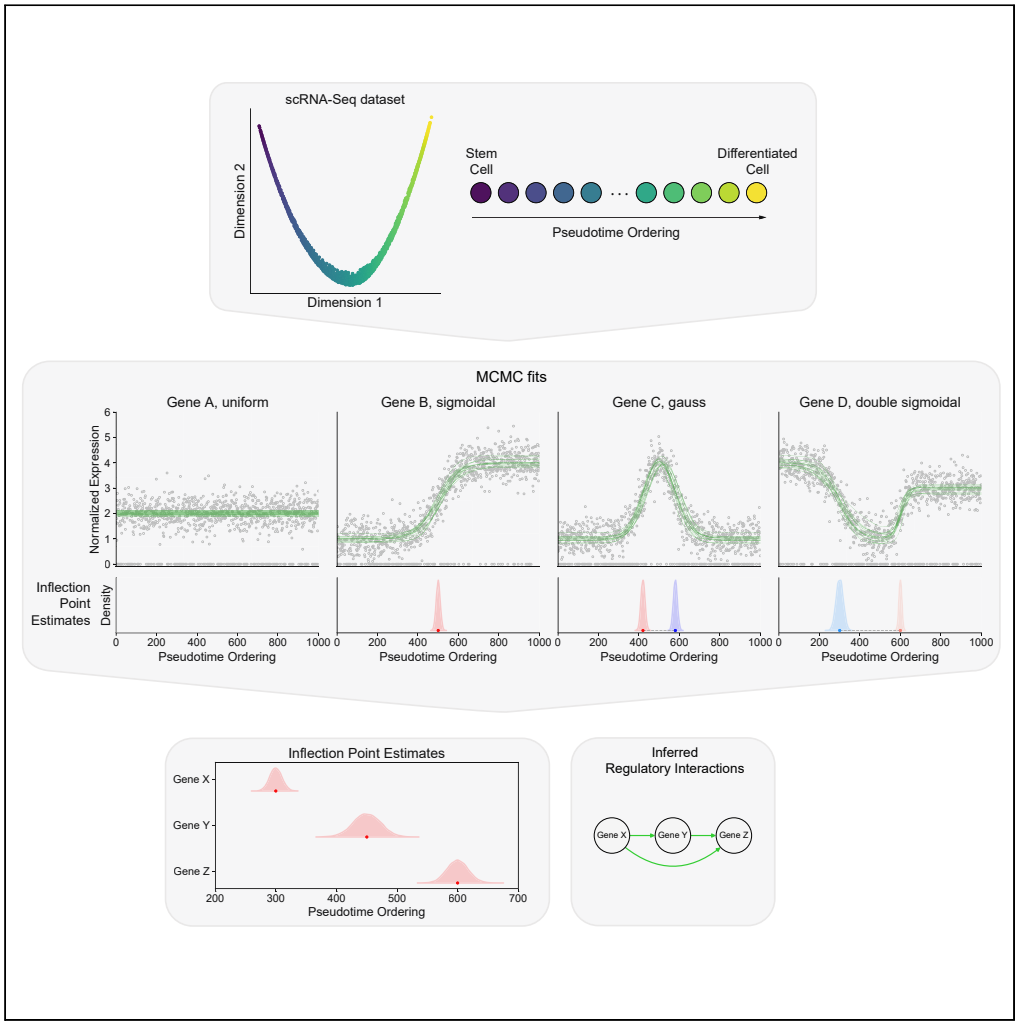
iScience

CellPress
OPEN ACCESS

Article

Modeling gene expression cascades during cell state transitions



Daniel Rosebrock, Martin Vingron, Peter F. Arndt

rosebroc@molgen.mpg.de (D.R.)
arndt@molgen.mpg.de (P.F.A.)

Highlights

Fitting pseudotime-ordered expression profiles to interpretable functional forms

Derivation of transcriptional cascades to define a pseudotime trajectory

Inference of directionality of regulatory interactions

Rosebrock et al., iScience 27, 109386
April 19, 2024
https://doi.org/10.1016/j.isci.2024.109386

iScience

CellPress
OPEN ACCESS

Article

Modeling gene expression cascades during cell state transitions

Daniel Rosebrock,[1,2,3,*] Martin Vingron,[1] and Peter F. Arndt[1,*]

SUMMARY

During cellular processes such as differentiation or response to external stimuli, cells exhibit dynamic changes in their gene expression profiles. Single-cell RNA sequencing (scRNA-seq) can be used to investigate these dynamic changes. To this end, cells are typically ordered along a pseudotemporal trajectory which recapitulates the progression of cells as they transition from one cell state to another. We infer transcriptional dynamics by modeling the gene expression profiles in pseudotemporally ordered cells using a Bayesian inference approach. This enables ordering genes along transcriptional cascades, estimating differences in the timing of gene expression dynamics, and deducing regulatory gene interactions. Here, we apply this approach to scRNA-seq datasets derived from mouse embryonic forebrain and pancreas samples. This analysis demonstrates the utility of the method to derive the ordering of gene dynamics and regulatory relationships critical for proper cellular differentiation and maturation across a variety of developmental contexts.

INTRODUCTION

Changes in gene expression underlie the intrinsic molecular processes governing differentiation, enabling cells to change their morphology and function. These changes can occur in part due to extrinsic cues from signaling molecules[1] or temperature and oxygen levels in the organism's environment,[2,3] as well as intrinsic mechanisms such as the asymmetric distribution of cellular components during cell division.[4] These processes result in modifying the expression levels of genes that are critical for cell fate specification, most importantly transcription factors, which can initiate or block the expression of downstream target genes, including other transcription factors. The sequential activation and repression of transcription factors and their target genes can give rise to a cascade of gene expression, whereby an initiating event can regulate a hierarchy of downstream genes essential for the cell to acquire subsequent cell states. For example, the *Pax6* → *Eomes* → *Tbr1* transcription factor cascade directs the progression of radial glia to intermediate progenitor to postmitotic projection neuron in the developing cortex,[5,6] and the transcription factor cascade initiated by *Neurog3* controls the differentiation of endocrine progenitor cells to mature pancreatic cells.[7,8] It is therefore critical to accurately deduce gene expression cascades in order to determine which genes are responsible for specific cell fate changes during differentiation and maturation.

Single-cell RNA sequencing (scRNA-seq) enables sampling the gene expression profile of thousands of cells in an individual sample. However, it is necessary to destroy the cell in order to measure its transcriptome, thereby making it impossible to observe how the cell and its gene expression profile would have altered in the future. Nonetheless, it is possible to order cells along a trajectory which accurately recapitulates the progression of cells as they transition from one cell state to another. This ordering of cells along a trajectory is known as pseudotime, which is essentially a mapping of single-cell transcriptomes to a developmental timeline. Pseudotime methods work under the assumption that cell state changes occur through transitional states, and that these can be measured as gradual shifts in gene expression in individual cells.[9–14]

Based on the ordering of cells along a pseudotemporal trajectory, it is possible to measure the dynamics of gene expression as cells undergo cell state transitions. Current algorithms typically model gene expression dynamics along pseudotemporal trajectories by fitting their expression profiles using generalized linear models,[12,15,16] with the ultimate goal of determining if gene expression significantly varies as a function of pseudotime. Other methods attempt to deduce pseudotime-dependent gene interactions by calculating a similarity measure between the expression levels of the "present" of one gene, and the "past" of another gene using correlation[17] or mutual information.[18] However, these methods do not calculate an explicit ordering of expression dynamics along a pseudotime trajectory, and require user-defined cutoffs for determining meaningful interactions.

Here, we present a method to better understand the cascade of gene expression dynamics underlying cell state transitions. We are interested in answering questions such as, if two genes are up-regulated during a cell state transition, is one gene up-regulated before the other, or are they up-regulated simultaneously? Furthermore, is it possible to estimate a certainty in the timing of their expression dynamics? In this paper, we address these questions by explicitly modeling gene expression over a pseudotime trajectory using a set of functions that reflect

[1]Department of Computational Molecular Biology, Max Planck Institute for Molecular Genetics, 14195 Berlin, Germany
[2]Present address: Evotec SE, Hamburg, Germany
[3]Lead contact
*Correspondence: rosebroc@molgen.mpg.de (D.R.), arndt@molgen.mpg.de (P.F.A.)
https://doi.org/10.1016/j.isci.2024.109386

biological state switches, and that model the dynamic behaviors of gene expression within cells as they differentiate. We formulate the problem using a Bayesian inference framework and use an ensemble sampler Monte Carlo Markov chain (MCMC) approach[19] to sample from the posterior distributions over the parameter spaces of the various functions, and determine which model best fits the data. This provides an explicit ordering of genes along a pseudotemporal trajectory based on inflection point estimates, enabling the description of expression dynamics in terms of transcriptional cascades, estimating differences in switch times of gene expression, and annotation of potentially causal gene interactions in gene regulatory networks.

We will introduce our modeling framework in general terms in the first section of the results. A more detailed description is provided in the STAR Methods section. We then apply our method in multiple developmental settings, in which we dissect the transcription factor cascades underlying cortical neurogenesis and pancreatic beta cell development across multiple scRNA-seq datasets. We also show how our method can be used to infer potential upstream regulators of a given gene of interest. Finally, we utilize our method to deduce the gene expression cascade of the Notch signaling pathway in the developing cortex in order to highlight the applicability of our method to gene sets beyond transcription factors. These examples demonstrate the ability of our method to accurately model the dynamics of gene expression during cell state transitions, and highlight the biological insights our method enables.

RESULTS

Modeling gene expression dynamics along pseudotime trajectories

The goal of the method presented here is to decide if a state switch (up- to down-regulation or down- to up-regulation) occurs along a pseudotemporal trajectory, and at what pseudotime these switches occur, in order to determine the timing and ordering of activation and repression during cell state transitions. In order to do this, we first define a set of functions which can model a wide variety of expression dynamics, and for which state changes are well defined and interpretable, namely at the inflection points of each function. The functions are then fit to the normalized expression levels for each gene across cells ordered by their relative pseudotempoal ordering. The functions used for fitting are defined as follows,

$$
\begin{aligned}
f_{\text{unif}}(t; b) &= b, \\
f_{\text{gauss}}(t; a, b, t_0, \sigma) &= a e^{-\frac{(t-t_0)^2}{\sigma^2}} + b, \\
f_{\text{sig}}(t; k, L, t_0, b_{min}) &= \frac{L}{1+e^{-k(t-t_0)}} + b_{min}, \\
f_{\text{dsig}}(t; k_1, k_2, t_1, t_2, b_{min}, b_{mid}, b_{max}) &= b_{min} + \frac{b_{mid} - b_{min}}{1+e^{-k_1(t-t_1)}} + \frac{b_{max} - b_{mid}}{1+e^{-k_2(t-t_2)}}.
\end{aligned}
\qquad \text{(Equation 1)}
$$

Here, f_{unif} is a uniform function with $b>0$, which models the absence of dynamics in gene expression along a pseudotime trajectory. f_{gauss} is a Gaussian function with parameter constraints $a>0,\ b>0, \sigma>0$, and $1 \le t_0 \le N$, with N = number of cells in the pseudotime trajectory. f_{sig} is a sigmoidal function with parameter constraints $L>0,\ b>0$, and $1 \le t_0 \le N$. Finally, f_{dsig} is a double sigmoidal function with the formulation described in the study by Baione et al.[20] and parameter constraints $b_{min}>0,\ b_{mid}>0,\ b_{max}>0,\ k_1>0,\ k_2>0$, and $1 \le t_1 < t_2 \le N$. The motivation for using these functions is based on observations from biological scenarios during development.[21] For instance, during differentiation, genes can display a shift from one steady state to another, which can be modeled using a sigmoidal function. They can also exhibit impulse patterns of up-regulation followed by a return to basal levels, which can be modeled using a Gaussian function. Finally, double sigmoidal functions can model impulse patterns with asymmetric increase and decrease rates and different initial and terminal basal levels, as well as stepwise up and stepwise down expression patterns (Figure S1). We formulate the problem of fitting gene expression profiles in cells ordered along a pseudotime trajectory as a Bayesian inference problem, and estimate parameters for each function using an ensemble sampler MCMC approach[19] (see STAR Methods). Based on the best-fitting function to the gene expression profiles, genes are ordered according to the relative occurrence of inflection point estimates to provide temporal estimates of gene expression cascades, and regulatory interactions between genes are deduced, enabling a detailed characterization of the molecular processes underlying cellular transitions.

Transcriptional cascades during cortical neuron differentiation

We first applied our method to differentiating forebrain dorsal neural stem cells during mouse development at embryonic stage e13.5. The input to the method consists of a set of cells ordered by pseudotime, $t = 1, \ldots, N$, and the expression levels (counts) of genes within those cells. Cells from the Atlas of the Developing Mouse Brain[22] were initially subset to non-dividing forebrain dorsal cells consisting of neural stem cells, intermediate progenitors (IPs), and neurons at embryonic stage e13.5. A pseudotime ordering was estimated using diffusion pseudotime[9] (Figure S2). All dividing cells were excluded for the pseudotime estimation due to their expression of a transcriptional program that is independent of the underlying cell type, potentially confounding pseudotime estimates.

In differentiating cells along the mouse e13.5 forebrain dorsal neural stem cell (NSC) → IP → neuron trajectory, 60 out of 510 (11.8%) transcription factors (derived from the study by Lambert et al.[23]) that were expressed in at least 1% of cells had a non-uniform fit (Figure 1; Table S1). Initially, *Gli3*, a gene that is required for maintaining cortical progenitors in active cell cycle,[24] was down-regulated in a state-switch manner with a sigmoidal fit, along with *Sox9* and *Hes1*, which are both required for neural stem cell maintenance.[25,26] Subsequently, other genes important for neural stem cell maintenance including *Sox1*, *Sox2*, *Hes5*, and *Pax6* were down-regulated. Genes exhibiting a state-switch or stepwise

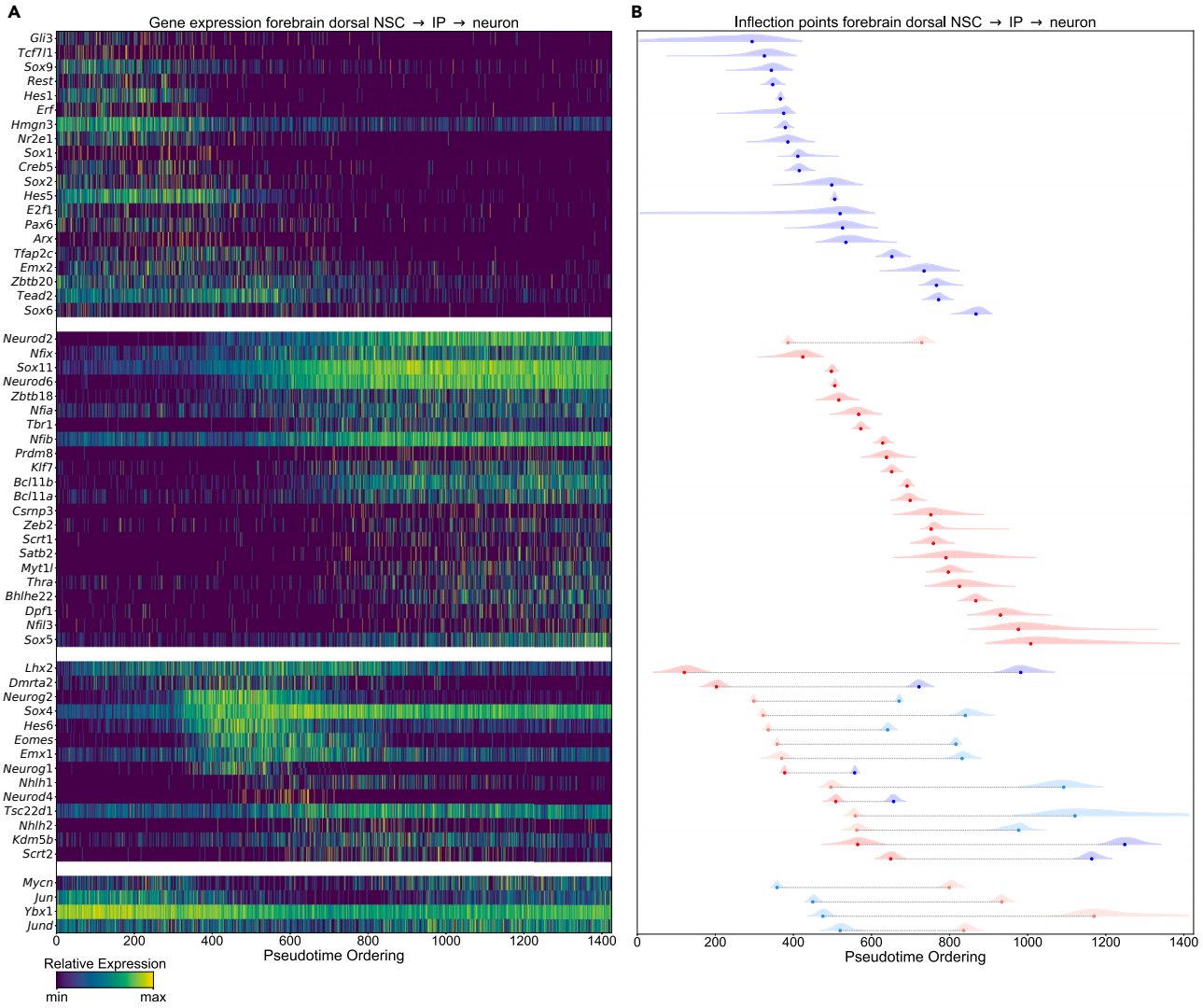


Figure 1. Transcriptional cascades in mouse 13.5 forebrain dorsal cells

(A) Gene expression profiles of transcription factors with non-uniform fits are displayed as a heatmap. Genes are grouped according to a state-switch from high to low expression (sigmoidal fit) or stepwise down-regulation (double sigmoidal fit), a state-switch from low to high expression (sigmoidal fit) or stepwise up-regulation (double sigmoidal fit), a transient up (Gaussian or double sigmoidal fit) expression pattern, and transient down (double sigmoidal fit) expression pattern. (B) The inflection point estimates are shown for the same genes as in (A). Inflection point estimates from double sigmoidal fits are shown in light blue and light red, and those from Gaussian and sigmoidal fits in blue and red.

up-regulation included *Neurod2*, *Sox11*, and *Neurod6*, which play a critical role in inducing cell-cycle arrest and neurogenic differentiation in the developing cortex,[27–29] followed by *Tbr1* and *Bcl11b*, markers of deep-layer cortical neurons generated during early cortical neurogenesis. Subsequently, *Satb2* and *Bhlhe22*, markers of upper-layer cortical neurons generated during later stages of neurogenesis,[30] were up-regulated. Interestingly, four transcription factors were found to be transiently down-regulated using a double sigmoidal fit, including *Mycn*, *Jun*, *Ybx1*, and *Jund*. Genes exhibiting a transient up-regulation (Gaussian or double sigmoidal fit) included *Hes6* and *Eomes*, markers of cortical IPs,[31] as well as *Neurog2* and *Sox4*, which are required for IP cell specification and maintenance via activation of *Eomes*.[32]

These results demonstrate that the functions which best fit the expression profiles of dynamically expressed genes (genes exhibiting a non-uniform fit) largely reflect the known biological role these genes play during differentiation. Furthermore, the relative ordering of inflection point estimates for dynamically expressed transcription factors along the mouse e13.5 forebrain dorsal NSC → IP → neuron trajectory accurately recapitulates known temporal orderings that are essential for the differentiation of cortical neurons. Finally, in order to justify the functional forms we used, we performed a PCA of the gene expression profiles. Genes with a non-uniform fit fill the extremes of the principal component space (Figure S3), indicating that the functional forms we used to model the pseudotime-ordered gene expression profiles are able to capture most of the variability in the data.

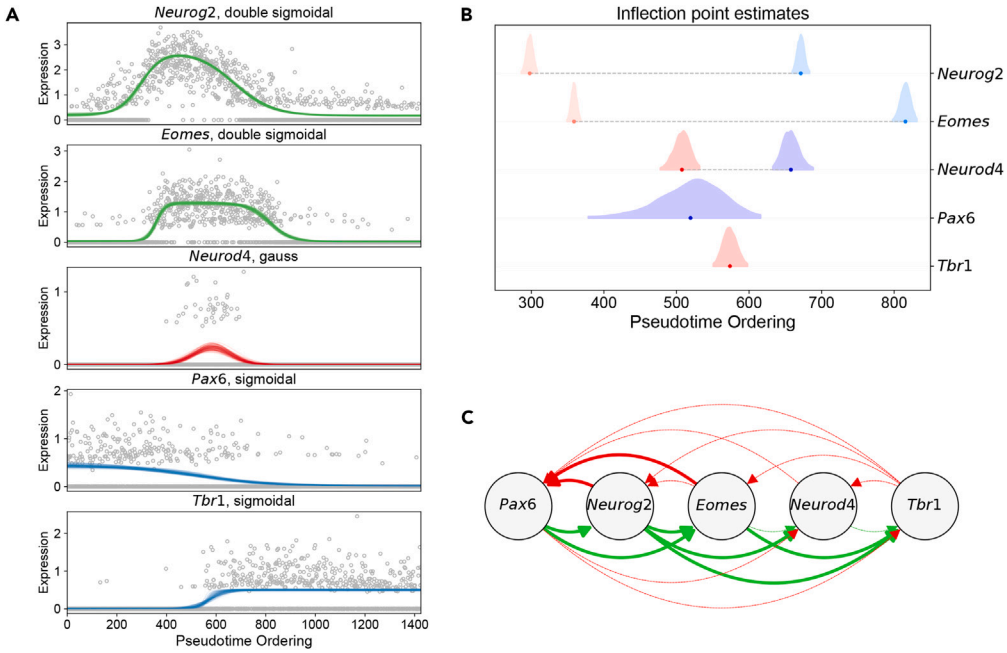


Figure 2. Reconstructing regulatory interactions during mouse e13.5 cortical development

(A) Normalized expression levels of essential genes — *Pax6*, *Neurog2*, *Eomes*, and *Tbr1* — forming a regulatory network underlying cortical neuron differentiation, as well as the neural lineage bHLH factor, *Neurod4*, across pseudotime-ordered cells are shown. The curves display a random sampling of the parameters from 100 iterations of the MCMC traces for the best-fitting model for each gene.

(B) Inflection point estimates for the genes highlighted in (A).

(C) A reconstructed gene regulatory network based on the comparison of inflection points. Positive regulatory interactions which have previously been validated are highlighted as a green solid line, and those which have not been validated as a green dashed line. Similarly, negative regulatory interactions which have previously been validated are highlighted as a red solid line, and those which have not been validated as a red dashed line.

Constructing regulatory interactions during cortical neurogenesis

We then compared a set of transcription factors forming an essential regulatory network underlying cortical neuronal differentiation including *Pax6*, *Neurog2*, *Eomes*, and *Tbr1*,[33] as well as the neural lineage bHLH factor, *Neurod4* (Figure 2A). *Neurog2* and *Eomes* exhibited a transient up-regulation, with both genes having a double sigmoidal fit. *Pax6* and *Tbr1* were fit using a sigmoidal function, with *Pax6* exhibiting a state-switch from high to low expression, and *Tbr1* from low to high expression. *Neurod4* was fit using a Gaussian function, and was specifically expressed transiently in mid-stage *Eomes*$^{+}$ cells.

These genes were then ordered according to the pseudotemporal occurrence of inflection point estimates (Figure 2B), whereby *Neurog2* was found to be up-regulated before *Eomes*, followed by the up-regulation of *Neurod4* and down-regulation of *Pax6*. Subsequently, *Tbr1* was up-regulated, followed by down-regulation of *Neurod4*, *Neurog2*, and finally *Eomes*. *Neurod4* exhibited a brief, transient impulse expression pattern within mid-stage *Eomes*$^{+}$ cells, reflecting previously studied expression patterns of *Neurod4*, which is only expressed in a subset of *Eomes*$^{+}$ cells in the mouse e14.5 cortex.[34]

By comparing inflection point estimates of these genes (see STAR Methods), we were able to reconstruct previously validated regulatory interactions (Figure 2C). The initial up-regulation of *Neurog2* just before *Eomes* up-regulation suggests that *Neurog2* initiates expression of *Eomes* in intermediate progenitors. This relationship has been shown in mouse e13 embryos via electroporation of *Neurog2* cDNA into the ganglionic eminence, where both *Neurog2* and *Eomes* are not expressed, resulting in ectopic expression of *Eomes*.[35] *Neurog2* has also been shown to directly activate *Neurod4* in cortical IP cells using a luciferase reporter assay,[36] which we also recapitulate based on the sequential up-regulation of *Neurog2* and *Neurod4*. Furthemore, it has been shown that both *Neurog2* and *Eomes* induce *Tbr1* expression,[36] which we also infer based on the up-regulation of *Tbr1* following both *Neurog2* and *Eomes*. Interestingly, directly after *Eomes* and *Neurog2* were up-regulated, *Pax6* was down-regulated, suggesting a negative feedback loop, whereby *Pax6* activates both *Eomes* and *Neurog2*, which then both in turn repress *Pax6*, a relationship which has been previously described in the developing mouse cortex.[37]

Inferring shared upstream regulators of *Eomes*

We next explored potential upstream regulators of *Eomes* in mouse e13.5 forebrain dorsal cells across two samples in order to deduce high confidence regulators of *Eomes* and determine how robust our method is across biological replicates. We applied our method to forebrain dorsal cells in a mouse e13.5 biological replicate (Figure S4; Table S2). Transcription factors with a positive inflection point occurring simultaneously with or before the first inflection point of *Eomes*, as well as those with a negative inflection point occurring after the first inflection

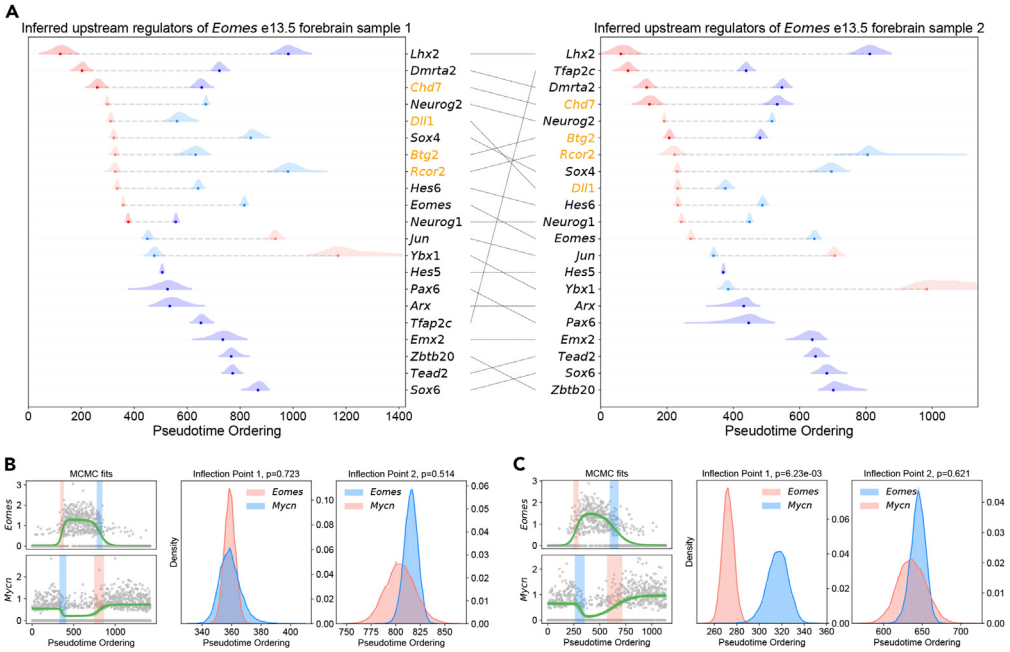


Figure 3. Inferring upstream regulators of *Eomes* across mouse e13.5 embryos

(A) The left and right plots show a transcriptional cascade of the shared potential positive regulators of *Eomes* in forebrain dorsal cells of mouse e13.5 embryos across biological replicates. Transcriptional co-activators and co-repressors (derived from the study by Siddappa et al.[38]) are shown in orange, and transcription factors (derived from the study by Lambert et al.[23]) are shown in black.

(B) The left panel in the plot displays a random sampling of the parameters from 100 iterations of the MCMC traces for the genes *Eomes* and *Mycn* using the double sigmoidal model, the best-fitting model for both genes. The full range of first and second inflection point estimates for both genes is highlighted as a shaded region, with blue indicating a negative inflection point and red a positive inflection point. The middle and right panels highlight the distribution of first and second inflection point estimates across MCMC iterations, respectively. The right panel highlights the distribution of second inflection point estimates across MCMC iterations. p values were estimated as the percentage of overlapping inflection point estimates across both genes after binning the inflection point estimates across all MCMC iterations to 100 equally spaced bins, starting at the minimum inflection point estimate and ending at the maximum inflection point estimate across both genes.

(C) The same plot for (B) in cortical cells of the biological replicate.

point of *Eomes*, were labeled as positive upstream regulators. We furthermore included all co-activators and co-repressors (derived from the study by Siddappa et al.[38]) that exhibited a transient up-regulation, with the first inflection point occurring simultaneously with or before the first inflection point of *Eomes*. In total, 25 positive upstream regulators were found in the first sample, and 27 were found in the second sample, with an overlap of 21 genes across the two (Figure 3A; Figure S5). Furthermore, the relative ordering of inflection points of these genes along the cortical differentiation trajectory strongly agrees across both datasets, with one exception being *Tfap2c*, which was fit to a sigmoidal function in the first sample, and Gaussian function in the second sample.

Within the set of inferred transcription factors regulating *Eomes* expression were *Neurog2* and *Pax6*, which are known to directly activate *Eomes* in the developing mouse neocortex, as described in the previous section. The co-regulators *Dll1*, a key ligand for activating Notch signaling, and *Chd7*, a chromatin remodeler, have also been implicated in the formation of IP cells,[39,40] although their role as a co-activator of *Eomes* has not been established to our knowledge. These results validate the utility of our method in discovering upstream regulators of a given gene of interest. The remaining potential activators of *Eomes* warrant further experimental validation.

Furthermore, the genes that repress *Eomes* in maturing IP cells, thereby enabling the differentiation of these cell types into neurons, are largely unknown.[33] The transcription factor *Mycn*, a gene critical for normal brain development,[41] has been shown to down-regulate *Eomes* in neuroblastoma cell lines[42]; however, its role in regulating *Eomes* expression in maturing IP cells is not well understood. In differentiating cells along the forebrain dorsal NSC → IP → neuron trajectory in both mouse e13.5 samples, *Mycn* was expressed in a transient down-regulation pattern and best fit using a double sigmoidal function (Figures 3B and 3C). In both samples, *Mycn* up-regulation occurred simultaneously with *Eomes* down-regulation, signifying that *Mycn* may play a role in the differentiation of cortical neurons by down-regulating *Eomes* in maturing IPs.

Dissecting Notch signaling during cortical neurogenesis

To demonstrate the applicability of our method to genes beyond transcription factors, we investigated the dynamics of Notch signaling along the forebrain dorsal NSC → IP → neuron trajectory in e13.5 mouse embryos. Shared dynamically expressed genes involving ligand-receptor pairs of Notch receptors from the study by Shao et al.[43] in both embryonic samples were estimated (Figure 4).

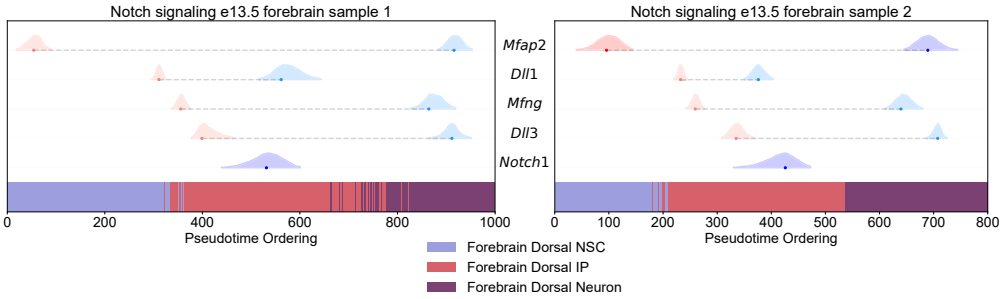


Figure 4. Notch signaling cascade in mouse e13.5 embryos

The left and right plots show a transcriptional cascade of the shared ligand-receptor pairs involved in Notch signaling in cells along the forebrain dorsal NSC → IP → neuron trajectories in mouse e13.5 embryos across biological replicates. Annotated cell types are highlighted below.

In both samples, *Mfap2*, which can interact with the extracellular domain of *Notch1*,[44] however whose role is poorly understood in the regulation and differentiation of cortical NSCs, was up-regulated within forebrain dorsal NSCs, and down-regulated in neuronal cells. This indicates that *Mfap2* may play a general role in Notch signaling within differentiating cortical NSCs, whose actions are not specific to a given cell type. *Dll1* was up-regulated in early IPs, followed by the up-regulation of *Dll3* in later stage IPs, confirming the selective basal expression of *Dll3* from *in vivo* studies.[33] Furthermore, *Mfng*, a glycosyltransferase which increases the ability of *Notch1* to bind to *Dll1*,[45] was up-regulated shortly after *Dll1* up-regulation in both samples within IPs, indicating that this gene becomes activated sequentially after the activation of *Dll1*. *Dll1* was then down-regulated within IPs, suggesting that this gene is not essential for further IP differentiation into neurons. Finally, *Notch1* was down-regulated in maturing IPs, followed by down-regulation of *Mfap2*, *Mfng*, and *Dll3* in neurons. These results highlight the ability of our method to dissect the complex dynamics of signaling pathways within differentiating cell types.

Transcriptional cascades in mouse pancreatic beta cell development

To demonstrate the utility of our method in other developmental contexts, we applied our method to a scRNA-seq dataset of pancreatic cells derived from mouse e14.5 embryos,[46] subsetting to cells belonging to the beta cell lineage. When measuring the expression dynamics of a set of genes known to play an essential role in the specification and maturation of pancreatic beta cells,[8] we find a well-defined transcriptional cascade which largely agrees with previously characterized gene expression cascades (Figure 5A). Interestingly, we find one exception to this cascade, *Neurod1*, which is up-regulated at a later stage of beta cell maturation than previously reported (Figures 5B and 5C). We are also able to measure the sequential up-regulation of *Pax6* and *Pdx1*, followed by *Mnx1*, and ending with the insulin gene expression regulator *Isl1*, thereby providing a more explicit ordering of the expression cascade in maturing beta cells than previously established. Furthermore, with this approach, we can model the expression dynamics of all transcription factors (Figure S6; Table S3), enabling a detailed overview of the full gene expression cascade underlying pancreatic beta cell differentiation.

DISCUSSION

In this paper, we explored an approach to model the gene expression dynamics in cells ordered by a pseudotime trajectory using a fully Bayesian framework. This framework enabled us to fit the gene expression profiles of cells undergoing cell state transitions to a set of functions that are able to model complex transcriptional dynamics. From these fits, we were able to order genes along a gene expression cascade which describes the molecular dynamics underlying cell state transitions, and deduce regulatory interactions.

We first applied the method to differentiating forebrain dorsal neural stem cells into neurons in mouse e13.5 embryos. By ordering transcription factors by the relative occurrence of inflection point estimates, we were able to reconstruct the transcriptional cascades underlying neuronal differentiation within the developing cortex, and model the dynamics of gene expression for all genes along the trajectory. However, genes can undergo further dynamic changes including post-transcriptional and post-translational modifications, and localization changes within the cell, all of which can have a large impact on function and regulation. While transcriptomics data are unable to identify these changes, the dynamics we uncover from gene expression data can still shed light on their regulatory roles.

By comparing the relative timing of expression dynamics of the transcription factors *Pax6*, *Neurog2*, *Eomes*, *Neurod4*, and *Tbr1*, which form a regulatory network underlying cortical neuron differentiation, we were able to infer known causal interactions. However, reconstructing a gene regulatory network using all genes with a non-uniform fit would lead to many false positives, in part due to the simultaneous activation of multiple pathways involving different genes. Thus, we believe one of the main utilities of our approach is to infer the directionality of regulatory interactions, especially in cases where an interaction has been measured but the directionality is unknown.

We then identified potential upstream positive regulators of *Eomes*, an essential gene for the formation of IPs. Subsetting to genes which have similar dynamics across biological replicates revealed a set of high-confidence potential upstream regulators. Not only did we recover validated activators of *Eomes*, such as *Pax6* and *Neurog2*, but we also detected a number of other transcription factors whose roles in *Eomes* activation have not been fully characterized. The enrichment of known DNA-binding motifs of these transcription factors in the promoter and

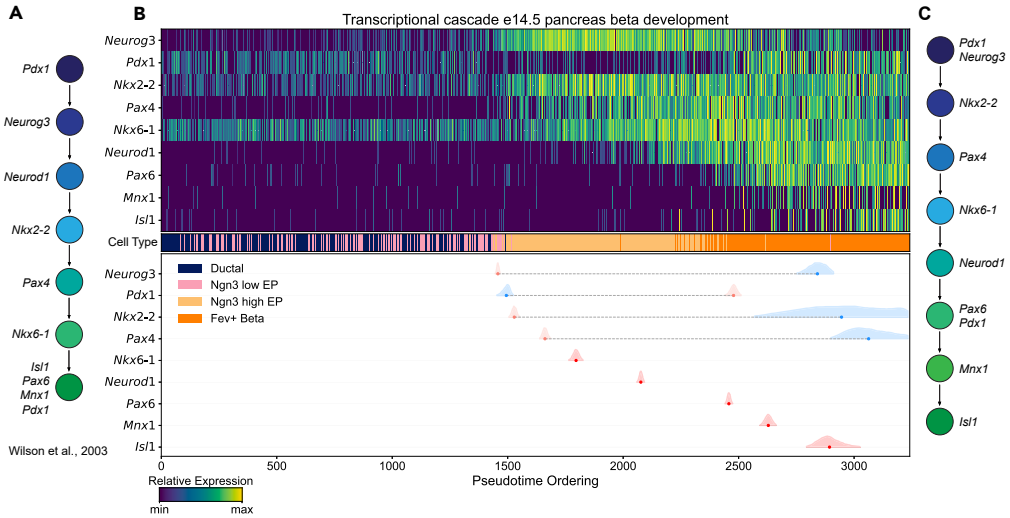


Figure 5. Gene expression cascades in developing mouse e14.5 pancreatic beta cells

(A) Schematic diagram of the previously characterized gene expression cascade in developing pancreatic beta cells, based on the study by Wilson et al.[8]

(B) The heatmap in the upper panel highlights the expression profiles of transcription factors ordered by the occurrence of their first inflection points. Inflection point estimates are highlighted in the plot below using the same ordering, with double sigmoidal fits shown in light blue and light red, and those from Gaussian and sigmoidal fits in blue and red. The annotated cell type for each cell in the trajectory is highlighted in the middle.

(C) Modified gene expression cascade based on inflection point estimates from (B).

enhancer regions of *Eomes* may provide further evidence for the regulatory role of these genes in *Eomes* expression. We also identified a potential negative regulator of *Eomes*, the transcription factor *Mycn*, whose role in cortical IP maturation has not been fully explored. Wet lab experiments, such as knockin or knockout experiments, or chromatin immunoprecipitation sequencing experiments, would need to be performed in order to validate the roles of these transcription factors in the regulation of *Eomes* expression.

We further demonstrated the applicability of our method to genes beyond transcription factors by comparing the expression dynamics of genes involved in the Notch signaling pathway. This analysis revealed a sequential up-regulation of the Notch receptor ligand *Dll1* in early IPs, followed by *Mfng*, and finally *Dll3* in maturing IPs. This activation cascade supported the selective expression of *Dll1* and *Dll3* in apical and basal IPs, respectively, further demonstrating the utility of comparing genes according to inflection point estimates to dissect signaling pathways.

We also applied our method to differentiating pancreatic beta cells in mouse e14.5 embryos. Based on this analysis, we were able to reconstruct a gene expression cascade that defines beta cell maturation. In this analysis, we highlighted a gene that deviated from the established literature, *Neurod1*, whose up-regulation along the cascade occurred later during beta cell development than previously established. Follow-up experiments are needed to validate these findings.

In order to place our method in a broader context, we compared our results with Monocle 3[12] and tradeSeq,[16] which perform statistical tests to determine if a gene is differentially expressed along a pseudotime trajectory, in cells from the e13.5 forebrain dorsal NSC → IP → neuron trajectory. While the overwhelming majority of genes with a non-uniform fit from our method were also found to be significantly differentially expressed by these two methods, both methods detected at least six times more genes to be significant compared to our method (Figure S7). Thus, we conclude that our method is more stringent in detecting genes exhibiting dynamic changes along a trajectory. Furthermore, while the relative ordering of gene expression dynamics along a trajectory is not readily available using these two methods, we are able to explicitly infer this using our method based on inflection point estimates. Similar to our method, the authors of the original diffusion pseudotime publication used derivative estimates of smoothed gene expression profiles to order gene dynamics along a pseudotime trajectory.[9] However, the authors only used derivative estimates to measure switch-like transitions and not transient up or down transitions, and only provide point estimates of these transitions. We are able to model a higher variety of transitions, and based on the MCMC samplings, quantify the uncertainty in the timing of these transitions using the posterior distribution of the parameter fits.

To measure the dependence of our method on the pseudotime method used to order cells, we ran our method on the pseudotime-ordered cells from the e13.5 forebrain dorsal NSC → IP → neuron trajectory using both Slingshot[10] and Monocle 3,[12] and compared them with the diffusion pseudotime estimates (Figure S8). Overall, the fits were largely consistent independent of the pseudotime method used to order the cells, indicating that our method is robust to fluctuations in pseudotime estimates and underlying pseudotime method.

While we focused specifically on cells along the forebrain dorsal NSC → IP → neuron trajectory, and pancreatic beta cell development, the method presented in this paper can be applied to any scRNA-seq dataset where cells can be ordered along a pseudotime trajectory. Our method is able to reconstruct transcriptional cascades in order to deduce critical genes for cell state transitions. It is also able to predict regulatory interactions, as well as gene interactions involved in different signaling pathways. Therefore, we believe this approach can provide useful insights into the molecular underpinnings involved in a variety of developmental biology contexts.

Limitations of the study

We do not perform any experiments to validate the derived regulatory interactions from differentiating mouse e13.5 forebrain dorsal neurons. Furthermore, deriving regulatory interactions based on all genes with a non-uniform fit along a trajectory would lead to many false positive interactions. Therefore, incorporating other databases and/or scATAC-seq datasets to measure the enrichment of DNA-binding motifs of a transcription factor in the promoter or enhancer regions of an inferred target would provide more evidence of the interaction, which we plan to incorporate in future research.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Processing scRNA-Seq of mouse e13.5 forebrain dorsal samples
  - Processing scRNA-Seq of mouse e14.5 pancreas development samples
  - Establishing a likelihood model
  - Model inference using MCMC
  - Model selection
  - MCMC diagnostics
  - Estimating inflection points
  - Comparing inflection points
  - Running Monocle 3, tradeSeq and Slingshot on mouse e13.5 forebrain dorsal cells

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2024.109386.

ACKNOWLEDGMENTS

We thank the IMPRS-CBSC doctoral program for their financial support. We also thank the IT group of the Max Planck Institute for Molecular Genetics for providing in-house computing infrastructure and support. Finally, we would like to thank Yechiel Elkabetz for introducing us to the topic of cell state transitions in cortical development.

AUTHOR CONTRIBUTIONS

D.R.: conceptualization, data curation, formal analysis, visualization, methodology, and writing. M.V.: conceptualization, supervision, methodology, resources, and writing – review and editing. P.F.A.: conceptualization, supervision, methodology, resources, and writing – review and editing.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: July 18, 2023
Revised: December 14, 2023
Accepted: February 27, 2024
Published: March 4, 2024

REFERENCES

1. Gilbert, S.F. (2009). Developmental Biology (Sinauer Associates).
2. Wang, X., Ni, L., Wan, S., Zhao, X., Ding, X., Dejean, A., and Dong, C. (2020). Febrile Temperature Critically Controls the Differentiation and Pathogenicity of T Helper 17 Cells. Immunity *52*, 328–341.e5. https://doi.org/10.1016/j.immuni.2020.01.006.
3. Holzwarth, C., Vaegler, M., Gieseke, F., Pfister, S.M., Handgretinger, R., Kerst, G., and Müller, I. (2010). Low physiologic oxygen tensions reduce proliferation and differentiation of human multipotent mesenchymal stromal cells. BMC Cell Biol. *11*, 11. https://doi.org/10.1186/1471-2121-11-11.
4. Morrison, S.J., and Kimble, J. (2006). Asymmetric and symmetric stem-cell divisions in development and cancer. Nature *441*, 1068–1074. https://doi.org/10.1038/nature04956.
5. Englund, C., Fink, A., Lau, C., Pham, D., Daza, R.A.M., Bulfone, A., Kowalczyk, T., and Hevner, R.F. (2005). Pax6, Tbr2, and Tbr1 are expressed sequentially by radial glia, intermediate progenitor cells, and postmitotic neurons in developing

neocortex. J. Neurosci. *25*, 247–251. https://doi.org/10.1523/JNEUROSCI.2899-04.2005.

6. Elsen, G.E., Hodge, R.D., Bedogni, F., Daza, R.A.M., Nelson, B.R., Shiba, N., Reiner, S.L., and Hevner, R.F. (2013). The protomap is propagated to cortical plate neurons through an Eomes-dependent intermediate map. Proc. Natl. Acad. Sci. USA *110*, 4081–4086. https://doi.org/10.1073/pnas.1209076110.
7. Gradwohl, G., Dierich, A., LeMeur, M., and Guillemot, F. (2000). neurogenin3 is required for the development of the four endocrine cell lineages of the pancreas. Proc. Natl. Acad. Sci. USA *97*, 1607–1611. https://doi.org/10.1073/pnas.97.4.1607.
8. Wilson, M.E., Scheel, D., and German, M.S. (2003). Gene expression cascades in pancreatic development. Mech. Dev. *120*, 65–80. https://doi.org/10.1016/S0925-4773(02)00333-7.
9. Haghverdi, L., Büttner, M., Wolf, F.A., Buettner, F., and Theis, F.J. (2016). Diffusion pseudotime robustly reconstructs lineage branching. Nat. Methods *13*, 845–848. https://doi.org/10.1038/nmeth.3971.
10. Street, K., Risso, D., Fletcher, R.B., Das, D., Ngai, J., Yosef, N., Purdom, E., and Dudoit, S. (2018). Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. BMC Genom. *19*, 477. https://doi.org/10.1186/s12864-018-4772-0.
11. Setty, M., Kiseliovas, V., Levine, J., Gayoso, A., Mazutis, L., and Pe'er, D. (2019). Characterization of cell fate probabilities in single-cell data with Palantir. Nat. Biotechnol. *37*, 451–460. https://doi.org/10.1038/s41587-019-0068-4.
12. Cao, J., Spielmann, M., Qiu, X., Huang, X., Ibrahim, D.M., Hill, A.J., Zhang, F., Mundlos, S., Christiansen, L., Steemers, F.J., et al. (2019). The single-cell transcriptional landscape of mammalian organogenesis. Nature *566*, 496–502. https://doi.org/10.1038/s41586-019-0969-x.
13. Campbell, K.R., and Yau, C. (2019). A descriptive marker gene approach to single-cell pseudotime inference. Bioinformatics *35*, 28–35. https://doi.org/10.1093/bioinformatics/bty498.
14. Lange, M., Bergen, V., Klein, M., Setty, M., Reuter, B., Bakhti, M., Lickert, H., Ansari, M., Schniering, J., Schiller, H.B., et al. (2022). CellRank for directed single-cell fate mapping. Nat. Methods *19*, 159–170. https://doi.org/10.1038/s41592-021-01346-6.
15. Ji, Z., and Ji, H. (2016). TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. Nucleic Acids Res. *44*, 117. https://doi.org/10.1093/nar/gkw430.
16. Van den Berge, K., Roux de Bézieux, H., Street, K., Saelens, W., Cannoodt, R., Saeys, Y., Dudoit, S., and Clement, L. (2020). Trajectory-based differential expression analysis for single-cell sequencing data. Nat. Commun. *11*, 1201. https://doi.org/10.1038/s41467-020-14766-3.
17. Specht, A.T., and Li, J. (2017). LEAP: constructing gene co-expression networks for single-cell RNA-sequencing data using pseudotime ordering. Bioinformatics *33*, 764–766. https://doi.org/10.1093/bioinformatics/btw729.
18. Qiu, X., Rahimzamani, A., Wang, L., Ren, B., Mao, Q., Durham, T., McFaline-Figueroa, J.L., Saunders, L., Trapnell, C., and Kannan, S. (2020). Inferring Causal Gene Regulatory Networks from Coupled Single-Cell Expression Dynamics Using Scribe. Cell Syst. *10*, 265–274.e11. https://doi.org/10.1016/j.cels.2020.02.003.
19. Goodman, J., and Weare, J. (2010). Ensemble samplers with affine invariance. Commun. Appl. Math. Comput. Sci. *5*, 65–80. https://doi.org/10.2140/camcos.2010.5.65.
20. Baione, F., Biancalana, D., and De Angelis, P. (2021). An application of Sigmoid and Double-Sigmoid functions for dynamic policyholder behaviour. Decis. Econ. Finance *44*, 5–22. https://doi.org/10.1007/s10203-020-00279-7.
21. Bar-Joseph, Z., Gitter, A., and Simon, I. (2012). Studying and modelling dynamic biological processes using time-series gene expression data. Nat. Rev. Genet. *13*, 552–564. https://doi.org/10.1038/nrg3244.
22. La Manno, G., Siletti, K., Furlan, A., Gyllborg, D., Vinsland, E., Mossi Albiach, A., Mattsson Langseth, C., Khven, I., Lederer, A.R., Dratva, L.M., et al. (2021). Molecular architecture of the developing mouse brain. Nature *596*, 92–96. https://doi.org/10.1038/s41586-021-03775-x.
23. Lambert, S.A., Jolma, A., Campitelli, L.F., Das, P.K., Yin, Y., Albu, M., Chen, X., Taipale, J., Hughes, T.R., and Weirauch, M.T. (2018). The Human Transcription Factors. Cell *172*, 650–665. https://doi.org/10.1016/j.cell.2018.01.029.
24. Wang, H., Ge, G., Uchida, Y., Luu, B., and Ahn, S. (2011). Gli3 Is Required for Maintenance and Fate Specification of Cortical Progenitors. J. Neurosci. *31*, 6440–6448. https://doi.org/10.1523/JNEUROSCI.4892-10.2011.
25. Scott, C.E., Wynn, S.L., Sesay, A., Cruz, C., Cheung, M., Gomez Gaviro, M.V., Booth, S., Gao, B., Cheah, K.S.E., Lovell-Badge, R., and Briscoe, J. (2010). SOX9 induces and maintains neural stem cells. Nat. Neurosci. *13*, 1181–1189. https://doi.org/10.1038/nn.2646.
26. Kageyama, R., Ohtsuka, T., and Kobayashi, T. (2008). Roles of Hes genes in neural development: Hes genes in neural development. Dev. Growth Differ. *50*, 97–103. https://doi.org/10.1111/j.1440-169X.2008.00993.x.
27. Olson, J.M., Asakura, A., Snider, L., Hawkes, R., Strand, A., Stoeck, J., Hallahan, A., Pritchard, J., and Tapscott, S.J. (2001). NeuroD2 Is Necessary for Development and Survival of Central Nervous System Neurons. Dev. Biol. *234*, 174–187. https://doi.org/10.1006/dbio.2001.0245.
28. Bergsland, M., Werme, M., Malewicz, M., Perlmann, T., and Muhr, J. (2006). The establishment of neuronal properties is controlled by Sox4 and Sox11. Genes Dev. *20*, 3475–3486. https://doi.org/10.1101/gad.403406.
29. Uittenbogaard, M., and Chiaramello, A. (2002). Constitutive overexpression of the basic helix-loop-helix Nex1/MATH-2 transcription factor promotes neuronal differentiation of PC12 cells and neurite regeneration. J. Neurosci. Res. *67*, 235–245. https://doi.org/10.1002/jnr.10119.
30. Molyneaux, B.J., Arlotta, P., Menezes, J.R.L., and Macklis, J.D. (2007). Neuronal subtype specification in the cerebral cortex. Nat. Rev. Neurosci. *8*, 427–437. https://doi.org/10.1038/nrn2151.
31. Bedogni, F., and Hevner, R.F. (2021). Cell-Type-Specific Gene Expression in Developing Mouse Neocortex: Intermediate Progenitors Implicated in Axon Development. Front. Mol. Neurosci. *14*, 686034. https://doi.org/10.3389/fnmol.2021.686034.
32. Chen, C., Lee, G.A., Pourmorady, A., Sock, E., and Donoghue, M.J. (2015). Orchestration of Neuronal Differentiation and Progenitor Pool Expansion in the Developing Cortex by SoxC Genes. J. Neurosci. *35*, 10629–10642. https://doi.org/10.1523/JNEUROSCI.1663-15.2015.
33. Hevner, R.F. (2019). Intermediate progenitors and Tbr2 in cortical development. J. Anat. *235*, 616–625. https://doi.org/10.1111/joa.12939.
34. Li, Z., Tyler, W.A., Zeldich, E., Santpere Baró, G., Okamoto, M., Gao, T., Li, M., Sestan, N., and Haydar, T.F. (2020). Transcriptional priming as a conserved mechanism of lineage diversification in the developing mouse and human neocortex. Sci. Adv. *6*, eabd2068. https://doi.org/10.1126/sciadv.abd2068.
35. Ochiai, W., Nakatani, S., Takahara, T., Kainuma, M., Masaoka, M., Minobe, S., Namihira, M., Nakashima, K., Sakakibara, A., Ogawa, M., and Miyata, T. (2009). Periventricular notch activation and asymmetric Ngn2 and Tbr2 expression in pair-generated neocortical daughter cells. Mol. Cell. Neurosci. *40*, 225–233. https://doi.org/10.1016/j.mcn.2008.10.007.
36. Sessa, A., Ciabatti, E., Drechsel, D., Massimino, L., Colasante, G., Giannelli, S., Satoh, T., Akira, S., Guillemot, F., and Broccoli, V. (2017). The Tbr2 Molecular Network Controls Cortical Neuronal Differentiation Through Complementary Genetic and Epigenetic Pathways. Cerebr. Cortex *27*, 3378–3396. https://doi.org/10.1093/cercor/bhw270.
37. Kovach, C., Dixit, R., Li, S., Mattar, P., Wilkinson, G., Elsen, G.E., Kurrasch, D.M., Hevner, R.F., and Schuurmans, C. (2013). Neurog2 Simultaneously Activates and Represses Alternative Gene Expression Programs in the Developing Neocortex. Cerebr. Cortex *23*, 1884–1900. https://doi.org/10.1093/cercor/bhs176.
38. Siddappa, M., Wani, S.A., Long, M.D., Leach, D.A., Mathé, E.A., Bevan, C.L., and Campbell, M.J. (2020). Identification of transcription factor co-regulators that drive prostate cancer progression. Sci. Rep. *10*, 20332. en. In. https://doi.org/10.1038/s41598-020-77055-5.
39. Nelson, B.R., Hodge, R.D., Bedogni, F., and Hevner, R.F. (2013). Dynamic Interactions between Intermediate Neurogenic Progenitors and Radial Glia in Embryonic Mouse Neocortex: Potential Role in Dll1-Notch Signaling. J. Neurosci. *33*, 9122–9139. https://doi.org/10.1523/JNEUROSCI.0791-13.2013.
40. Ohta, S., Yaguchi, T., Okuno, H., Chneiweiss, H., Kawakami, Y., and Okano, H. (2016). CHD7 promotes proliferation of neural stem cells mediated by MIF. Mol. Brain *9*, 96. https://doi.org/10.1186/s13041-016-0275-6.
41. Knoepfler, P.S., Cheng, P.F., and Eisenman, R.N. (2002). N-myc is essential during neurogenesis for the rapid expansion of progenitor cell populations and the inhibition of neuronal differentiation. Genes Dev. *16*, 2699–2712. https://doi.org/10.1101/gad.1021202.
42. Hsu, C.L., Chang, H.Y., Chang, J.Y., Hsu, W.M., Huang, H.C., and Juan, H.F. (2016). Unveiling MYCN regulatory networks in neuroblastoma via integrative analysis of heterogeneous genomics data. Oncotarget *7*, 36293–36310. https://doi.org/10.18632/oncotarget.9202.

43. Shao, X., Liao, J., Li, C., Lu, X., Cheng, J., and Fan, X. (2021). CellTalkDB: a manually curated database of ligand–receptor interactions in humans and mice. Briefings Bioinf. *22*, bbaa269. https://doi.org/10.1093/bib/bbaa269.

44. Miyamoto, A., Lau, R., Hein, P.W., Shipley, J.M., and Weinmaster, G. (2006). Microfibrillar Proteins MAGP-1 and MAGP-2 Induce Notch1 Extracellular Domain Dissociation and Receptor Activation. J. Biol. Chem. *281*, 10089–10097. https://doi.org/10.1074/jbc.M600298200.

45. Moloney, D.J., Panin, V.M., Johnston, S.H., Chen, J., Shao, L., Wilson, R., Wang, Y., Stanley, P., Irvine, K.D., Haltiwanger, R.S., and Vogt, T.F. (2000). Fringe is a glycosyltransferase that modifies Notch. Nature *406*, 369–375. https://doi.org/10.1038/35019000.

46. Bastidas-Ponce, A., Tritschler, S., Dony, L., Scheibner, K., Tarquis-Medina, M., Salinno, C., Schirge, S., Burtscher, I., Böttcher, A., Theis, F.J., et al. (2019). Massive single-cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. Development *146*, dev.173849. https://doi.org/10.1242/dev.173849.

47. Wolf, F.A., Angerer, P., and Theis, F.J. (2018). SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. *19*, 15. https://doi.org/10.1186/s13059-017-1382-0.

48. McGinnis, C.S., Murrow, L.M., and Gartner, Z.J. (2019). DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors. Cell Syst. *8*, 329–337.e4. https://doi.org/10.1016/j.cels.2019.03.003.

49. McInnes, L., Healy, J., and Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. Preprint at arXiv. https://doi.org/10.48550/ARXIV.1802.03426.

50. Blondel, V.D., Guillaume, J.L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. J. Stat. Mech. *2008*, 10008. https://doi.org/10.1088/1742-5468/2008/10/P10008.

51. Svensson, V. (2020). Droplet scRNA-seq is not zero-inflated. Nat. Biotechnol. *38*, 147–150. https://doi.org/10.1038/s41587-019-0379-5.

52. Lause, J., Berens, P., and Kobak, D. (2021). Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data. Genome Biol. *22*, 258. https://doi.org/10.1186/s13059-021-02451-7.

53. Foreman-Mackey, D., Hogg, D.W., Lang, D., and Goodman, J. (2013). emcee: The MCMC Hammer. Publ. Astron. Soc. Pac. *125*, 306–312. https://doi.org/10.1086/670067.

54. Hou, F., Goodman, J., Hogg, D.W., Weare, J., and Schwab, C. (2012). An Affine-Invariant Sampler for Exoplanet Fitting and Discovery in Radial Velocity Data. Astrophys. J. *745*, 198. https://doi.org/10.1088/0004-637X/745/2/198.

55. Schwarz, G. (1978). Estimating the Dimension of a Model. Ann. Stat. *6*, 461–464. https://doi.org/10.1214/aos/1176344136.

56. Hogg, D.W., and Foreman-Mackey, D. (2018). Data Analysis Recipes: Using Markov Chain Monte Carlo. Astrophys. J. Suppl. *236*, 11. https://doi.org/10.3847/1538-4365/aab76e.

57. Link, W.A., and Eaton, M.J. (2012). On thinning of chains in MCMC: Thinning of MCMC chains. Methods Ecol. Evol. *3*, 112–115. https://doi.org/10.1111/j.2041-210X.2011.00131.x.

58. Harms, R.L., and Roebroeck, A. (2018). Robust and Fast Markov Chain Monte Carlo Sampling of Diffusion MRI Microstructure Models. Front. Neuroinf. *12*, 97. https://doi.org/10.3389/fninf.2018.00097.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Atlas of the Developing Mouse Brain	La Manno et al.[22]	http://mousebrain.org/development/downloads.html
Mouse Pancreas Endocrinogenesis Dataset	GSE132188[46]	http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE132188
Software and algorithms		
python	www.python.org	https://www.python.org/downloads/release/python-397/
scanpy	pypi.org	https://pypi.org/project/scanpy/1.9.1/
matplotlib	pypi.org	https://pypi.org/project/matplotlib/3.5.1/
numpy	pypi.org	https://pypi.org/project/numpy/1.21.6/
scipy	pypi.org	https://pypi.org/project/scipy/1.8.0/
emcee	emcee.readthedocs.io	https://emcee.readthedocs.io/en/stable/user/install/
R	cran.r-project.org	https://cran.r-project.org/src/base/R-4/R-4.0.2.tar.gz
Monocle 3	bioconductor.org	https://cole-trapnell-lab.github.io/monocle3/docs/installation/
Slingshot	bioconductor.org	https://bioconductor.org/packages/devel/bioc/vignettes/slingshot/inst/doc/vignette.html
tradeSeq	bioconductor.org	https://bioconductor.org/packages/devel/bioc/vignettes/tradeSeq/inst/doc/tradeSeq.html

RESOURCE AVAILABILITY

Lead contact

Further information and requests should be directed to and will be fulfilled by the lead contact, Daniel Rosebrock (rosebroc@molgen.mpg.de).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- This paper analyzes existing, publicly available data. These accession numbers for the datasets are listed in the key resources table.
- All original code has been deposited at https://github.com/daniel-rosebrock/transcriptional_cascades and is publicly available as of the date of publication.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Processing scRNA-Seq of mouse e13.5 forebrain dorsal samples

The raw count data from the Atlas of the Developing Mouse Brain[22] was downloaded from http://mousebrain.org/development/downloads.html. The raw count data was loaded into scanpy[47] for downstream analyses. Cells were then initially subset to samples corresponding to e13.5 embryos derived from the forebrain dorsal tissue (labeled as 'ForebrainDorsal' in the metadata), and further subset to 'Radial glia' and 'Neuron' cell types. The first sample ('SampleName' = 'G23') and second sample ('SampleName' = 'G9') were analyzed separately. Initially, in both sample, cells with a DoubletFinderPCA[48] score above 0.5 were filtered to remove potential doublets. Following this, the count data was normalized using scanpy's 'normalize_total' function, followed by a natural log transformation and adding a pseudocount of 1. Highly variable genes were estimated using scanpy's 'highly_variable_genes' function, after which a principal component analysis was run using the highly variable genes. A kNN graph was estimated from the top 50 principal components using k = 15 nearest neighbors based on the UMAP neighborhood selection approach.[49] Following this, Louvain clustering[50] was performed using a resolution parameter of 1.5. Clusters

exhibiting high expression levels of G2M cell cycle genes were subsequently filtered, as well as clusters with a subpallium (ventral cortical) identity, hippocampal identity, and Cajal-Retzius neurons. The above procedure was re-run until the only subsequent populations in the sample consisted of forebrain dorsal NSCs, IP cells, or neurons based on the expression of known marker genes for the respective populations. Diffusion pseudotime estimates[9] for each cell were then estimated was after running a diffusion map embedding and assigning a starting cell. The raw count data across all cells ordered by diffusion pseudotime were then stored and the MCMC procedure was run on the resulting count matrix.

Processing scRNA-Seq of mouse e14.5 pancreas development samples

The raw count data for the pancreas endocrinogenesis dataset[46] was downloaded from http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE132188. The raw count data was loaded into scanpy[47] for downstream analyses. Cells were then initially subset to samples corresponding to e14.5 embryos. All cells with a positive G2M score in the metadata were initially filtered. Following this, the count data was processed in a similar fashion to the Atlas of the Developing Mouse Brain dataset using scanpy. Diffusion pseudotime estimates were calculated and the raw count data across all cells ordered by diffusion pseudotime were then stored and the MCMC procedure was run on the resulting count matrix.

Establishing a likelihood model

The negative binomial distribution has been shown to accurately describe the count data generated in scRNA-Seq experiments without the need to account for zero-inflation resulting from "dropout" events.[51] The probability mass function for the negative binomial distribution can be parameterized using the mean, $\mu \in \mathbb{R}^+$, and dispersion parameter, $\varphi \in \mathbb{R}^+$, with $y \in \mathbb{N}$, as follows,

$$p(y|\mu,\varphi) = \binom{y+\varphi-1}{y}\left(\frac{\mu}{\mu+\varphi}\right)^{y}\left(\frac{\varphi}{\mu+\varphi}\right)^{\varphi}. \quad \text{(Equation 2)}$$

The mean and variance of the random variable $Y \sim \mathrm{NB}(\mu,\varphi)$ which follows a negative binomial distribution is then $\mathbb{E}[Y] = \mu$ and $\mathrm{Var}[Y] = \mu + \frac{\mu^2}{\varphi}$. For a gene g with measured counts of $\vec{Y}_g = \{y_{gt}\}_{t=1,\ldots,N}$ along a pseudotime trajectory with fixed pseudotime-step interval, $\vec{\mu}_g = \{\mu_{gt}\}_{t=1,\ldots,N}$ and $\vec{\varphi}_g = \{\varphi_{gt}\}_{t=1,\ldots,N}$ the mean and dispersion at corresponding pseudotimes, the full likelihood of observing $\vec{Y}_g$ is:

$$\mathcal{L}\left(\vec{\mu}_g \middle| \vec{Y}_g, \vec{\varphi}_g\right) = \prod_{t=1}^{N} p\left(y_{gt} \middle| \mu_{gt}, \varphi_{gt}\right), \quad \text{(Equation 3)}$$

where $p(y_{gt}|\mu_{gt},\varphi_{gt})$ is the negative binomial probability mass function. The full log-likelihood is then:

$$\ln\left(\mathcal{L}\left(\vec{\mu}_g \middle| \vec{Y}_g, \vec{\varphi}_g\right)\right) = \sum_{t=1}^{N} \ln\left(p\left(y_{gt} \middle| \mu_{gt}, \varphi_{gt}\right)\right). \quad \text{(Equation 4)}$$

It was shown that when fitting scRNA-Seq UMI count data to a negative binomial model, data are consistent with a global dispersion parameter independent of the expression level of a given gene, and that fitting a dispersion parameter to each gene individually leads to overfitting.[52] Therefore, a global estimate of φ can be used for every gene independent of pseudotime, and $\vec{\varphi_g} = \{\varphi_{gt}\}_{t=1,\ldots,N}$ is replaced with a constant φ in Equation 4. A dataset specific φ using genes which exhibit lower levels of overdispersion is estimated, since the expression levels in these genes reflect the technical rather than the biological variability. To do this, the log10 mean counts for each gene are binned into five equally spaced bins, and a linear fit between log10 mean and log10 variance of counts in each bin is estimated. Genes within the top 20th percentile of the difference between the estimated variance and the expected variance using the linear fit in each bin are then filtered. The remaining genes are used to fit the non-linear relationship between the mean (μ) and variance ($\sigma^2 = \mu + \frac{\mu^2}{\varphi}$) using unconstrained non-linear least squares (Figure S9).

Here, φ estimates the dispersion based on genes which do not exhibit high variability in the dataset, and therefore captures the technical variability in the dataset. This technical variability is in large part driven by the varying number of UMI counts captured in each cell, as well as other factors including library quality and amplification bias. Thus, the full log-likelihood of observing counts $\vec{Y}_g = \{y_{gt}\}_{t=1,\ldots,N}$ for gene g along a pseudotime trajectory given the mean at corresponding pseudotime points $\vec{\mu}_g = \{\mu_{gt}\}_{t=1,\ldots,N}$, becomes,

$$\ln\left(\mathcal{L}\left(\vec{\mu}_g \middle| \vec{Y}_g, \varphi\right)\right) = \sum_{t=1}^{N} \ln\left(p\left(y_{gt} \middle| \mu_{gt}, \varphi\right)\right), \quad \text{(Equation 5)}$$

where φ is a global parameter estimated using the procedure described above.

For scRNA-Seq methods which sequence only from one end of the transcript and not full-length protocols, normalization does not need to account for the total transcript length. In this case, for a given cell i, let M_i be the number of UMIs in cell i, and y_{gi} be the number of UMIs for gene g in cell i. In this paper, we use the median number of UMIs across all cells in the dataset as a size factor $\tilde{M}$, that is, $\tilde{M} = \mathrm{med}\{M_i\}_{i=1,\ldots,N}$. Then, the log-normalized expression levels for gene g in cell i is defined by the following mapping,

$$h\left(y_{gi}\right) = \tilde{y}_{gi} = \ln\left(\frac{y_{gi}}{M_i}\tilde{M} + 1\right). \quad \text{(Equation 6)}$$

The functions $(f_{\text{unif}}, f_{\text{gauss}}, f_{\text{sig}}, f_{\text{dsig}})$ described in Equation 1 are then fit to the pseudotemporally ordered expression profile for gene g, $\{\tilde{y}_{gt}\}_{t=1,\ldots,N}$, in the log-normalized expression space with the objective function to maximize defined by the likelihood in Equation 5. The means $\vec{\mu}_g = \{\mu_{gt}\}_{t=1,\ldots,N}$ are then calculated by mapping the function values evaluated at $t = 1, \ldots, N$ back to count space using the inverse of Equation 6. The full log-likelihood estimate is then evaluated by plugging in the $\vec{\mu}_g$ values and global estimate for φ into Equation 5.

This procedure can be summarized as follows. We want to solve for $f_\alpha(t;\theta)$, which maximizes the following likelihood,

$$\ln\left(\mathcal{L}\left(\vec{\mu}_g \middle| \vec{Y}_g, \varphi\right)\right) = \sum_{t=1}^{N} \ln\left(p\left(y_{gt} \middle| h^{-1}(f_\alpha(t;\theta)), \varphi\right)\right), \quad \text{(Equation 7)}$$

where $f_\alpha \in (f_{\text{unif}}, f_{\text{gauss}}, f_{\text{sig}}, f_{\text{dsig}})$.

Model inference using MCMC

Under the framework presented above, solving for $f_\alpha(t;\theta))$ can be formulated as a Bayesian inference problem, which we solve using an ensemble sampler MCMC approach.[19] This provides an estimate of the posterior distribution over the parameter space for each of the parameters in the different functions $(f_{\text{unif}}, f_{\text{gauss}}, f_{\text{sig}}, f_{\text{dsig}})$ described in Equation 1. For each of the models, the priors used for the different parameters are summarized in Table S4.

Note, in Table S4, the folded normal distribution is parameterized by $\mu > 0$ and $\sigma > 0$ with probability density function,

$$p(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} + \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x+\mu)^2}{2\sigma^2}}. \quad \text{(Equation 8)}$$

The uniform priors in Table S4 are uninformative, however, they provide bounds on the parameters to keep them in interpretable and meaningful ranges. The slope parameters k in the sigmoidal function, and k_1 and k_2 in the double sigmoidal function, have a folded normal prior with 0-mean and 0.1 variance, which is used to ensure that the slope has a low magnitude. This prior is used because differences in the function once the slope becomes relatively large are minimal. Finally, the folded normal prior on σ in the Gaussian with 0-mean and $N/10$ variance is used to ensure that the curve does not become very flat.

In this paper, we use the ensemble sampler MCMC proposed by Goodman & Weare in 2010[19] with implementation by Foreman–Mackey et al.[53] An initial guess is needed as a starting point from which a walker begins in the ensemble sampler. For the Gaussian and sigmoidal functions, initial guesses are derived from a non-linear least squares fit for each function on the log-normalized pseudotime expression levels using scipy's 'curve_fit' function, with added Gaussian noise. For the double sigmoidal function, initial guesses are randomly chosen to cover the varieties of different forms the functions can have. For the uniform function, initial guesses are randomly chosen from a uniform distribution over the interval 0.01 and maximum expression level for the gene of interest. The number of walkers used is four times the number of parameters for each function — 28 for the double sigmoidal fit, 16 for the Gaussian fit, 16 for the sigmoidal fit, and 4 for the uniform fit. This enables a wide sampling across the search space of parameters.

The MCMC is then run for a total of 10,000 iterations. There is generally no consensus on how many iterations to run an MCMC algorithm.[53] Thousands of iterations are typically desirable to allow the process to reach a steady-state. After reaching the steady-state, the MCMC will sample from the posterior distribution over the parameter space, enabling an estimate of the posterior distribution for each parameter. Iterations before reaching the steady-state are discarded, as these are not sampled from the target distribution. This is called the "burn-in" phase. For this implementation, a burn-in of 5,000 iterations was used (Figure S10).

Some MCMC walkers can get stuck near a local maximum. These walkers typically have a low acceptance rate, that is the proportion of moves for which the MCMC sampler generated parameter values that differed from the previous sample. One common practice is to prune these walkers from the final MCMC output. For example, walkers can be pruned which get stuck in irrelevant local optima by clustering the likelihood of the walkers and removing the clusters with lower likelihoods.[54] For this implementation, half of the MCMC walkers are pruned with the lowest acceptance rate in order to remove potentially stuck walkers (Figure S11).

Model selection

We use a probabilistic model selection technique, the bayesian information criterion (BIC)[55] to score the different models, and select the model with the best score. The BIC is defined as follows,

$$\text{BIC} = k\ln(n) - 2\ln(\widehat{L}), \quad \text{(Equation 9)}$$

where n = number of data points, k = number of parameters in the model, and $\widehat{L}$ = maximized value of the likelihood function. In the original formulation of the BIC, the value $\widehat{L}$ was derived from maximum likelihood estimation. When using an MCMC for model inference, the output consists of a sampling or distribution over the parameter space. It is advantageous to use a likelihood estimate which more closely reflects the optimal parameter regime estimated from the MCMC instead of the parameter regime which maximizes the likelihood. To this end, $\widehat{L}$ in the

BIC equation in Equation 9 is replaced with $P(y|\langle\theta\rangle)$, the likelihood of observing the data given $\langle\theta\rangle$, where $\langle\theta\rangle$ = mean over the parameter estimates across all MCMC iterations.

To improve the generalizability of a model fit to the dataset, and remove the bias of outliers, we developed a variation of cross-validation for model selection, described in Algorithm 1.

Algorithm 1. Perform model selection based on MCMC runs

1: Measure average parameter estimates, $\langle\theta\rangle$, across MCMC runs for each model.
2: Remove 2% of the data chosen randomly (y_{sub}), and estimate BIC for each model using $P(y_{sub}|\langle\theta\rangle)$.
3: Repeat Step 2. for 10,000 subsets. Define BIC_x as the set of BIC estimates across all 10,000 subsets for a given fit, and $\langle BIC_x\rangle$ as the mean BIC estimate across all 10,000 subsets.
4: **if** $\max(BIC_{\text{double sigmoidal}}) < \min(BIC_{\text{uniform}})$ & $\langle BIC_{\text{double sigmoidal}}\rangle < \langle BIC_{\text{gauss}}\rangle$ & $\langle BIC_{\text{double sigmoidal}}\rangle < \langle BIC_{\text{sigmoidal}}\rangle$ **then**
5: Set best fit to double sigmoidal.
6: **else if** $\max(BIC_{\text{sigmoidal}}) < \min(BIC_{\text{uniform}})$ & $\langle BIC_{\text{sigmoidal}}\rangle < \langle BIC_{\text{gauss}}\rangle$ **then**
7: Set best fit to sigmoidal.
8: **else if** $\max(BIC_{\text{gauss}}) < \min(BIC_{\text{uniform}})$ & $\langle BIC_{\text{gauss}}\rangle < \langle BIC_{\text{sigmoidal}}\rangle$ **then**
9: Set best fit to Gaussian.
10: **else**.
11: Set best fit to uniform.
12: **end if**.

Note, instead of cross-validating a model estimated from a training set on a test set, the full dataset is used for model inference and tested on random subsets of the dataset. Figure S12 highlights a random sampling of the parameters over the MCMC runs using a double sigmoidal, Gaussian, sigmoidal and uniform model, as well the BIC estimates on random 98% subsets of the data.

It is worth noting that the double sigmoidal function can also closely take the form of the Gaussian and sigmoidal functions. It would be possible to use the double sigmoidal function alone, instead of including the Gaussian and sigmoidal functions, to model the dynamics of gene expression. However, the double sigmoidal function will force the presence of two inflection points, whereas with the sigmoidal function will only have one inflection point, which in many cases more accurately models the gene expression dynamics of single a state-switch. Finally, a simpler model is often more favorable to use than a more complex model to prevent overfitting, and in the cases where a Gaussian function provides an equally good fit as the double sigmoidal function, then the selection of the simpler Gaussian model is preferred.

MCMC diagnostics

In order to ensure that the MCMC adequately approximates the posterior distribution over the parameter space, a variety of heuristics exist. The MCMC trace plot (Figure S10) provides a visual inspection of whether the MCMC appears to have reached a steady-state. Also, the acceptance fraction across MCMC chains (Figure S11) is used to filter potentially stuck MCMC walkers. In general, there is no way to prove convergence of an MCMC sampler,[56] and therefore diagnostics are used to measure how well an MCMC run has converged to an equilibrium or steady-state. A few diagnostics are highlighted in this section to show the ability of the ensemble sampler described above to adequately converge to the posterior distribution over the parameter space.

One diagnostic metric relies on the estimate of the integrated autocorrelation time, which estimates the number of iterations needed for the MCMC to draw an independent sample. In the case of samples generated by an MCMC, the samples are not independent. This is due to the nature of the Markov process used to sample from the posterior distribution, which is dependent on the previous sampling of parameters, by definition. The integrated autocorrelation time is defined as,

$$\tau_f = \sum_{\tau=-\infty}^{\infty} \rho_f(\tau) = 1 + 2\sum_{\tau=1}^{\infty} \rho_f, \quad \text{(Equation 10)}$$

where $\rho_f(\tau)$ is the autocorrelation function at time delay τ. Then, the effective sample size (ESS), i.e. the number of i.i.d. draws from the posterior distribution, for an ensemble sampler can be calculated as,

$$\text{ESS} = \frac{MN}{\tau_f}, \quad \text{(Equation 11)}$$

where M = number of walkers, and N = number of MCMC iterations used after discarding the burn-in. In order to estimate τ_f, the marginal autocorrelation function for each parameter in the model can be estimated separately out to a certain time delay, T, using the average estimate across all walkers, and taking the maximum estimate of τ_f over all T, defined as

$$\widehat{\tau}_f = \max_T \left(1 + 2\sum_{\tau=1}^{T} < \rho_f(\tau) > \right). \quad \text{(Equation 12)}$$

Here, $T \in [0, 1000]$ enables an accurate estimate of $\widehat{\tau}_f$ under the assumption that $\rho_f(\tau)$ approaches 0 by $\tau = T$ for each parameter. The autocorrelation function (Figure S13) and autocorrelation time (Figure S14) is estimated for each parameter separately.

For a general comparison, the autocorrelation times were estimated for all genes using the model with the best fit in the mouse e13.5 forebrain sample (Figure S15). The autocorrelation times increase with the complexity of the model (i.e. number of parameters specified in each model). This is in part expected, since a model with more parameters will generally have a lower acceptance rate due to the higher number of dimensions in which the MCMC has to make proposal moves, leading to higher autocorrelations for each parameter. Nonetheless, the autocorrelation times are fairly robust for each model.

Thinning is an approach to use every k-th iteration of the MCMC walkers, where $k = \tau_f$ would represent an i.i.d. sampling of the posterior distribution. However, various publications indicate that thinning is often unnecessary and results in reduced precision.[57,58] Therefore, no thinning of the MCMC walkers was used in this analysis.

Another way to visualize the posterior distribution over the parameter space derived from an MCMC is a corner plot (Figure S16). The corner plot highlights both the two dimensional projections over the parameter space across iterations of the MCMC, as well as the marginal posterior distribution for each individual parameter (highlighted in the upper plots). Some parameters are more correlated with each other than others, indicating underlying covariates within the model parameters. However, the marginal posterior distributions do not appear to be multimodal.

These heuristics provide some insight into the ability of the ensemble MCMC sampler to provide an accurate sampling of the posterior distribution over the parameter space.

Estimating inflection points

Inflection points occur where the curvature of a function changes sign. At inflection points, the first-order derivative, or rate of change, of a function reaches a local maximum or local minimum. At an inflection point, the second-derivative of a function passes through 0 with the second derivative changing sign from positive (concave upward) to negative (concave downward) or vice versa. The inflection points of the Gaussian, sigmoidal and double sigmoidal fits can be used to compare the relative timing of when genes exhibit a state transition along a pseudotime trajectory. To estimate the inflection points of the different functions, first solve for x at which the second-derivative of the function is zero. For the Gaussian function, $f_{\text{gauss}}(t)$, sigmoidal function $f_{\text{sig}}(t)$, and double simgoidal function $f_{\text{dsig}}(t)$ defined in Equation 1, the second derivatives are

$$f''_{\text{gauss}}(t) = \frac{a}{\sigma^4} e^{-\frac{(t-t_0)^2}{2\sigma^2}} (t - (t_0 - \sigma))(t - (t_0 + \sigma)),$$

$$f''_{\text{sig}}(t) = k^2 L \frac{e^{-k(t-t_0)}\left(e^{-k(t-t_0)} - 1\right)}{\left(1+e^{-k(t-t_0)}\right)^3},$$

$$f''_{\text{dsig}}(t) = k_1^2 (b_{mid} - b_{min}) \frac{e^{-k_1(t-t_1)}\left(e^{-k_1(t-t_1)} - 1\right)}{\left(1+e^{-k_1(t-t_1)}\right)^3} + k_2^2 (b_{max} - b_{mid}) \frac{e^{-k_2(t-t_2)}\left(e^{-k_2(t-t_2)} - 1\right)}{\left(1+e^{-k_2(t-t_2)}\right)^3}.$$

For the Gaussian function, $f_{\text{gauss}}(t)$, two inflection points occur at $t \in (t_0 - \sigma, t_0 + \sigma)$. For the sigmoidal function, $f_{\text{sig}}(t)$, one inflection point occurs at $t = t_0$. The estimates for the inflection points are then measured from the parameters $(t_0 - \sigma, t_0 + \sigma)$ for the case of the Gaussian and t_0 for the case of sigmoidal function at each MCMC iteration. Finally, for the double sigmoidal function, $f_{\text{dsig}}(t)$, the number of inflection points can vary. However, if all parameters are fixed besides k_1, then, $f''_{\text{dsig}}(t) \to 0$ as k_1 increases. Similarly, if all parameters are fixed besides t_1, then $f''_{\text{dsig}}(t) \to 0$ as t_1 decreases. That is, for $k_1 \gg 0$, i.e. the transition from b_{min} to b_{mid} occurs rapidly, then an inflection point will occur very close to t_1. Similarly, for $k_2 \gg 0$, i.e. the transition from b_{mid} to b_{max} occurs rapidly, then an inflection point will occur very close to t_2. Also, the further apart t_1 and t_2 are from each other, the closer the inflection points are to t_1 and t_2. To ensure the inflection points occur very close to t_1 and t_2, at each iteration of the MCMC, a move is only accepted in cases where $\text{sign}(f''_{\text{dsig}}(t_1 - dt)) * \text{sign}(f''_{\text{dsig}}(t_1 + dt)) < 0$ and $\text{sign}(f''_{\text{dsig}}(t_2 - dt)) * \text{sign}(f''_{\text{dsig}}(t_2 + dt)) < 0$ for $dt = 1$. The estimates for the inflection points are then calculated from the parameters t_1 and t_2 at each MCMC iteration.

Comparing inflection points

Regulatory interactions were inferred based on the relative timing of inflection point estimates (Figure 2). If there was an overlap of at least 1% in the inflection point estimates between two genes across MCMC iterations, then these were assumed to have a simultaneous switch state. A regulatory interaction between the two was mutually positive if the inflection points had the same sign, and mutually negative if the inflection points differed in sign. The overlap between two inflection points is estimated by binning the inflection point estimates across all MCMC iterations to 100 equally spaced bins, starting at the minimum inflection point estimate across both genes and ending at the maximum inflection point estimate across both genes. Let $\{x_i\}_{i \in [1,100]}$ represent this binning domain. If $p_A(x_i)$ is the percent of counts in the histogram in bin x_i for gene A, and $p_B(x_i)$ is the percent of counts in the histogram in bin x_i for gene B, then the overlap between the two, $P(A = B)$, is

$$P(A = B) = \sum_{i=1}^{100} \min\left(p_A(x_i), p_B(x_i)\right). \quad \text{(Equation 13)}$$

If the inflection point estimates were non-overlapping (*i.e.* inflection point overlap was less than 1%), then the following relationships were constructed. If the first gene (*i.e.* earlier inflection point) had a positive inflection point and the second gene (*i.e.* later inflection point) also had a positive inflection point, then the first gene positively regulates the second gene. If the first gene had a positive inflection point and the second gene had a negative inflection point, then the first gene negatively regulates the second gene, and the second gene positively regulates the first. If the first gene had a negative inflection point and the second gene had a negative inflection point, then the first gene positively regulates the second gene. If the first gene had a negative inflection point and the second gene had a positive inflection point, then no relationship is given.

Running Monocle 3, tradeSeq and Slingshot on mouse e13.5 forebrain dorsal cells

We tested whether genes were differentially expressed along the e13.5 forebrain dorsal NSC → IP → neuron trajectory using Monocle 3 and tradeSeq. tradeSeq works by fitting a negative binomial generalized additive model (GAM) to the pseudotime-ordered counts for each gene separately. We used the associationTest() from tradeSeq, which tests the null hypothesis that all smoother coefficients in the GAM are equal to each other. We passed in the raw counts and pseudotime ordering from diffusion pseudotime as input, specifying the number of knots used for the GAM fitting to 3. To test whether genes were differentially expressed along the trajectory using Monocle 3, we used the graph_test() function, passing in the principal_graph estimated by the learn_graph() function, which estimates a pseudotime trajectory by fitting a principal graph through the cells. Finally, to compare the affect of input pseudotime method, we also estimated a pseudotime ordering of e13.5 forebrain dorsal NSC → IP → neurons using Slingshot, which fits a principal curve through the data. As input, we passed in the 2-dimensional umap embedding of these cells and log-normalized expression data, using the getLineages() function to estimate the pseudotime ordering.