monocle 2, DPT, Wishbone, to construct branched trajectories for each fraction, which are compared to the corresponding trajectory built from the full dataset. ARI, Pearson's Rho, Kendal's Tau for all cases are then calculated as above.

In order to assess the robustness of Monocle 2 over different parameters choices, we run Monocle 2 and sample a large range for each parameter used in DDRTree, including, Dimension, lambda, maxIter, ncenter, param:gamm, sigma while keeping other parameters as default and compare the result to the ordering obtained by running Monocle 2 with all default parameters. Pearson's Rho and ARI are used to calculate the robustness.

Comparing different algorithms to a marker-based ordering

In order to test the accuracy of each trajectory reconstruction algorithm, we compared their trajec-tories to an empirical ordering based on marker genes. Relying on results from Paul et al

(Paul et al. 2015), we first select *Pf4, Apoe, Flt3, Cd74* as CMP specific genes, *Hba-a2, Car2, Cited4, Klf1* as MEP specific genes and *Mpo, Prg2, Prtn3, Ctsg* as GMP specific genes. Following the approach of Tirosh et al (Tirosh et al. 2016a), we then select 100 other genes with expression correlated to these marker genes to calculate a stemness score and GMP or MEP lineage score. We define cells with stemness score larger than 0 as CMP cell and any cells with positive lineage score as MEP cells and negative score as GMP cells. This grouping of cells is used for branch assignment accuracy evaluation in **Supplementary Figure 9**. We then define the reference pseudotime for each cell as:

$$\varphi_(b_x, s_i) = \begin{cases} d(s_i, \mathbf{0}), & \text{if } i \in \{\text{CMP}\} \\ \max\limits_{j \in \{\text{CMP}\}} \|s_j, \mathbf{0})\|_2 + \|l_i, \mathbf{0}\|_2, & \text{otherwise} \end{cases}$$

where $\mathbf{0}$ corresponds to the origin $(0, 0)$, s_i corresponds to the stemness score and l_i the lineage score for the lineage to which each cell is assigned, $d(\cdot, \cdot)$ represents the Euclidean distance between two points, and $\{\text{CMP}\}$ indicates the set of CMP cells.

Pseudotime correlations were computed on the paths from the root to each fate based on the reference ordering separately and then averaged. Since the empirical ordering based on marker genes is not perfect, we also investigate the accuracy of the ordering in terms of the absolute lag-1 autocorrelation of fitted spline curve for the selected marker genes. We first select the trajectory segments corresponding to the transition from the CMP cells to either MEP or GMP cells and then fit a kinetic curve for each marker gene for each transition with a spline

curve with three degree of freedom. We then calculate the the absolute lag-1 autocorrelation r, which is defined as following:

$$r_i = \frac{\sum_{i=1}^{N-1} |(Y_i - \mu)(Y_{i+1} - \mu)|}{\sum_{i=1}^{N} (Y_i - \mu)^2}$$

where Y_i represents the gene expression at time stamp i, and μ is the mean expression across the pseudotime series for that gene. Higher autocorrelation value implies smoother gene expression dynamics based on the ordering. Those 300 cells are also used to calculate the accuracy of branch assignment with the branch assignment from the marker-based ordering.

Although a reference ordering based on markers from literature can serve as a reasonable gold-standard, it also introduces bias in a benchmarking analysis. Algorithms that order cells based on a small set of informative genes (which include or correlate the marker genes) will likely match it better than algorithms that order cells based on all genes. We therefore explored orthogonal means of measuring accuracy of each programs ordering based on the neuron simulation data (see **Supplementary Method**).

Reconstruct complex haemopoiesis hierarchy

We check the scree plot to choose ten dimension as the intrinsic dimensions to reconstruct the developmental trajectory for the Paul dataset (cells used in **Figure 1** of the original study (Paul et al. 2015)). Five branch points and six terminal lineages (monocytes, neutrophils or eosinophil, basophils, dendritic cells, megakaryocytes, and erythrocytes) are revealed. We ordered the cells using genes Paul et al. used to cluster their data rather than the genes from dpFeature, for the

sake of consistency with their clusetering analysis. Similarly, we reconstruct Olsson datasets in four dimensions. The major bifurcation between the granulocyte and monocyte branch (GMP) as well as the intricate branch between GMP and megakaryocyte erythrocyte (EryMeg) are revealed. Top 1, 000 genes from dpFeature based on WT cells are used in both of the WT and full datasets. The distribution (related to confusion matrix) of percentages of cells in each cluster from the original papers over each segment (state in Monocle 2) of the principal graph are calculated and visualized in the heatmap.

We applied BEAM analysis to identify genes significantly bifurcating between EryMeg and GMP branch on the Olsson wildtype dataset. We then calculate the instant log ratios (ILRs) of gene expression between EryMeg and GMP branch and find genes have mean ILR larger than 0.5. The ILRs are defined as:

$$ILR_t = log(\frac{y_1^t}{y_2^t})$$

So ILR_t is calculated as the log ratio of fitted value at interpolated pseudotime point for the EryMeg lineage and that for the GMP lineage. Those genes are used to calculate the lineage score (simply calculated as average expression of those genes in each cell, same as stemness score below) for both of the Olsson and the Paul dataset which is used to color the cells in a tree plot transformed from the high dimensional principal graph (see **Supplementary Notes**). The same genes are used to create the multi-way heatmap for both of the Paul and Olsson dataset (see *plot multiple_branches_heatmap* function). Critical functional genes from this procedure are identified. *Car1, Car2* (important erythroid functional genes for reversible hydration of carbon

dioxide) as well as *Elane, Prtn3* (important proteases hydrolyze proteins within specialized neutrophil lysosomes as well as proteins of the extracellular matrix) are randomly chosen as example for creating multi-lineage kinetic curves in both of the Olsson and Paul dataset (see *plot_multiple_branches_pseudotime* function).

In addition, pseudotime dependent genes for the EryMeg and GMP branch are identified in the Olsson wildtype dataset. All genes that always have lower expression from both lineages than the average in the progenitor cells are selected. Those genes are used to calculate the stemness score for both of the Olsson and the Paul dataset which is used to color the cells in the tree plot.

Chapter 3. BEAM (Branch Expression Analysis Models) detect significant branching genes during lineage bifurcation

A version of this chapter has been previously published as part of the following paper:

Single-cell mRNA quantification and differential analysis with Census. **X Qiu**, A Hill, J Packer, D Lin, YA Ma, C Trapnell ***Nature methods*** 14 (3), 309

3.1 Introduction

Differential gene expression analysis, typically powered by statistical regression, is central to nearly all single-cell transcriptomic studies. As experiments now capture tens of thousands of cells[1,2], such regressions could in principle be used to detect gene regulatory changes across individual cells as a function of developmental progression, position in an embryo, or genetic sequence. However, they report measurements with high variability, frustrating efforts to build regression models that can detect such changes[3,4]. For example, numerous studies have reported high rates of "drop-out", wherein some cells of a nominally homogeneous population express high levels of a gene and others none at all. Drop-outs have spurred the deployment of hurdle models[5] that overcome limitations over simpler regression approaches, typically at a cost in speed, numerical stability, or design flexibility for the user.

3.2 Differential analysis of branch points in developmental trajectories reveals regulators of cell fate

Many single-cell gene expression studies aim to identify gene regulatory circuits that control cell-fate decisions made during development[17,18]. We recently developed Monocle 2, an algorithm that organizes single cells along trajectories and can describe the gene expression changes executed during cell differentiation. Monocle introduced the concept of "pseudotime", which quantifies each cell's progress through development. Pseudotime resolves cascades of gene regulatory changes that accompany differentiation and other dynamic cellular programs[15]. Monocle produces more reliable tests for differential expression along a trajectory when provided with Census transcript counts than with relative expression values (**Supplementary Figure 7**).

Single-cell trajectories can have multiple outcomes, such as during the generation of alternative developmental lineages[19]. Analyzing cells at branch points where cells are diverted along two or more mutually exclusive paths could identify genes differentially regulated in response to a cell fate decision and reveal the mechanisms by which such decisions are made. For example, scrutinizing genes upregulated in common myeloid progenitors but downregulated in common lymphoid progenitors has shed light on the molecular regulation of cell fate in hematopoiesis[20,21]. Therefore, accurately identifying genes differentially regulated following a fate decision requires a statistical analysis that detects trajectory-dependent changes in gene expression while controlling for the noise inherent to single-cell RNA-Seq experiments.

To explore a cell fate decision made during development at single-cell resolution, we reanalyzed single-cell RNA-seq data from a recent study investigating the specification of the distal lung epithelium[10]. Treutlein *et al.* sequenced developing epithelial cells to define the cellular intermediates giving rise to type I (AT1) and type II (AT2) pneumocytes. Development of these cell types, which mediate gas exchange and surfactant production, respectively, remains the subject of intense study as they are at the center of serious lung diseases in infants (see Whitsett *et al.* for a recent review[22]). We sought to expand on prior analysis by examining the gene expression kinetics that characterize developmental transitions between each stage of differentiation. Monocle reconstructed a trajectory with a single branch point leading from progenitors to two outcomes corresponding to the AT1 and AT2 fates. The beginning of the trajectory contained cells with high levels of markers of active proliferation[23] (*Ccnb2*, *Cdk1*), whereas these genes were expressed at much lower levels after the branch point (**Figure 3a**). High expression of a known marker of AT1 cells[24] (*Pdpn*) was restricted to cells on one branch of the tree, whereas cells expressing an AT2 marker[25] (*Sftpb*) at high levels were located on the other branch. Cells classified as AT1 and AT2 according to known markers by Treutlein *et al.* fell exclusively along the branches, with what the authors termed "bipotent progenitors (BP)" at or near the branch point. (**Supplementary Figure 8**) Monocle thus accurately reconstructed the pneumocyte fate specification trajectory, enabling analysis of its developmental kinetics at pseudotemporal resolution.

We sought to identify all genes dynamically regulated during pneumocyte specification. To detect branch-dependent genes, we developed BEAM, a generalized linear modeling (GLM) [26] strategy for analyzing branched single-cell trajectories (**Figure 3b; Supplementary Figure 9,**

10, see Methods). BEAM identified 1,219 genes (FDR < 5%) as either AT1- or AT2- lineage dependent, including canonical markers[24] such as as *Pdpn* and *Sftpb* (**Figure 3c, Supplemental Figure 11**). AT1-restricted genes were strongly enriched for ontological terms related to tube development, cytoskeletal remodeling, and cell morphogenesis (**Figure 3d, Supplementary Table 1**), while AT2-restricted genes were enriched for terms related to lipid processing, consistent with the production of lipid-rich surfactant by AT2 cells in the mature lung. Annotating these genes with potential upstream regulators based on sequence motifs in proximal DNA regulatory elements identified 74 transcription factors with significant binding motif enrichment. Seventeen of these factors, several of which are well known regulators of lung epithelial differentiation[27-32], themselves exhibited significant branch-dependent expression (**Figure 4e, Supplementary Figure 11)**. For example, the motif for *Tcf7l2Tcf4* is highly enriched at promoters of AT1-specific genes, and *Tcf7l2* mRNA is rapidly restricted to AT1 cells after the branch point in the trajectory. *Tcf7l2* is a major mediator of Wnt signaling, which drives the development of the distal lung eptithelium[28,32]. These analyses demonstrate that branches in single-cell trajectories can point to potentially important developmental regulators governing cell fate specification**.**

3.3 Disruption of interferon signaling induces a branch in the dendritic cell LPS stimulation trajectory

Branch points in single-cell trajectories represent steps in a program of transcriptional change in which cells must choose between one of several mutually exclusive gene expression programs. In theory, such alternative programs could arise not only during development, but also in

response to loss- or gain-of-function mutations, treatment with drugs or small molecules, or other cellular perturbations. We re-analyzed a recent study[33] from Shalek and colleagues, which dissected the transcriptional response of murine bone marrow-derived dendritic cells (BMDCs) to lipopolysaccharide (LPS) (**Figure 4a**). In BMDCs, LPS triggers a paracrine feedback loop of type I interferon signaling mediated in part by *Stat1*[34-36]. The authors compared BMDCs from wild-type (WT) mice to those from mice that lack the receptor for *Interferon alpha* (*Ifnar1*--) or *Stat1* (*Stat1*--). Monocle recovered a trajectory with a single branch point, with cells from *Infar*--Stat1-- mice distributed on an alternative trajectory in response to LPS stimulation compared with those from WT mice (**Figure 4b**). BEAM identified 870 genes (FDR < 5%) dependent on this branch, including a core of 226 genes enriched for functions related to interferon signaling (**Figure 4c**). Down-sampling the number of cells in the dataset and re-running Monocle 2's trajectory analysis showed that precision remained high even with few cells (**Supplementary Figure 12**).

We investigated whether genes reported by BEAM were directly downstream of interferon signaling by scrutinizing their regulatory DNA elements for common transcription factor binding sites. Recently, Lavin *et al.* cataloged regulatory elements in tissue-resident macrophages, which have many functions in common with BDMCs, using a battery of genome-wide epigenetic assays[37]. Peaks corresponding to open chromatin from this catalog proximal to branch-dependent genes up-regulated in the WT BMDCs are enriched for *Stat12* and *Irf12* binding motifs (**Figure 4d**). These factors were themselves significantly branch-dependent, with branching pseudotimes substantially earlier than their putative targets, confirming that BEAM can distinguish the regulatory factors that drive branching in single-cell trajectories from

genes downstream (**Figure 4e, f**). Monocle 2 and BEAM demonstrated that loss of a key paracrine loop generates an “alternative trajectory”, suggesting that single-cell trajectory analysis can be useful for defining how a signaling pathway regulates a larger process.

3.4 Discussion

Efforts to detect changes in gene regulation in development have grappled with high technical and biological variability, demanding specialized statistical methods that explicitly model drop-outs and other nuisance variation. Here, we show that analyzing changes in normalized transcript counts leads to dramatic reductions in apparent technical variability compared to normalized read counts, making single-cell RNA-Seq compatible with widely used regression techniques. We have developed Census, a normalization algorithm that can convert relative expression levels from read counts into per-cell transcript counts without the need for spike-in standards or UMIs. The algorithm requires only that genes are most frequently present at 1 cDNA molecule in each cell’s library. We show through reanalysis of several datasets that this is the case with most current protocols, owing to mRNA capture rates lower than 50% and their generation of full-length cDNAs during reverse transcription. Census cannot control for amplification biases, and thus does not produce estimates of lysate mRNA abundances that perfectly match those derived with spike-ins or UMIs. When spike-ins or UMIs are available, transcript counts should be recovered using them rather than Census. However, we show through extensive benchmarking that differential analysis results with Census counts are highly concordant with those from spike-ins. Importantly, tools widely used for bulk RNA-Seq analysis

that perform poorly when provided with read counts work vastly better with Census counts, alleviating the need for software tailored for single-cell RNA-Seq.

Census makes transcript count analysis available in a wide range of single-cell RNA-Seq experimental designs. To illustrate its power, we have developed three regression-based methods for detecting changes in transcript counts from Census associated with lineage commitment, alternative splicing, and allele-specific expression, respectively.

The first, BEAM, builds on our previous work tracking gene expression changes in single-cell trajectories, helping pinpoint the moment at which cell-fate decisions occur in a complex biological process. BEAM detects genes that become expressed in a lineage-dependent manner following such decisions by encoding branch assignments as regression variables. BEAM identified thousands of genes differentially regulated during specification of the type I and type II pneumocytes in the alveolar epithelium. The branched single-cell trajectory is driven by genes that cluster into groups whose regulatory elements are highly enriched for distinct transcription factor binding sites. Some of these factors, such as *Foxp2*, are themselves branch-dependent and known to be involved in mediating the pneumocyte specification. Surprisingly, branched cell trajectories arise not only in development, but also in response to genetic perturbations, suggesting that branch analysis may be useful in many biological contexts. Our unsupervised reanalysis of dendritic cells undergoing immune activation reconstructed a linear trajectory that strongly agrees with the current understanding of this process. Dendritic cells lacking receptors or signal transducers central in mediating immune activation follow an alternative trajectory. Understanding the genes that control passage beyond this point may yield

crucial insights regarding the molecular regulation of the innate immune response and cellular decision-making.

The second method uses Census counts to find genes undergoing pseudotime-dependent changes in splicing. Reanalysis of differentiating myoblasts showed widespread alteration in isoform ratios in genes involved in muscle contraction and cytoskeletal structure. The genes included the tropomyosins, which are well-characterized models of alternative splicing in development. Census revealed a sequence of pseudotime-dependent changes in TPM1, with exon 6b inclusion preceding that of exon 9b, rather than a simultaneous shift at these splice sites.

The third method captures changes in allelic transcript counts derived with Census. By reanalyzing data from pre-implantation embryos, we detected transcription from the genome in 2-cell embryos, with allelic balance equilibrating to 50% quickly thereafter. We subsequently used this method to survey gene silencing on the X chromosome and confirmed the escape of several well characterized genes such as *Xist*. In contrast to the original study, we do not see substantial evidence of random, monoallelic expression on the autosomes, and attribute this observation to inadequate modeling of dropouts in normalized read counts. Monoallelic expression at the transcript count level was in line with expectations under a simple overdispersed binomial regression model.

Together, our analyses show that single-cell differential expression analyses conducted at the level of normalized transcript counts are more robust and accurate than analyses of normalized read counts. We provide a new algorithm, Census, that makes transcript count analysis widely accessible, as well as examples of regression models that leverage them for

high-resolution dissection of gene regulation. We expect that such techniques will continue to unveil new mechanisms of gene regulation, including at the allele and isoform level, in development and disease.

Figures

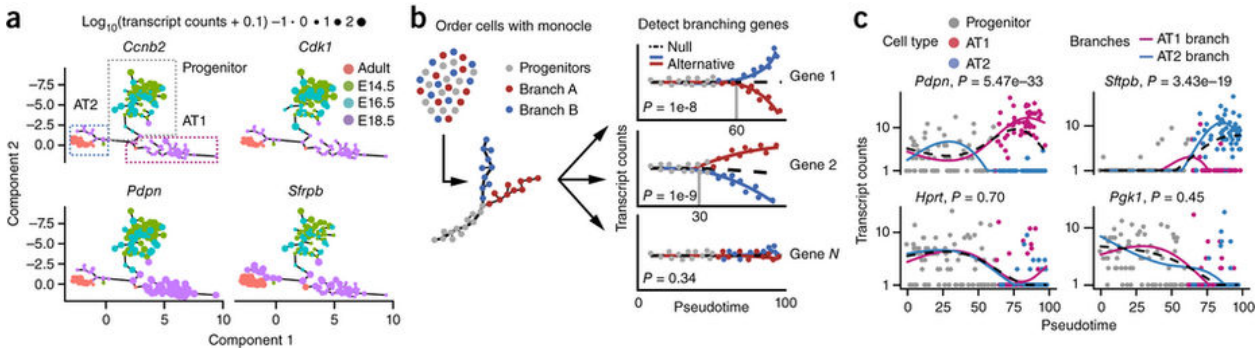


Figure 3.1. BEAM identifies genes with branch-dependent expression and potential drivers during lung epithelial fate specification. (a) Monocle 2 recovers a branched single-cell trajectory beginning with bronchoalveolar progenitors (BP) and terminating at type I (AT1) and type II (AT2) pneumocytes. High expression of known markers of proliferation (*Ccnb2*, *Cdk2*) is restricted to progenitor cells, whereas high expression of known AT1 (Pdpn) and AT2 (Sftpb) markers is restricted to their corresponding lineages. Size of circles denotes level of expression. (b) Branching Expression Analysis Modeling (BEAM) is a statistical framework for identifying genes with expression that changes over a single-cell trajectory in a branch-dependent manner. BEAM first uses generalized linear models with natural splines to perform a regression on the data in which the branch assignments of the cells are known (alternative model), fitting a separate curve for each branch. It also performs another regression in which the branch

assignments are not known (null model), fitting a single curve for all the data, and then compares these models via a likelihood ratio test. (c) Null and alternative model fits for the AT12 markers (*Ager* *Sftpb*) and housekeeping genes (*Hprt* and *Pgk1*). Solid lines indicate the smoothed expression curves for each branch in the alternative model while dashed line corresponding to the fitted curve in the null model used in the BEAM test. (d) Hierarchical clustering of normalized expression in terms of transcript counts for markers of pneumocyte specification as defined by Treutlein *et al.* and cell cycle genes. Each column represents a cell ordered along the trajectory. The center of the heatmap corresponds to the beginning of the trajectory. Moving left proceeds down the AT1 branch, whereas moving right proceeds down the AT2 branch. Each row represents the smoothed BEAM expression curve for a gene on each branch. Rows are transformed to Z-scores prior to hierarchically clustering using Pearson's correlation with Ward's method. (e) Branch kinetics for selected transcription factors with binding motifs enriched at regulatory elements for genes in panel **c**, shown with *p* values from BEAM. See **Supplementary Figure 11a** for the full list of transcription factors.

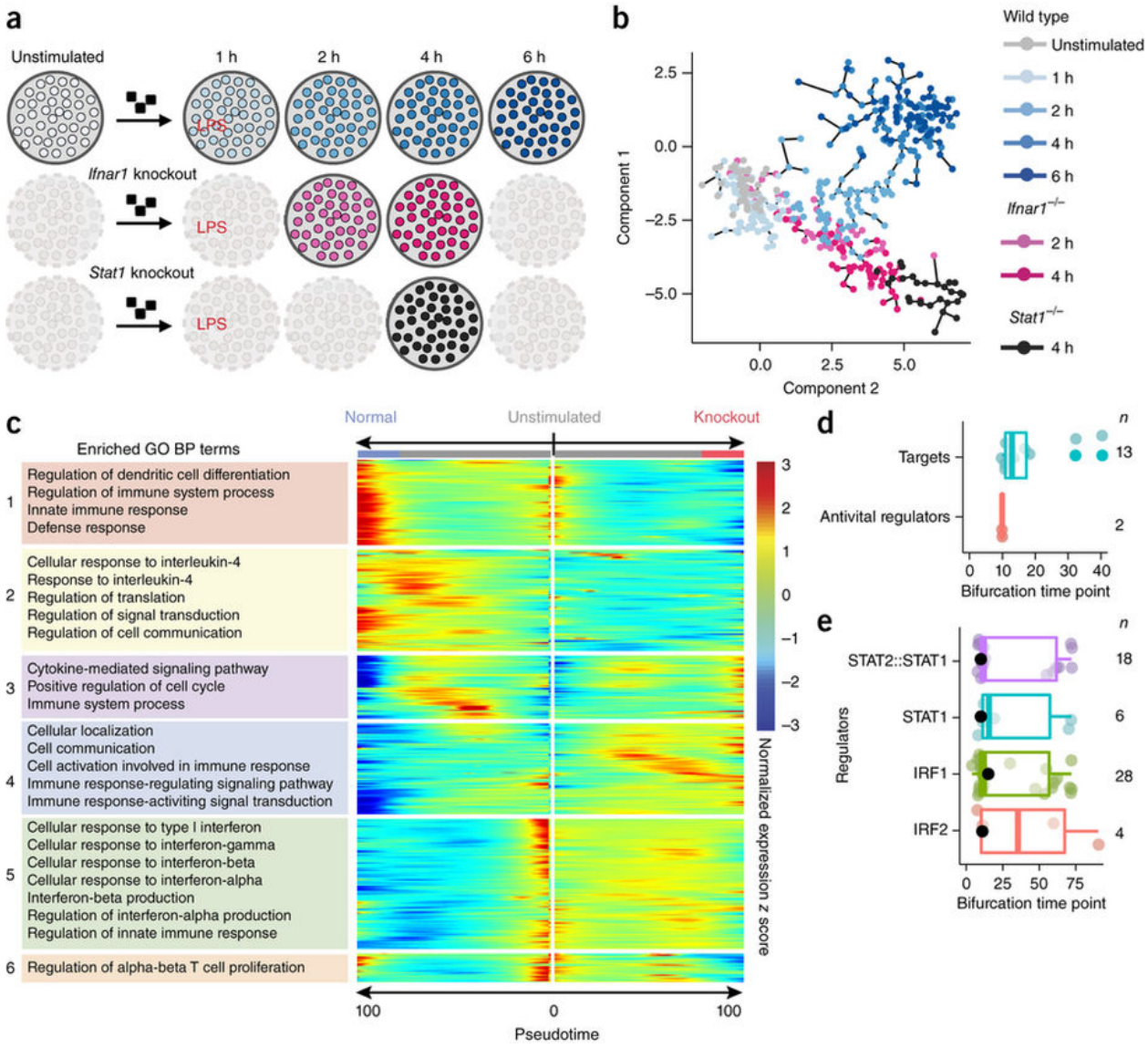
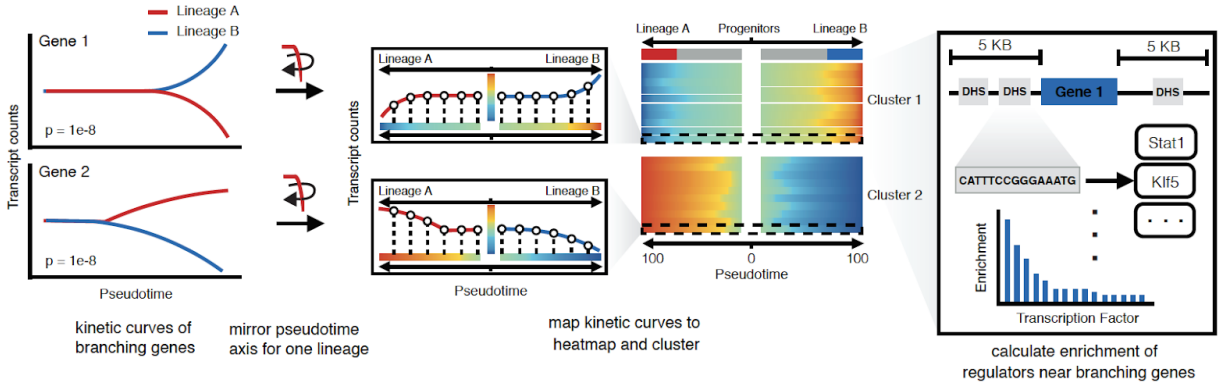


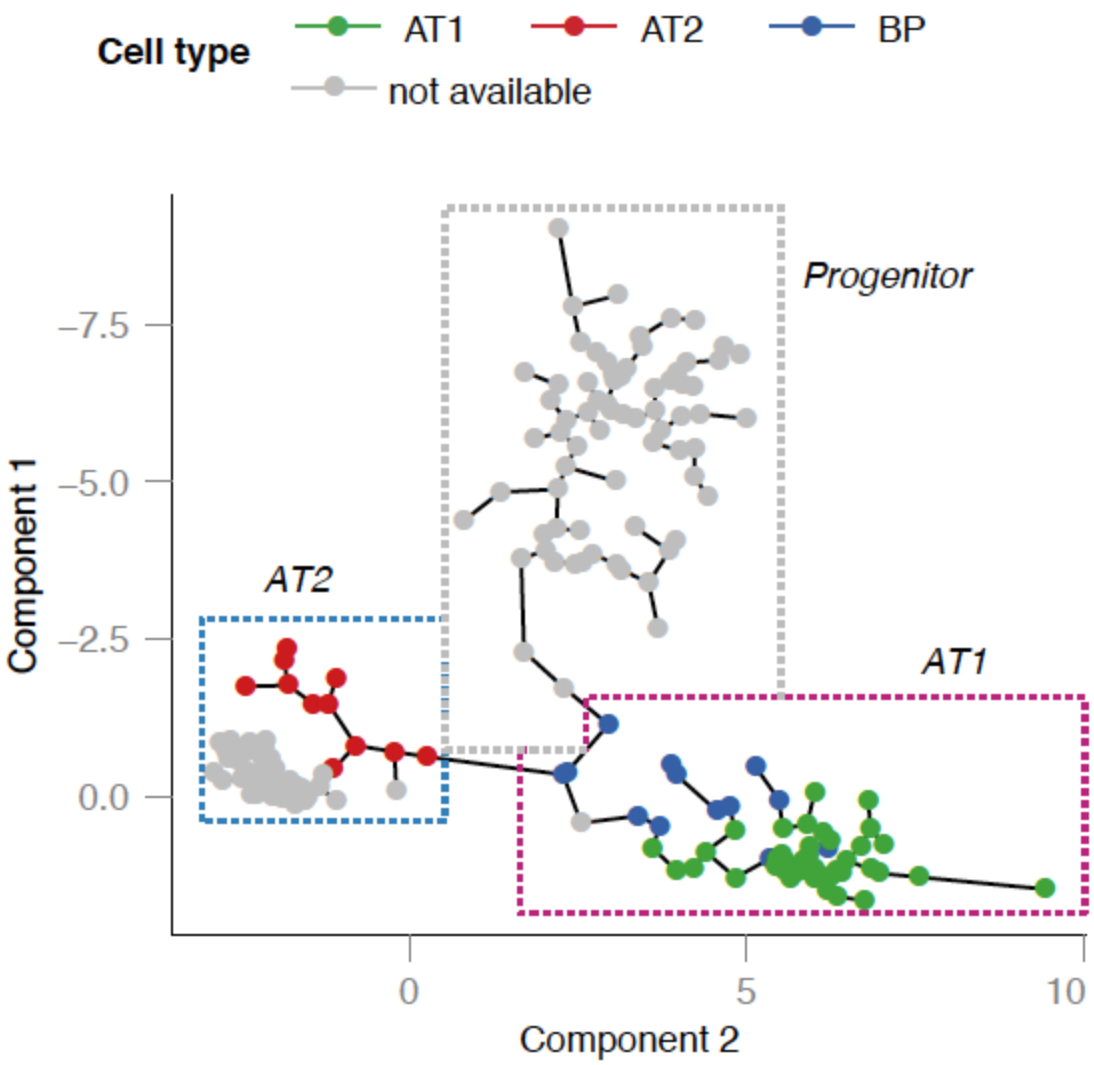
Figure 3.2. Loss of interferon signaling generates a branch in the trajectory followed by immune-stimulated dendritic cells. (a) Experimental design used by Shalek *et al.* to compare BMDCs from *Ifnar1*-- and *Stat1*-- knockout mice against the wild type as they respond to LPS. (b) Single-cell trajectory recovered by Monocle 2. (c) Kinetic clusters of branch-dependent genes identified by BEAM are functionally enriched for interferon signaling and other immune-related processes. (d) Transcription factor binding motifs significantly enriched in regulatory elements upstream of genes in cluster 3 (hypergeometric test, FDR < 10%. See methods). (e) Branch time

point for the antiviral regulators and their targets collected from Fig. 4 of ref. [46]) (f) Branch time points for the annotated TFs highlighted in panel **d** and their targets in genes from cluster 3 (panel **c**). Black dots represent the branch time point of each factor itself.

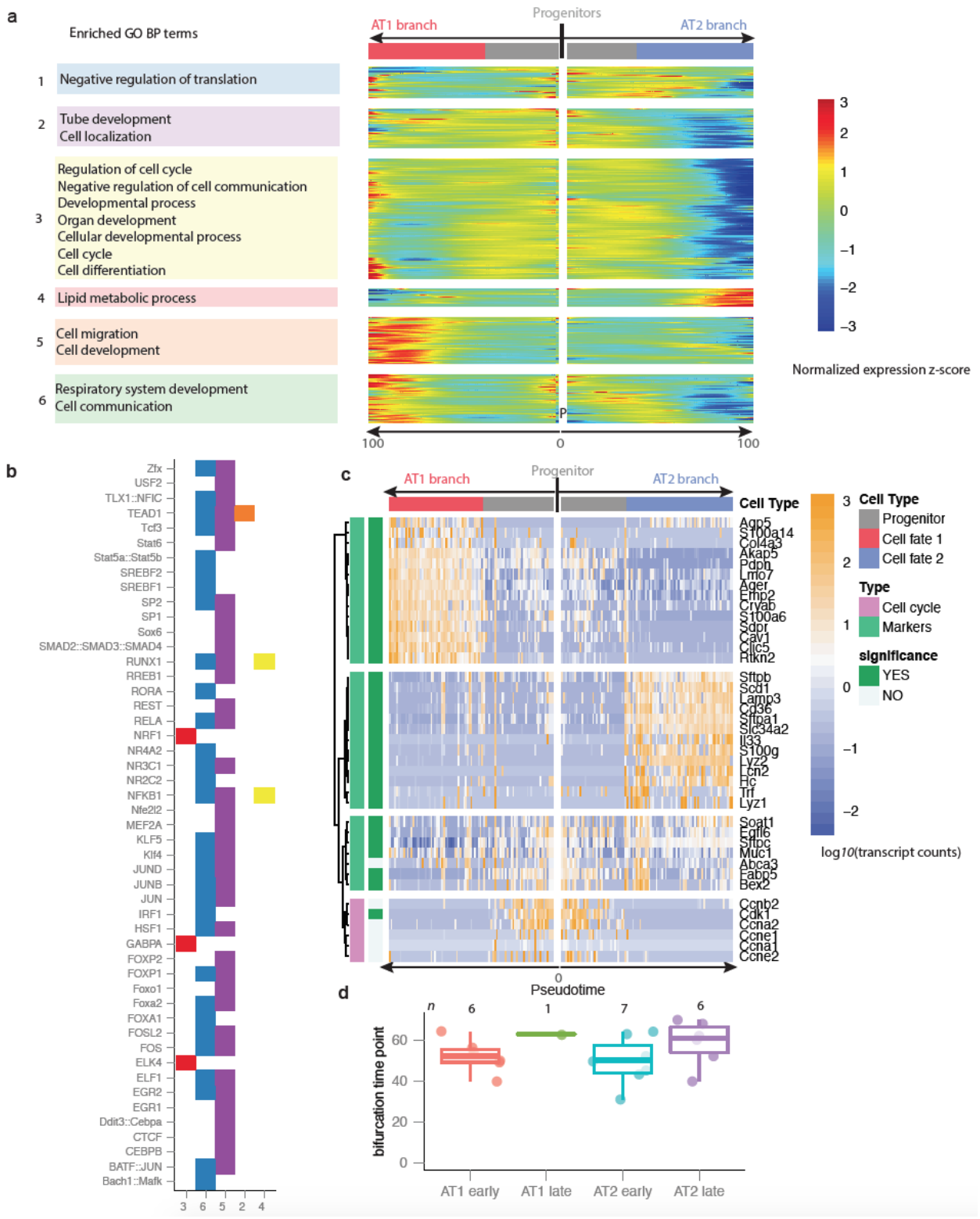
Supplemental Figure



Supplementary Figure 1. Clustering and motif enrichment analysis of branch-dependent genes according to bilineage expression kinetics. Branch-dependent genes identified by BEAM can be further assessed for common functions and potential upstream regulators. The expression for each gene along the two trajectories is used to cluster genes into groups that share branch-dependent expression kinetics. Regulatory elements (e.g. defined by DNaseI-Seq) can be collected for each group and tested for enrichment with specific transcription factor binding site motifs.

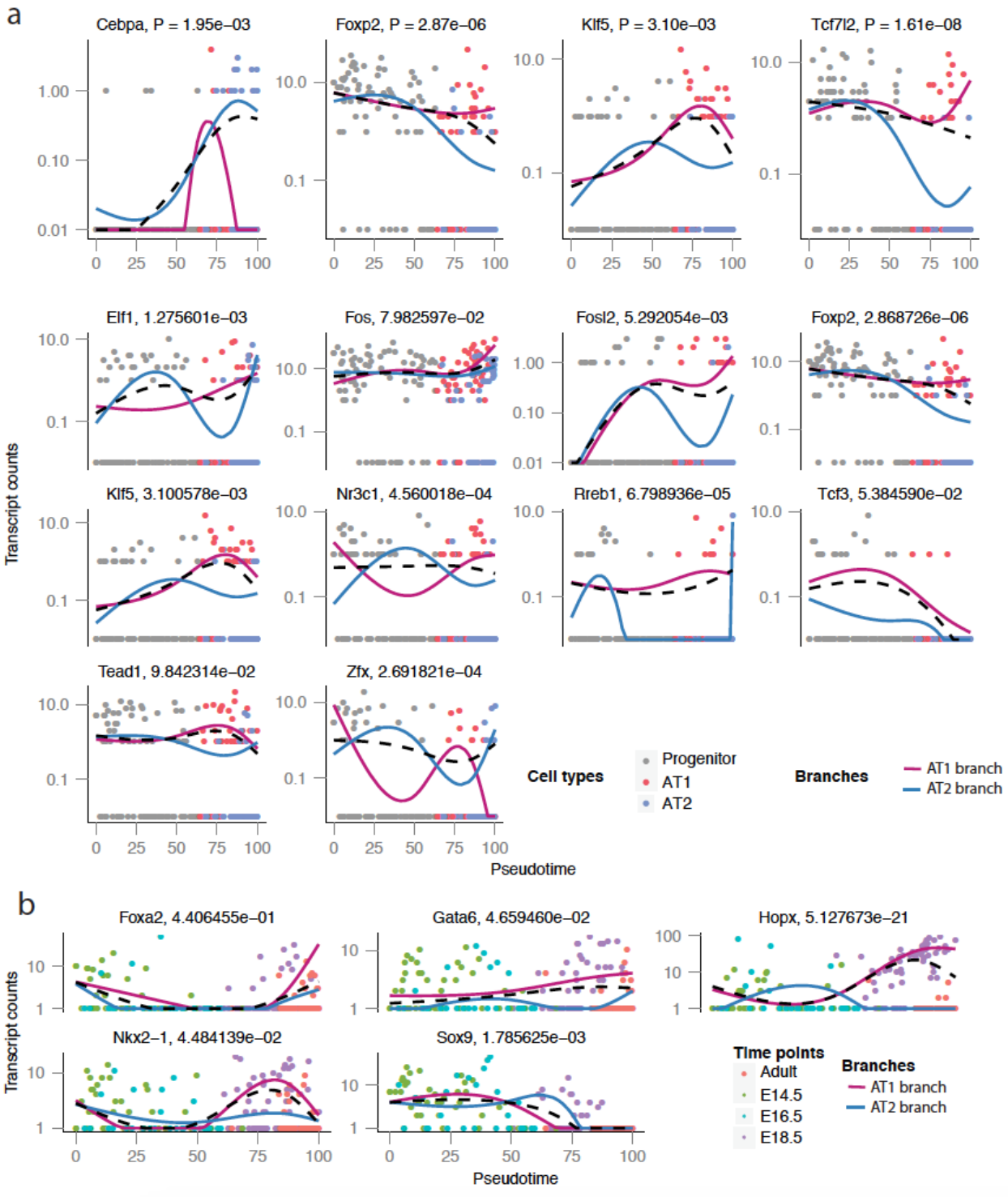


Supplementary Figure 2. Monocle's Single-cell trajectory of developing murine pneumocytes. Cells are colored according to type assignments made by Truetlein *et al*. Only cells collected at E18.5 were assigned in the original study. Bounding boxes show Monocle's assignment of cells to types based on the topology of the trajectory.



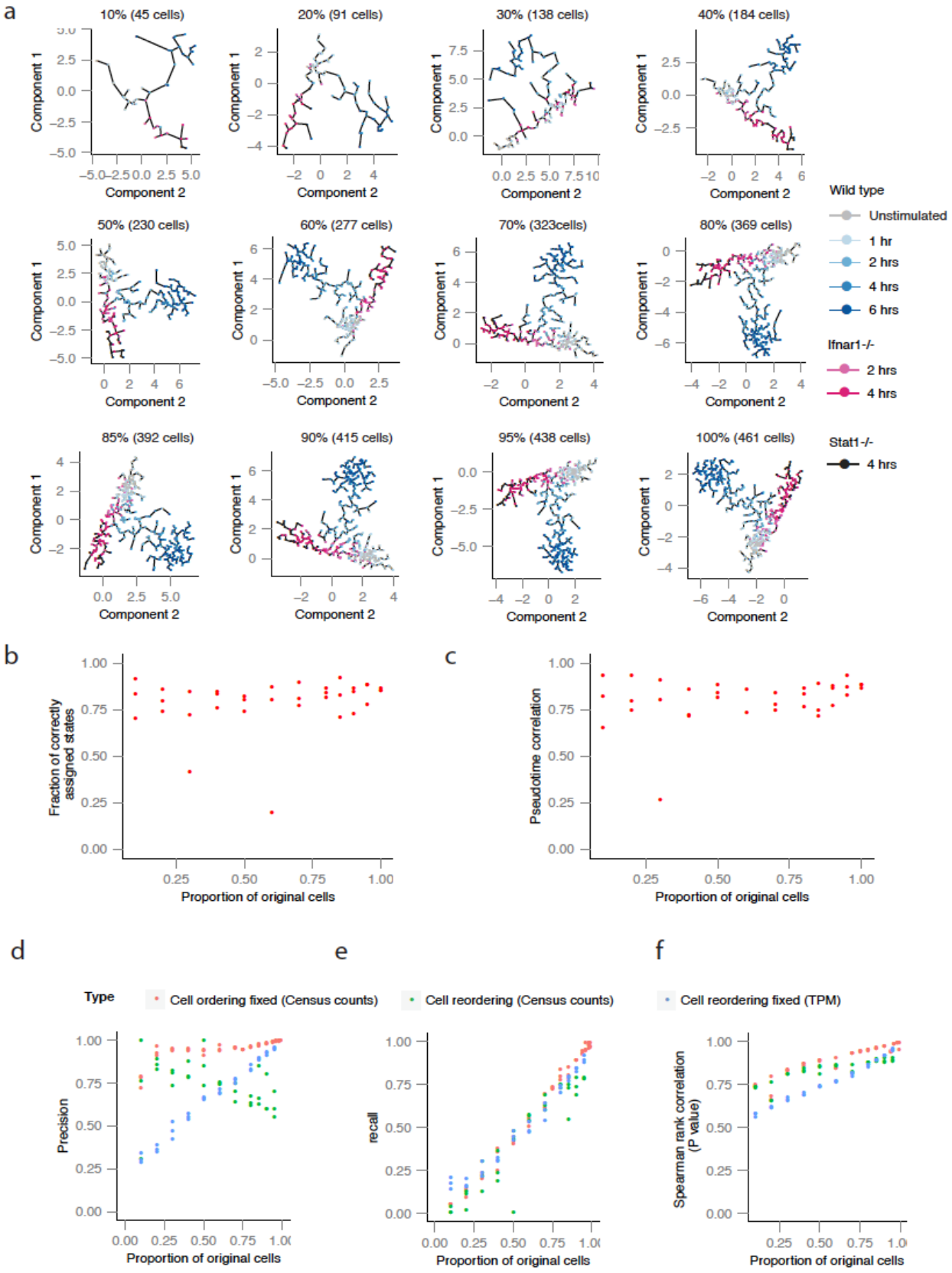
Supplementary Figure 3. BEAM analysis reveals branch-dependent gene expression during

lung epithelial specification. (a) Hierarchical clustering of normalized expression in terms of transcript counts for markers of pneumocyte specification as defined by Treutlein *et al.* and cell cycle genes. Each column represents a cell ordered along the trajectory. The center of the heatmap corresponds to the beginning of the trajectory. Moving left proceeds down the AT1 branch, whereas moving right proceeds down the AT2 branch. Each row represents the transcript counts for a gene on each branch. Rows are hierarchically clustered using Pearson's correlation with Ward's method. (b) Pseudotime distribution of branch points for markers of early and late pneumocyte specification as defined by Truetlein *et al.* (c) TF motif enrichment within the hypersensitive sites proximal to the branching genes in each of the kinetic clusters. Motifs corresponding to transcription factors that are themselves significantly branch dependent are labeled in red.



Supplementary Figure 4. Branching expression kinetics for predicted and known regulators of lung epithelial cell fate and function. (a) Significant branching TFs which also

have binding motifs enriched in the upstream of significant branching genes shown in figure 4b. (b) Known regulators of lung epithelial cell fate specification.


a
10% (45 cells)
20% (91 cells)
30% (138 cells)
40% (184 cells)
50% (230 cells)
60% (277 cells)
70% (323cells)
80% (369 cells)
85% (392 cells)
90% (415 cells)
95% (438 cells)
100% (461 cells)
Component 1
Component 2
Wild type
Unstimulated
1 hr
2 hrs
4 hrs
6 hrs
Ifnar1-/-
2 hrs
4 hrs
Stat1-/-
4 hrs
b
Fraction of correctly assigned states
Proportion of original cells
c
Pseudotime correlation
Proportion of original cells
d
e
f
Type
Cell ordering fixed (Census counts)
Cell reordering (Census counts)
Cell reordering fixed (TPM)
Precision
recall
Spearman rank correlation (P value)
Proportion of original cells

Supplementary figure 5. Robustness of BEAM analysis to number of cells collected. The original set of cells from the KO experiment from Shalek et al. were downsampled randomly at progressively lower proportions of the original dataset. For each subset, cells were re-ordered with Monocle 2, and then BEAM tests were then performed. Three different subsets were performed per proportion. (a) Example trajectories for each proportion. (b) Fraction of cells assigned to the same branch as in the full dataset. (c) Pearson correlation of pseudotime for cells included in each sample compared to the relative ordering of these cells in the full dataset. (d) Precision and (e) Recall of genes reported as significantly branch-dependent by BEAM for each sample of the cells, using the branch-dependent genes from the full dataset as the "ground truth". (f) Spearman rank correlation of p-values returned by BEAM for all genes compared to their p-values when BEAM is given the full dataset. Red (blue) dots in panels d, e, f correspond to cell downsampling with cell ordering fixed using transcript count data (TPM expression data), green dots cell downsampling with cell re-ordering as in panels a, b. For TPM data, the same ordering as for transcript counts is used.

Supplementary method

Testing for branch-dependent expression

Monocle assigns each cell a pseudotime value and a "State" encoding the segment of the trajectory it resides upon based on the PQ-tree algorithm (see the supplemental material for Trapnell and Cacchiarelli et al for further information[15]). Transcript counts values were variance-stabilized[48] via the technique described by Anders and Huber prior to tree construction. Monocle assigns each cell a pseudotime value and a "State" encoding the segment of the

trajectory it resides upon based on the PQ-tree algorithm (see the supplemental material for Trapnell and Cacchiarelli et al for further information[15]). Transcript counts values were variance-stabilized[48] via the technique described by Anders and Huber prior to tree construction.

In Monocle 2, we extended the capability to test for branch-dependent gene expression by formulating the problem as a contrast between two negative binomial GLMs.

The null model

$$NB(Transcript\ counts)\tilde{}sm.ns(Pseudotime)$$

for the test assumes the gene being tested is not a branch specific gene, whereas the alternative model:

$$NB(Transcript\ counts)\tilde{}sm.ns(Pseudotime)+Branch+sm.ns(Pseudotime):Branch$$

assumes that the gene is a branch specific gene where : represents an interaction term between branch and transformed pseudotime, NB means negative binomial distribution. Each model includes a natural spline (here with three degrees of freedom) describing smooth changes in mean expression as a function of pseudotime. The null model fits only a single curve, whereas the alternative will fit a distinct curve for each branch. Our current implementation of Monocle 2 relies on VGAM's "smart" spline fitting functionality, hence the use of the *sm.ns()* function instead of the more widely used *ns()* function from the splines package in R[49]. Likelihood ratio testing was performed with the VGAM *lrtest()* function, similar to Monocle's other differential expression tests[49]. A significant branch-dependent genes means that the gene has distinct expression dynamics along each branch, with smoothed curves that have different shapes.

To fit the full model, each cell must be assigned to the appropriate branch, which is coded through the factor "Branch" in the above model formula. Monocle's function for testing branch dependence accepts an argument specifying which branches are to be compared. These arguments are specified using the 'State' attribute assigned by Monocle during trajectory reconstructions. For example, in our analysis of the Truetlein *et al* data [10], Monocle reconstructed a trajectory with two branches (L_{AT1}, L_{AT2} for AT1 and AT2 lineages, respectively), and three states ((S_{BP}, S_{AT1}, S_{AT2} for progenitor, AT1, or AT2 cells). The user specifies that he or she wants to compare L_{AT1} and L_{AT2} by providing S_{AT2} and S_{AT2} as arguments to the function. Monocle then assigns all the cells with state S_{AT1} to branch L_{AT1} and similarly for the cells. However, the cells with must be members of both branches, because they are on the path from each branch back to the root of the tree. In order to ensure the independence of data points required for the LRT as well as the robustness and stability of our algorithm, we implemented a strategy to partition the progenitor cells into two groups, with each branch receiving a group. The groups are computed by simply ranking the progenitor cells by pseudotime and assigning the odd-numbered cells to one group and the even numbered cells to the other. We assign the first progenitor to both branches to ensure they start at the same time which is required for downstream spline fitting and clustering. The branch plots in Figure 3d visualize the branch specific spline curves fit by this method.

Branch time point detection

The branching time point for each gene can be quantified by fitting a separate spline curves for each branch from all the progenitor to each cell fate. To robustly detect the pseudotime point (t$^{t^i_\beta}$

) when a gene i with a branching expression pattern starts to diverge between two cell fates L_1, L_2, we developed the branch time point detection algorithm. The algorithm starts from the end of stretched pseudotime (pseudotime $t = 100$, see below) to calculate the divergence ($D_i(t = 100) = x_{L_1}(t = 100) - x_{L_2}(t = 100)$) of gene $i(x_{L_1}(t = 100), x_{L_2}(t = 100)$'s expression between two cell fates, L_1, L_2, (for a branching gene, the divergence at this moment should be large if not the largest across pseudotime). It then moves backwards to find the latest intersection point between two fitted spline curves, which corresponds to the time when the gene starts to diverge between two branches. To add further flexibility, the algorithm moves forward to find the time point when the gene expression diverges up to a user controllable threshold (ϵ), or $D_i(t) \leq \epsilon(t)$, and defines this time point as the branch time point, y_{\beta}^i, for that particular gene i.

Stretching raw pseudotime to account for heterochrony between branches

From reconstructed trajectories, we observe that branches of a trajectory can have different lengths of pseudotime, which implies different rate of transcriptomic changes for each branch (assuming there is even sampling during the biological branching process). This observation is an example of *heterochrony*, and we address it by simply "stretching" the shorter branch and the longer branch uniformly to 100 (assuming 100 percentage completion at the end of each branch). This approach can help us to consider both branches and avoid excluding cells on the longer branch that extend beyond the edge of the shorter one's pseudotime scale. Our current method amounts to a naive alignment between two pseudotime series when branches are of approximately equal length. We leave more sophisticated scaling techniques as future work. The

transformation of the raw pseudotime into scaled pseudotime is shown in (Eq. 8). We first scale the raw pseudotime (t) of the longest branch (L_l) to a range of pseudotime (T) with values from 0 − 100 (including all the progenitor cells located between pseudotime 0 and the branch pseudotime, t_β^c which corresponds to the raw pseudotime of the latest progenitor cell), of the trajectory ($0 - t_\beta$)). The time range between the branch pseudotime and the end of the shorter branch ($t_\beta^c \leq t_{L_s} \leq t_{L_s}^E$) is stretched to a maximum value of 100 to match the longer branch (L_l).

$$T = \begin{cases} t_{L_l} \cdot \dfrac{100}{t_{L_l}^E} & (0 \leq t_{L_l} \leq t_{L_l}^E | 0 \leq t_{L_s} \leq t_\beta^c) \\ \dfrac{100 - t_\beta^c}{t_{t_s}^E} \cdot t_{L_s} & (t_\beta^c \leq t_{L_s} \leq t_{L_s}^E) \end{cases}$$

We should note that there is only one branching time point, t_β^c for the reconstructed trajectories but the branching time point for each gene i, t_β^i can be different although it tends to center around t_β^c.

Clustering genes by branch-dependent expression kinetics

All significantly branching genes can be used to make the branched heatmap. The branched expression dynamics for all the branching genes are firstly obtained from fitted spline curves. Then, the expression values from the fitted curve are variance stabilized[2] and transformed to Z-scores. The scaled values are also truncated at 3 or -3 to ensure better visualization of the data and to avoid the outliers. On the heatmap, the progenitor cells are located towards the center and cells in each branch appear towards the left and right. The *ward.D2 clustering* method is then

applied on the correlation matrix for the transformed data between all the genes (the number of clusters was chosen to be six for all datasets). To obtain the enriched GO BP or Reactome terms, we then performed the hypergeometric test on the corresponding Gene Matrix Transposed file format (GMT) file for each cluster of genes based on the piano package[3].

Transcription factor motif enrichment analysis

Regulatory regions of genome were identified as peaks, either downloaded from mouse ENCODE database (for lung epithelium data) or obtained using MACS package[4] (for dendritic cell data). We link the regulatory elements to genes by proximity of those elements to genes in upstream / downstream 5 kb of the specific gene using bedtools[5]. Motif data from JASPAR is then used to scan DHS peaks which generates a group of potential targets for the genes having that specific motif. The motif enrichment for each cluster of branching genes is performed based on hypergeometric test using the genes with the motifs targeting that particular cluster of genes against all the genes with motifs and their potential targets as the background. We then retrieved the genes whose motifs are enriched in certain cluster (FDR < 10%), Benjamini-Hochberg correction) but also belong to significantly branch-dependent genes.

Measuring relative expression with transcript counts

Although in principle per-cell transcript counts allow us to track changes in absolute gene expression, we often wish to use these values to test for changes in relative expression. We could of course do so by simply dividing the expression level for a gene by the total transcript recovered from a cell . However, in practice, we find there are alternative means of normalizing for variation in total transcript recovery across cells that result in better results in downstream

analysis. We tested two schemes for computing *size factors*. The first is the method proposed by Anders and Huber for library size normalization:

$$S_i = median_j \frac{x_{ij}}{(\Pi_{v=1}^{n} x_{vj})^{\frac{1}{n}}}$$

in which x_{ij} is the transcript count of gene j in cell i, n is the number of cells, and C is the number of genes. We found, however, that this scheme often produces poor results or fails entirely to compute finite size factors. The underlying problem with this method is that the calculation must exclude all genes except those that are detectably expressed in all cells. As has been discussed here and elsewhere, current single-cell RNA-Seq experiments are enriched with zeros (arising from "dropout" effects) [6], such that there may not be a single gene that is non-zero in all cells. When the above procedure works, it typically includes only the most abundantly expressed genes in the calculation, raising concerns about bias.

We instead chose to use the following alternative equation for size factor calculation:

$$S_i = \frac{\sum_{j=1}^{C} x_{ij}}{(\Pi_{v=1}^{n} x_{vj})^{\frac{1}{n}}}$$

This equation will return finite values provided that there are no cells with zero expression for all genes in the experiment. For such cells, we simply define the size factor to be 1.

Choosing a distribution to model single-cell expression

Parametric tests for differential expression between two or more conditions or as a function of one or more covariates typically require that expression be described accurately by well-defined

probability distribution. Various choices, such as the Poisson, quasi-Poisson, negative binomial, Tobit, have been used to model RNA-seq data [5,7,8]. The choice of distribution depends in part on whether expression is measured by read counts, TPM values, or some other scale. To identify an appropriate distribution for modeling single-cell expression data, we performed a series of chi-squared goodness of fit tests for every gene using the fitdistrplus package. For each gene, we fit two negative binomial distributions to its expression values in Truetlein *et al*. For the first, we measured expression in read counts. For the second, we provided expression as transcript counts. We also fitted zero-inflated negative binomial distribution to the same data to assess whether explicit modeling of dropouts would improve downstream analysis. All fits were conducted with the fitdistrplus package. Genes were considered to pass the goodness of fit test when they generated p-values greater than 0.1. For many genes, fitdistrplus threw numerical exceptions, indicating an internal failure to fit a particular distribution to a particular gene. We tracked these to assess the tradeoff between an increased quality of fit and lower numerical robustness of the fitting algorithm.

Differential gene expression tests on transcript counts in Monocle 2

We used cells collected at E14.5d (44 cells) and E18.5d (66 cells) to perform the two-group test. To benchmark the performance of Monocle 2 on differential gene tests, we test on relative abundance expression, the raw read counts data, the transcript counts data recovered from spike-in regression, Census, TPM scaled to 100, 000 total transcripts, TPM using negative binomial distribution and TPM scaled to the true total calculated from the spike-in regression.

The underlying GLM models the expression values for each gene using relative abundance expression as Tobit distributions and from the other three as negative binomial distributions.

Monocle originally accepted only relative expression values (e.g. TPMs) as input. Monocle then used the VGAM package[9] to model expression of each gene across the cells in an experiment with a Tobit distribution. To support analysis of transcript counts values, in Monocle 2, we use a negative binomial distribution. If transcript counts values are used, Monocle 2 tests for differential expression using a likelihood ratio test on two negative binomial models. The size parameter is modeled as an intercept only by setting the argument in the negbinomial function from VGAM, which amounts to assuming that dispersion is similar between conditions. We also provide a starting estimate regarding the value of the size parameter. To do so, we first fit a gamma generalized linear model (identical to the method used by DESeq[2]) to the entire data matrix prior to testing, and then look up the size parameter for each gene using this curve. This increases the accuracy of the test and reduces numerical exceptions thrown by VGAM. To consider extreme cases in which the negative binomial cannot fit the data at all, we relax the convergence criteria for the negative binomial distribution fit (from a VGAM "epsilon" parameter value of 1e-7 to 1e-1) to ensure that most genes can be effectively tested.

Benchmarking differential expression analysis

To benchmark the accuracy of differential gene expression tests between two groups of cells and for pseudotime-dependence, we first defined a "ground truth" set of differentially expressed genes (DEGs). For the two-group test, we implement a permutation test for log mean fold changes in spike-in derived expression levels between E18.5d cells and E14.5d cells as well

E16.5d cells and E14.5d cells. Cell labels were permuted 10,000 times to generate the distribution of log fold changes under the null hypothesis of no difference between the groups. In order to maximize agreement between DEG test tools, which all include some form of size factor normalization, we calculated size factors for each cell as described above and scale the RNA counts by these factors before the permutation test. Tools were provided with relative expression levels, normalized read counts, and transcript counts estimated with spike-ins, Census counts, TPM (true total) counts which are derived by scaling TPM values by the correct per-cell total RNA. TPM (true total) control shares Census' inability to control for amplification bias, but begins with the same total per-cell transcript counts available through spike-ins. Comparing this control to spike-based regression reveals the impact of amplification bias on differential analysis in single cells. Comparing it to Census assesses how error in estimating total transcript counts translates into error in differential analysis. To perform the permutation test for genes that change as a function of pseudotime, we modified the glm.perm package to permute the residual for the regressors (corresponding to three degrees of freedom for the spline curve in the design matrix of the GLM models) of the pseudotime spline for inferring the *P* value[10]. Because we defined permutation-based ground truth DEGs for each measurement type. We then ran Monocle 2, DESeq2 (version 1.8.1) [11], edgeR (version 3.10.5)[7], SCDE (version 1.99.0)[6] for the two-group test. For benchmarking the pseudotime test, we adjusted DESeq1 for performing pseudotime test. We compared the programs on the basis of four criteria: 1) the area under the ROC curve, 2) precision, 3) recall and 4) F1 score (as defined below). Precision, recall, and F1 were assessed when the programs were run with target false discovery rates of 10%. All programs were run

with default input parameters while setting the size factors calculated by Monocle 2. For edgeR and DESeq2, we used likelihood ratio tests.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

We also added a benchmark analysis based on the bulk RNA-seq data for the HSMM experiment, which was collected at same time points as single cells. Of genes expressed above 10 cells, the top 1,000 most significant genes from the Cuffdiff test between time 0 and and 72 hours are used as true positives while the bottom 1, 000 least significant genes are used as true negatives. We benchmark Monocle 2, DESeq2, edgeR, and SCDE with the above ground truth using the either TPM, read counts or census counts as the measurement type. For details on benchmarking the ESC dataset, see below.

Obtaining data and estimating expression from single-cell RNA-Seq reads

For the lung epithelial cell data analysis, all data from Treutlein *et al.*[12], which include 199 samples in total, was downloaded from GEO[12] (accession ID: GSE52583) and grouped by time (E14.5d, E16.5d, E18.5d and Adult AT2 cells) and annotated according to metadata available on GEO.

For the neuron dataset analysis, all data was downloaded from GEO (accession ID: GSE67310) and grouped by cell according to metadata available on GEO [13].

For the Shalek *et al.*[14] data analysis, all data was downloaded from GEO (accession ID: GSE48968) and grouped by cell according to metadata available on GEO.

All SRA files were converted to FASTQ format and aligned to the mm10 (both of lung epithelial cell data, neuron and dendritic cell data) reference using Tophat[15]. GENCODE M1 with spike-in controls (lung epithelial cell data) or GENCODE M2 (dendritic cell data) transcript annotations were provided to Tophat during alignment. Gene and isoform-level expression matrices were generated for all cells using cuffquant and cuffnorm[1]. For the lung epithelial cells, relative expression values were first converted to transcript counts based the regression relationship between spike-in transcript counts and relative abundance in each cell. We also converted relative expression values into transcript counts using the spike-in free algorithm.

We used the script read_distribution.py from RSeQC v2.6.2[16] to collect read mapping statistics as a measure of library quality for the lung epithelial data and dendritic cell data.

Analysis of lung epithelial cell and neuron reprogramming data

All cells annotated as ciliated cells, clara cells or bulk sample from Supplementary Data 5 in Treutlein et al. were excluded, yielding 183 cells for downstream analysis. The developmental trajectory in lung epithelial data was reconstructed based on the entire set of marker genes, highest PCA loading genes (218 in total) collected from their paper. We used log-transformed

TPM values as input data to the dimensionality reduction with ICA, to maximize consistency with the original study.

We then applied BEAM to detect all the genes that are significantly branch-dependent between the AT1 and AT2 branches based on the transcript counts recovered using spike-in standards. State 1 cells inferred by Monocle 2 corresponded to progenitor cells, whereas State 2 and State 3 cells corresponded to AT1 and AT2 branches respectively.

All significantly branching (FDR < 5%) genes are used to make the branched heatmap. The branched heatmap is generated as described in *Clustering Genes by branch-dependent expression kinetics*.

For the motif enrichment, DHS narrow peaks for the lung, retrieved from mouse ENCODE dataset (file names: wgEncodeUwDnaseLungC57bl6MAdult8wksPkRep1-3.narrowPeak.gz) Enrichment analysis is performed as described in *Transcription factor motif enrichment analysis*.

For benchmarking the Census vs. the spike-in recovery for the lung dataset, we run Census with all default parameters.

For the neuron dataset, all 405 cells from day 0, 2, 5 ,20, 22 upon Ascl1 induction during reprogramming from MEFs to iN are used. Cells from day 0 and day 20 are used to benchmark the differential expression performance of census compared to spike-in based regression. Note that in this reprogramming experiment, external RNA spike-in transcripts (ERCC spike-in Mix,

Ambion) were added to all single-cell lysis reactions at a dilution of 1:40,000 (and 1:4,000 in case of day 5 experiment) [13].

Analysis of dendritic cells from knockout mice

Before analysis, all cells that do not appear in the sample metadata sheet provided on GEO (because they were filtered for failing quality control) are removed from the expression matrix, yielding 1787 cells in cuffnorm output matrices. Relative abundances are then converted into transcript counts using Census with default parameters with parameter estimated from the relative abundance data for each cell.

We selected cells (510 in total) annotated as unstimulated replicate (normal unstimulated cells were observed to have low RNA library quality), LPS stimulated cells without any perturbations, and LPS stimulated cells with Stat1 and Ifnar1 knocked out taken at each of the included time points.

The genes used for dimensionality reduction and pseudotime ordering were expressed in at least 50 cells and differentially expressed between wild-type and knockout cells and across LPS stimulation time with q values < 1e-34 (Benjamini-Hochberg correction) using the transcript counts data recovered from the spike-in free algorithm (no spike-in controls were present in this dataset). Transcript counts for these genes were then variance stabilized using the procedure introduced by Anders and Huber[2]. Briefly, each gene's empirical dispersion expression relative to the Poisson is estimated by the method of moments. A gamma-family generalized linear model is then fit to capture how dispersion varies with the mean across cells. This model has two

parameters: the "extra poisson" dispersion q and the "asymptotic" dispersion a. We then transform each transcript count x by the function

$$vst(x) = log(\frac{1 + q + 2ax + 2\sqrt{(ax(1 + q + ax))}}{4a})\frac{1}{log(2)}$$

The resulting transformed counts are more homoscedastic, which generally improves downstream analysis with PCA or ICA. For the knockout cells we followed the VST with a linear transformation that removes variation attributable to mRNA recovery efficiency using the removeBatchEffects function of the limma package. This acts to exclude some technical variation from the data, so it doesn't drive dimensionality reduction and trajectory analysis.

BEAM was applied to the recovered transcript count data after correcting for size factor normalization (see the section on *Measuring relative expression with transcript counts*) to identify genes that are significantly branching between the normal cells and the knockout cells.

State 1 cells as inferred by Monocle 2 corresponded to unstimulated DC cells, State 2 and State 3 corresponded to the normal and knockout lineages respectively. The heatmap of significantly branching genes (q values less than 0.01, Benjamini-Hochberg correction) and motif enrichment are performed as described in *Clustering Genes by branch-dependent expression kinetics* and *Transcription factor motif enrichment analysis*. We used the Gene Matrix Transposed file format) file from the Reactome database to perform the gene set enrichment analysis on each cluster of the branched heatmap.

For the motif enrichment analysis, we used the union of the ATAC-seq peaks from Ido Amit's group [17] (GEO accession GSE63341). For the bar plot of the motif enrichment on cluster

4 (Figure 5d) from the heatmap, all the motif enrichment scores (-log10(q-val) are used but only TFs with known Interferon signal pathway related annotations are labeled with gene names.

Analysis of UMI data

Grün *et al*[18] recently published a comparison of the global gene expression variability between growing J1 mouse embryonic stem cells (mESCs) in serum/LIF medium and that in 2i medium. The UMI transcript counts data matrix was downloaded from GEO (accession ID: GSE54695). mESCs in serum/LIF medium (annotated as SC_2i from the metadata) and J1 mESCs in 2i medium (annotated as serum_2i) were used. Pool and split controls from the study were removed, yielding 131 cells for downstream analysis. mESCs in serum medium and mESCs in 2i medium are used as two groups for performing the two group tests, and the benchmark gold sets were generated from the permutation tests as described previously.

For benchmarking the Census vs. the spike-in recovery for the UMI dataset, we use the 50 ERCC spike-in transcripts detected in this dataset. We run Census with all default parameters.

Analysis of ESC data

Sophie *et al*[19] recently used sc RNA-seq to study the X chromosome expression dynamics during human embryogenesis. In this study, the single-cell was manually picked by glass, following by using SMART2-seq for preparing the cDNA library. The rpkm table, ercc annotation files are downloaded from ArrayExpress (accession ID: E-MTAB-3929). Poor cells based total transcript counts calculated from spike-in regression are removed which leads to 1380 cells out of a total of

1529 for benchmarking Census vs. the spike-in. Capture rate is estimated based on the spike-in. We run Cenus with all default parameters.

Robustness of tree reconstruction and BEAM test to number of cells included in analysis

In order to test robustness of BEAM and the preceding dimensionality reduction and tree construction to the number of cells collected (simulating how results would change with smaller original experiments), we performed cell downsampling simulations on the Shalek *et al.* KO data. Starting with the original dataset., we generated subsets containing fractions of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.85, 0.9, 0.95, and 1.0 of the original dataset. Sampling was performed without replacement and three different subsets were generated for each proportion to serve as replicates. For each subset, we completely reran the differential expression analysis to obtain a set of ordering genes, recalculated dimensionality reduction, and reordered cells using the same procedure as for the original dataset. Ordering was performed using the same number of top most significant genes as for our main analysis of the Shalek *et al.* KO data rather than using a p-value cutoff to account for the progressively reduced power of differential expression tests in smaller subsets. As the arbitrary numeric assignment of which branches in the tree correspond to state 1, 2, and 3 respectively can vary during these runs, we swapped the numeric assignment of state values to maximize overlap with the original tree if necessary. This does not change which cells are assigned to a common branch; only which number (1, 2, or 3) each group of cells is assigned for consistency in downstream analysis. Finally, we performed the BEAM test on each ordered subset. A summary of the resulting tree reconstruction and BEAM test results is shown in *Supplementary Figure 12*.

In addition to test the robustness of BEAM solely contributed to the cell downampling while avoid the conflation from the the dimension reduction, state and pseudotime assignment, we performed another benchmark where only the cells are downsampled but the ordering of cells is kept fixed. This second analysis provides an upper bound on the accuracy of the test, and excludes the impact of variability arising from the trajectory itself. We also included a downsampling BEAM test for the TPM data while fixing the cell ordering as that inferred based on the transcript counts data.

Generality of census on testing differential gene expression

In order to demonstrate that Census is generally applicable, four additional datasets with ERCC spike-in transcripts (Pancreas: E-MTAB-5061[20], Cortex: http://linnarssonlab.org/cortex/ [21], marker-free: GSE54006 [22], quantitative assessment data: GSE51254 [23]) were downloaded from the original studies. We converted each dataset into TPM values and calculate absolute transcript counts based on the spike-in regression. Capture rate and mode of corresponding transcript counts either in cDNA or cell lysate space are estimated as described above. Census is then run by default parameters to obtain the estimated transcript counts. Based on the annotation of the cells from each study, we select two groups of cells for making the MA plot as well as the log fold-change plot. For the quantitative assessment dataset, which doesn't have biological conditions, two random sample sets of cells from the 96 cells using nanoliter volume sample processing on a microfluidic platform are selected for making the MA and log fold-change plot.

(a) Census related variables:

Variables	Descriptions	Remarks
	Relative abundance (FPKM, TPM) of gene expression in each cell	(here and below)
	Transcript counts in each cell	lowercase is on log scale; hat represents estimates based on Census
	Mode of relative abundance in log-scale based on Gaussian kernel	
	Corresponding copy number of the mode of relative abundance in log-scale	hat represents estimates based on the calibration procedure in Census
	Mode of the transcript counts	Mode is calculated on log-scale based on Gaussian kernel and converted back to original scale
	Degradation rate (including loss) of RNA from cell into cell lysate; degradation rate of spike-in RNA added into the cell lysate	is a unknown parameters in single-cell RNA-seq experiment. is assumed to be one or there is no loss for the spike-in RNA.
	Capture rate of RNA transcripts from cell lysate to cDNA	Capture rate can be estimated based on spike-in ladders using a binomial distribution
	Amplification rate from cDNA to sequencing reads; number of total sequencing reads	This value is normally large than 1 when sequence depth is enough which means that all the cDNA are observed

		in the sequencing machine
	Slope and intercept of the regression relationship from relative abundance into transcript counts	hat represents estimates based on the optimization procedure in Census
	Total mRNAs counts in cell lysate (including spike-in transcripts), total spike-in mRNAs added into cell lysate , total endogenous mRNAs counts from cell in cell lysate	lowercase is on log scale
	Total mRNAs in the cell	
	Number of gene expressed	
	Cell or gene index	
	Probability to capture for a single transcript; the probability to observe at least once for a spike-in transcript; probability for all spike-in transcripts with copies have non-zero FPKM values	
	The relative abundance of each spike-in transcript to the total spike-in transcript counts; a constant introduced from the specific relative abundance metric; the fraction of total spike-in transcript counts relevant to the total RNA	

(b) BEAM related variables:

Variable	Description	Remarks
	Pseudotime for each cell estimated by monocle	
	Lineages or branches for cells assigned by monocle	
	States for cells assigned by monocle	State means a segment of a tree structure returned by the PQ-tree algorithm in monocle. A single bifurcation tree, like the lung data, has two branches (lineages) and three states
	Monocle detected branch time point for a gene	
	Extended pseudotime in monocle accounting for the heterochrony between different branches	

(c) Isoform switch analysis related variables:

Variable	**Description**	**Remarks**
	count of isoform of a gene	count of isoform is recovered from Census algorithm
	The frequency of the	

	isoform	
	Over-dispersion parameters in Dirichlet distribution	

(d) Allele switch analysis related variables:

Variable	Description	Remarks
	Probability that an RNA originated from the maternal allele	
	Maternal and paternal RNA counts	Allele-specific relative gene expression values were estimated by Kallisto and converted into transcript counts by Census

Supplementary table 2. Variables used in Census, BEAM, isoform and allele switch analysis. (a) A table describing all variables used in the Census algorithm. (b) The same for BEAM. (c) The same for isoform switch analysis. (d) The same for allele switch analysis.

Chapter 4. Detect causal interaction from single-cell measurements

A version of this chapter will be submitted as a manuscript

Detecting causal regulations from single-cell measurements. **Xiaojie Qiu**, Arman Rahimzamani, Li Wang, Qi Mao, Timothy Durham, José L McFaline-Figueroa, Lauren Saunders, Cole Trapnell, Sreeram Kannan

4.1 Introduction

Most biological processes, either in development or disease progression, are governed by complicated gene regulatory networks. It is thus of great interest to computationally reverse engineer the network from observational data. In the past few decades, various network inference algorithms(Faith et al. 2007; Margolin et al. 2006; Meyer, Lafitte, and Bontempi 2008; Friedman et al. 2000; Langfelder and Horvath 2008) have been developed. However, they are designed specifically for bulk gene expression measurements. Single-cell measurements reveal the natural variation of expression dynamics in a large population of cells and provide unprecedented resolution of the gene expression cascades required for accurate inference of regulatory networks(Liu and Trapnell 2016).

Recently, we developed algorithms to accurately order cells along a "pseudotime"(Trapnell et al. 2014a; Qiu, Mao, et al. 2017a) trajectory based on the learned "structure" of their transcriptome. Our algorithms can also reveal gene bifurcation hierarchy(Qiu, Hill, et al. 2017c) from single-cell (sc-)RNA-seq data associated with lineage bifurcation (where one progenitor cell type commits to two distinct cell fates). These results reinforce the finding

that dynamic expression changes of putative regulators often precede that of their downstream targets(Bar-Joseph, Gitter, and Simon 2012), suggesting that *causal* gene regulations can be inferred from reconstructed pseudotime-series data. In this study, we define causality in terms of the strength of information transferred from one variable, a potential regulator, to another time-delayed response variable, a potential target.

Various causality inference methods have been proposed. Causal Inference based on Granger causality (GC)(Granger 1969), a statistical hypothesis test for determining whether one time series (X_1) is useful in forecasting another (X_2), has been applied to biological networks. However, its assumption of linear causality is violated in biological settings(Hill et al. 2016). Recently, an exciting complement, Convergent Cross-Mapping (CCM)(Sugihara et al. 2012) has been proposed for ecological system. CCM is based on state-space reconstruction(Takens 1981) and can detect pairwise non-linear interactions. Unfortunately, this method is limited to *deterministic* systems.

Recently, network inference has been applied to single-cell genomics. The SCENIC method(Aibar et al. 2017) combines GENIE3(Huynh-Thu et al. 2010) with regulatory binding motif enrichment to simultaneously cluster cells and infer regulatory networks. Others studies have inferred regulatory networks from scRNA-seq data using for example, differential equations, information measures or linear regression techniques(Huynh-Thu et al. 2010; Ocone et al. 2015; Chan, Stumpf, and Babtie 2017; Matsumoto et al. 2017; Hamey et al. 2017; Wei et al. 2017; Sanchez-Castillo et al. 2017; Papili Gao et al. 2017; Fiers et al. 2018; Babtie, Chan, and Stumpf 2017). However, most of those methods don't explicitly leverage time-series data, and

more importantly, may fail to recover the correct network even in simple settings(Fiers et al. 2018; Babtie, Chan, and Stumpf 2017) (see **Methods**).

Here we introduce Scribe, a scalable toolkit, that relies on *Restricted Directed Information* (RDI)(A. Rahimzamani and Kannan 2016), to accurately and efficiently infer causal regulatory networks from single-cell genomics datasets. The causal network inferred by Scribe is discussed in contrast to correlation networks as we explicitly consider the response of the targets to their putative regulators with some time delay. In contrast to GC and CCM, Scribe learns both linear and non-linear causality in deterministic and stochastic systems by measuring the information transferred from the potential regulator to their putative target. It incorporates rigorous procedures to alleviate sampling bias and builds upon novel estimators and regularization techniques to facilitate inference of large-scale directed causal networks. Additionally, Scribe provides intuitive approaches to directly visualize the responses, causality, and combinatorial causal regulations. Scribe is generally applicable to most time-series data including pseudotime and "RNA velocity"(La Manno et al. 2017), to detect causal interactions. To demonstrate the versatility of our method, we applied Scribe to real-time confocal imaging data of *C. elegans* early embryogenesis(La Manno et al. 2017; J. I. Murray et al. 2012) and built a compendium of temporal causal regulatory networks related to a whole organism's developmental history. As we move towards building organismal cell atlases using single-cell genomics, Scribe provides a platform from which to infer detailed regulatory networks governing cell lineage differentiation across all cell types.

4.2 Scribe, a toolkit for inferring and visualizing causal regulations from single-cell genomics datasets.

An upstream regulator's expression dynamics precede the response of its downstream targets(Bar-Joseph, Gitter, and Simon 2012). This fact can be leveraged to detect causal interactions between gene-pairs. Previously, we proposed RDI as a novel information metric to accurately and efficiently quantify causality(A. Rahimzamani and Kannan 2016; Arman Rahimzamani and Kannan 2017). Built upon RDI, we developed a toolkit, Scribe, that is designed for the analysis of time-series datasets, and is specially tailored for single-cell genomics, in particular, scRNA-seq datasets (**Supplementary Figure 1**, **Fig 1A**).

Scribe analyzes single-cell expression kinetics as cells progress along with a pseudotime trajectory (real time series, like the live cell imaging data). It then estimates causality scores (where a higher score implies stronger evidence for a causal interaction and *vice versa*) from this pseudotime-series using RDI for sets of gene-pairs that is then used to build a causal network (**Fig 1A, B**). Specifically, the RDI between two genes is formulated and quantified as the mutual information of the regulator's past state (x_{t-d}) and the target's current state (y_t) conditioned over the target's history (y_{t-1}) (or formally, $I(x_{t-d}; y_t|y_{t-1})$) (**Fig 1A**). Since pair-wise causal interaction may result from indirect interactions, Scribe supports conditioning on other potential regulators to remove any indirect causal interactions. To account for the sparsity of single-cell genomic data(Krishnaswamy et al. 2014), Scribe also uses a novel estimator recently reported by us(Krishnaswamy et al. 2014). As an optional technique to alleviate sampling biases from single-cell measurements, Scribe provides a rigorous approach to quantify the *potential*

causality(Arman Rahimzamani and Kannan 2017) (how much influence a regulator can potentially exert on target *without cognizance* to the regulator's distribution) (**Supplementary Figure 1)**. Scribe refines the inferred network using Context Likelihood of Relatedness (CLR)(Faith et al. 2007) and opts for a new method for directed graph regularization (see **Methods** or **supplementary materials** for more details). Scribe then plots the causal network and identifies regulatory motifs (**Fig 1B**). Finally, Scribe also provides a variety of methods to visualize complex causal regulations (**Fig 1B**).

To test the performance of Scribe in inferring causal networks, we simulated the differentiation of central nervous system with a minimal regulatory network through a set of stochastic differential equations (SDEs)(Qiu, Ding, and Shi 2012d) and generated multiple time-series. We then took those time-series as input to infer the causal network using Scribe and compared our results to other established causal inference approaches, including GC and CCM (See details in **Supplementary Methods**). When we use the minimal network as the reference to calculate the AUC (Area Under Curve) score of ROC (Receiver Operating Characteristics) curve for each network returned by each algorithm, we find Scribe outperforms alternative techniques across various settings on this simulation datasets (**Fig 1, Supplementary Figure 2**), thus confirming it is favorable to GC and CCM. In addition, we also benchmark with other algorithms reported for the DREAM, and find Scribe perform similarly to the reported best algorithm.

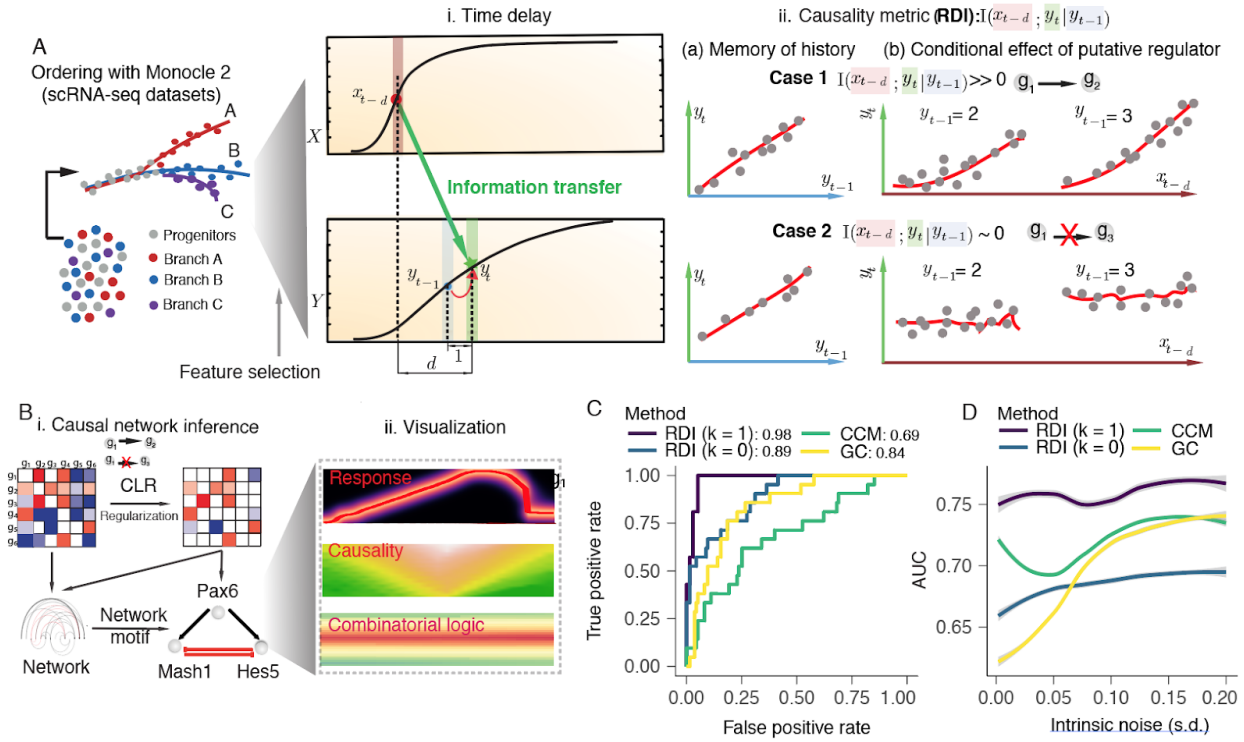


Figure 1: Scribe, a toolkit for inferring and visualizing causal regulations. (**A**). Scribe detects causality from pseudotime-ordered scRNA-seq datasets with a novel information metric, restricted direct information (RDI). We will first utilize Monocle 2 to resolve the trajectory for the scRNA-seq dataset, from which we will obtain a pseudotime-series. We can then apply feature selection procedures, for example BEAM(Qiu, Hill, et al. 2017c), to retrieve genes directly related to the regulatory mechanism of the biological process of interest. For a putative regulator-target pair, the current state of the target (y_t) receives information from the regulator's previous expression dynamics (x_{t-d}) along a pseudotime-series trajectory while also having the memory of its own intermediate previous state (y_{t-1}) (*i. Time delay*). Scribe relies on RDI(Qiu, Hill, et al. 2017c; A. Rahimzamani and Kannan 2016) to quantify the information transferred from the potential regulator to the target under some time delay while conditioned over its past

on this pseudotime-series data (*II. Causality metric*). A gene often has strong memory to its intermediate previous state (y_{t-1}) but RDI will only give highly positive causality score from the putative regulator to target in cases where there still is strong relationship between regulator's history and target's present conditioning on target's history (*II. Causality metric, Case 1 vs. Case 2*). (**B**) Inferring causal regulatory network and visualizing causal regulations with Scribe. Scribe infers a causal network via calculating the causal strength of all genes-pairs set automatically or by the user. Scribe then prunes the network first with a well established technique, Context Likelihood Relatedness (CLR), and opts for a new directed graph regularization method (See **supplementary method**) to reveal statistically significant edges while also satisfying certain well-known in/out-degree distribution constraints (*I. Causal network inference*). Scribe also incorporates a visualization framework to visualize the response function, causal interaction as well as combinatorial regulation between gene pairs (*II. Visualization*). (**C**) Receiver Operating Curves or ROC for Scribe, CCM and GC on a linear system. Standard deviation (s.d.) of independent additive noise injected into each gene at each time point and propagate through this system (intrinsic noise) is set to be 0.01. (**D**) Area Under Curve or AUC for Scribe, CCM and GC on the non-linear neurogenesis system under different s.d. of intrinsic independent additive noise. See **Methods** on details of the simulation setup.

4.3 Scribe visualizes transcriptional response, causal regulation, and combinatorial regulation logic

Visualization of gene regulations aids hypothesis generation; we thus provide in Scribe a set of intuitive visualizations of the associated information transfer (corresponds to causality) and the combinatorial regulation of multiple regulators.

Scribe visualizes causal regulations as a heatmap which encodes the expectation of target's current state given its immediate past and a putative regulator's previous state ($\mathbb{E}[Y(t)|Y(t-1), X(t-d)]$). From the heatmap, the horizontal dimension corresponds to the response function while the vertical dimension corresponds to the target's "memory of its history". Applying this visualization to simulated neuron lineage commitment data based on the minimal CNS network, we correctly recover a sigmoid function and a threshold-inhibition function for the delayed response of *Tuj1* to *Brn2* and that of *Mash1* to *Hes5*, respectively (**Fig 2**). Scribe is able to visualize various response functions and a multitude of patterns of information transfer in the CNS network as well as in other two-gene motifs (**Fig 2, Supplementary Figure 2-3**). Visualizing the current target's expected expression given two putative regulators' previous states through a Gaussian kernel ($\mathbb{E}[Y(t)|X(t-d), X(t-d)]$), we find Scribe is able to dissect additive regulations in the CNS network (**Fig 2A, D**), other common two-input combinatorial regulation logics (**Fig 2E, Supplementary Figure 3**) as well as direct or indirect regulators (**Fig 2A, D**). Additionally, Scribe also visualizes the time-delayed response function of the target to the regulator, similar to that of the previously described DREVI visualization(Krishnaswamy et al. 2014) **(Supplementary Figure 2-3**). These results

demonstrate Scribe's power to guide the interpretations of complex gene regulations using the described visualizations tools.

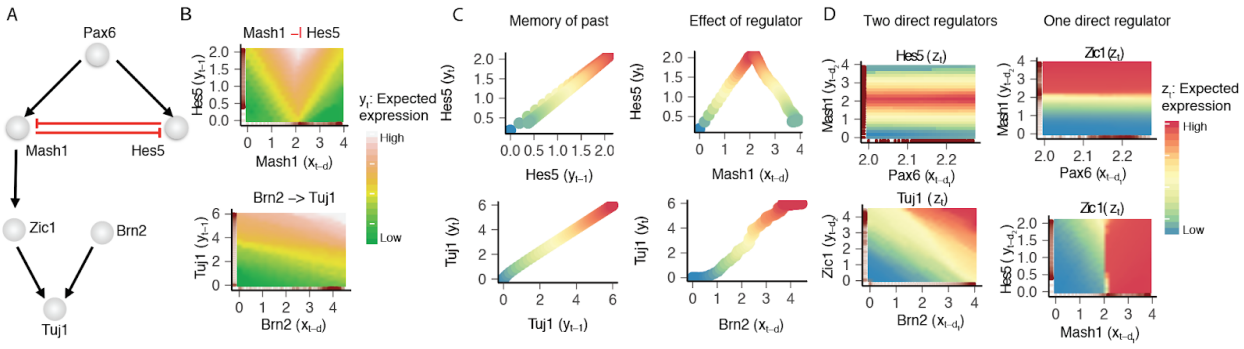


Figure 2: Visualize gene regulations with Scribe. (**A**) Examples of three-gene regulatory motifs in a minimal network describing neurogenesis. Top: *Pax6* activates both of *Mash1* and *Hes5* and there is a mutual inhibition between *Mash1* and *Hes5;* Bottom: *Zic1* and *Brn2* both activate *Tuj1*. The two network motifs are connected by an edge from Mash1 to Zic1. For the full regulatory network, see **Supplementary Figure 2**(Krishnaswamy et al. 2014; Qiu, Ding, and Shi 2012d). For panels **B-C**, all panels on the top correspond to the top motif in panel **A** and *vice versa*. (**B**) A *causality* visualization reveals the information transfer from one gene to another. The x-axis corresponds to the regulator's previous expression with a time lag d while the y-axis corresponds to the target gene's latest expression (immediate previous state). The heatmap corresponds to the expectation of the target gene's current expression given the target's latest expression and regulator expression with a time lag d or $\mathbb{E}[Y(t)|Y(t-1), X(t-d)]$. Each column indicates the response for the target gene's current state to its latest state (*memory of past*) given a particular value for the regulator while for each row, indicates the response of the regulator to its target given the latest state of the target itself (*effect from the regulator*). By

considering both dimensions of causality visualization, the information transferred from the regulator to the target given its history ($RDI_d(X \to Y)$ or equivalently $I[X(t-d); Y(t)|Y(t-1)]$) can be intuitively grasped. (**C**) Scatterplot describes relationships of gene regulations with time delay. (**D**) A *combinatorial regulation* visualization reveals the combinatorial gene regulation from two regulators to a target gene and direct or indirect/irrelevant regulator. The x-axis corresponds to the regulator's previous expression with a time lag d_1 while y-axis corresponds to another regulator's gene expression with another time lag d_2. The heatmap corresponds to the expected value of the target gene's current expression given both of the regulators' expressions with the time lags (d_1 or d_2) or $\mathbb{E}[Y(t)|X(t-d), X(t-d)]$. For the first column, it describes the relationship from two direct regulators to the target (top: *Pax6* activates and *Mash1* inhibits *Hes5*; bottom: both *Zic1* and *Brn2* activates *Tuj1*). The second column describes the relationship from one direct regulator and other indirect regulator or irrelevant gene to the target (top: *Mash1* directly while *Pax6* indirectly activates *Zic1*; bottom: *Mash1* directly activates while *Hes5* regulates *Zic1*). For this simulation, all the time delays d, d_1 or d_2 are set to be 1. Obvious color changes on either the causality or combinatorial logic visualization on either horizontal or vertical dimension implies potential strong information transfer.

4.4 Scribe recovers regulation direction from groups of putative regulators to their potential targets

In order to validate the performance of Scribe using single-cell genomic data, we first tested whether Scribe is able to recover previously characterized potential regulatory hierarchies from

putative regulators to targets in scRNA-seq data. We hypothesize that the sum of outgoing edges' causality *strength* (the score of causal interaction calculated from RDI) inferred by Scribe should be higher in regulators than that in their downstream targets. Previously, Treutlein and colleagues(Treutlein et al. 2014) categorized genes as either early or late AT1 or AT2 lineage-specific genes, some of which may represent putative regulator-target pairs, during lung epithelium differentiation. Analyzing the interactions of these early and late AT1/AT2 genes using Scribe, we find that the sum of the causality scores of the outgoing edges for early AT1/AT2 genes is considerably higher than that for late genes (**Fig 3A**). Similarly, applying Scribe to a scRNA-seq dataset of the response of dendritic cells to LPS stimulation(Shalek et al. 2014), we find that known regulators(Amit et al. 2009) involved in the LPS response have higher outgoing causality scores than that of their corresponding target genes (**Fig 3B**). Lastly, to validate Scribe's ability to identify regulators and targets, we used a network we recently built for myelopoiesis based on genes that significantly diverging between monocyte and granulocyte lineages, as well as enriched motif sequences in DNase hypersensitive sites (DHS) from 5-kb regions upstream or downstream of the open reading frame of those genes(Qiu, Mao, et al. 2017a). This network has three layers; the first layer contains the master regulators (*Gfi1* or *Irf8*), the second layer contains transcription factors targeted by the master regulators, and the third layer contains the downstream targets. Scribe correctly infers the ordering of sum RDI score for the three-tiered groups of genes (**Fig 3C**). In contrast, CCM can only recover the two-tiered hierarchy in the Treutlin dataset and part of the hierarchy in our myelopoiesis network, while GC fails to find the correct hierarchy for any of the three datasets (**Supplementary Figure 4**). These

results indicate that Scribe can correctly reveal the direction of the regulation in real datasets across various biological settings.

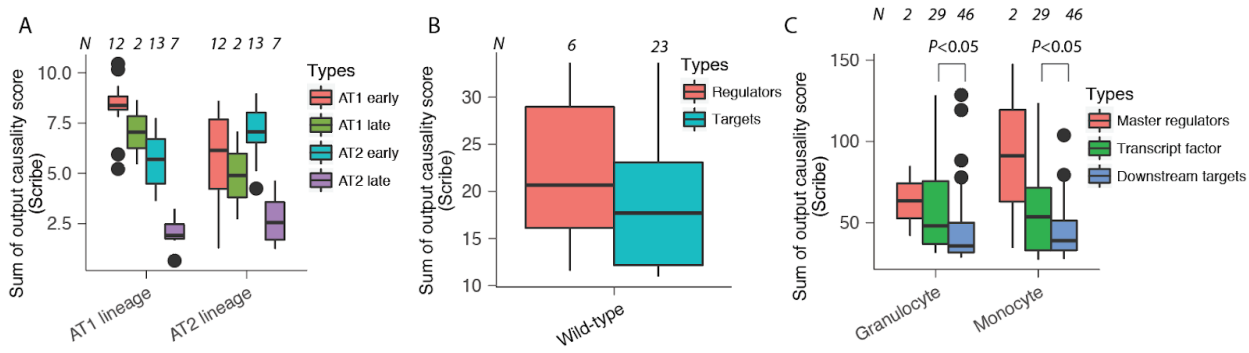


Fig 3: Scribe correctly reveals the ordering of the sum of outgoing RDI for a variety of single-cell RNA-seq datasets. In order to demonstrate Scribe's power in detecting the direction of causal regulation, we test the hypothesis that the sum of outgoing edges' causality score should be higher in groups of potential regulators than in their targets on three different datasets (lung(Treutlein et al. 2014), LPS(Shalek et al. 2014) and blood datasets(Olsson et al. 2016)). (**A**) Total outgoing causality scores of the putative regulators is higher compared to that of the target genes across AT1 or AT2 branch. (**B**) Same as in panel **A** but for the LPS data (only wild-type cells are chosen from this dataset to avoid testing on disrupted LPS response network in the knockout cells). (**C**) The master regulators have the highest total outgoing causality scores compared to the putative direct targets (transcription factors) and then the putative secondary targets (downstream targets). To obtain total outgoing causal scores, causal scores between all gene pairs are calculated with RDI and then processed by the CLR algorithm, followed by summing up all outgoing edges' scores for each gene. Integers (*N*) above each boxplot

corresponds to the number of genes used for creating the plot. An unpaired two-sample *t*-test is used to test each pair of hierarchical groups of genes. Only pairs of genes detected as significantly different ($p < 0.05$), which also include larger number of genes, are labelled.

4.5 Scribe recovers a core regulatory network responsible for myelopoiesis

We next explored whether Scribe can accurately reconstruct causal regulatory networks. Recently, Olsson and colleagues suggested a core network of transcription factors for regulating myelopoiesis(Olsson et al. 2016) by performing bulk ATAC-seq, ChIP-seq, perturbation experiments and profiling the transcriptomes of 382 cells from flow-sorted populations undergoing the transition (**Fig 4A**). We used Scribe to calculate causal scores for each regulator-target pair from the *Irf8* and *Gfi1* master regulators of the monocyte or granulocyte lineage, respectively, to the other six genes in the core network. We hypothesize that Scribe will return strong causal scores for the targets ascribed to each regulator but not others. We observed that expression kinetics over pseudotime correctly reflect the network architecture (**Fig 4A, B**). For example, as *Irf8* expression increases in the monocyte lineage, *Zeb2* expression also increases while expression of *Per3* decreases (**Fig 4B)**, reflecting the fact that *Irf8* activates expression of *Zeb2* while inhibiting *Per3*'s expression. We represent the causal network inferred by Scribe as a heatmap where each row corresponds to the causal score from the regulator to all other genes and the color corresponds to the magnitude of the causal score (**Fig 4C**). Scribe assigns a high causality score for all targets of *Irf8* (*Gfi1*, *Irf5*, *Klf4, Per3*, *Zeb2*) but lowest causality score to *Irf8* and *Ets1* which are not its direct targets. Similarly, Scribe assigns high causality score for the majority of *Gfi1*'s targets (*Irf8*, *Klf4, Per3*) even though *Gfi1* has low

expression values (**Fig 4C**). Visualization of the combinatorial regulation of *Irf8* and *Gfi1* to either *Zeb2* or *Per3*, based on the Scribe visualization toolkit, captures the conflicting regulation pattern between two regulators and their two targets (**Fig 4D**). Additionally, Scribe discovers evidence for temporal causal regulations in myelopoiesis (**Supplementary Figure 5**). Lastly, using the manually curated networks(Su et al. 2017) as well networks reconstructed based on ATAC-seq/ChIP-seq data(Su et al. 2017; Olsson et al. 2016) as a network gold-standard, we benchmark Scribe against other network inference algorithms and find that Scribe consistently outperforms other methods (**Supplementary Figure 6**).

To determine Scribe's capabilities to reconstruct transcriptome-level causal networks containing edges between transcription factors (TFs) as well as from TFs to putative downstream targets, we applied Scribe to the scRNA-seq data for hematopoiesis (Paul et al. 2015). We find that the lineage-specific genes tend to have high total outgoing RDI sum among all significant transcription factors (**Fig 4E**). When restricting to a small subset of previously identified erythropoiesis associated TFs, we find putative causal regulation of *Gata1* to *Zfpm1*, *Gfi1b*, *Tcf3* and *Klf1*, relationships supported by previous studies based on STRING database (**Fig 4F**, **Supplementary Figure 6**). These results highlight that Scribe's workflow for scRNA-seq data provides a useful strategy to identify regulations at either global or small scale, aiding in the understanding of global regulatory hierarchy or small core networks during complicated lineage commitment processes like hematopoiesis.

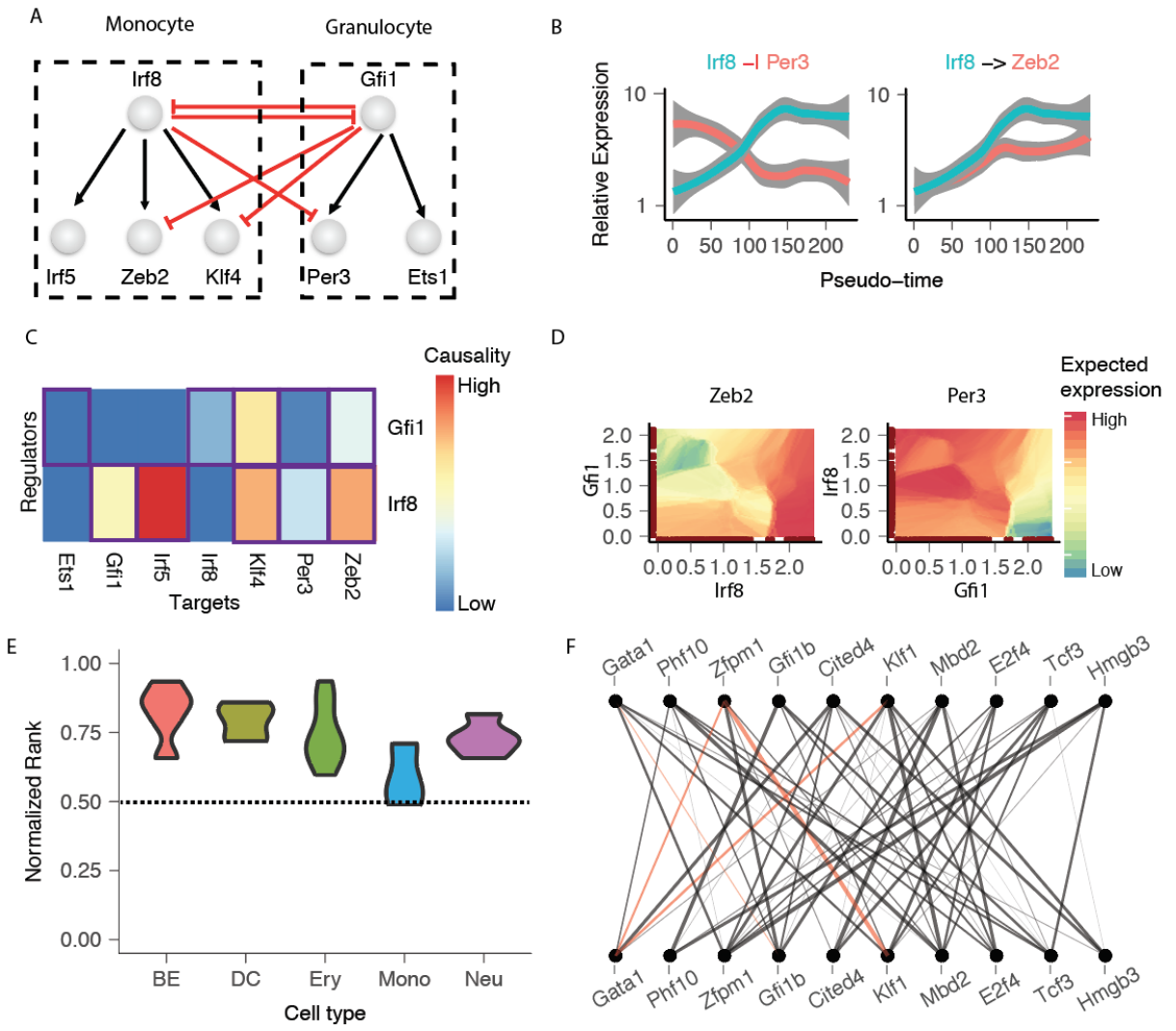


Fig 4: Scribe recovers a core regulatory network responsible for myelopoiesis. (**A**) A core network describes key regulators during the specification of monocytes and granulocytes based on data collected from perturbation experiments, bulk ATAC-seq and ChIP-seq data(Olsson et al. 2016). (**B**) Examples of gene-target pair kinetic curves over pseudotime along the monocyte lineage. See **Supplementary Figure 5** for other examples of the gene-target pair along different lineages. (**C**) Scribe infers the expected core regulatory network interactions for myelopoiesis. Causal scores from regulators to all other genes are calculated using RDI and are then

normalized using the CLR algorithm. (**D**) Visualization of combinatorial gene regulation from *Irf8* and *Gfi1* to *Zeb2* or *Per3*. Gene expression values are denoised through reversed graph embedding and calculated as a local average. Values are then rasterized to plot as a two-dimensional heatmap (See **methods** for details). (**E**) Normalized rank of lineage-specific genes' total outgoing RDI sum. Total outgoing RDI sum is calculated for all lineage-specific genes for each lineage as in **Figure 4**. The normalized rank is calculated based on the order of each lineage-specific TF among all significant branching TFs divided by its total number. When the normalized rank is close to 1, the corresponding gene is close to having the highest sum of outgoing RDI scores. The dashed line indicates the average rank (0.5) for a random gene. (**F**) Lineage-specific network of significant regulators during erythropoiesis. Edges supported by SPRING database is colored as red lines. For panels **E (F)**, BEAM analysis was used to identify significant branching genes associated with the four (one) lineage bifurcation events shown in the hematopoietic trajectory from ref. (Qiu, Mao, et al. 2017a) based on the paul dataset (Paul et al. 2015). The top 1,000 differentially expressed genes associated with each bifurcation were chosen to build a causal network for each relevant lineage. A set of TFs relevant to specific lineages described previously are used for panel **E** or **F**. Neu: Neutrophil; Ery: Erythroid, Mk: Megakaryocyte; Mono: Monocyte; DC: Dendritic Cell; BE: Basophil / Eosinophil.

4.6 Scribe incorporates RNA-velocity to aid causal network inference

Although pseudotime ordering is powerful, it has inherent limitations for performing network inference: *1) gene expression fluctuations of a regulator do not propagate to its target in pseudotime-series as each cell is distinct; 2) even if pseudotime ordering is accurate, the scale of*

pseudotime is arbitrary as it measures the aggregate transcriptomic differences and is thus incapable of accounting for different rates of transcription at different stages of cell fate transitions; 3) pseudotime ordering is suitable for datasets with a clear structure but challenging for those with complex dynamics, for example, processes involving both cell cycle and cell fate bifurcation. By incorporation of "RNA-velocity"(La Manno et al. 2017), Scribe alleviates these limitations and extends its use to complex manifolds(La Manno et al. 2017).

RNA-velocity offers the ability to perform causal inference via RDI based on two data points from the *same* cell (**Fig 5A**). Recently La Manno and colleagues applied RNA-velocity to capture the trend and the rate of chromaffin fate specification, as well as the associated cell cycle dynamics. We use this chromaffin dataset as a proof-of-principle for incorporating "RNA velocity" into Scribe. We first reconstructed a developmental trajectory with spliced mRNAs expression from each cell in this dataset and then applied BEAM to identify genes that significantly bifurcate between Schwann and chromaffin cell branches (**Fig 5B, C**). We find that these genes are enriched in processes related to neuron differentiation, adaptive immune, etc. along the path from SCPs (Schwann Cell Progenitors) to mature chromaffin cells (**Fig 5C**). To reveal the overall causal interactions between the top 1, 000 differentially expressed TFs and potential targets, we partitioned transcription factors and other informative genes into two or ten clusters, respectively, based on their expression dynamics along the chromaffin branch (**Fig 5C**). A causal network constructed from clusters of transcription factors to clusters of other genes reveal complex gene regulatory patterns (**Fig 5C, D**). Reconstructing causal networks of the top 1, 000 genes based on either pseudo-time-series or RNA-velocity, Scribe identifies considerable concordance between the pseudo-time and RNA-velocity network structures (**Fig SI7**). We also

built a core network between fourteen TFs responsible for chromaffin cell specification as suggested by the original study(Furlan et al. 2017). Only the RNA-velocity based core network recovers two feed-forward loop (FFL) motifs(Alon 2007) (*Eya1-Phox2a-Erbb3* and *Gata3-Phox2a-Notch1*) (**Fig 5E**). These motifs are partially confirmed from existing molecular interactions(Szklarczyk et al. 2017) retrieved from STRING database and await future experimental validations (**Supplementary Figure 7)**. From the RNA-velocity network, we also find that SCPs related TFs, such as *Sh3tc2*, tend to have a stronger causal regulation to other genes compared to that of chromaffin cell related TFs, including *Chga* and *Th*, reflecting the transition from SCPs to chromaffin cells (**Fig 5E**). We believe that the "RNA velocity" based causal network inference will help reveal mechanisms for fate specification of chromaffin cells, and it is highly generalizable for other cell types and processes.

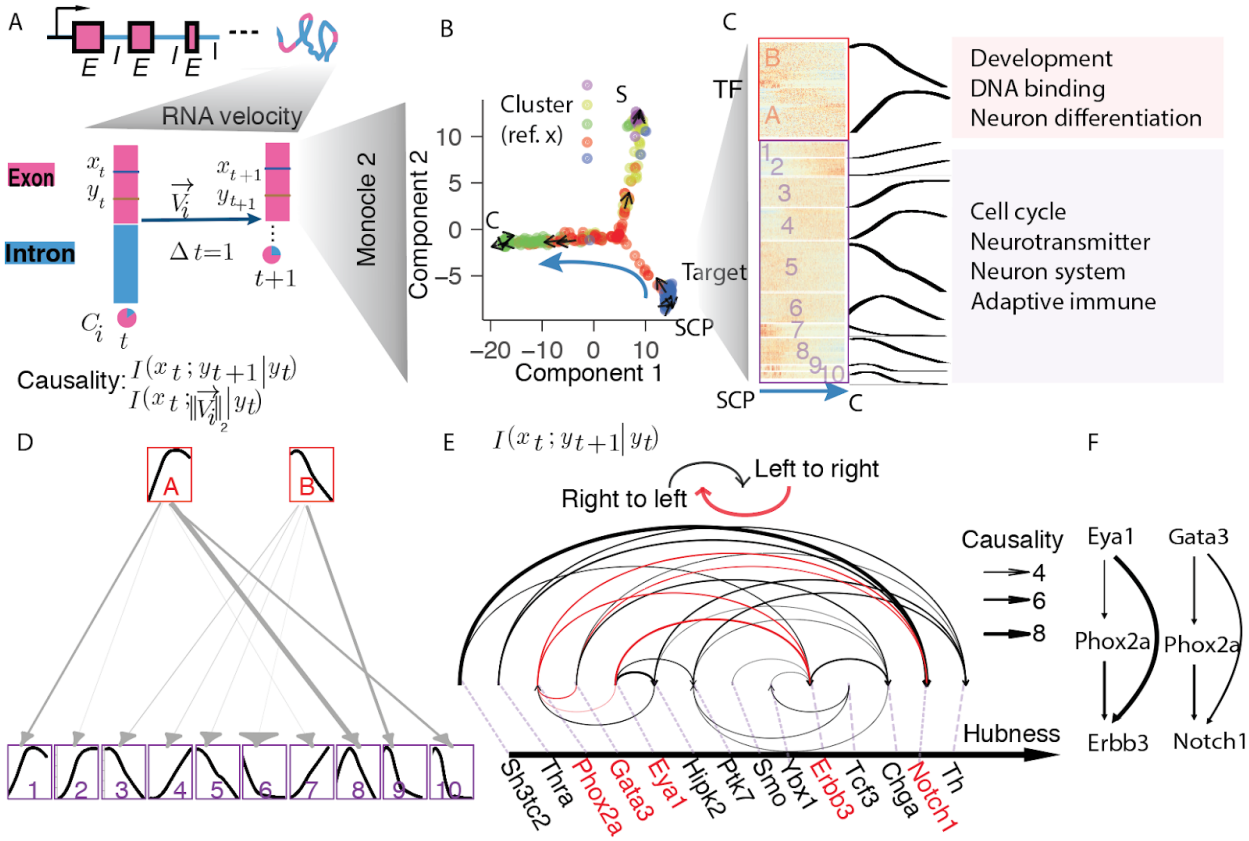


Fig 5: Scribe overcomes limitations from pseudotime-based causality inference with RNA-velocity. (**A**) Incorporating RNA velocity analysis into Scribe for causality inference. A gene with multiple exons (pink box, ***E***) and introns (blue line, ***I***) is transcribed into immature RNA and then spliced into mature RNA, both of which can be quantified by scRNA-seq. The RNA-velocity analysis framework estimates both exon and intron expression levels for each cell i or C_i. It then calculates the RNA-velocity for each gene $\vec{V_i}$ and predicts the future exon expression of $E^{predict}$ after $\Delta t = 1$. Assuming the time delays from all regulators to their putative targets are the same as Δt (or 1), Scribe calculates causality from the potential regulator to the target with the conditional mutual information between the current regulator's exon expression x_t to the predicted target exon expression y_{t+1} (or the estimated magnitude of RNA

velocity $||\overrightarrow{V_i}||_2$) conditioned on the current target exon expression y_t or by the default formula $I(x_t; y_{t+1}|y_t)$ (or alternatively $I(x_t; ||\overrightarrow{V_i}||_2|y_t)$). Since $x_t, y_{t+1}(||\overrightarrow{V_i}||_2), y_t$ are all estimated from the same cell, this approach overcomes several limitations from pseudotime-based causality inference (see **methods**). (**B**) RNA-velocity vector projected onto the first two latent dimensions. A small subset of arrows is used to visualize the velocity field of the cells. **S**: Sympathoblasts; **C**: Chromaffin. **SCP**: Schwann Cell Progenitor. The color of each cell corresponds to the cluster id from **Fig 5B** of ref. (Furlan et al. 2017). Only the exon values from RNA-velocity framework are used to reconstruct the developmental trajectory. (**C**) Clusters of TFs or potential targets from the chromaffin cell path (as indicated by the blue arrow in panel **B**) enriched in relevant biological pathways. Expression dynamics of significant branching genes between *C* and *S* fates are clustered (TF: 2 clusters; Targets: 10 clusters) to obtain cluster-specific average expression kinetics. The enriched pathways are based on gene ontology and Reactome pathway database. (**D**) The causal strength between clusters of TFs and clusters of target genes. Red boxes correspond to TF clusters (on the top) while purple boxes target clusters. The curve inside each box corresponds to the average expression kinetics in **B**. Edge width corresponds to estimated causality strength. (**E**) A core causal network for chromaffin cell commitment inferred based on RNA-velocity. Gene set is collected from ref. (Furlan et al. 2017). CLR (context likelihood relevance) is used to remove spurious causal edges (quantified with $I(x_t; y_{t+1}|y_t)$) in the network (see **methods**). Network is layouted as an arc-plot where all the genes are ordered on a line, sorted by the hub centrality score (see **methods**) decreasing from left to right. The edges above (below) the line indicate the interactions from the left (right) genes to the right (left). The width of the edge corresponds to the normalized causality score returned after applying CLR on

RDI values. Genes are labeled below the horizontal line. (**F**) Two potential coherent FFL (feed-forward loop) motifs of chromaffin differentiation are discovered from the core network.

4.7 Scribe dissects causal networks of *C. elegans*' early embryogenesis

In principle, Scribe is not limited to scRNA-seq data and is capable of inferring causality for various types of time-series data. Here we demonstrate its generality by inferring causal regulatory networks from live-specimen confocal imaging data of *Caenorhabditis elegans* embryos. This dataset consists of 265 time-series that track the expression dynamics of individual transcription factors using transcription factor fluorescent reporter constructs with the intensity of fluorescence measured at one minute intervals in every cell of the developing embryo for the first ~350 minutes of embryogenesis (**Fig 6A)**.

Known master regulators of intestine (*end-3/1*, *elt-2/7*), muscle (*hlh-1*, *hnd-1*, *unc-120*) and epidermis (*elt-1/3*, *lin26*, *nhr25*) cells as well as that for particular lineages, for example, *AB* or *MS* lineages follow the expected expression dynamics in this dataset(J. I. Murray et al. 2012) (**Fig 6B, Supplementary Figure 8,9**). Application of Scribe to this fluorescent reporter time-course correctly captures the known causal interactions among these master regulators(Owraghi et al. 2010) (**Fig 6C, Supplementary Figure 8C**). Focusing on one cell lineage, the intestine cell *Ealap*, we found that transcription factors can be clustered into distinctive temporal gene expression patterns (**Fig 6D**). We then used Scribe to perform an integrative analysis focused on inferring the causal regulatory networks that allow for maintenance of cell identity as well as those involved in initial cell fate commitment in the same manner as a recent work from Zhuo and colleagues(Du et al. 2014) which however relies on

extensive genetic perturbations (**Fig 6E, F**). Known regulators of *E* lineage, *end-1, end-3, ref-1, elt-2*, and *elt-7* sequentially appear in our cell maintenance network ordered by their known onset timing(J. I. Murray et al. 2012). Interestingly, this analysis also suggests a putative regulation of *ref-1* by *end-1* which is further causally regulated by *aly-1* (a factor thought to be related to mRNA binding and exportation(Fortes et al. 2007)). In general, we find that both the maintenance and the commitment networks become increasingly complex as cells divide along the lineage tree. As each cell or cell division is associated with a particular time period and space dimension, the causal networks recovered by Scribe recover both time and spatially-dependent gene regulation (**Fig 6G, Supplementary Figure 9D**). For example, an *E2F* like gene *elt-3* has strong putative causal interaction in both *Eal* or *Ear*'s commitment networks and gains additional causal interactions in the left axis of the worm (*Eal*) including a key neuronal TF *unc-86* and a previously identified posterior-anterior asymmetric gene *lin-39*, indicating potential novel regulation in the left-right axis of *E* lineage for the latter. Scribe also identifies lineage-specific causal networks and reveals that similar lineages (for example, *E* and *MS* lineages) share similar causal regulations (**Fig 6GF**). This analysis demonstrates the power of Scribe in building a compendium of the causal regulatory network for early *C. elegans* embryogenesis and further highlights the usefulness of Scribe for casual regulatory network inference from diverse time-series datasets (**Fig 6F, Supplementary Figure 9**).

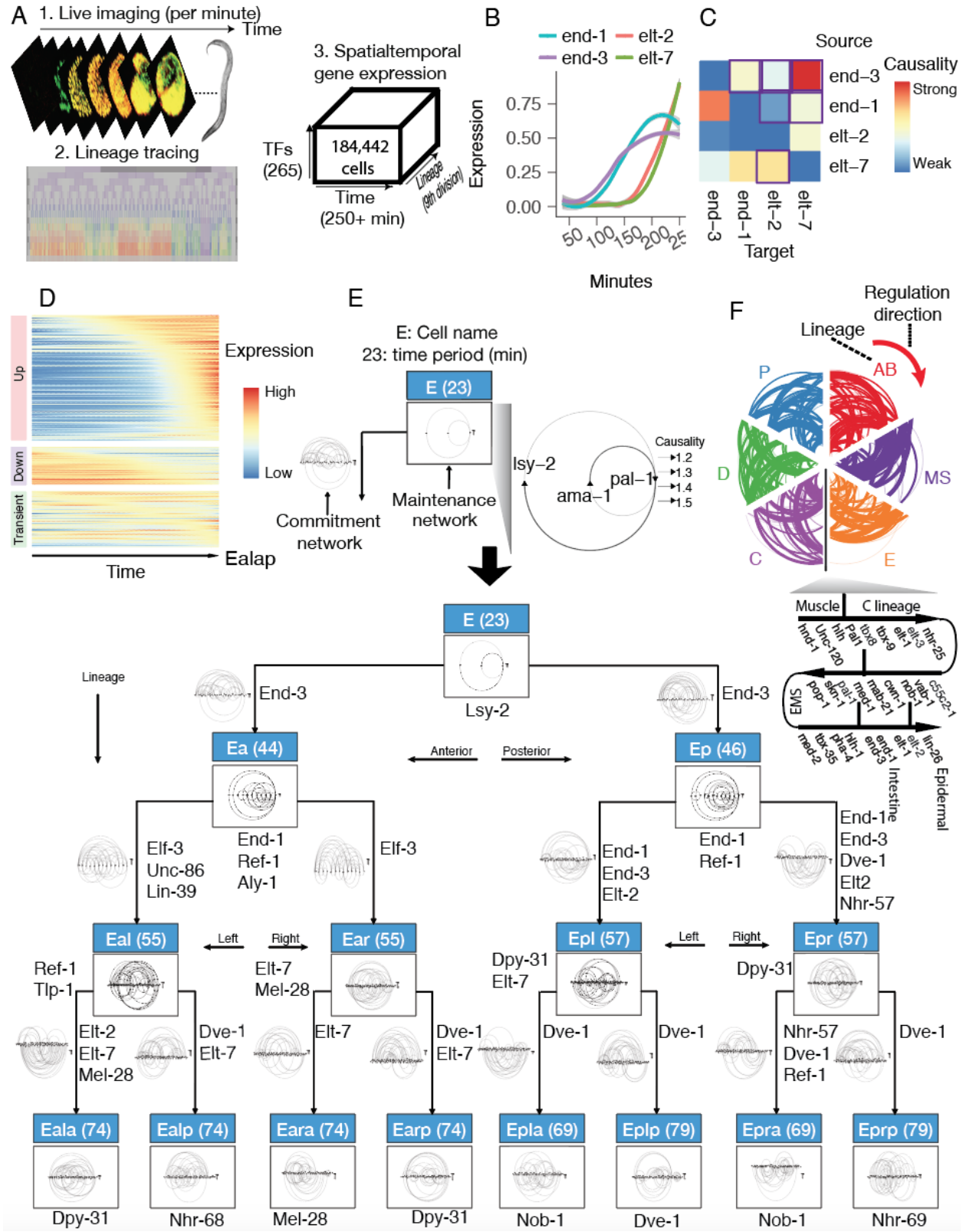

A
1. Live imaging (per minute)
Time
2. Lineage tracing
3. Spatialtemporal gene expression
TFs (265)
184,442 cells
Time (250+ min)
Lineage (9th division)
B
end-1
elt-2
end-3
elt-7
Expression
0.75
0.50
0.25
0.00
50
100
150
200
Minutes
C
Source
end–3
end–1
elt–2
elt–7
Causality
Strong
Weak
Target
D
Up
Down
Transient
Expression
High
Low
Time
Ealap
E
E: Cell name
23: time period (min)
E (23)
Commitment network
Maintenance network
lsy–2
ama–1
pal–1
Causality
1.2
1.3
1.4
1.5
F
Regulation direction
Lineage
P
AB
D
MS
C
E
E (23)
Lsy-2
End-3
End-3
Lineage
Ea (44)
Ep (46)
Anterior
Posterior
Elf-3
Unc-86
Lin-39
End-1
Ref-1
Aly-1
Elf-3
End-1
End-3
Elt-2
End-1
Ref-1
End-1
End-3
Dve-1
Elt2
Nhr-57
Eal (55)
Left
Right
Ear (55)
Epl (57)
Left
Right
Epr (57)
Ref-1
Tlp-1
Elt-7
Mel-28
Dpy-31
Elt-7
Dpy-31
Elt-2
Elt-7
Mel-28
Dve-1
Elt-7
Elt-7
Dve-1
Elt-7
Dve-1
Dve-1
Nhr-57
Dve-1
Ref-1
Dve-1
Eala (74)
Ealp (74)
Eara (74)
Earp (74)
Epla (69)
Eplp (79)
Epra (69)
Eprp (79)
Dpy-31
Nhr-68
Mel-28
Dpy-31
Nob-1
Dve-1
Nob-1
Nhr-69
Muscle
C lineage
hnd-1
Unc-120
hlh
Pal1
tbx-8
tbx-9
elt-1
elt-3
nhr-25
EMS
pop-1
skn-1
pal-1
med-1
mab-21
cwn-1
nob-1
vab-1
c55c2-1
med-2
tbx-35
pha-4
hlh-1
end-3
end-1
elt-1
elt-2
lin-26
Intestine
Epidermal

Fig 6: Reconstructing causal regulatory network of *C. elegans*' early embryogenesis with Scribe. (**A**) Procedure for measuring TFs (transcription factors) protein expression dynamics in real-time for every cell during early *C. elegans* embryogenesis. (1) Protein-RFP fusions reporters are used to measure transcription factor protein expression levels with 3D live imaging every minute in each cell while a ubiquitous histone-GFP marker is used to trace the *C. elegans* cell lineage. (2) Reporter fluorescence data was then mapped based on methods from Murray and colleagues(John Isaac Murray et al. 2008) onto the invariant cell lineage. By combining expression measures in each corresponding cell from each embryo, we obtain a 3D tensor with dimensions of 265 genes X 550 time-points X 1365 (in total more than 180, 000 data points, after removing those with invalid measurements). Note that, for measuring transcription factor protein expression level, there are two types of reporters: a transcription factor gene directly tagged with a fluorescent protein (protein-fusion) or a fluorescently-tagged histone protein driven by the promoter of that transcription factor(J. I. Murray et al. 2012). (**B**) single-cell lineage-resolved fluorescence data capture temporal dynamics of *E* lineage master regulators during *C. elegans* embryogenesis. The expression for each gene is scaled to be between 0 and 1 and then smoothed using LOESS regression. (**C**) Scribe reconstructs the causal regulatory network for the four master regulators (*end-1/3*, *elt-2/7*). (**D**) Expression dynamics for 265 report TFs along the lineage to becoming the *Ealap* cell. The entire developmental lineage from the first *E* cell all the way to the *Ealap* cell in each embryo for each TF reporter is used to make the heatmap. The raw fluorescence intensity is scaled to be between 0 and 1 and then smoothed using LOESS regression. The order of genes in each row is calculated as previously described(Pliner et al. 2017). See **Supplementary Figure 8,9** for similar plots to visualize

master regulators' dynamics as well as all gene expression dynamics along a particular single-cell lineage commitment in muscle, epidermis and neuron cell types. (**E**) An integrative multiscale model for the *E* lineage specification. Above the thick arrow: scheme for the multi-scale network. The *C. elegans* cell name is denoted within the blue box with the number in the parenthesis pertaining to the time period for that cell in minutes. The network within the box corresponds to the causal network maintained during the lifetime of that cell (denoted as **maintenance network**). The network on the left of the arrows corresponds to the causal network relevant to the cell fate commitment from the progenitor cell dividing into the daughter cells (denoted as **commitment network**). A zoomed-in version of the cell fate commitment network for E cell is shown on the right of the box. For constructing the maintenance network, we first identify gene pairs with mutual information larger than a threshold 1 based on raw fluorescence values in each cell. Then we calculate the causality score for those pairs. For constructing the commitment network, we first identify genes with the expression significantly different between the two daughter cells and use all those genes to calculate the pairwise causality score as well as filter gene pairs with mutual information smaller than a threshold 1. We then apply the CLR algorithm(Faith et al. 2007) to remove spurious causal interactions. The same arc plot visualization from **Fig 5** is applied here. To avoid overplotting, only the top 50 edges are visualized. (**F**) Lineage-specific (*AB*, *P*, *MS*, *E*, *D*, *C*) causal networks in hive-plot for the manually curated master regulators constructed with Scribe. Regulatory interactions (i.e., from the regulator to the regulated) are shown by the edges in clockwise orientation. The width of the edge corresponds to the causality score from the regulator to the target estimated from Scribe.

Similarly, only up to top 50 edges with strongest causality score are visualized. All transcription factors are arranged in the same order along each axis, as shown below the hive-plot.

4. 8 Discussion

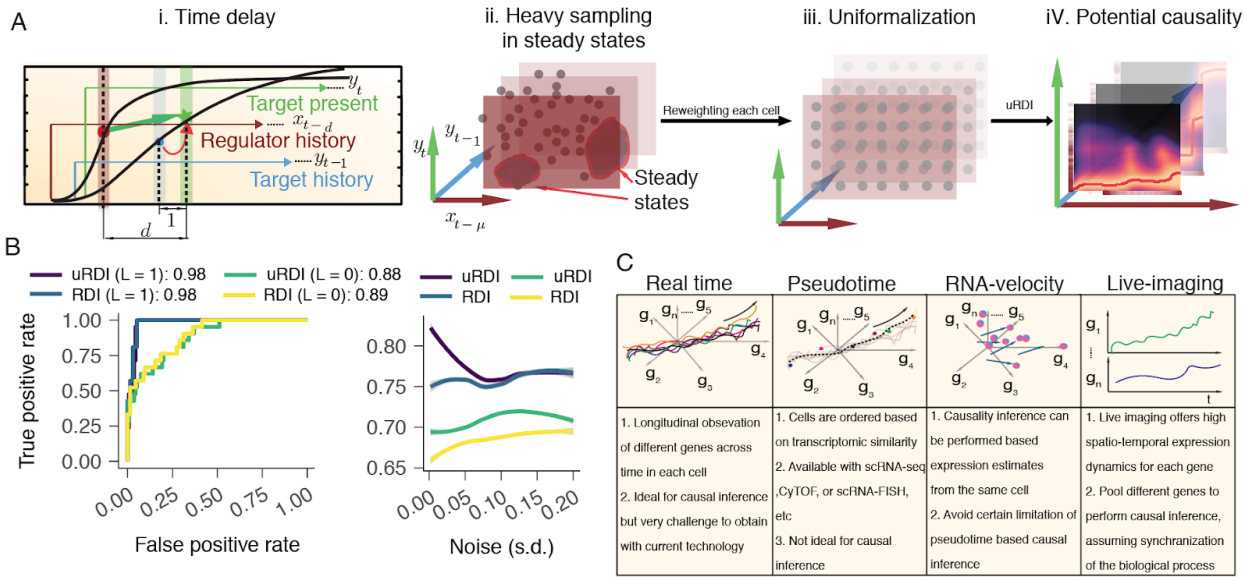
scRNA-seq affords an unprecedented high-resolution view of cellular dynamics which has sparked an explosion of network inference algorithm development. Methods based on mutual information and pseudotime ordering have been reported(Huynh-Thu et al. 2010; Ocone et al. 2015; Chan, Stumpf, and Babtie 2017; Matsumoto et al. 2017; Hamey et al. 2017; Wei et al. 2017; Sanchez-Castillo et al. 2017; Papili Gao et al. 2017); however, most of those methods only report statistical dependence(Huynh-Thu et al. 2010; Chan, Stumpf, and Babtie 2017; Papili Gao et al. 2017; Sanchez-Castillo et al. 2017), have poor performance(Matsumoto et al. 2017) or are difficult to scale up(Ocone et al. 2015; Wei et al. 2017; Hamey et al. 2017). More importantly, all pseudotime based network inference methods neglect several limiting facts.

Scribe builds upon a few advances in information theory to specifically dissect complex *casual* regulations from single-cell measurements at scale. Firstly, Scribe employs Restricted Direct Information (RDI) overcoming limitations inherent to Granger Causality (GC) and Convergent Cross-Mapping (CCM). Through extensive evaluation, we demonstrate that Scribe performs favorably compared to GC and CCM across synthetic and real scRNA-seq data benchmarks. Furthermore, Scribe provides a rigorous technique, uniformized RDI (uRDI) as an option to quantify potential influence, greatly aiding in the unfolding of causal regulations happen at rare transition states. Scribe also provides a framework to intuitively visualize causal information transfer and combinatorial regulation. Finally, Scribe explicitly considers the

limitation of pseudotime based network inference and leverages “RNA-velocity” to directly infer causal regulations through estimating information transfer from the regulator to target in the same cell. We apply Scribe to a variety of scRNA-seq datasets and find it accurately resolves putative gene regulatory hierarchies including lung epithelium cell bifurcation and dendritic cell response to LPS stimulation. Scribe correctly infers key regulatory interactions of myelopoiesis and reveals lineage-specific regulatory network in hematopoiesis. By taking advantage of RNA-velocity, Scribe identified two potential feed-forward loops associated with the commitment of chromaffin cells. Finally, Scribe provides an avenue to reconstruct high-resolution causal network for every cell division during *C elegans* early embryogenesis with lineage-resolved live-imaging data.

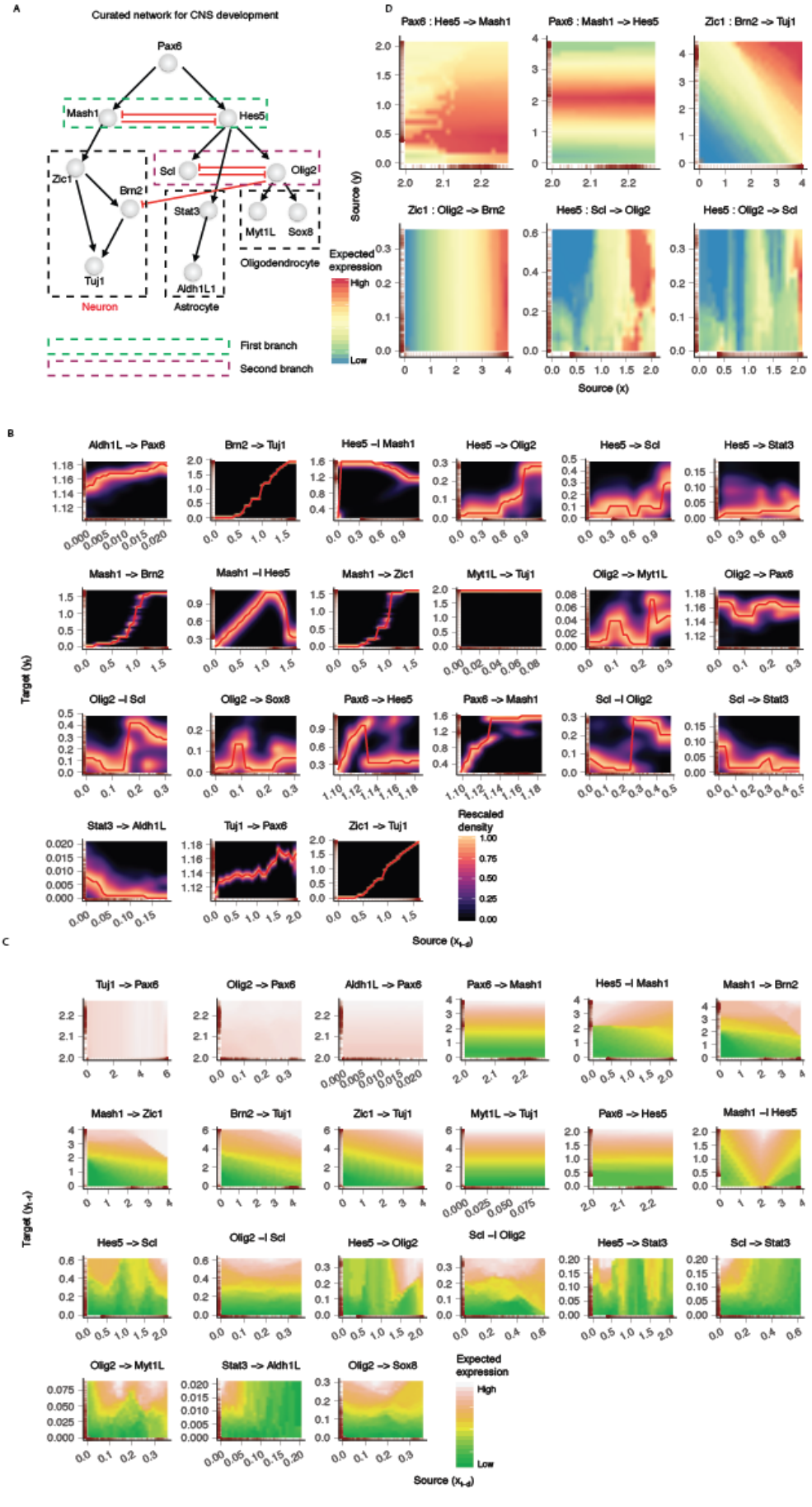
The increase in scale in scRNA-seq experiments has launched several initiatives for building a comprehensive cell atlas of various organisms, including *C. elegans*, mouse and human(Cao et al. 2017a). By integrating these large datasets with novel single-cell measurements, lineage tracing, and live imaging, we anticipate that Scribe will provide a foundation to identify the causal networks that govern cell type specification.

Supplementary Figure Legends:

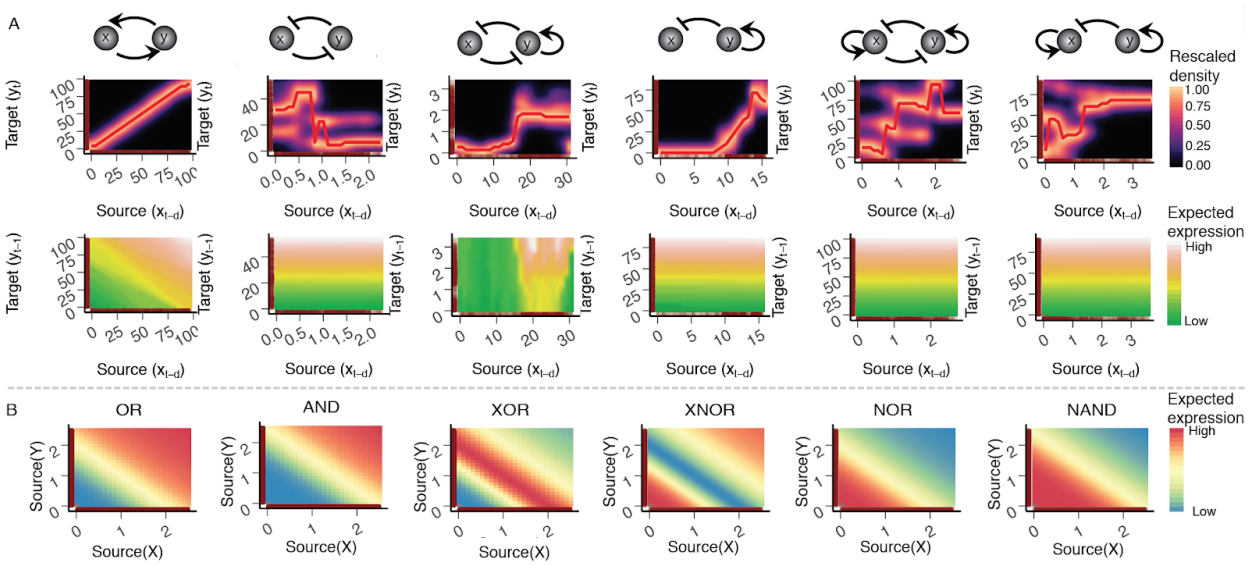


Supplementary Figure 1: Scribe recovers causal interactions at rare transition states and generalizes to various types of scRNA-seq datasets. (**A**). Scribe leverages a rigorous technique of uniformalization to detect potential causality. Cells often reside in the steady states but rarely in the transition state. This leads to a heavier sampling of single-cell measurements in steady states (for example, the low or high expression regions at the beginning or end of the kinetic curves as shown in panel A) than transition state. More intuitively, the biased sampling in data can be seen from the state space formed with the expression values of the regulator, target and target's history (*II. Heavy sampling in steady states*). In order to account for sampling biases from single-cell measures, Scribe integrates uRDI (uniformization of RDI) by reweighting each cell and thus replacing the biased sampling distribution with a uniform distribution (*II. Uniformalization*) to rigorously quantify the *potential causality(Arman Rahimzamani and*

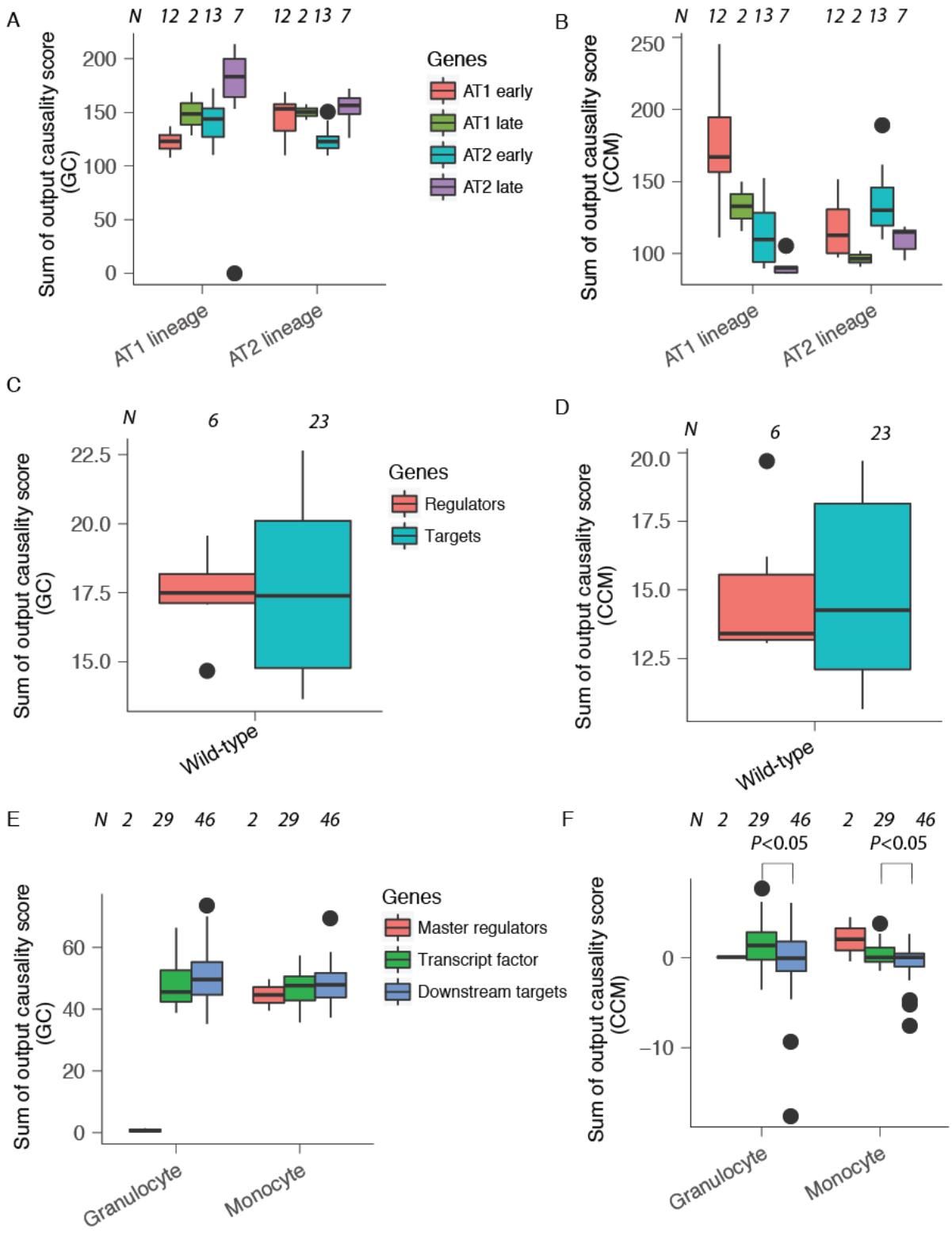
Kannan 2017) (how much influence a regulator can potentially exert on target *without cognizance* to the regulator's distribution). For a strong causal interaction, RDI requires the response of the target (see more details in **Fig 2**) to the regulator evolves under different historical states of the target (*III. Potential causality)*. (**B**). uRDI improves the recovery of causal regulations. **Left**: Receiver Operating Curves or ROC for different methods in Scribe on causal inference with or without uniformalization for the linear system. **Right**: Area Under Curve or AUC for different methods in Scribe on causal inference with or without uniformalization for the neurogenesis system. Noise for each system is treated as the same in the **Main Figure 1C**. (**C**). Scribe is applicable on any general time-series datasets. Each column in the table corresponds to different types of time-series data. The first column corresponds to real time-series datasets, where, for example, the transcriptome for each cell is followed over time longitudinally. The second column corresponds to the pseudotime-series datasets, where the transcriptome for a population of cells at different developmental stages is captured with scRNA-seq. Using computational algorithms, for example, Monocle 2, cells are ordered to obtain pseudotime-series data. The third column corresponds to the datasets estimated from the "RNA velocity" analysis framework where the current or future mature mRNA expression, etc are estimated for each cell. For the first three cartoons, each axis corresponds to one gene dimension where each curve corresponds to the expression dynamics for each individual cells in the full gene space over time. The arrow points to the direction of cell differentiation and the dashed line from the second figure corresponds to the inferred pseudotime trajectory. The text below provides more information for each scenario in the context of causal network inference.


A
Curated network for CNS development
Pax6
Mash1
Hes5
Zic1
Scl
Olig2
Brn2
Stat3
Myt1L
Sox8
Tuj1
Aldh1L1
Neuron
Astrocyte
Oligodendrocyte
First branch
Second branch
D
Pax6 : Hes5 -> Mash1
Pax6 : Mash1 -> Hes5
Zic1 : Brn2 -> Tuj1
Zic1 : Olig2 -> Brn2
Hes5 : Scl -> Olig2
Hes5 : Olig2 -> Scl
Source (y)
Source (x)
Expected expression
High
Low
B
Aldh1L -> Pax6
Brn2 -> Tuj1
Hes5 -| Mash1
Hes5 -> Olig2
Hes5 -> Scl
Hes5 -> Stat3
Mash1 -> Brn2
Mash1 -| Hes5
Mash1 -> Zic1
Myt1L -> Tuj1
Olig2 -> Myt1L
Olig2 -> Pax6
Olig2 -| Scl
Olig2 -> Sox8
Pax6 -> Hes5
Pax6 -> Mash1
Scl -| Olig2
Scl -> Stat3
Stat3 -> Aldh1L
Tuj1 -> Pax6
Zic1 -> Tuj1
Target (yt)
Rescaled density
Source (xt-g)
C
Tuj1 -> Pax6
Olig2 -> Pax6
Aldh1L -> Pax6
Pax6 -> Mash1
Hes5 -| Mash1
Mash1 -> Brn2
Mash1 -> Zic1
Brn2 -> Tuj1
Zic1 -> Tuj1
Myt1L -> Tuj1
Pax6 -> Hes5
Mash1 -| Hes5
Hes5 -> Scl
Olig2 -| Scl
Hes5 -> Olig2
Scl -| Olig2
Hes5 -> Stat3
Scl -> Stat3
Olig2 -> Myt1L
Stat3 -> Aldh1L
Olig2 -> Sox8
Target (yt+g)
Expected expression
High
Low
Source (xt-g)

Supplementary Figure 2: A gallery of regulatory patterns visualized by the response, causality and combinatorial regulation visualizations from Scribe on the simulated neurogenesis dataset. (**A**) The manually curated network used to simulate the three-way cell fate specification of the central nervous system(Qiu, Ding, and Shi 2012d). The network consists of two key mutual-inhibition gene pairs. Initializing this network with small amounts of stochastic noise and following expression kinetics over time simulates the trajectory followed by a single cell leading to the fate of either neuron, astrocyte or oligodendrocyte. For simplicity, only a simulation leading to the neuron fate is used for the analysis presented in panels **B-D**. (**B**) Response visualization plots for all the genes pairs in the network. *Response* visualization reveals the regulatory response of the target to the regulator. The x-axis corresponds to the regulator's previous expression with a time lag d (x_{t-d}) while y-axis corresponds to target's current expression (y_t). Here, theresponse of *Brn2* to *Tuj1* is a sigmoid function suggesting positive regulation while the response of *Mash1* to *Hes5* is a threshold function suggesting threshold mutual repression. The heatmap corresponds to the rescaled normalized conditional density for two genes ($P(y_t|x_{t-d})$), similar to the DREVI plot from ref.(Krishnaswamy et al. 2014). The red line represents the most probable value for the target given a regulator's expression. The rug plot on the axis corresponds to the density of cells at a particular value. –(**C**) Causality visualization plots for all genes pairs in the network. (**D**) Combinatorial logic visualization plots for all six two-input combinatorial regulations cases in the network.

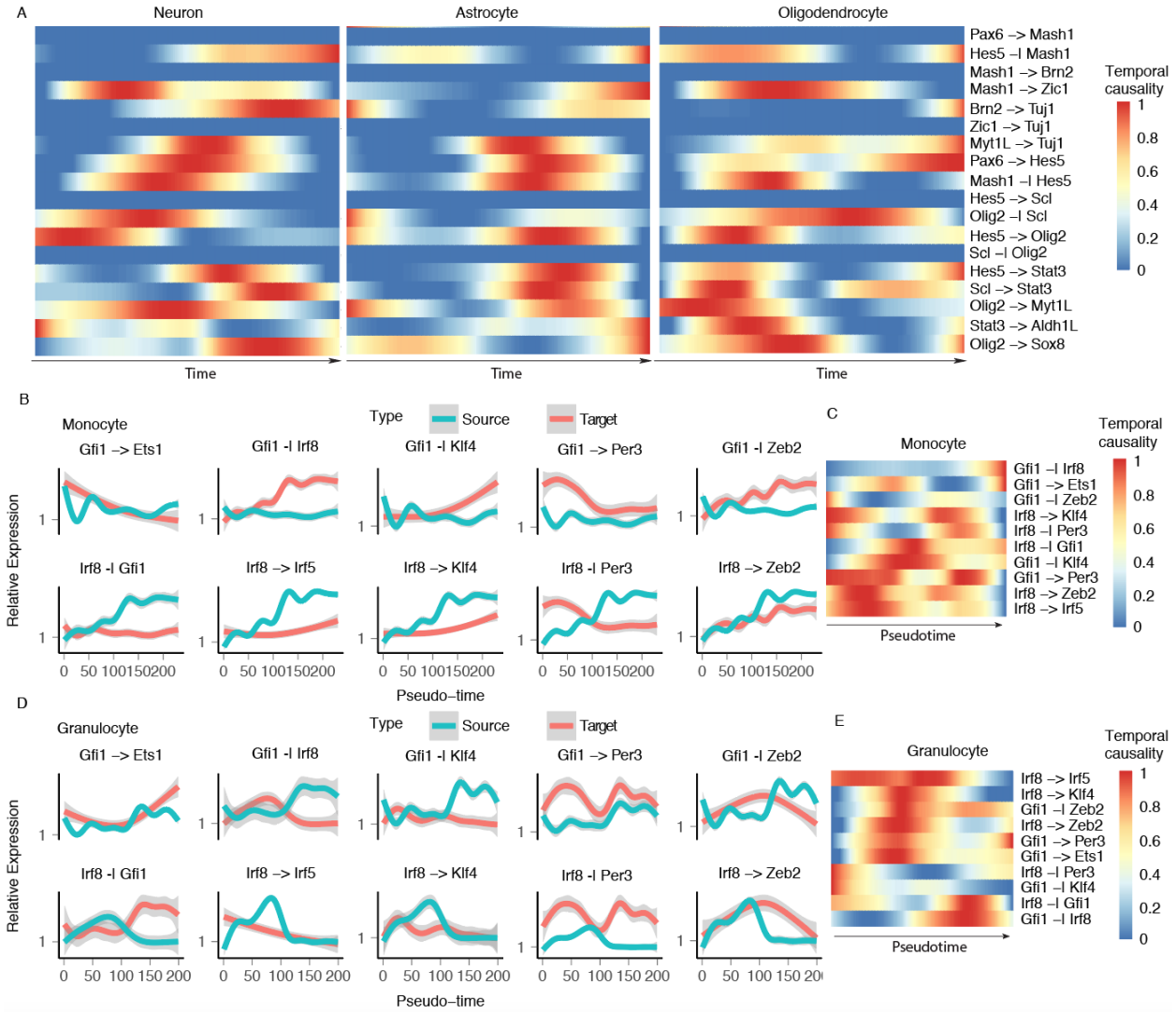


Supplementary Figure 3: Visualizing pairwise interactions from robust two-gene network motifs and combinatorial regulations from common two-input logic gates with Scribe. (A) Visualizing response and information transfer (causality) for robust two-gene network motifs. Top: robust network motifs from(Ma et al. 2009); middle: corresponding response visualization plots; bottom: corresponding causality visualization plots. The first node in the motif plot corresponds to the source (x-axis) while the second the target (y-axis). (**B**) Visualizing combinatorial regulations for six two-input logic gates.



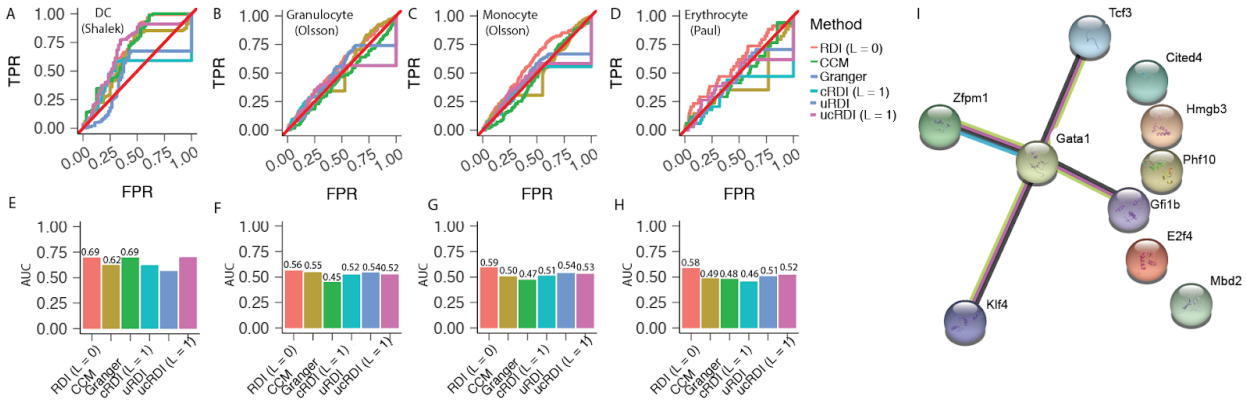
Supplementary Figure 4: Poor performance of other causality inference algorithms in

resolving gene regulatory hierarchy. To test whether or not other well-known causality detection algorithms (Granger Causality (GC) and Convergent Cross-Mapping (CCM)) can also infer the correct regulatory hierarchy, we calculated the total outgoing causality scores inferred from them for the known regulators and targets on the same datasets as used in main text **Fig 3**. (**A, B**) Distribution of total outgoing causality scores compared to that of the target genes across AT1 or AT2 branch based on GC (**A, left**) or CCM (**B, right**). (**C, D**) the same as in (**A, B)** but for the LPS data (wild-type cell subset). (**E, F**) The same as in (**A**, **B**) but across granulocyte and monocyte branches of the Olsson dataset (wild-type cell subset). To calculate total causality score, causal strength between all the genes are calculated with RDI which is then processed with CLR algorithm.



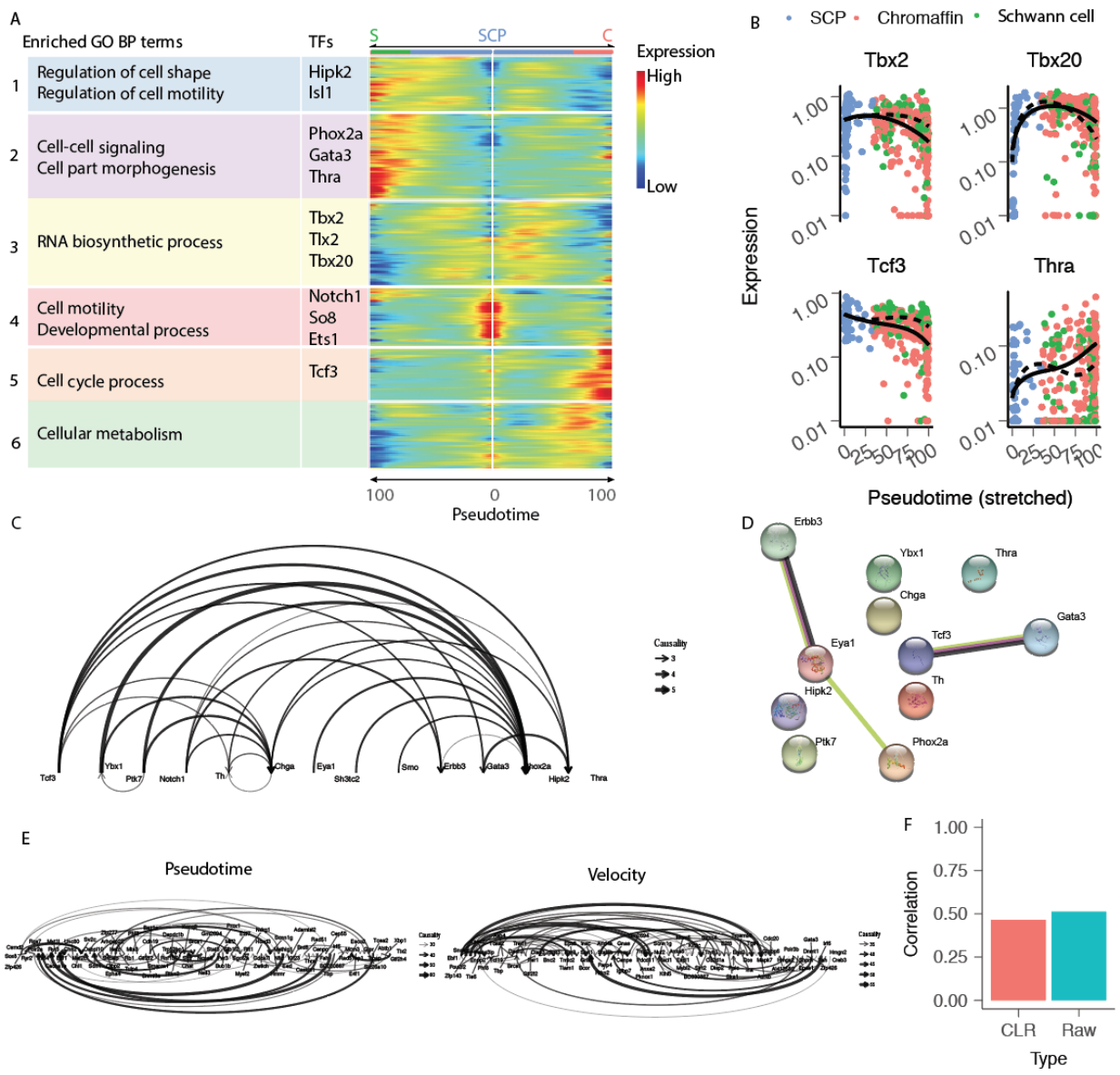
Supplementary Figure 5: Scribe detects temporal causal regulations. (**A**) Temporal causal strength from all gene-pairs (See **Fig SI1a**) in the simulation system across neuron, astrocyte and oligodendrocyte lineages. (**B, D**). Kinetic curves for all the gene pairs from the core network (See **Fig 4a**) during myelopoiesis across the monocyte or granulocyte lineage. (**C, E**) Temporal causal strength between all gene pairs from the core network of myelopoiesis across monocyte or granulocyte lineage. The temporal causal score is calculated through a moving window with

window size as 50. For creating heatmaps in panels **A, C, E**, raw causal scores are smoothed with LOWESS function and then scaled to be between 0 and 1.



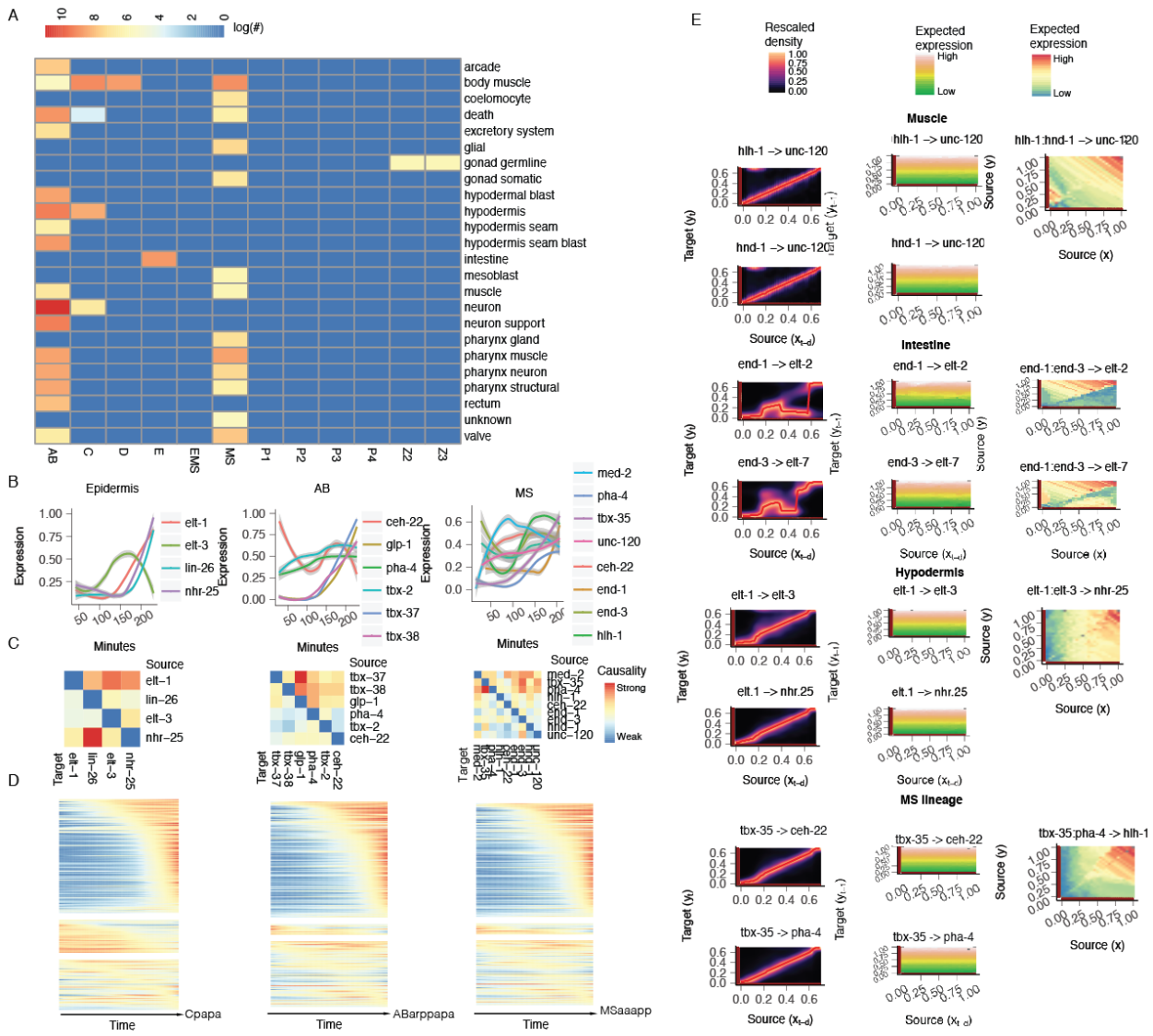
Supplementary Figure 6: Benchmark Scribe with other algorithms. (**A, E**) Receiver Operating Curves or ROC (**A, top**) and Area Under Curve or AUC (**E, bottom**) of the inferred causal network based on Scribe, GC, and CCM on the Dendritic Cells (DC) dataset. Four different variants of causality inference implemented in Scribe are tested: *RDI (L = 0)*: the default RDI method without conditioning on any other gene; *RDI (L = 1)*: the RDI method based on conditioning on the incoming gene with highest causality score, except the current target; *uRDI*: the method based on the uniformization technique applied on the actual distribution in RDI; *uRDI (L = 1)*: the uRDI method but also with the conditioning on the incoming gene with the highest causality score, except the current target. The network from(Amit et al. 2009) based on XXX is used as benchmark gold-standard. (**B, F**) The same as in (**A, E**) but for the granulocyte branch of the Olsson dataset. (**C, G**) The same as in (**A, E**) but for the monocyte branch of the Olsson dataset. The manually curated network for the myeloid differentiation from(Su et al. 2017) is used as the benchmark gold-standard. (**D, H**) The same as in (**A, E**) but

for the erythroid branch of the Paul dataset. All wild-type cells are pooled to reconstruct the developmental trajectory and a subset of CMP and erythroid branch cells are used to estimate the causal network for the erythroid branch. The manually curated network for the erythroid differentiation from Ref. 48 is used as benchmark gold-standard. (I) The network of the gene-set as included in the panel (Fig 4F) retrieved from the STRING database. See https://string-db.org/cgi/network.pl?taskId=2OGoh9uvYIdY for more details.



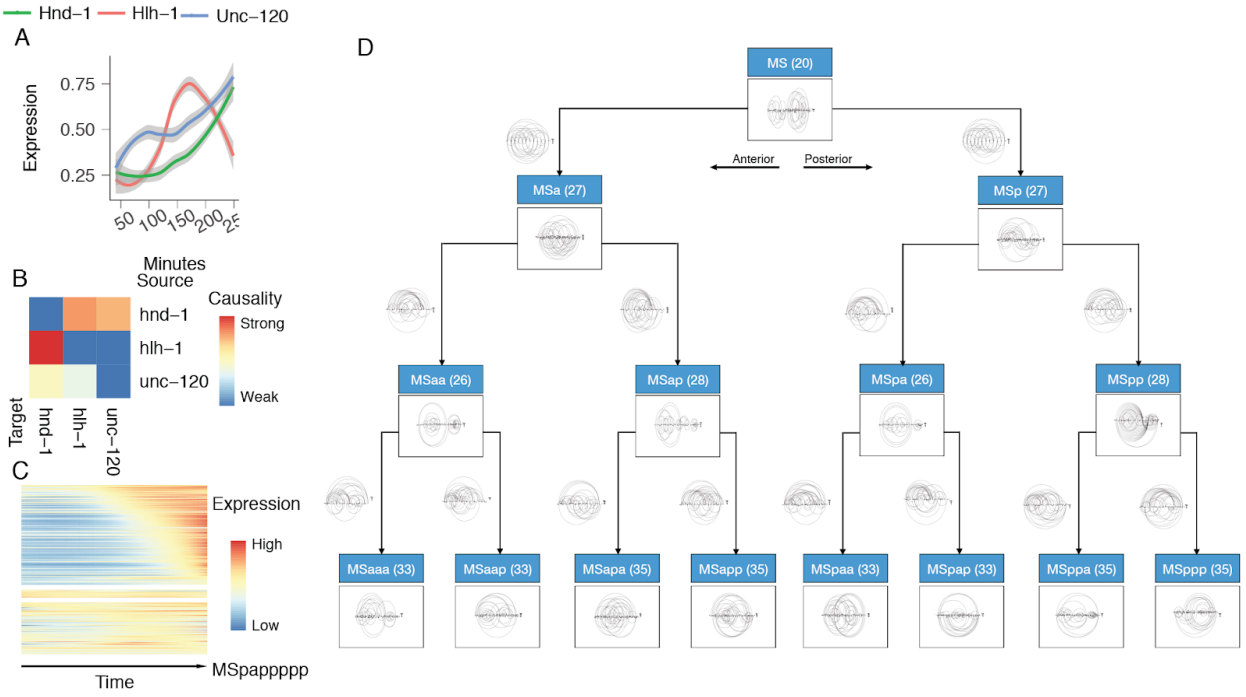
Supplementary Figure 7: Comparing causality inference based on pseudotime or RNA-velocity. (**A**) Clusters of significant branching genes reveal distinct transcriptional programs between Schwann and chromaffin cell lineages. (**B**) Gene expression dynamics of representative significant branching transcription factors from (**A**). (**C**) Some figure as in Fig 5E but inferred based on pseudotime-series data. (**D**) The network of the gene-set as included in the panel (**C**) retrieved from the STRING database. (**E**) Top 100 causal edges between significant

branching transcription factors (TFs) as well as from TFs to the potential targets in chromaffin lineage inferred with pseudotime (top) or RNA-velocity (bottom). (**F**) Spearman correlation of causality strength of pseudotime network with that of "RNA-velocity" network (either with raw causality or normalized CLR scores).



Supplementary Figure 8: Expression dynamics of and causal interactions between master regulators in different cell types or lineages of *C. elegans* embryogenesis. (**A**) The number of cells corresponding to a particular cell type (rows) which all come from the same founder cell (columns). For example, all cells produced by the *E* founder cell are intestine cells. (**B**) Single-cell lineage tracing fluorescence data captures temporal dynamics of the epidermis, AB or MS lineage master regulators during *C. elegans* embryogenesis. (**C**) Scribe reconstructs the

causal regulatory network for the corresponding master regulators shown in **B**. (**D**) Expression dynamics for 265 report TFs along the lineage to become cell *Cpapa*, *ABarppapa* or *MSaaapp*, each coming from the specified cell type or the lineage shown in **B**. (**E**) Scribe visualizes the response functions (first column in each section), causal interactions (second column) and combinatorial regulations (third column) for master regulators belonging to either muscle, intestine, hypodermis cell types or MS lineage.



Supplementary Figure 9: Inferring causal network for MS lineage of *C. elegans* embryogenesis. (**A**) Single-cell lineage tracing fluorescence data captures temporal dynamics of MS (muscle) lineage master regulators during *C. elegans* embryogenesis. (**B**) Scribe reconstructs the causal regulatory network for the three master regulators (*hnd-1*, *hlh-1*, and *unc-120*). (**C**)

Expression dynamics for 265 report TFs along the lineage to become the MSpapaapap cell. (**D**) An integrative multiscale model for the MS lineage specification.

Method

The problem of causal regulatory network inference

In this work, we refer to causality in terms of information transferred from one variable (a potential regulator) to another time-delayed response variable (a potential target). Moreover, in the context of single-cell genomics (e.g. scRNA-seq, live cell imaging), we define the problem of causal regulatory network inference for a dynamic biological process where the associated genes expression dynamics is profiled as: how can we reconstruct a regulatory network consisting of causal regulations that accurately describe the gene expression dynamics and the associated cell fate transitions?

Causality Inference

Various techniques, including Granger Causality and CCM, each associated with different assumptions has been proposed to detect causality. In the following, we briefly summarize these methods and introduce RDI, the method we developed and used in this study.

Granger causality

In order to determine whether one time series (X_1) is useful in forecasting another (X_2) in economics, Clive Granger first proposed Granger Causality (GC) in 1969 (Su et al. 2017; Granger 1969). According to GC, if X_1 "Granger causes" X_2, then the predictability of X_2 based on past values of X_2 alone is significantly weaker than that of also based on the past

values of X_1. GC in its original formulation, however, is only able to detect linear causality. Furthermore, GC assumes that the information of the causative factor is separable from the effects and the system is treated as pieces at a time rather than as a whole.

Convergent Cross-Mapping

In order to detect pairwise non-linear interactions in deterministic ecology systems, George Sugihara proposed Convergent Cross-Mapping (CCM) which is based on state-space reconstruction (Sugihara et al. 2012). CCM avoids GC's assumption of separability. One fundamental and somewhat counterintuitive idea of CCM, distinct from GC, is that it is possible to estimate X_1 from X_2, but not the other way if causation is from X_1 to X_2. CCM first constructs shadow manifolds M_{X_2} and M_{X_1} from lagged coordinates of the time-series X_2 and X_1. It then tests whether states in the shadow manifold M_{X_2} can be used for estimating the states in M_{X_1} and *vice versa* via mapping through nearest neighbors (*cross mapping*). Another key idea of CCM is *convergence* which means that as the length of the time-series increases, the shadow manifolds become denser and the ellipsoid or space formed by nearest neighbors shrinks, leading to improvement of cross-map estimates. Although CCM is appealing, it cannot be generalized to stochastic systems as the Taken's theorem, the cornerstone of CCM, will break down in such scenarios (Takens 1981).

Restricted Directed Information (RDI)

As mentioned earlier, the causal inference method in Scribe is based on Restricted Directed Information (RDI). This measure determines the amount of *statistical inter-dependence* (or more

formally the *mutual information*) between the past state of the regulator and current state of the target gene conditioned on the target's immediate previous state.

Cell state transitions are controlled by hierarchical regulatory networks (Peter and Davidson 2011). In such networks, as the expression of regulator changes, their downstream target responds accordingly after some time delay d. A very well-known measure of mutual dependence which accounts for both linear and nonlinear associations between two genes (or more generally, two random variables), X, Y, is mutual information (MI)(Takens 1981; Cover 2006). MI is symmetric and can quantify the "amount of information" obtained about gene X or Y, through the other gene Y or X. It essentially determines how similar the joint distribution ($p_{XY}(x, y)$) of the two genes X, Y is to the products of factored marginal distribution $p_X(x)p_Y(y)$, or formally:

$$I(X;Y) = \sum_{x,y} p_{XY}(x,y) \log \frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}$$

If $I(X;Y)$ is zero, then the two genes X, Y are independent; otherwise it implies there exists some dependence between them (e.g. in the case of a regulator and its target). It is often useful to quantify the mutual dependence between two random variables (for example, regulator X and target Y) while removing the effect of a third random variable (for example another regulator Z or the history state of the target). This leads to the developing of conditional mutual information, which is defined as:

$$I(X;Y|Z) = \sum_{x,y,z} p_{XYZ}(x,y,z) \log \frac{p_{XY|Z}(x,y|z)}{p_{X|Z}(x|z)p_{Y|Z}(y|z)}$$

MI provides a powerful approach to quantify the symmetric interdependence between genes. However, a favorable approach would be to measure the causal score from a potential regulator to its target. We can achieve this by considering the time-series of regulators and targets ($\underline{X}^t$, $\underline{Y}^t$) and quantifying the information transfer from the past state(s) of X to the current state of the variable Y denoted by $Y(t)$. The time-series can be obtained either through pseudotime ordering of scRNA-seq or other single-cell genomics datasets, including live imaging data.

Previously, T. Schreiber reported *Directed Information (DI)* as a measure for the amount of information flowing from the past state(s) of X, the regulator, to the current state of the variable Y, the target (Schreiber 2000). DI is defined as:

$$DI(X \to Y) = \sum_{t=1}^{T} I(\underline{X}^{t-1}; Y(t)|\underline{Y}^{t-1})$$

In order to remove indirect interactions, we can calculate the information transferred from the regulator to the target while conditioning on all the other genes ($\{X_i, X_j\}^c$), which is,

$$DI(X_i \to X_j|\{X_i, X_j\}^c) = \sum_{t=1}^{T} I\left(\underline{X_i}^{t-1}; X_j(t)|\underline{X_j}^{t-1}, \{\underline{X_l}^{t-1}\}_{l\in\{X_i,X_j\}^c}\right)$$

Furthermore, for a set of genes of interest, $X_1, X_2, \ldots, X_N$, from a single-cell genomics dataset, we can infer a Directed Information graph, $G_{DI} = (V, E)$ where the vertex set V corresponds to the genes $\{X_1, X_2, \ldots, X_N\}$ and the edge $e_{ij} = (X_i, X_j)$ from gene X_i to X_j exists if and only if $DI(X_i \to X_j | \{X_i, X_j\}^c) \neq 0$ and the edge weight corresponds to the quantified DI value $DI(X_i \to X_j | \{X_i, X_j\}^c)$.

It was shown that if a system is not purely deterministic, the directed information graph G_{DI} inferred from DI will correctly recover the true causal graph G_C (the network which includes all causal interactions as directed edges) (Schreiber 2000). Although DI is able to detect both of linear and non-linear causality as opposed to the linear Granger causality and is applicable to stochastic system, it (1) can not deal with deterministic systems which may be of interests for certain scenarios and (2) poses huge computational burden because it conditions on all possible previous states of the regulator or target and requires enormous amount of data which is not affordable even with current single-cell genomic datasets.

We recently proposed a novel formulation of DI to alleviate those issues by employing only the immediate past of the target or regulators instead of the all past states assuming a first-order Markov system, which is generally applicable to most biological processes. We term this novel method as Restricted Direct Information (RDI) and define it as,

$$DI_d(X \to Y) = I\left(X(t-d); Y(t) | Y(t-1)\right)$$

and similarly, we can define the the conditioned version of it as:

$$DI_{d_1}\left(X \to Y | Z(t-d_2)\right) = I\left(X(t-d_1); Y(t) | Y(t-1), Z(t-d_2)\right)$$

In ref (A. Rahimzamani and Kannan 2016), it's shown that RDI works in many stochastic or deterministic cases and under some mild assumptions is capable of inferring the correct causal graph G_C. Moreover, we have shown that if the conditions are violated, no other method will be able to recover the correct graph.

4.10.6 Uniformization method for adjusting sampling bias

As opposed to the more empirical DREMI method(Krishnaswamy et al. 2014), Scribe accounts for sampling biases from single-cell measures (for example, small number of cells in transition states with the majority of cells occupying steady states), by integrating uRDI (uniformized RDI) as a user option to provide a rigorous approach to replace the biased sampling distribution with a uniform distribution to quantify potential causality(Arman Rahimzamani and Kannan 2017) (how much influence a regulator can potentially exert on target without cognizance to the regulator's distribution).

As shown above, RDI requires calculating conditional mutual information, $I(X(t-d);Y(t)|Y(t-1))$, which is a function of the joint distribution $p(x_{t-d}, y_t, y_{t-1}) = p(y_t|x_{t-d}, y_{t-1})p(x_{t-d}, y_{t-1})$. In reality, we are often much more interested in the potential causality from the regulator X and the target Y given the target's own history $y(t-1)$ without cognizance of the actual distribution $p(x_{t-d}, y_{t-1})$; that is, the causality should be fully defined by the conditional distribution $p(y_t|x_{t-d}, y_{t-1})$. As single-cell genomics methods often heavily sample cells from steady states while rarely capturing cells in transition states, this can dramatically decrease the performance of RDI, especially considering that the transition states imply critical regulation events.

We were thus motivated to develop a new method, uniformized conditional mutual information (uCMI) , to replace the actual distribution $p(x_{t-d}, y_{t-1})$ with a uniform distribution $u(x_{t-d}, y_{t-1})$ to remove the aforementioned bias from the data and calculate the conditional mutual information for the new distribution $p(y_t|x_{t-d}, y_{t-1})u(x_{t-d}, y_{t-1})$ instead of the actual distribution $p(x_{t-d}, y_t, y_{t-1})$. This is possible thanks to the concept of potential Conditional Mutual Information (qCMI)(Arman Rahimzamani and Kannan 2017) and a novel proposed estimator, in which the actual distribution $p(x_{t-d}, y_{t-1})$ of samples is replaced by any arbitrary distribution $q(x_{t-d}, y_{t-1})$ before estimating the conditional mutual information. uCMI is thus a special case of qCMI, in which the replacement distribution $q(x_{t-d}, y_{t-1})$ is uniform. By replacing the conditional mutual information (CMI) in RDI with uCMI, we obtain a new method called uniformized RDI (uRDI), which measures the potential causality score.

4.10.7 Scribe: a toolkit for visualization and detection of complex causal regulation from single-cell genomics datasets

Although Scribe is applicable to any time-series datasets, it is specifically designed for visualizing and detecting complex gene regulation from single-cell genomics datasets (e.g. scRNA-seq). Scribe relies on (uniformized) restricted directed information to detect causality but also supports other methods, including the well-known mutual information, Granger causality and the more recent CCM. Scribe starts with time-series data, which can be based on “pseudotime-series”of a developmental trajectory reconstructed from scRNA-seq data such as those constructed using Monocle 2, live imaging data or datasets with current and predicted

spliced RNA expression estimated using RNA-velocity. Scribe provides three main types of analysis:

1. Visualization and estimation of causal gene regulation;
2. Inferring temporal causality of gene regulation;
3. Reconstruction of large-scale sparse causal regulatory networks.

4.10.7.1 Preparing pseudotime-series for scRNA-seq datasets

Scribe does not provide any built-in functionalities for pseudotime-series construction and relies on Monocle 2 or other tools, such as dpt(Haghverdi et al. 2016c) or wishbone(Setty et al. 2016), for reconstructing the single-cell trajectory before inferring causal networks. In relation to physical time, pseudotime has an arbitrary scale, thus Scribe doesn't consider pseudotime value themselves instead using the ordering of each cell in pseudotime for causal network inference.

4.10.7.2 Visualizing pairwise gene interaction

In order to intuitively visualize casual regulations between genes, Scribe provides different strategies to visualize the **response**, **causality** and **combinatorial regulatory logic between gene pairs**. The response visualization is similar to the DREVI approach as proposed by Smita Krishnaswamy, et. al(Krishnaswamy et al. 2014) with the exception that it considers time delay to visualize the expected expression of potential targets given a potential regulator's expression after a time delay. Response visualization thus additionally aids in visualizing commonly appeared time-delayed regulations involved in cell differentiation(Alon 2007).

One limitation of response visualization is that it ignores the effects of a gene's previous state to the current state or memory of its history. In order to also capture this effect and thus intuitively visualize causality, Scribe is thus equipped with causality visualization. Essentially, this approach visualizes the causal regulation by considering the information transfer from the time delayed potential regulator to the target's current expression, conditioned on the target's previous state to remove effects from auto-regulation. Causality visualization is a heatmap consisting of the expected value of the target's current expression given the target's immediate past expression (y-axis) and regulator's expression with a time lag d (x-axis). Each column represents the relationship for the target's expression at the previous time point to the current state (memory of the history or "auto-regulation") given a fixed regulator value, while for each row, the information transfer from the regulator to its targets given the previous target state.

4.10.7.3 Visualizing combinatorial gene regulation

It is of great interest to understand the combinatorial gene regulation as it often determines how cells make decisions to choose a particular cell fate or adapt to external stimuli(Qiu, Ding, and Shi 2012d; Alon 2007). In order to visualize two-input combinatorial regulation, Scribe provides a third visualization tool. This visualization is a heatmap consisting of the expected value of the target's current expression given knowledge of both of the regulators' expressions with a time lag (x/y-axis). For both of the causality and the combinatorial logic visualizations, the corresponding expected value is calculated through a local average with a Gaussian kernel.

We noticed that gene regulation *directly* affects the rate of the target gene which then results in gene expression changes. For example, if a gene x is negatively regulated by gene y.

We may define the rate function of x as $\frac{dx_t}{dt} = 1/(x_{t-1}^2 + y_{t-\mu}^2)$. Therefore, visualizing the expected rate of a target at its current state given knowledge of both the regulators' expression with the a time lag (x/y-axis) allows better intuition of regulations (see **supplementary figure 3B**). Although we won't have accurate estimates of the rate of gene expression with pseudotime-series data, the RNA-velocity method can be used to obtain those estimates.

We also noted that combinatorial gene regulation visualization is especially useful to help us visually identify potential direct/indirect regulators. For example, if we have a regulatory pathway $x->y->z$, we will see that the expected gene expression of z is only dependent on the direct regulator y instead of the indirect regulator x from the visualization (**Figure 2D**). That is, combinatorial gene regulation visualization indeed provides visual intuitions for the conditional RDI (in the above case, the row corresponds to $DI_{d_1}(x->z|y(t-d_2))$ while the column $DI_{d_1}(y->z|x(t-d_2))$).

4.10.8 Causal network inference: an RDI-based algorithm

Causal inference in Scribe is based on RDI, which is an extension of directed information under the assumption that the underlying processes can be described by a first-order Markov model. The method we implemented basically tries to calculate the RDI value for each pair of genes (i, j) conditioned over the top L genes (default is 0 or no conditioning and 1 for cases where we used conditioning) which are candidates of being regulators of the gene j.

To reach this goal, it first calculates all the pairwise *unconditioned* RDI values, for all the potential delays specified by the user in vector d (by default, it is a vector includes 5, 10, 20, 25).

Then for each pair (i, j), it treats the delay corresponding to the largest RDI value as the "*true" delay of effect*, i.e. the actual time delay by which the effect of i appears in j. Having identified the "true" delays, the method then re-calculates the pairwise RDI values for each pair of genes (i, j), this time conditioned over the top L (L can be specified by the user) genes with the highest incoming RDI values to j associated with their corresponding true delays, treating them as the potential regulators of j.

The algorithm of causal inference in Scribe is as follows:

Input: gene expression time-series (either based on pseudotime-series, "RNA-velocity" or living imaging data, among others) X_i for each gene i

Output: A matrix of pairwise causality scores

Parameters: d: vector of delays, L: number of conditioning genes

Pseudocode:

1. For each pair of genes (i, j):

   - For all delays $\delta \in d$: Calculate $RDI_\delta(X_i \to X_j)$

   - Set $\delta_{i,j}^{\max} := \text{argmax}_{\delta \in d} RDI_\delta(X_i \to X_j)$

2. For each gene j:

   - For all i: sort $RDI_{\delta_{i,j}^{\max}}(X_i \to X_j)$ values descendingly

- According to the sorting above, take the $L+1$ nodes i with the highest incoming RDI values to j and store them in a set as $\text{inc}_j^{\max}$. Store their corresponding delays $\delta_{i,j}^{\max}$ in a set $d_j^{\max}$.

3. For each pair of genes (i,j):

- If $i \in \text{inc}_j^{\max}$, remove i from $\text{inc}_j^{\max}$. Otherwise, remove the node with the lowest $RDI_{\delta_{i,j}^{\max}}(X_i \to X_j)$ from $\text{inc}_j^{\max}$.

4. Output $RDI_{\delta_{i,j}^{\max}}\left(X_i \to X_j | \{X_l(t-\delta_{l,j}^{\max})\}_{l \in \text{inc}_j^{\max}}\right)$

The estimation of the mutual information is inspired by Kraskov's method (Kraskov, Stögbauer, and Grassberger 2004), which builds on counting nearest-neighbor points. In the R implementation of Scribe, nearest-neighbor points are identified with a modified RANN package while in the python implementation, it relies on the sklearn module.

To calculate the causal network with uRDI, we apply the same algorithm as above but simply replace RDI with uRDI. In addition to what required in RDI, uRDI also needs to estimate the actual distribution, $p(x_{t-d}, y_{t-1})$, which relies on kernel density estimation (KDE). We use standard Gaussian kernels from R or python in the Scribe package to calculate KDE.

4.10.9 Assessing temporal causal gene regulation

Gene regulation is dynamic and the timing of maximal regulation (regulation peak) between gene pairs varies dramatically. Scribe provides a convenient interface to estimate the temporal causal regulation based on a moving window across the time series, that is, we calculate the causality within a time interval, for example, every 50 time-points in time-series data. This method helps to identify the period of the strongest gene regulation which may have important clinical application as it suggests a window for genetic inference.

4.10.10 Inferring and visualizing transcriptomic gene regulatory network

Scribe can estimate a causal network from a set of known TFs (and among the TFs) to a set of target of interest (select through, for example, the BEAM test), or estimate the pairwise causality among all the genes in a set of genes of interest. For the first scenario, Scribe estimates causality between all pairs of TFs and the causality from each TF to each putative target; for the second scenario, Scribe estimates causality for any pair of genes in both directions. In order to retrieve significant causal edges while removing promiscuous edges and reconstruct a sparse causal regulatory network that satisfies known properties of biology networks, Scribe relies on a modified CLR method (*Context Likelihood Relatedness*) and the novel directed network regularization inspired by some biological assumptions (see section **Network sparsifer** below).

In order to facilitate the visualization of complex networks, Scribe provides a variety of approaches to visualize the RDI network either through a heatmap, a hierarchical layout, an arc

diagram or a hive plot, implemented based on igraph, netbiov, ggraph, arcdiagram as well as the HiveR R packages.

In addition to the core causality detection feature based on (uniformalized) restricted direction information, Scribe also supports various methods for inferring the regulatory relationships including mutual information, Granger causality and CCM implemented based on *parmigene*, *vars* and the *rEDM* packages, respectively. We also provide a python package for most of estimation methods, although without extensive support for visualization.

4.10.11 Parameters of RDI

Parameter	**Type**	**Effect of tuning parameters**
d	**Vector of positive integers**	Default: 5, 20, 40 The vector of potential delays, for which the corresponding RDI values are calculated. Setting this argument too small may limit the ability of Scribe to detect causal relationships, while setting it too large can result in discovery of incorrect or indirect causal relationships, resulting in false delays and conditioning.
L	**Non-negativ**	Default: 0

	e Integer	The number of the top incoming node(s) to the target, excluding the source, over which RDI is conditioned. 0 corresponds to no conditioning (Plain pair-wise RDI). Conditioning over more nodes approaches the theoretical prerequisite of conditioning over all genes, excluding the source and target, needed for inferring the true causal network, however it imposes more computational burden and undesirably reduces the accuracy of the RDI estimator with fixed number of samples N, as it exponentially increases the dimension of the state space used to calculate the k-nearest neighbors.
k	**Positive Integer**	Default: 5 The number of the nearest neighbors in kNN estimator for the conditional mutual information. The parameter should be set in such a way so the neighborhood captures adequate number of samples for a good estimate of the probability corresponding to each sample.
Uniformization	**Boolean**	Default: False If True, uRDI instead of RDI will be used. While imposing higher computational burden over the same data than RDI,

		uRDI is expected to improve the causal inference in the cases with highly-biased sampling distributions.

Algorithm complexity

Algorithm	**Methodology**	**Parameters**	**Complexity** **N: the number of samples;** **d: the dimension of the manifolds (default 2);** **k: the number of nearest neighbors** **L: number of** **I: the dimension of the features data**

CCM	Determining the causality from X to Y based on how well one can reconstruct the cross-mapped estimate of X from the nearest neighbors determined on Y space	**E:** The number of lags embedded in the shadow manifold **Tau:** The time lag between each consecutive pair of time samples (default: 1)	O(2Nd+2kN)+O(N)
Granger Causality	Determining the causality from X to Y based on how much the past samples of X contribute in linearly estimating the current state of Y, compared to when the Y is estimated based merely upon its own past	**Maxlag:** The number of lags of past sample included in estimating the current state of Y	O(l^2 N + l^3 + l^2 N + l N)

RDI	Determining the causality from X to Y based on the amount of mutual information between the past of X and the current state of Y conditoned over the past of (potentially) all other variables than X	**k:** The number of neighbors for kNN estimation of mutual information **Delays:** The lags for which the mutual information from the lagged source to the current state of target is estimated. **L:** The number of the conditioning nodes other than X and Y	O(dN logN +kN) + O(N)
uRDI	Same as RDI method, but including the replacement of the empirical distribution of the past samples with a uniform distribution	**All Parameters from RDI plus:** **BW:** The bandwidth of the kernel estimator	O(dN logN +kN) + + (complexity of kde) + O(N)

4.10.12 Limitation of pseudotime based causality inference

Although pseudotime provides us a general trend of the gene expression dynamics, it has some inherent limitations which restrict its performance and accuracy in causal inference. Firstly, the information fluctuation doesn't necessarily transfer from one cell to another nearby cell in pseudotime trajectory as they are different cells. Secondly, the pseudotime is often arbitrary as it is purely based on the distance between cells in the gene expression space. Scribe can avoid these limitations by taking advantage of the old idea of exploiting detection afforded by microarray of unspliced mRNAs as surrogate for ongoing or recent transcription, and hence to derive an approximation of the first time-derivative of transcript abundance - this old idea has been re-used in the recent analysis framework "RNA velocity"(La Manno et al. 2017) to measure the information transfer based on current expression state and predicted future expression from the same cell. As reported the time delay between the current transcript abundance and predicted transcript abundance is close to a particular physical time (for example, 2 hours in reference (La Manno et al. 2017)). Meanwhile, real-time confocal imaging data for about 200 transcription factors' expression dynamics over the entire *C. elegans* embryogenesis(La Manno et al. 2017; J. I. Murray et al. 2012) also provides us a great resource of real-time series data for each gene in each *C. elegans* cell.

4.10.12 Network sparsifier: CLR and directed graph regularization

In theory, Scribe can remove potential indirect causal gene regulation from one gene x to another gene y by conditioning on all other genes in the transcriptome excepting x. However,

this is impractical as it requires infinite samples and causes enormous computational burden. Therefore, we seek for alternative approaches based on statistical significance and reasonable assumptions of biological structures to remove potential indirect edges. The first method we applied is the CLR or *Context Likelihood Relatedness*. After computing the causality score with RDI (uRDI) between all gene-pairs, CLR calculates a normalized score based on the z-score (or 0 if the z-score is less than 0) from all the input edges to the potential target and all the output edges from the potential regulator of the gene pair. This normalized score is used as a statistical likelihood of each causal edge regarding its network context. More formally, denoting the asymmetric matrix R corresponds to all raw causality scores calculated with Scribe, with R_{ij} being the causality score from gene i to gene j, we can calculate the z-score z_i based on all gene i's output causality scores and z_j all gene j's input causality scores. The normalized score of R_{ij}, $\hat{R}_{ij}$ is defined as:

$$\hat{R}_{ij} = \sqrt{(max(0, z_i))^2 + (max(0, z_j))^2}/2.$$

The user can either use the normalized score or choose a threshold of the normalized scores and treat the edges above the threshold as significant or real regulation compared to the background distribution of the causality scores. As discussed in the original study, CLR removes many of the false regulations in the network by eliminating "promiscuous" cases, where one regulator weakly co-varies with a large number of genes, or one gene weakly co-varies with many transcription factors which may arise when the assayed conditions are inadequately or unevenly sampled. We note that, however, the original CLR is only applied on a symmetric mutual information based matrix while we are dealing with an asymmetric matrix of causality scores. After applying CLR,

the network may be still dense and contain spurious edges. Previous studies have shown that the biological networks have some special properties distinct from those of random networks; for example, the network's out-degree distribution is well approximated by a power law distribution where its in-degree distribution is almost an exponential distribution. Based on those assumptions, we proposed a new regularization method for a directed graph.

The goal of our method is to learn a sparse directed graph from a dense asymmetric causality network (retrieved after applying CLR) satisfying two aforementioned properties. The directed graph's structure is represented by an indicator matrix denoted by $\Theta \in \{0,1\}^{N\times N}$, where $\theta_{i,j} = 1$ stands for the existence of edge i to j, and 0 otherwise . Since the entries are indicators, the in-degree and out-degree of each node in the network can be easily formulated. Specifically, the out-degree of the ith node can be represented by $h_{out}(i) = ||\theta_i||_1$ and the in-degree of the ith gene is correspondingly represented by $h_{in}(i) = ||\theta^i||_1$, where θ_i and θ^i are the ith row and ith column of Θ, and ℓ_1 counts the number of nonzero elements since $\theta_{i,j} \in \{0,1\}$. Given the asymmetric matrix of causality score R with the (i,j)th entry as R_{ij}, the following optimization problem is formulated to learn the structure of the network:

$$\min_{\Theta\in\mathcal{A}} -\sum_{i,j}\theta_{i,j}R_{i,j} + \alpha\sum_{i=1}^{N}\log(||\theta_i||_1 + \xi) + \lambda\sum_{i=1}^{N}||\theta^i||_1$$

where the feasible set of the network structure is

$$\mathcal{A} = \{\Theta \in \{0,1\}^{N\times N} : \sum_i\sum_j \theta_{i,j} \geq B\}$$

The intuition of the objective function comes directly from the above three assumptions: the first term of the objective is to select the edge with large value of R_{ij}; the second term is the negative log likelihood of the power law distribution for the out-degree of each gene; the last term is the negative log likelihood of the exponential distribution for the in-degree of each gene. The budget parameter B is introduced to prevent trivial solutions, and a small positive value ξ is used to prevent the numerical issue of log function. Parameter α is the exponent of the power law distribution and λ is the parameter of the exponential distribution.

4.10.13 Benchmark Scribe with alternative algorithms on inferring causal regulatory network

We follow the same procedure as reported previously(Qiu, Ding, and Shi 2012d) to simulate the differentiation of central nervous system, excepting to replace the correlated noise in the previous study with independent additive noise for the purpose of simplicity. For creating **Fig 1C**, we set the time step as 0.1, samples per simulation as 100, the total number of simulations as 20. We then infer the causal network based on all the 2000 samples using CCM, GC and RDI or uRDI either without conditioning or conditioning on one gene that has the maximal input causality other than the current regulator to the target. The time delay between regulator and target used in all those algorithms is set to be 1. We compare the inferred network with the known network to calculate the AUC (area under the curve) and the result is repeated for 25 times to ensure reliable conclusions. We also increase the standard deviation of the intrinsic noise all the way from 0 to 0.2. ROC (Receiver Operating Characteristic) curve in **Fig 1B** is obtained similarly while setting the simulation based on a linear system where the transition matrix A is generated according to the network with non-zero coefficients randomly taken from a uniform

distribution $u(0.75, 1.25)$. The A matrix is then normalized to $max(eigen(A)) * 1.01$ to avoid the divergence of the system. The intrinsic noise standard deviation (s.d) is set to be equal to 0.01. All the genes are initialized with a random value $u(0.5, 2)$. To infer the causal network, we take 100 samples per simulation and perform the simulation five times, then apply Scribe, CCM, and GC on those simulated data points.

To visualize the response, causality and combinatorial regulations as in **Fig 1, Fig SI2**, a single simulation leading to the neuron fate is used. To create the response and the causality visualization for the two-node motifs (Qiu, Ding, and Shi 2012d), the network motifs is firstly converted into a set of SDE functions using similar formulations as that used in the above simulation for neuronal differentiation. The expression dynamics is then simulated by setting the initial expression for both genes as 0.01 and followed based on the set of SDE equations (**Fig SI2a**). We used similar procedures to simulate expression of genes under combinatorial regulations with different logic gates and then create the combinatorial regulation visualizations (**Fig SI2b**).

4.10.14 Practical suggestions on running Scribe

Users can often simply run Scribe with default settings to visualize gene regulation and infer causal regulatory network. By default, we scan a vector of time delays of 5, 20, 41 to infer causal network. We also set the default maximal number of genes used for conditioning to be 1 although most of the analysis in this work doesn't use conditioning because it requires a lot of data sample and computation time. If the users have a better idea of the time delay between the regulator to

the target or enough data points, they may consider adapt the *delay* argument or increase the number of genes for conditioning when running Scribe for the causal network inference.

The nonlinear interactions between the regulator (causal variable) and its targets (effect variable) often prevents us to conclude a simple positive or negative regulation because the regulator may appear to positively relate to the target at some time points but negatively to the target at some other time points (Kraskov, Stögbauer, and Grassberger 2004; Sugihara et al. 2012). The causal network we inferred with Scribe thus doesn't include a sign. Sometime user may want to assign the sign of the causal interaction simply based on the sign of correlation calculated between regulator to target.

4.11 Details on analyzing datasets used in this study

4.11.1 Infer causal network with pseudotime ordered scRNA-seq datasets.

Lung data is processed as described previously (Qiu, Mao, et al. 2017a). Categorization of pneumocyte specification markers into either early and late groups used for benchmarking is based on references(Qiu, Mao, et al. 2017a; Treutlein et al. 2014).

LPS data is pre-processed as described previously (Qiu, Hill, et al. 2017c) while the trajectory is reconstructed with the new method of reversed graph embedding on the same set of ordering gene used in the previous study. Only the path with wild-type cells is used for causal network inference. Regulators and targets, and the regulatory network used for benchmarking is collected from references (Qiu, Hill, et al. 2017c; Amit et al. 2009) and reference (Garber et al. 2012), respectively.

Olsson data is processed as described previously. The master regulators, transcription factors and downstream targets, and the regulatory network used for benchmarking are collected from reference (Qiu, Mao, et al. 2017a) and references (Qiu, Mao, et al. 2017a; Su et al. 2017), respectively.

Paul data is processed as described previously. Only the path leading to the erythrocytic fate is used for reconstructing the causal regulatory network. The regulatory network responsible for differentiation of erythrocyte cells used for benchmarking is collected from \cite{erythroid lineage}.

4.11.2 Infer causal network with RNA-velocity.

The data of the chromaffin cell "RNA-velocity" analysis is retrieved from (http://pklab.med.harvard.edu/velocyto/notebooks/R/chromaffin.nb.html). We use the estimated exon expression to reconstruct the trajectory for the chromaffin cell commitment. Only cells on the path from the Schwann cell progenitors to mature chromaffin cells are used to infer the causal network. Two different formulations, $I(x_t; y_{t+1}|y_t)$ (or $I(x_t; ||\overrightarrow{V_i}||_2|y_t)$), can be used to infer causal networks with data from RNA-velocity. In this study, we mostly apply the first formulation.

4.11.3 Infer causal network with live-image data.

Lineage-resolved live-imaging data for *C. elegans* early embryogenesis is obtained from Waterston lab. Raw fluorescence intensity signal is directly used for causal network inference. We note two caveats to analyzing the reporter data with Scribe. First, although the

promoter-fusion data sheds light on the induction kinetics of the TF of interest, once the fluorescent reporter is expressed it follows the trafficking and degradation kinetics of the histone protein and not the TF. Second, the time series for each TF was captured in a different embryo, so this may introduce noise that obscures the regulator/target relationships between the TFs although the *C. elegans* development process is highly robust. Nevertheless, this data set represents an unprecedented view of TF activity at high spatiotemporal resolution during the early development of a complex organism (**Supplementary Figure 8**).

Prospective

The single-cell genomics field is still rapidly developing. Various new directions are emerging, which leads to a multitude of computational or experimental challenges. Chief among them includes: how to develop algorithms to analyze very large and complex datasets generated from cell atlas projects; how to analyze datasets with spatial gene expression information; how to analyze multi-omic dataset which may include data from, for example, transcriptome, chromatin accessibility and proteomics at the same time; how to analyze perturb-seq like dataset to infer causal gene regulatory networks directly from perturbation experiments; how to directly experimentally track cell lineage and gene regulatory network hierarchy at the same time. Therefore, moving forward, I would like to continue working on following directions.

1. **Cell atlas datasets analysis**: Manipulating, visualizing and learning complex trajectories and cell states for big cell atlas datasets.
2. **Causal network inference**: Inferring causal gene regulatory networks with the emerging technique of CRISPR-cas8 based genetic screening with single-cell genomics readout.
3. **Tracing lineage and regulatory hierarchy with molecular recorders**: Integrating CRISPR-cas9 based molecular recorders with single-cell genomics to understand the intricate lineage hierarchy and regulatory complexity.

Cell atlas datasets analysis

Cell atlas projects have already generated an enormous amount of data and will generate thousand times more datasets, thus they pose great computational challenges for data

manipulation, visualization as well as analyzation. The first but also often neglected challenge in this direction is the storage and memory burdens related to analyzing big single-cell genomics datasets. To address this and many other emerging computational challenges, new toolkits can be developed to handle cell atlas datasets with millions of cells through effective data storage and downsampling techniques, followed by new dimension reduction and generalized trajectory inference methods to understand the cellular heterogeneity and dynamics. For the data-storage, the HDF5 file format will be applied by taking advantage of the sparsity of the scRNA-seq data. For data downsampling, two parallel strategies can be used. Firstly, the k-nearest neighbor graph or kNN-graph(Naidan, Boytsov, and Nyberg 2015) can be applied to convolve gene expression values to achieve hundreds-fold data compression (can also be used to identify cell groups with Louvain clustering algorithm(Levine et al. 2015)). Secondly, landmark-based approaches(Mao, Zheng, et al. 2015) can be developed to identify a small subset of representatives while also ensure comprehensive downsampling of the space of cellular heterogeneity. The downsampled data can then be integrated seamlessly with downstream analysis of data visualization, clustering as well trajectory inference. Newest algorithms with a limited assumption of continuity of gene expression space have been developed and shown to be able to accurately learn a smooth skeleton(L. Wang, Mao, and Tsang 2017) from the high dimension noisy data. Aided by the smooth skeleton structure learning, a L1 graph method(Mao et al. 2016b) based on L1 regularization can be used to learn a general graph thus not limiting to the tree structure as other state-of-the-art single-cell analysis tools(Qiu, Mao, et al. 2017b; Haghverdi et al. 2016c; Setty et al. 2016),(Satija et al. 2015) (**Fig. 1**). Importantly, after organizing the data into a graph of cell states or cell fate specification, improved differential gene expression tests will be developed to identify

associated biomarkers or key transcription factors(Qiu, Hill, et al. 2017b). In particular, method from geostatistics, like Moran's *I* test may be applied to detect significant gene expression changes on the low dimensional manifold without assuming any particular structure of the manifold, therefore is generally applicable to various complex developmental structure and potentially applied to any spatial transcriptomics like those measured by MERFISH, Seq-fish or in-situ sequencing. Additional ideas include gene expression entropy(Teschendorff and Enver 2017) can be incorporated to automatically identify progenitors and terminal cell types as well as the direction of cell fate specification on the reconstructed graph.

Causal network inference

In this thesis, I discussed approaches to infer the causal network based on quantifying the information transfer involved in putative target's time-delayed response to putative regulator's gene expression changes. The arrival of CRISPR-Cas9 based targeted mutagenesis for genetic screening combined with single-cell genomics readout provides another alternative to directly reconstruct the causal network by observing the gene expression changes after combinatorial knockout of regulators(Dixit et al. 2016)(Adamson et al. 2016; Jaitin et al. 2016; Datlinger et al. 2017). Methods include elastic net regression or RIDS (Robust IDentification of Sparse network with perturbation expression) can be used to accurately identify the regulatory network.

Tracing lineage and regulatory hierarchy with molecular recorders

Although the computational algorithm of developmental trajectories reconstruction I discussed is a powerful approach in mapping the detailed developmental dynamics, the destructive nature of

single-cell genomics prevents longitudinal observations and thus the computational results are nevertheless biased by the sampling procedure and is also compromised by the low capture efficiency (especially for the lowly expressed key transcription factors in development process) in the data collection procedure which we have limited control. It is therefore essential to develop new technologies to directly map lineage relationship and resolve regulatory hierarchy of key transcription factors.

Another emerging theme in single-cell transcriptomics is the repurposing of CRISPR-Cas9 system for recording molecular events as well as lineage relationships(McKenna et al. 2016; Kalhor, Mali, and Church 2017; Junker et al. 2017; Perli, Cui, and Lu 2016; Shipman et al. 2016; Sheth et al. 2017). A number of papers(McKenna et al. 2016; Kalhor, Mali, and Church 2017; Junker et al. 2017; Perli, Cui, and Lu 2016; Shipman et al. 2016; Sheth et al. 2017) recently reported such molecular recorders achieved by exploiting the versatile CRISPR-Cas9 system to map the zebrafish development lineage or to measure the concentration of metabolic molecules.

It is possible to use such a molecular recorder to record the lineage relationship, and more importantly the timing of activity peak of master regulators during lineage specification, which can for example address the controversy around the lineage relationship during hematopoiesis(Pei et al. 2017; Velten et al. 2017; Perié et al. 2015). A CRISPR-cas9 system can be designed to record the activity peak timing of master regulators, including *Cebpa*, *Pu. 1*, *Gata 1*, *Eklf*, *Fli1*, *etc*. This system can be transfected into a small number of murine hematopoietic stem cells which are then injected into total-body irradiated mice to repopulate the entire hematopoietic system. The scRNA-seq will be applied to measure both the recording sequence

(which directly records the gene regulatory hierarchy) as well as the whole transcriptome for the repopulated blood cells. By the transcriptomes we can identify the corresponding cell types while combining with recording sequence we will be able to also identify the gene regulatory hierarchy and the lineage relationship at the same time.

As the technologies continue to emerge and mature, I expect we will be able to visualize the spatial distribution of transcriptome either through single-cell RNA-FISH (Fluorescence In Situ Hybridization) or in-situ sequencing; to measure all RNAs and proteins as well as their various modifications in single cells through native sequencing, for example nanopore or SMRT (Single Molecule Real Time) sequencing; to track the transcriptome over time through novel live imaging technologies. Those new technologies will produce new datasets and thus lead to novel computational problems. It is imperative for us to work at the forefront of computational biology and technology development to attack those emerging computational challenges by developing novel state-of-the-art machine learning toolkits and aim toward the ultimate goal of understanding the fundamental mechanism of developmental processes as well as the evolution of development.

BIBLIOGRAPHY

Adamson, Britt, Thomas M. Norman, Marco Jost, Min Y. Cho, James K. Nuñez, Yuwen Chen, Jacqueline E. Villalta, et al. 2016. "A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response." *Cell* 167 (7): 1867–82.e21.

Aibar, Sara, Carmen Bravo González-Blas, Thomas Moerman, Vân Anh Huynh-Thu, Hana Imrichova, Gert Hulselmans, Florian Rambow, et al. 2017. "SCENIC: Single-Cell Regulatory Network Inference and Clustering." *Nature Methods* 14 (11): 1083–86.

Alon, Uri. 2007. "Network Motifs: Theory and Experimental Approaches." *Nature Reviews. Genetics* 8 (6): 450–61.

Amit, I., M. Garber, N. Chevrier, A. P. Leite, Y. Donner, T. Eisenhaure, M. Guttman, et al. 2009. "Unbiased Reconstruction of a Mammalian Transcriptional Network Mediating Pathogen Responses." *Science* 326 (5950): 257–63.

Ao, Ping. 2009. "Global View of Bionetwork Dynamics: Adaptive Landscape." *Journal of Genetics and Genomics = Yi Chuan Xue Bao* 36 (2): 63–73.

Babtie, Ann C., Thalia E. Chan, and Michael P. H. Stumpf. 2017. "Learning Regulatory Models for Cell Development from Single Cell Transcriptomic Data." *Current Opinion in Systems Biology* 5: 72–81.

Bar-Joseph, Ziv, Anthony Gitter, and Itamar Simon. 2012. "Studying and Modelling Dynamic Biological Processes Using Time-Series Gene Expression Data." *Nature Reviews. Genetics* 13 (8): 552–64.

Bellman, Richard. 1954. "The Theory of Dynamic Programming." DTIC Document. http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=AD0604386.

Buenrostro, Jason D., Beijing Wu, Ulrike M. Litzenburger, Dave Ruff, Michael L. Gonzales, Michael P. Snyder, Howard Y. Chang, and William J. Greenleaf. 2015. "Single-Cell Chromatin Accessibility Reveals Principles of Regulatory Variation." *Nature* 523 (7561): 486–90.

Cao, Junyue, Jonathan S. Packer, Vijay Ramani, Darren A. Cusanovich, Chau Huynh, Riza Daza, Xiaojie Qiu, et al. 2017a. "Comprehensive Single Cell Transcriptional Profiling of a Multicellular Organism by Combinatorial Indexing." *bioRxiv*. https://doi.org/10.1101/104844.

———. 2017b. "Comprehensive Single-Cell Transcriptional Profiling of a Multicellular Organism." *Science* 357 (6352): 661–67.

Chan, Thalia E., Michael P. H. Stumpf, and Ann C. Babtie. 2017. "Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures." *Cell Systems* 5 (3): 251–67.e3.

Cover. 2006. *Elements of Information Theory*. John Wiley & Sons.

Cusanovich, Darren A., Riza Daza, Andrew Adey, Hannah A. Pliner, Lena Christiansen, Kevin L. Gunderson, Frank J. Steemers, Cole Trapnell, and Jay Shendure. 2015. "Multiplex Single Cell Profiling of Chromatin Accessibility by Combinatorial Cellular Indexing." *Science* 348 (6237): 910–14.

Datlinger, Paul, André F. Rendeiro, Christian Schmidl, Thomas Krausgruber, Peter Traxler,

Johanna Klughammer, Linda C. Schuster, Amelie Kuchler, Donat Alpar, and Christoph Bock. 2017. “Pooled CRISPR Screening with Single-Cell Transcriptome Readout.” *Nature Methods* 14 (3): 297–301.

Dixit, Atray, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, et al. 2016. “Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens.” *Cell* 167 (7): 1853–66.e17.

Du, Zhuo, Anthony Santella, Fei He, Michael Tiongson, and Zhirong Bao. 2014. “De Novo Inference of Systems-Level Mechanistic Models of Development from Live-Imaging-Based Phenotype Analysis.” *Cell* 156 (1-2): 359–72.

Faith, Jeremiah J., Boris Hayete, Joshua T. Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J. Collins, and Timothy S. Gardner. 2007. “Large-Scale Mapping and Validation of Escherichia Coli Transcriptional Regulation from a Compendium of Expression Profiles.” *PLoS Biology* 5 (1): e8.

Fiers, Mark W. E. J., Mark W E, Liesbeth Minnoye, Sara Aibar, Carmen Bravo González-Blas, Zeynep Kalender Atak, and Stein Aerts. 2018. “Mapping Gene Regulatory Networks from Single-Cell Omics Data.” *Briefings in Functional Genomics*. https://doi.org/10.1093/bfgp/elx046.

Fortes, Puri, Dasa Longman, Susan McCracken, Joanna Y. Ip, Raymond Poot, Iain W. Mattaj, Javier F. Cáceres, and Benjamin J. Blencowe. 2007. “Identification and Characterization of RED120: A Conserved PWI Domain Protein with Links to Splicing and 3’-End Formation.” *FEBS Letters* 581 (16): 3087–97.

Frieda, Kirsten L., James M. Linton, Sahand Hormoz, Joonhyuk Choi, Ke-Huan K. Chow, Zakary S. Singer, Mark W. Budde, Michael B. Elowitz, and Long Cai. 2017. “Synthetic Recording and in Situ Readout of Lineage Information in Single Cells.” *Nature* 541 (7635): 107–11.

Friedman, Nir, Michal Linial, Iftach Nachman, and Dana Pe’er. 2000. “Using Bayesian Networks to Analyze Expression Data.” In *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology - RECOMB ’00*. https://doi.org/10.1145/332306.332355.

Furlan, Alessandro, Vyacheslav Dyachuk, Maria Eleni Kastriti, Laura Calvo-Enrique, Hind Abdo, Saida Hadjab, Tatiana Chontorotzea, et al. 2017. “Multipotent Peripheral Glial Cells Generate Neuroendocrine Cells of the Adrenal Medulla.” *Science* 357 (6346). https://doi.org/10.1126/science.aal3753.

Garber, Manuel, Nir Yosef, Alon Goren, Raktima Raychowdhury, Anne Thielke, Mitchell Guttman, James Robinson, et al. 2012. “A High-Throughput Chromatin Immunoprecipitation Approach Reveals Principles of Dynamic Gene Regulation in Mammals.” *Molecular Cell* 47 (5): 810–22.

Gorban, Alexander N., and Andrei Y. Zinovyev. 2009. “Principal Graphs and Manifolds.” *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques*. Information Science Reference, IGI Global: Hershey, PA, USA, 28–59.

Granger, C. W. J. 1969. “Investigating Causal Relations by Econometric Models and Cross-Spectral Methods.” *Econometrica: Journal of the Econometric Society* 37 (3): 424.

Haghverdi, Laleh, Maren Büttner, F. Alexander Wolf, Florian Buettner, and Fabian J. Theis. 2016a. “Diffusion Pseudotime Robustly Reconstructs Lineage Branching.” *Nature Methods*

13 (10): 845–48.
———. 2016b. “Diffusion Pseudotime Robustly Reconstructs Lineage Branching.” *Nature Methods* 13 (10): 845–48.
———. 2016c. “Diffusion Pseudotime Robustly Reconstructs Lineage Branching.” *Nature Methods* 13 (10): 845–48.
Hamey, Fiona K., Sonia Nestorowa, Sarah J. Kinston, David G. Kent, Nicola K. Wilson, and Berthold Göttgens. 2017. “Reconstructing Blood Stem Cell Regulatory Network Models from Single-Cell Molecular Profiles.” *Proceedings of the National Academy of Sciences of the United States of America* 114 (23): 5822–29.
Hastie, Trevor, and Werner Stuetzle. 1989. “Principal Curves.” *Journal of the American Statistical Association* 84 (406). Taylor & Francis: 502–16.
Hill, Steven M., Laura M. Heiser, Thomas Cokelaer, Michael Unger, Nicole K. Nesser, Daniel E. Carlin, Yang Zhang, et al. 2016. “Inferring Causal Molecular Networks: Empirical Assessment through a Community-Based Effort.” *Nature Methods* 13 (4): 310–18.
Huynh-Thu, Vân Anh, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. 2010. “Inferring Regulatory Networks from Expression Data Using Tree-Based Methods.” *PloS One* 5 (9). https://doi.org/10.1371/journal.pone.0012776.
Jaitin, Diego Adhemar, Assaf Weiner, Ido Yofe, David Lara-Astiaso, Hadas Keren-Shaul, Eyal David, Tomer Meir Salame, Amos Tanay, Alexander van Oudenaarden, and Ido Amit. 2016. “Dissecting Immune Circuits by Linking CRISPR-Pooled Screens with Single-Cell RNA-Seq.” *Cell* 167 (7): 1883–96.e15.
Junker, Jan Philipp, Bastiaan Spanjaard, Josi Peterson-Maduro, Anna Alemany, Bo Hu, Maria Florescu, and Alexander van Oudenaarden. 2017. “Massively Parallel Clonal Analysis Using CRISPR/Cas9 Induced Genetic Scars.” *bioRxiv*. https://doi.org/10.1101/056499.
Kalhor, Reza, Prashant Mali, and George M. Church. 2017. “Rapidly Evolving Homing CRISPR Barcodes.” *Nature Methods* 14 (2): 195–200.
Keller, R. E. 1975. “Vital Dye Mapping of the Gastrula and Neurula of Xenopus Laevis. I. Prospective Areas and Morphogenetic Movements of the Superficial Layer.” *Developmental Biology* 42 (2): 222–41.
Kim, Jong Kyoung, Aleksandra A. Kolodziejczyk, Tomislav Ilicic, Tomislav Illicic, Sarah A. Teichmann, and John C. Marioni. 2015. “Characterizing Noise Structure in Single-Cell RNA-Seq Distinguishes Genuine from Technical Stochastic Allelic Expression.” *Nature Communications* 6 (October): 8687.
Kraskov, Alexander, Harald Stögbauer, and Peter Grassberger. 2004. “Estimating Mutual Information.” *Physical Review E* 69 (6). https://doi.org/10.1103/physreve.69.066138.
Krishnaswamy, Smita, Matthew H. Spitzer, Michael Mingueneau, Sean C. Bendall, Oren Litvin, Erica Stone, Dana Pe’er, and Garry P. Nolan. 2014. “Systems Biology. Conditional Density-Based Analysis of T Cell Signaling in Single-Cell Data.” *Science* 346 (6213): 1250689.
Kumar, Pavithra, Yuqi Tan, and Patrick Cahan. 2017. “Understanding Development and Stem Cells Using Single Cell-Based Analyses of Gene Expression.” *Development* 144 (1): 17–32.
La Manno, Gioele, Ruslan Soldatov, Hannah Hochgerner, Amit Zeisel, Viktor Petukhov, Maria Kastriti, Peter Lonnerberg, et al. 2017. “RNA Velocity in Single Cells.” *bioRxiv*. https://doi.org/10.1101/206052.

Langfelder, Peter, and Steve Horvath. 2008. “WGCNA: An R Package for Weighted Correlation Network Analysis.” *BMC Bioinformatics* 9 (1): 559.

Levine, Jacob H., Erin F. Simonds, Sean C. Bendall, Kara L. Davis, El-Ad D. Amir, Michelle D. Tadmor, Oren Litvin, et al. 2015. “Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells That Correlate with Prognosis.” *Cell* 162 (1): 184–97.

Liu, Serena, and Cole Trapnell. 2016. “Single-Cell Transcriptome Sequencing: Recent Advances and Remaining Challenges.” *F1000Research* 5 (February). https://doi.org/10.12688/f1000research.7223.1.

Mao, Qi, Li Wang, Steve Goodison, and Yijun Sun. 2015. “Dimensionality Reduction Via Graph Structure Learning.” In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 765–74. KDD ’15. New York, NY, USA: ACM.

Mao, Qi, Li Wang, Ivor Tsang, and Yijun Sun. 2016a. “Principal Graph and Structure Learning Based on Reversed Graph Embedding.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, December. https://doi.org/10.1109/TPAMI.2016.2635657.

———. 2016b. “Principal Graph and Structure Learning Based on Reversed Graph Embedding.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, December. ieeexplore.ieee.org. https://doi.org/10.1109/TPAMI.2016.2635657.

Mao, Qi, Le Yang, Li Wang, Steve Goodison, and Yijun Sun. n.d. “SimplePPT: A Simple Principal Tree Algorithm.” In *Proceedings of the 2015 SIAM International Conference on Data Mining*, 792–800.

Mao, Qi, Wei Zheng, Li Wang, Yunpeng Cai, Volker Mai, and Yijun Sun. 2015. “Parallel Hierarchical Clustering in Linearithmic Time for Large-Scale Sequence Analysis.” In *2015 IEEE International Conference on Data Mining*. https://doi.org/10.1109/icdm.2015.90.

Margolin, Adam A., Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. 2006. “ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context.” *BMC Bioinformatics* 7 Suppl 1 (March): S7.

Matsumoto, Hirotaka, Hisanori Kiryu, Chikara Furusawa, Minoru S. H. Ko, Shigeru B. H. Ko, Norio Gouda, Tetsutaro Hayashi, and Itoshi Nikaido. 2017. “SCODE: An Efficient Regulatory Network Inference Algorithm from Single-Cell RNA-Seq during Differentiation.” *Bioinformatics* 33 (15): 2314–21.

Ma, Wenzhe, Ala Trusina, Hana El-Samad, Wendell A. Lim, and Chao Tang. 2009. “Defining Network Topologies That Can Achieve Biochemical Adaptation.” *Cell* 138 (4): 760–73.

McKenna, Aaron, Gregory M. Findlay, James A. Gagnon, Marshall S. Horwitz, Alexander F. Schier, and Jay Shendure. 2016. “Whole-Organism Lineage Tracing by Combinatorial and Cumulative Genome Editing.” *Science* 353 (6298): aaf7907.

Meyer, Patrick E., Frédéric Lafitte, and Gianluca Bontempi. 2008. “Minet: A R/Bioconductor Package for Inferring Large Transcriptional Networks Using Mutual Information.” *BMC Bioinformatics* 9 (1): 461.

Murray, J. I., T. J. Boyle, E. Preston, D. Vafeados, B. Mericle, P. Weisdepp, Z. Zhao, Z. Bao, M. Boeck, and R. H. Waterston. 2012. “Multidimensional Regulation of Gene Expression in the C. Elegans Embryo.” *Genome Research* 22 (7): 1282–94.

Murray, John Isaac, Zhirong Bao, Thomas J. Boyle, Max E. Boeck, Barbara L. Mericle, Thomas J. Nicholas, Zhongying Zhao, Matthew J. Sandel, and Robert H. Waterston. 2008. “Automated Analysis of Embryonic Gene Expression with Cellular Resolution in C.

Elegans." *Nature Methods* 5 (8): 703–9.
Naidan, Bilegsaikhan, Leonid Boytsov, and Eric Nyberg. 2015. "Permutation Search Methods Are Efficient, yet Faster Search Is Possible." *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases* 8 (12): 1618–29.
Ocone, Andrea, Laleh Haghverdi, Nikola S. Mueller, and Fabian J. Theis. 2015. "Reconstructing Gene Regulatory Dynamics from High-Dimensional Single-Cell Snapshot Data." *Bioinformatics* 31 (12): i89–96.
Olsson, Andre, Meenakshi Venkatasubramanian, Viren K. Chaudhri, Bruce J. Aronow, Nathan Salomonis, Harinder Singh, and H. Leighton Grimes. 2016. "Single-Cell Analysis of Mixed-Lineage States Leading to a Binary Cell Fate Choice." *Nature* 537 (7622): 698–702.
Owraghi, Melissa, Gina Broitman-Maduro, Thomas Luu, Heather Roberson, and Morris F. Maduro. 2010. "Roles of the Wnt Effector POP-1/TCF in the C. Elegans Endomesoderm Specification Gene Network." *Developmental Biology* 340 (2): 209–21.
Papili Gao, Nan, S. M. Minhaz Ud-Dean, Olivier Gandrillon, and Rudiyanto Gunawan. 2017. "SINCERITIES: Inferring Gene Regulatory Networks from Time-Stamped Single Cell Transcriptional Expression Profiles." *Bioinformatics* , September. https://doi.org/10.1093/bioinformatics/btx575.
Paul, Franziska, Ya 'ara Arkin, Amir Giladi, Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Deborah Winter, et al. 2015. "Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors." *Cell* 163 (7): 1663–77.
Pei, Weike, Thorsten B. Feyerabend, Jens Rössler, Xi Wang, Daniel Postrach, Katrin Busch, Immanuel Rode, et al. 2017. "Polylox Barcoding Reveals Haematopoietic Stem Cell Fates Realized in Vivo." *Nature* 548 (7668): 456–60.
Perié, Leïla, Ken R. Duffy, Lianne Kok, Rob J. de Boer, and Ton N. Schumacher. 2015. "The Branching Point in Erythro-Myeloid Differentiation." *Cell* 163 (7): 1655–62.
Perli, Samuel D., Cheryl H. Cui, and Timothy K. Lu. 2016. "Continuous Genetic Recording with Self-Targeting CRISPR-Cas in Human Cells." *Science* 353 (6304). https://doi.org/10.1126/science.aag0511.
Peter, Isabelle S., and Eric H. Davidson. 2011. "A Gene Regulatory Network Controlling the Embryonic Specification of Endoderm." *Nature* 474 (7353): 635–39.
Pliner, Hannah, Jonathan Packer, Jose McFaline-Figueroa, Darren Cusanovich, Riza Daza, Sanjay Srivatsan, Xiaojie Qiu, et al. 2017. "Chromatin Accessibility Dynamics of Myogenesis at Single Cell Resolution." https://doi.org/10.1101/155473.
Qiu, Xiaojie, Shanshan Ding, and Tieliu Shi. 2012a. "From Understanding the Development Landscape of the Canonical Fate-Switch Pair to Constructing a Dynamic Landscape for Two-Step Neural Differentiation." *PloS One* 7 (12): e49271.
———. 2012b. "From Understanding the Development Landscape of the Canonical Fate-Switch Pair to Constructing a Dynamic Landscape for Two-Step Neural Differentiation." *PloS One* 7 (12): e49271.
———. 2012c. "From Understanding the Development Landscape of the Canonical Fate-Switch Pair to Constructing a Dynamic Landscape for Two-Step Neural Differentiation." *PloS One* 7 (12): e49271.
———. 2012d. "From Understanding the Development Landscape of the Canonical Fate-Switch Pair to Constructing a Dynamic Landscape for Two-Step Neural Differentiation." *PloS One* 7 (12): e49271.

Qiu, Xiaojie, Andrew Hill, Jonathan Packer, Dejun Lin, Yi-An Ma, and Cole Trapnell. 2017a. "Single-Cell mRNA Quantification and Differential Analysis with Census." *Nature Methods*, January. Nature Research. https://doi.org/10.1038/nmeth.4150.

———. 2017b. "Single-Cell mRNA Quantification and Differential Analysis with Census." *Nature Methods* 14 (3): 309–15.

———. 2017c. "Single-Cell mRNA Quantification and Differential Analysis with Census." *Nature Methods* 14 (3): 309–15.

Qiu, Xiaojie, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah A. Pliner, and Cole Trapnell. 2017a. "Reversed Graph Embedding Resolves Complex Single-Cell Trajectories." *Nature Methods* 14 (10): 979–82.

Qiu, Xiaojie, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah Pliner, and Cole Trapnell. 2017b. "Reversed Graph Embedding Resolves Complex Single-Cell Developmental Trajectories." https://doi.org/10.1101/110668.

Rahimzamani, A., and S. Kannan. 2016. "Network Inference Using Directed Information: The Deterministic Limit." In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 156–63.

Rahimzamani, Arman, and Sreeram Kannan. 2017. "Potential Conditional Mutual Information: Estimators, Properties and Applications." *arXiv [cs.IT]*. arXiv. http://arxiv.org/abs/1710.05012.

Ramani, Vijay, Xinxian Deng, Ruolan Qiu, Kevin L. Gunderson, Frank J. Steemers, Christine M. Disteche, William S. Noble, Zhijun Duan, and Jay Shendure. 2017. "Massively Multiplex Single-Cell Hi-C." *Nature Methods*, January. https://doi.org/10.1038/nmeth.4155.

Rand, William M. 1971. "Objective Criteria for the Evaluation of Clustering Methods." *Journal of the American Statistical Association* 66 (336): 846–50.

Sanchez-Castillo, M., D. Blanco, I. M. Tienda-Luna, M. C. Carrion, and Yufei Huang. 2017. "A Bayesian Framework for the Inference of Gene Regulatory Networks from Time and Pseudo-Time Series Data." *Bioinformatics* , September. https://doi.org/10.1093/bioinformatics/btx605.

Satija, Rahul, Jeffrey A. Farrell, David Gennert, Alexander F. Schier, and Aviv Regev. 2015. "Spatial Reconstruction of Single-Cell Gene Expression Data." *Nature Biotechnology* 33 (5): 495–502.

Schreiber, Thomas. 2000. "Measuring Information Transfer." *Physical Review Letters* 85 (2): 461–64.

Setty, Manu, Michelle D. Tadmor, Shlomit Reich-Zeliger, Omer Angel, Tomer Meir Salame, Pooja Kathail, Kristy Choi, Sean Bendall, Nir Friedman, and Dana Pe'er. 2016. "Wishbone Identifies Bifurcating Developmental Trajectories from Single-Cell Data." *Nature Biotechnology* 34 (6): 637–45.

Shalek, Alex K., Rahul Satija, Joe Shuga, John J. Trombetta, Dave Gennert, Diana Lu, Peilin Chen, et al. 2014. "Single-Cell RNA-Seq Reveals Dynamic Paracrine Control of Cellular Variation." *Nature* 510 (7505): 363–69.

Sheth, Ravi U., Sung Sun Yim, Felix L. Wu, and Harris H. Wang. 2017. "Multiplex Recording of Cellular Events over Time on CRISPR Biological Tape." *Science* 358 (6369): 1457–61.

Shipman, Seth L., Jeff Nivala, Jeffrey D. Macklis, and George M. Church. 2016. "Molecular Recordings by Directed CRISPR Spacer Acquisition." *Science* 353 (6298): aaf1175.

Stoeckius, Marlon, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K.

Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. 2017. “Simultaneous Epitope and Transcriptome Measurement in Single Cells.” *Nature Methods* 14 (9): 865–68.
Sugihara, George, Robert May, Hao Ye, Chih-Hao Hsieh, Ethan Deyle, Michael Fogarty, and Stephan Munch. 2012. “Detecting Causality in Complex Ecosystems.” *Science* 338 (6106): 496–500.
Su, Hang, Gaowei Wang, Ruoshi Yuan, Junqiang Wang, Ying Tang, Ping Ao, and Xiaomei Zhu. 2017. “Decoding Early Myelopoiesis from Dynamics of Core Endogenous Network.” *Science China. Life Sciences* 60 (6): 627–46.
Sulston, J. E., E. Schierenberg, J. G. White, and J. N. Thomson. 1983. “The Embryonic Cell Lineage of the Nematode Caenorhabditis Elegans.” *Developmental Biology* 100 (1): 64–119.
Svensson, Valentine, Roser Vento-Tormo, and Sarah A. Teichmann. 2018. “Exponential Scaling of Single-Cell RNA-Seq in the Past Decade.” *Nature Protocols* 13 (4): 599–604.
Szklarczyk, Damian, John H. Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, et al. 2017. “The STRING Database in 2017: Quality-Controlled Protein-Protein Association Networks, Made Broadly Accessible.” *Nucleic Acids Research* 45 (D1): D362–68.
Takens, Floris. 1981. “Detecting Strange Attractors in Turbulence.” In *Lecture Notes in Mathematics*, 366–81.
Tang, Fuchou, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, et al. 2009. “mRNA-Seq Whole-Transcriptome Analysis of a Single Cell.” *Nature Methods* 6 (5): 377–82.
Tang, Ying, Ruoshi Yuan, and Ping Ao. 2014. “Summing over Trajectories of Stochastic Dynamics with Multiplicative Noise.” *The Journal of Chemical Physics* 141 (4): 044125.
Tang, Ying, Ruoshi Yuan, Gaowei Wang, Xiaomei Zhu, and Ping Ao. 2016. “Potential Landscape of High Dimensional Nonlinear Stochastic Dynamics and Rare Transitions with Large Noise.” *arXiv [cond-Mat.stat-Mech]*. arXiv. http://arxiv.org/abs/1611.07140.
Teschendorff, Andrew E., and Tariq Enver. 2017. “Single-Cell Entropy for Accurate Estimation of Differentiation Potency from a Cell’s Transcriptome.” *Nature Communications* 8: 15599.
Tirosh, Itay, Andrew S. Venteicher, Christine Hebert, Leah E. Escalante, Anoop P. Patel, Keren Yizhak, Jonathan M. Fisher, et al. 2016a. “Single-Cell RNA-Seq Supports a Developmental Hierarchy in Human Oligodendroglioma.” *Nature* 539 (7628): 309–13.
———. 2016b. “Single-Cell RNA-Seq Supports a Developmental Hierarchy in Human Oligodendroglioma.” *Nature* 539 (7628): 309–13.
Trapnell, Cole, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J. Lennon, Kenneth J. Livak, Tarjei S. Mikkelsen, and John L. Rinn. 2014a. “The Dynamics and Regulators of Cell Fate Decisions Are Revealed by Pseudotemporal Ordering of Single Cells.” *Nature Biotechnology* 32 (4): 381–86.
———. 2014b. “The Dynamics and Regulators of Cell Fate Decisions Are Revealed by Pseudotemporal Ordering of Single Cells.” *Nature Biotechnology* 32 (4): 381–86.
Treutlein, Barbara, 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. 2014. “Reconstructing Lineage Hierarchies of the Distal Lung Epithelium Using Single-Cell RNA-Seq.” *Nature* 509 (7500): 371–75.
Velten, Lars, Simon F. Haas, Simon Raffel, Sandra Blaszkiewicz, Saiful Islam, Bianca P. Hennig,

Christoph Hirche, et al. 2017. “Human Haematopoietic Stem Cell Lineage Commitment Is a Continuous Process.” *Nature Cell Biology* 19 (4): 271–81.
Waddington, C. H. 2014. *The Strategy of the Genes*. Routledge.
Wang, Jin, Kun Zhang, Li Xu, and Erkang Wang. 2011. “Quantifying the Waddington Landscape and Biological Paths for Development and Differentiation.” *Proceedings of the National Academy of Sciences of the United States of America* 108 (20): 8257–62.
Wang, L., Q. Mao, and I. W. Tsang. 2017. “Latent Smooth Skeleton Embedding.” *AAAI*. aaai.org. http://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/download/14303/14465.
Wei, Jiangyong, Xiaohua Hu, Xiufen Zou, and Tianhai Tian. 2017. “Reverse-Engineering of Gene Networks for Regulating Early Blood Development from Single-Cell Measurements.” *BMC Medical Genomics* 10 (Suppl 5): 72.
Weisblat, D. A., R. T. Sawyer, and G. S. Stent. 1978. “Cell Lineage Analysis by Intracellular Injection of a Tracer Enzyme.” *Science* 202 (4374): 1295–98.
Welch, Joshua D., Alexander J. Hartemink, and Jan F. Prins. 2016. “SLICER: Inferring Branched, Nonlinear Cellular Trajectories from Single Cell RNA-Seq Data.” *Genome Biology* 17 (1): 106.
Zinyk, D. L., E. H. Mercer, E. Harris, D. J. Anderson, and A. L. Joyner. 1998. “Fate Mapping of the Mouse Midbrain-Hindbrain Constriction Using a Site-Specific Recombination System.” *Current Biology: CB* 8 (11): 665–68.

VITA

Xiaojie Qiu was raised in a small village of southern China. He then attended Changchun University of Technology where he completed an undergraduate degree in bioengineering. Afterwards he earned a Masters in bioinformatics from East China Normal University in Shanghai. During his Masters, he applied dynamic systems approaches to understand the irreversibility of cell fate transitions. After a brief stint with Dr. Sui Huang, working on simulating evolution of developmental regulatory networks, at the Institute for Systems Biology (Seattle), Xiaojie started his PhD in the Molecular and Cellular Biology program at the University of Washington. Excited by the promise of single-cell genomics, Xiaojie joined Dr. Cole Trapnell's lab in the department of Genome Sciences as his first graduate student, to develop computational methods for single-cell genomics. Xiaojie's PhD work has made a few key contributions to the field of single-cell genomics. For example, he developed the popular single-cell genomics analysis toolkit, Monocle 2, to accurately and robustly reconstruct complex developmental trajectories. He also proposed BEAM (branch expression analysis modeling), a statistical framework that identifies genes significantly diverge between different lineages and pinpoints the precise timing of lineage specification events. Recently, in close collaboration with Dr. Sreeram Kannan, he has been developing and applying information theory techniques to detect casual interactions responsible for cell fate decisions with single-cell genomics datasets.