Simulating Gene Regulatory Networks for Validation of scRNA-seq Data Alignment Methods

Jennifer Chen

April 2025

Abstract

Accurately modeling gene regulatory networks (GRNs) is critical for understanding cell differentiation and development. However, this remains a significant challenge due to the difficulty of aligning single-cell expression data across heterogeneous cell populations and the inherent complexity of GRNs, which makes obtaining ground truth information rare and often infeasible. We present a simulation pipeline that generates biologically realistic and customizable single-cell expression data for synthetic GRNs. Building on and extending previous approaches, our pipeline integrates a GRN-based expression simulator with support for RNA splicing, pseudotime trajectory inference, and lineage branching. We introduce a flexible GRN generation script, allowing users to design or perturb regulatory networks to test specific hypotheses. Through a series of perturbation studies, we demonstrate how modifications to GRN structure and regulatory parameters impact downstream inference tasks, highlighting both strengths and limitations of current computational tools. This work establishes a controlled, extensible platform for benchmarking GRN reconstruction and single-cell analysis methods.

Contents

Introduction

Hundreds of different cell types compose the tissues and biological systems that create complex organisms. Cells are characterized by the unique combination and magnitude of gene expression which controls the functionality of cells. This is especially important in cell development, such as when a stem cell evolves to a specialized cell such as a neuron, when certain genes are upregulated and downregulated to achieve cell type differentiation. Genes participate in a highly complex orchestration of interactions, forming cascades, hierarchies, and feedback loops that control cell behavior. Computational methodologies are being created to infer these gene regulatory networks (GRNs) using high-dimensional, single-cell data such as OTVelo, a framework that leverages time-series single-cell gene expression data to reconstruct gene regulation across temporal stages. However, the scarcity of single-cell datasets complete with RNA splicing and time-stamp information, due to the high cost of generating such high-dimensional data, limits the opportunity to thoroughly test and validate the method. Moreover, ground truth for GRNs is challenging to quantify and obtain for real single-cell RNA sequencing (scRNA-seq) datasets due to the complex nature of biological systems, making it difficult to evaluate the accuracy of GRN inference methods.

The objective of this experimentation is to develop a pipeline for creating realistic, synthetic single-cell datasets with ground truth to provide a reliable procedure for benchmarking GRN inference methods such as OTVelo. We evaluate existing techniques for simulating RNA splicing, cell lineages, and pseudotime in GRNs. We also introduce a GRN generation script that allows users to easily customize networks for their experiments. By integrating and documenting these components, our pipeline provides a controlled and flexible environment for validating computational tools used in GRN reconstruction.

Background

Gene Regulatory Networks

Gene Regulatory Networks (GRNs)represent the relationships between genes, such as upregulation–the promotion of transcription of a gene, or downregulation–the suppression of gene expression. Capturing these systems rely on identifying the upstream and downstream gene actors in the regulatory process, as well as the purposes and strength of the regulation. By identifying GRNs, we can better understand the dynamics of the cellular system and the mechanisms that govern cell identity, function, and response to biological signals.

Single-Cell Data

Inferring GRNs has been an enduring challenge and the increase ofscRNA-seq and co-assay multimodal data collection on single cells has led to an abundance of biological information available for the exploration and extraction of GRNs [1]. However, during the single-cell processing procedure, observed cells are destroyed to gather data and subsequently the relations between the cells in these datasets are often lost and nonexistent.

This phenomena creates an obstacle to understanding a cell's gene expression trajectory throughout time. After a cell is processed, it is impossible to observe how the sequenced cell will evolve over time and how its cellular expression contributes to its growth. Capturing the gene expression activity over time of a cell is crucial for understanding its influence on cell differentiation patterns and the growth of other genes.

Another limitation in single-cell studies is the lack of robust ground-truth labeling for cell types. Because of the large scale of scRNA-seq data, manual annotations of datasets calls for significant time and labor. The quality of annotations can also fluctuate depending on what annotations are based on, introducing subjectivity that directly impacts downstream applications [4, 5].

The Motivation for Synthetic Data

An alternative to single-cell data is to synthesize single-cell expression datasets with statistical properties that parallel observed patterns in experimental data and biological relationships. Moreover, having the option of synthesizing your own datasets provides the opportunity to perturb the biological and technical parameters, enabling systematic evaluation of how these factors influence the performance of analytical tools. Additionally, by generating the GRN used in simulation, we obtain precise ground truth labels to benchmark GRN inference methods.

Related Works

SERGIO

SERGIO is a single-cell gene expression data simulator and the first single-cell expression simulator to explicitly integrate the MR-to-gene relationships in its simulation framework. SERGIO processes a user-defined GRN as input and can simulate both steady-state and differentiating cell expression data. The model computes the expression of each gene, excluding the MRs, by determining a production rate that incorporates contributions from its upstream regulators and a fixed decay rate. This explicit modeling of MR-to-gene interactions allows SERGIO to simulate realistic gene expression trajectories and cell differentiation processes. Additionally, SERGIO simulates both spliced and unspliced gene expression data which is useful for tasks such as calculating more accurate pseudotime estimates [3].

GRouNDGAN

GRouNDGAN is a deep learning framework for simulating scRNA-seq datasets using GRNs. Compared to traditional model-based simulators that rely on assumptions about coordinated gene regulation, GRouNdGAN learns complex regulatory patterns through a generative adversarial network (GAN) architecture. It uses a user-defined GRN and generates transcription factor expression values which are subsequently used to simulate target gene expressions. This ensures that the simulated data reflects causal regulatory relationships. Additional deep learning agents help maintain realism and enforce dependence on transcription factor data. However, GRouNdGAN does not seem to be able to generate more granular gene expression data such as spliced and unspliced RNA transcription levels, which are sometimes critical for downstream tasks. [8]

OTVelo

OTVelo is a dynamic GRN inference method that leverages optimal transport (OT) to estimate gene velocities from time-stamped single-cell RNA-seq data without requiring splicing measurements. OT is used to model the probabilistic transition of cells between consecutive time points, predicting each cell's gene expression in past and future states. These predictions allow the calculation of gene velocities, achieving single-cell resolution for every gene. OTVelo then infers gene interactions by applying either time-lagged correlation or Granger causality to the computed velocities. Importantly, these methods are adapted to use OT-based transition probabilities, accommodating the inherent uncertainty in cell trajectories.

OTVelo was validated on synthetic GRNs and real experimental datasets. Notably, the method generalizes to settings where only pseudotime is available, extending its range of applications to single-cell datasets. [7]

Methods

To generate synthetic GRNs and simulate RNA splicing and gene expression over time, several components are required:

1. **GRN Simulator:** Generates the projected gene expression data of a given GRN.
2. **GRN Generator:** Randomly creates realistic and contained biological systems to input into the above components.
3. **Pseudotime Calculator:** Computes the most plausible time-stamps for the input data.
4. **Ground Truth:** To assess the accuracy of cell trajectory alignment methods, we not only need simulated GRNs but also the ground truth mappings of the GRN to compare predictions to.

GRN Simulator

We chose to use SERGIO for GRN simulation because it explicitly models RNA velocity by simulating both unspliced and spliced transcription levels per gene. Furthermore, SERGIO applies transformations to the simulated data to incorporate technical noise that mimics the variability observed in real scRNA-seq experiments. SERGIO allows users to create their own GRN systems as input to the simulator. This is advantageous to developing and perturbing GRNs to create controlled experimentation to evaluate the performance of GRN inference methods

Specifically, the simulator requires separate files detailing the differentiation graph of cells in the GRN, the gene-to-gene regulatory relationships, and the MR genes. This information reflects the biological organization of gene regulatory systems: the differentiation graph captures the developmental trajectories between cell types, the regulatory interaction file encodes how genes influence each other's expression, and the master regulator file specifies key drivers of cell identity whose expression patterns vary across different cell types. Together, these inputs provide a comprehensive representation of the dynamic and hierarchical structure of GRNs.

We configured SERGIO with the following parameters for simulation experiments:

- `noise_params = 0.2`: Controls the magnitude of technical noise added to the expression matrix. A moderate value was chosen to reflect realistic levels of measurement variability in scRNA-seq data while preserving underlying biological structure.
- `decays = 0.8`: Sets the decay rate of gene expression over time, preventing unbounded transcript accumulation. A value of 0.8 ensures highe mRNA turnover rates to avoid getting stuck in minima and never converging.
- `noise_type = 'dpd'`: Indicates that dual-probabilistic dropout noise should be used, which more accurately mimics the dropout patterns observed in real single-cell datasets by modeling both gene- and cell-specific dropout effects.
- `dynamics = True`: Enables the use of stochastic differential equations (SDEs) to simulate dynamic gene expression over time, including both unspliced and spliced transcript levels, thereby allowing RNA velocity to be inferred from the output.

- `bifurcation_matrix = bMat`: Extracted from the user-provided cell differentiation transition matrix. This matrix governs the overall lineage topology simulated by SERGIO.
- `number_sc`: The number of single cells simulated per cell type. We usually used 300 cells per cell type for our experiments unless otherwise indicated.

Cell Differentiation

The cell differentiation file describes a transition matrix–a tab-separated $Y \times Y$ matrix, where Y is the number of cell types. Each element in row i and column j, denoted B_{ij}, represents the transition rate $r_{i,j}$ from cell type i to cell type j. When $r_{i,j} = 0$, cell type i never differentiates to cell type j. According to the documentation, an increased value of $r_{i,j}$ will increase the density of simulated cells transitioning from type i to type j, though it may also lead to increased latency in simulation, likely due to increased computational cost.

Gene Interactions

The gene interactions file follows the following format:

`targetidx, nRegs, reg1, reg2, ..., reg_N, K1, K2, ..., K_N`, where:

- `targetidx` is the index of the target gene being regulated
- `nRegs` is the number of regulators for the target gene
- `reg1` to `reg_N` are the indices of the regulator genes
- `K1` to `K_N` are the regulatory strength constant corresponding to each regulator. This variable also denotes if the interaction is upregulating or downregulating depending on if the sign is positive or negative, respectively

Master Regulators

Master regulator genes are commonly understood to be genes without upstream regulators, or genes that expert broad impact over the development of cell lineages [2]. These genes represent initial sources of regulatory activity, driving the expression of downstream targets over time. In SERGIO, the expression dynamics of master regulators are governed only by constant production and decay rates. Moreover, in the simulator the cell types are defined by the average expression levels of master regulators. Thus, the differentiation of cell types are directly influenced by variations in the user-provided production rates of these genes. By simulating their expression independently of other genes, the temporal evolution of the gene regulatory network is influenced more by the master regulatory inputs. The master regulators file follows the format:

`regulator, celltype1, celltype2, ..., celltypeN`

Each row specifies the production rates for a single master regulator gene across different cell types.

- `regulator`: the index of the master regulator gene
- `celltype1` through `celltypeN`: the constant production rates of the gene in each respective cell type

GRN Generator

The complex orchestration of genes and cell types in the construction of GRNs can make manually generating a simple GRN a challenging task. To address this, we developed a Python-based framework for generating random GRNs that mimic plausible gene or cell interaction structures and the respective informative input files described above for simulation on SERGIO.

SERGIO can only simulate acyclic, directed graph (DAG) inputs for both cell differentiation trajectories and gene interaction networks. The script supports two distinct simulation modes—gene mode and cell mode. It generates the DAG and the corresponding information respective to the type of system being generated and saves the network information to file for streamlined use in SERGIO's simulator.

The DAG algorithm takes as input the total number of nodes, the number of master regulators, and other parameters described later according to the different modes that the script supports. It guarantees the production of an acyclic by enforcing a global topological order on all edges. It also guarantees global connectivity by merging any disconnected components, as GRNs and cell trajectories should all be connected to each other if they are an interaction network.

In the first phase of the algorithm, the script constructs a single-parent backbone while following a topological ordering of the nodes, which help maintain the acyclic nature of the resulting graph as edges are drawn in order of the nodes. With a user-specified probability p, the script choose a random earlier regular gene as parent.Otherwise, with probability $1 - p$, it links the current node to the immediate predecessor. In the 'gene' mode described in detail later in the paper, if a master has no children the script attaches it to a random regular gene, and if a regular gene ends up not having a parent, it is connected from a random master.

In the second phase of the algorithm, we merge any isolated nodes or subgraphs that result from the first phase. It constructs a label ordering of the isolated components by their lowest-index member, then for each adjacent pair adds one edge from a node in the earlier component to one in the later. Because every new link runs from a lower-index to a higher-index node, the global ordering—and thus acyclicity—is preserved while guaranteeing full connectivity of the graph.

Together, these phases construct a minimally-branched, fully connected DAG of N genes with M designated master regulators, providing a scaffold for quantitative edge weight assignment.

In the script, for 'gene' mode, the generated DAG describes the gene regulatory interactions of the GRN, where each node represents a gene and directed edges signify regulatory impact. The graph is constructed such that each node (except designated master regulators) has a configurable `percent` probability of connecting to nodes from earlier layers, allowing for both strict hierarchical and randomly diverging topologies (see Table 1). Genes may have multiple regulators, in other words the GRN is surjective. Edges and edge weights represent interactions and regulatory magnitudes, respectively, and are sampled randomly from a uniform distribution and can be either activating or inhibitory based on a user-defined inhibition percentage `inhibit_pct`. Resulting gene interactions are exported in the format required by SERGIO as described above.

In GRN generation via 'gene' mode, a number of genes designated by the user using parameter `master_regulators` are selected randomly from the gene indices and used as master regulator

Parameter	Description
total_nodes*	Total number of genes (nodes) in the gene regulatory network.
min_per_rank*	Minimum number of nodes per layer (rank) in the DAG.
max_per_rank*	Maximum number of nodes per layer (rank) in the DAG.
min_ranks*	Minimum number of layers (tree depth) in the DAG.
max_ranks*	Maximum number of layers (tree depth) in the DAG.
percent	Probability (%) that a node connects to any previous node instead of strictly following the default sequential parent selection.
inhibit_pct	Probability (%) that an edge represents an inhibitory (negative) interaction.
master_regulators*	Number of nodes designated as master regulators (no incoming edges).
num_celltypes*	Number of cell types to sample master regulator expression profiles across.

Table 1: Parameter descriptions for gene mode DAG generation. * Required parameters.

genes. These genes are excluded from regulatory edge assignment and are functionally the most upstream sources of regulation. For each master regulator, expression values are generated independently for each cell type and written in SERGIO-compatible format to file.

In 'cell' mode, the script generates a DAG that describes the cell differentiation trajectory of the biological system containing the GRN, where each node represents a cell and directed edges and edge weights signify transitions from parent to child cell types and their migration rates, respectively. The graph also utilizes the configurable percent probability of connecting new nodes to nodes from previous tree layers (see Table 2). SERGIO currently does not support multi-parent cell types, in other words two nodes cannot map to the same child node, so cell differentiation systems must be injective. An adjacency matrix is extracted from the graph and migration rates are sampled from a bounded range to emulate biological variability. The range of values is set to 0.8-1 because values near 1 yield desirable differentiation patterns by balancing cell density and simulation efficiency, as recommended in SERGIO's documentation. This matrix is saved in a tab-separated format in accordance to SERGIO requirements.

Parameter	Description
min_per_rank*	Minimum number of nodes per layer (rank) in the DAG.
max_per_rank*	Maximum number of nodes per layer (rank) in the DAG.
min_ranks*	Minimum number of layers (tree depth) in the DAG.
max_ranks*	Maximum number of layers (tree depth) in the DAG.
percent	Probability (%) that a node connects to any previous node instead of strictly following the default sequential parent selection.

Table 2: Parameter descriptions for cell mode DAG generation. * Required parameters.

After running the script, the user has obtained the necessary files for beginning GRN simulation.

Pseudotime Calculator

Running GRN inference methods like OTVelo also requires pseudotimes to describe the pattern of cell development and evolution over time. We chose to simulate pseudotimes on synthetic GRNs using the Slingshot package, which takes clusters of cell data as input and infers cell lineages of the cell types by mapping a minimum spanning tree (MST) over the cell clusters.

The simulator calculates pseudotimes for each cell, which are one-dimensional variables that describe the cell's transcriptional status to the terminal state.

We decided to utilize Slingshot for its robust noise-filtering and reliable prediction of branching cell ancestry. The method incorporates the use of MST generation, covariance-scaled Euclidean distance for identifying cell clusters, and allows the user to incorporate some supervision for the simulated lineages. For example, users can indicate the starting and terminal cell types, which can guide the simulator to a more accurate result.

Moreover, Slingshot is a preferred method of obtaining pseudotimes because its *simultaneous principal curve* method for assigning time-stamps across multiple and branching lineages has been shown to effectively fit continuous, branching trajectories to stable cell pseudotime predictions. This is important as RNA-Seq data is prone to high levels of noise and variation, such as transcriptional bursting and drop-out, due to uncontrolled biological and technical sequencing phenomena. Synthetic data from SERGIO simulations also simulate this noise, so it is important to utilize a framework created to manage these challenges.

Slingshot operates on an `AnnData` object, which is a data structure designed to store high-dimensional biological data. The `AnnData` object is initialized with the simulated gene expression data in the form of a matrix where each row corresponds to a gene and each column corresponds to a cell type.

Moreover, Slingshot requires high-dimensional gene expression data to be first preprocessed into a low-dimensional space for accurate visualization and trajectory inference. This is necessary because dimensionality reduction methods help identify the significant features and variation of the data which prepares the foundations for modeling clear and accurate lineages and pseudotimes. Thus, we apply a log transformation to the gene expression data to stabilize variance and subsequently perform Principal Component Analysis (PCA). We also manually cluster the data points based on SERGIO's sequential simulation of different cell types. This manual clustering approach has been observed to be more effective than methods like K-Means, as SERGIO generates cells in a structured, ordered manner, allowing us to accurately partition the data into distinct cell type bins.

The expression matrix is stored in `Anndata.X`, while additional information such as cell type labels and low-dimensional embeddings are stored in `Anndata.obs` and `Anndata.obsm`, respectively.

In addition to providing the `AnnData` object as input, we also specify the start node to offer further guidance for trajectory inference in certain GRNs. However, this may not always be accurate, as the start node index provided may not align with the index interpreted by Slingshot. Therefore, it is generally more advisable to omit the specification of a start node.

Ground Truth

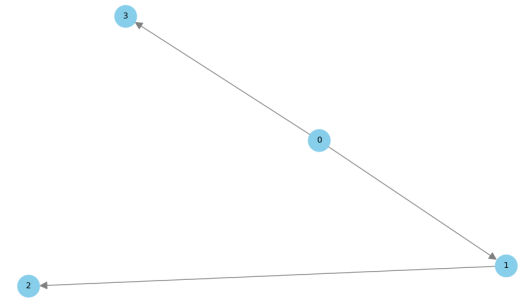
We also created a script `data/postprocessing/grn_to_gt.py` to extract the GRN from the generated gene interactions DAG, following SERGIO's ground truth interactions files. We map all affector genes, including upstream regulators, to target genes for validation of GRN inferences in method testing.

Results

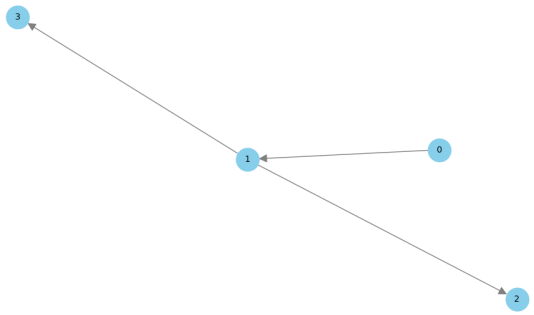
GRN Generation

We were able to produce a script that reliably produces random, simulation-ready GRNs based on user-specified parameters, allowing users to customize network structure and complexity according to their modeling needs.

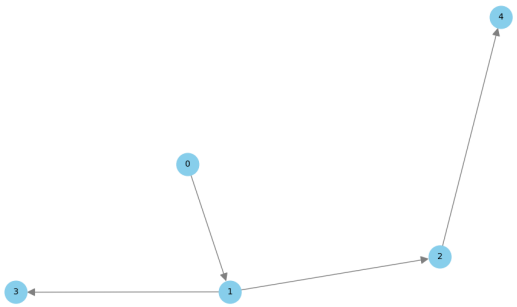
Below are several examples of DAGs with 3-5 cell types randomly generated by our script (Figure 1) using parameters `min_per_rank=1`, `max_per_rank=4`, `min_ranks=1`, `max_ranks=3`, `percent=50` . As the number of total nodes increases, the variation in differentiation pathways increases. The script performs as expected and no two nodes converge on the same target. Although this is a biologically plausible scenario, SERGIO's simulator is not equipped to handle multiple-parent to child cell type relationships. Moreover, no nodes are without edges and the maximum number of outgoing edges fall within the user-specified minimum and maximum values for nodes per level.



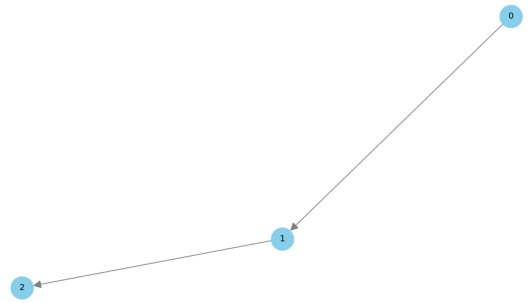
(a) System of 4 cells in a bifurcating lineage



(b) System of 4 cells in a bifurcating lineage that differs from network (a)



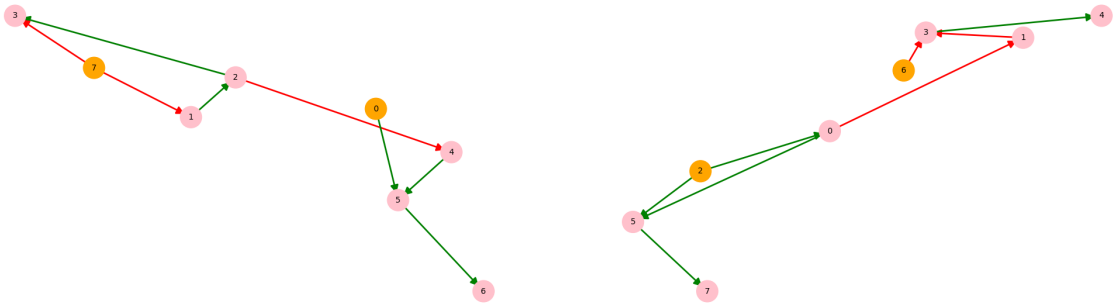
(c) System of 5 cells in a bifurcating lineage



(d) System of 3 cells in one differentiation trajectory

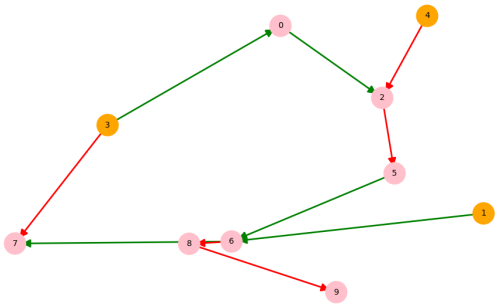
Figure 1: Randomly generated cell differentiation DAG networks from the GRN generation script

Figure 2 shows several randomly generated gene interaction networks with parameters `min_per_rank=2`, `max_per_rank=3`, `min_ranks=1`, `max_ranks=3`. While some generated graphs may exhibit a bias toward predominantly upregulating or downregulating interactions, we observe that the script consistently produces a wide variation of complex gene interaction networks with branching lineages and both one-to-one and multiple-to-one gene interactions.

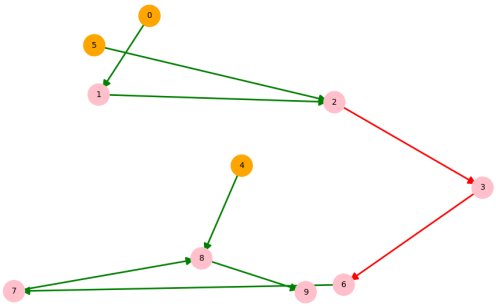


(a) 8 genes, 2 MRs

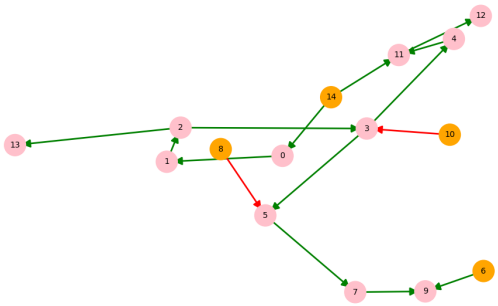
(b) 8 genes, 4 MRs



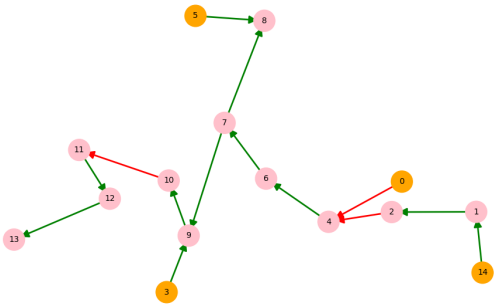
(c) 10 genes, 3 MRs



(d) 10 genes, 3 MRs



(e) 15 genes, 4 MRs



(f) 15 genes, 4 MRs

Figure 2: Examples of generated DAG gene interaction networks with varying parameter values for master regulators. Master regulators are colored in orange and regular genes are colored in pink. Green arrows indicate upregulation, red arrows indicate downregulation

Pseudotime Calculation

The GRN generation process successfully captures pseudotime and GRN dynamics that are consistent with the ground truth gene regulatory network and cell type differentiation trajectories. However, variability in graph structure and inaccuracies in pseudotime estimation can arise, primarily due to the reduced stability of cell type separation in PCA space, which can lead to suboptimal trajectory inference. Through perturbation studies involving modifying PCA projection parameters, the number of master regulators, and the number of cells simulated, we examined how different factors impact the development of a robust synthetic GRN. We found that the variation in their production rate combinations across cell types play a critical role in the stability of the simulated GRN which directly impacts the accuracy of inferred

pseudotimes. Greater diversity in MR expression profiles between cell types can sometimes lead to more distinguishable differentiation trajectories which is consistent with observations and recommendations from SERGIO's experimentation. However, in some perturbations, such as using MR expression profiles with greater dichotomies, we found that cell clusters can be projected to overlap in a reduced space which negatively impacts pseudotime prediction.

We observed this phenomena in the below perturbation studies.

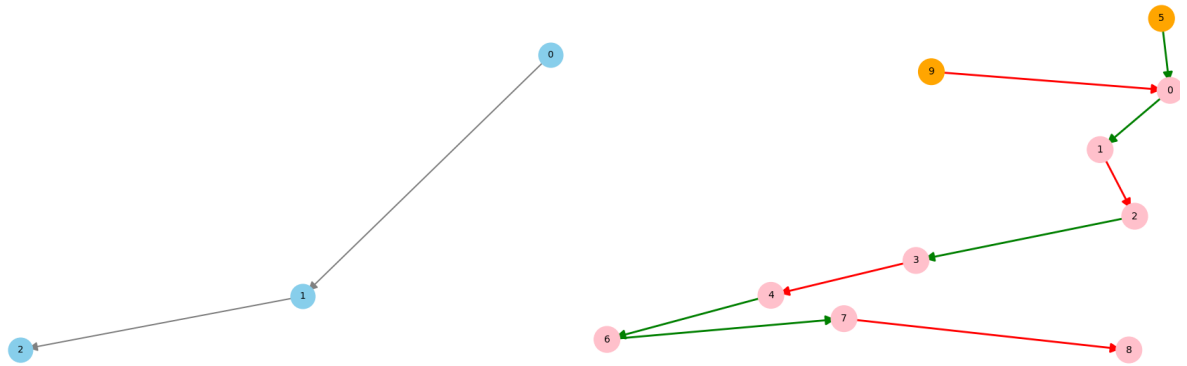
Case Studies

We examined four perturbations for two different setups:

- **(a) Original Generated GRN**: Trials *a* used the original setup without any modifications. We utilized the regulation rates and coefficients generated by the script.
- **(b) Decreasing Principal Components**: We decreased the number of principal components used in the PCA projection to 6 in order to enhance the separation of true biological signal from technical noise, resulting in clearer clustering.
- **(c) Increasing Number of Simulated Cells**: We increased the number of simulated cells from 300 to 500 per cell type to observe if the cell clusters will become more connected in the PCA projection and support more robust downstream trajectory inference.
- **(d) Modifying MR Production Rates**: Using the same MRs introduced in trial (a), we increased the differences in their production rates across cell types, highlighting the variation of regulatory influence across the two lineages.

Perturbation Study 1

The setup for this experiment includes one cell lineage of 3 cells that differentiate into each other. The GRN includes 10 genes with 2 MR. (Figure 3).

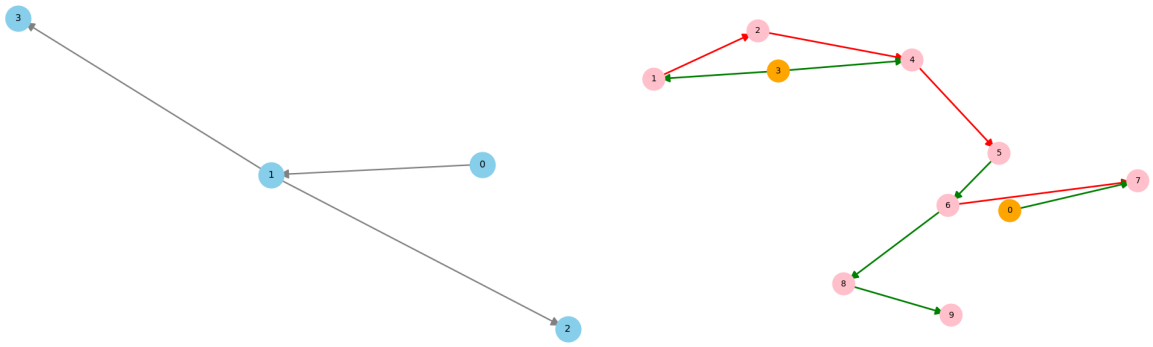


(a) 3 cell types in a one branch differentiation pattern (b) 10 genes with 2 MR

Figure 3: A generated GRN with a one branch cell trajectory, 3 cell types, 10 genes and 2 MR

Perturbation Study 2

The setup for this experiment includes one bifurcating cell trajectory of 4 cells that differentiate into each other. The GRN also includes 10 genes with 2 MR, with the interaction edge weights and migration rates generated using the script. (Figure 4)



(a) 4 cell types in a bifurcation differentiation pattern (b) 10 genes with 2 MR

Figure 4: A generated GRN with a one branch cell trajectory, 3 cell types, 10 genes and 2 MR

Pseudotime Simulation Performance

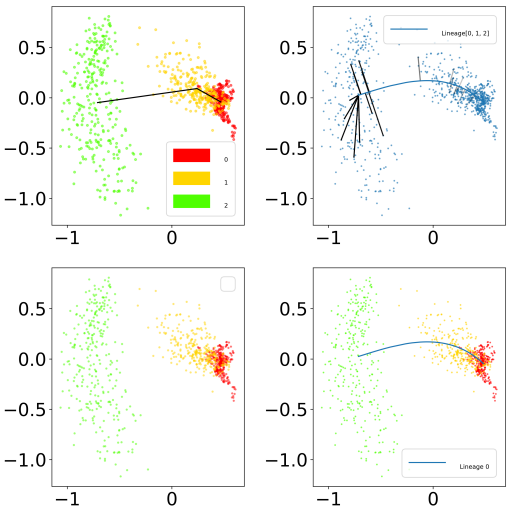
The resulting Slingshot pseudotime calculation process plots, represented in Figure 5 and Figure 7, consists of four panels describing the trajectory inference process:

- **(i) Top-left panel**: Displays the UMAP projection with cells colored by their annotated cell types.
- **(ii) Top-right panel**: Shows the initial lineage MST graph overlaid on the cell clusters, representing Slingshot's preliminary linkage between cell states. The dark lines represent the distance between a cell's true position in reduced space and its projected position along the inferred lineage trajectory. Larger lines indicate greater misalignment, suggesting inaccuracies in the pseudotime prediction or lineage inference.
- **(iv) Bottom-right panel**: Illustrates the fitted lineage curves, capturing the smoothed trajectories that model potential developmental pathways.

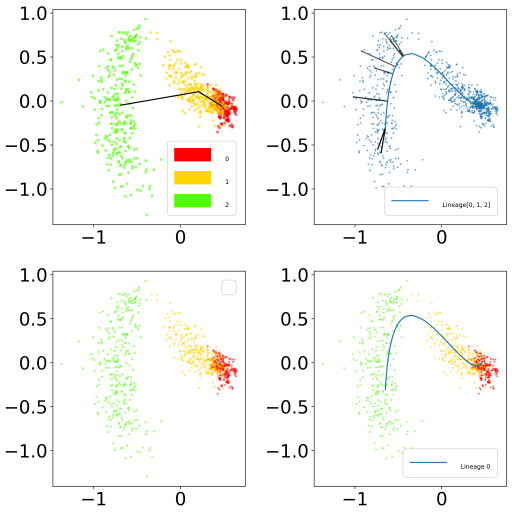
For perturbation study 1, we observed the following results in pseudotime calculation performance using Slingshot (Figure 5):

- **(a) No modification**: Slingshot was able to capture the correct number of branches and directionality of the cell differentiation lineage. However, the projected line does not follow the slope of the cell clusters very well, which indicates that the pseudotime and cell lineage fitting is not as accurate as it can be.
- **(b) Decreasing Principal Components to 6**: When the number of principal components used in the PCA projection decreased to 6, we observe less misalignment errors and a predicted cell trajectory that aligns with the cell clusters.
- **(c) Increasing Number of Simulated Cells to 500**: When we increase the number of simulated cells to 500 per cell, we found that there was slightly less sparsity and empty spaces across the cell cluster projections, though this may be mostly the product of having more cell density and not necessarily that the unrepresented area between clusters in the reduced space is being simulated and accounted for. The resulting predicted cell trajectory is more fitted to the directionality of the cell clusters.

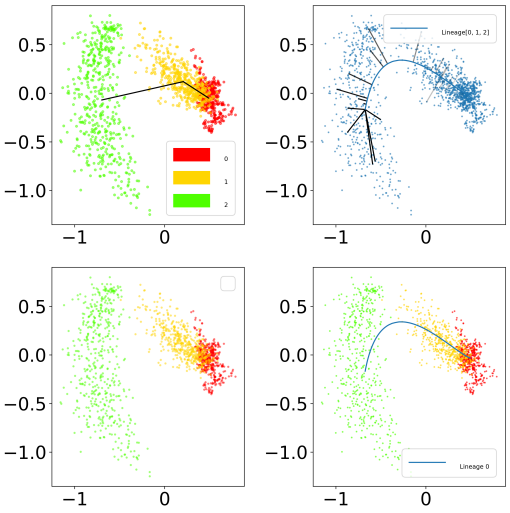
- **(d) Modifying MR Production Rates**: This setup resulted in a PCA embedding where the cell clusters are arranged differently than before and were narrower and sharper. This created a more defined cell trajectory. Slingshot was able to fit an accurate cell lineage and pseudotime to this PCA projection.



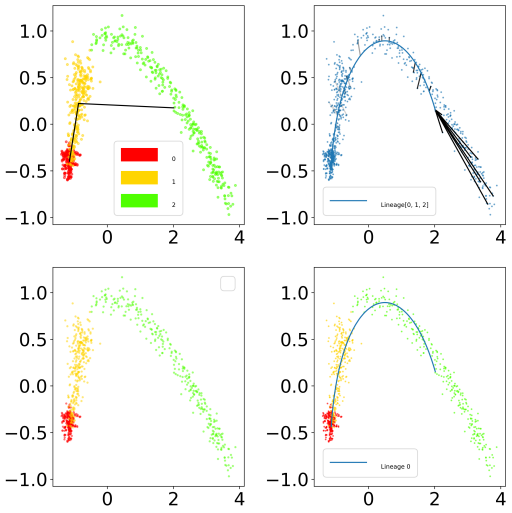
(a) Synthetic GRN with 10 genes and 2 master regulators, no modifications



(b) Synthetic GRN with 10 genes and 2 master regulators, plotted using PCA with 6 principal components



(c) Synthetic GRN with 10 genes and 2 master regulators, simulated with 500 cells per cell type

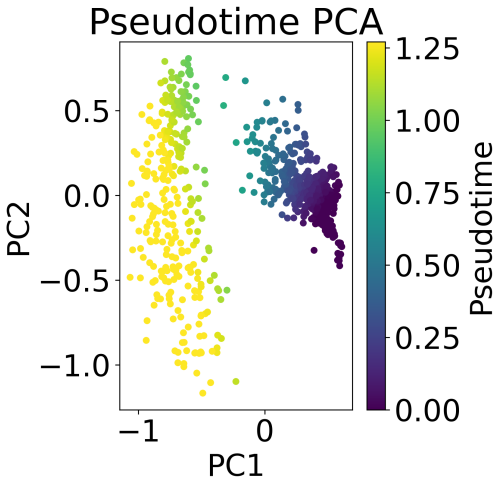


(d) Synthetic GRN with 10 genes and 2 master regulators with altered migration rates

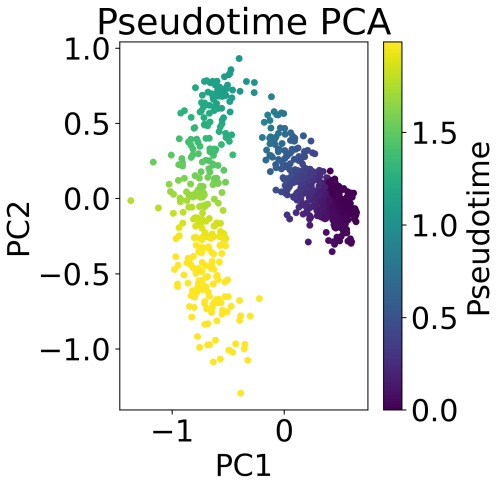
Figure 5: Pseudotime simulation from perturbation study 1

Figure 6 displays the corresponding final pseudotime results for the perturbation studies described above. For all setups, the pseudotimes were mostly accurate and aligned with the cell clusters. For setup a, the pseudotimes are measured in smaller increments, suggesting higher resolution across pseudotime which provides insight on more fine-grain gene expression changes.

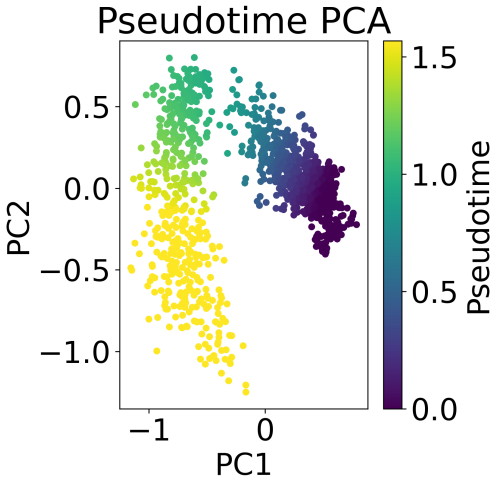
The range of pseudotimes is also proportional to the cell trajectory, so a larger range corresponds to a longer cell trajectory simulated.



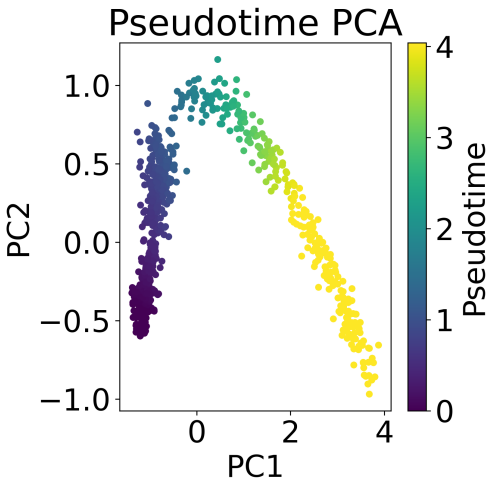
(a) Synthetic GRN with 10 genes and 2 master regulators without manual adjustment



(b) Synthetic GRN with 10 genes and 2 MR, plotted using PCA with 6 principal components



(c) Synthetic GRN with 10 genes and 2 MR, simulated with 500 cells per cell type



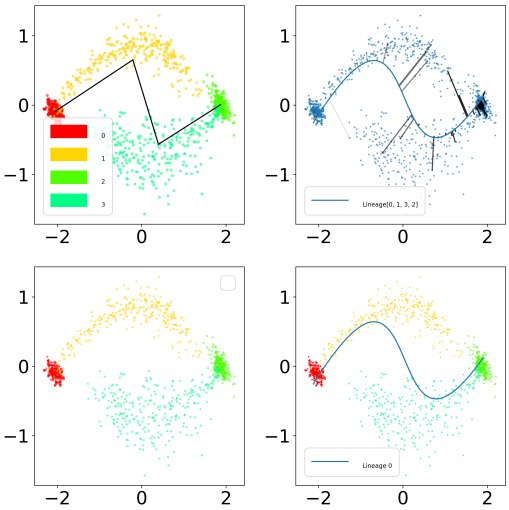
(d) Synthetic GRN with 10 genes and 2 master regulators with altered migration rates compared to (a)

Figure 6: Pseudotimes from perturbation study 1

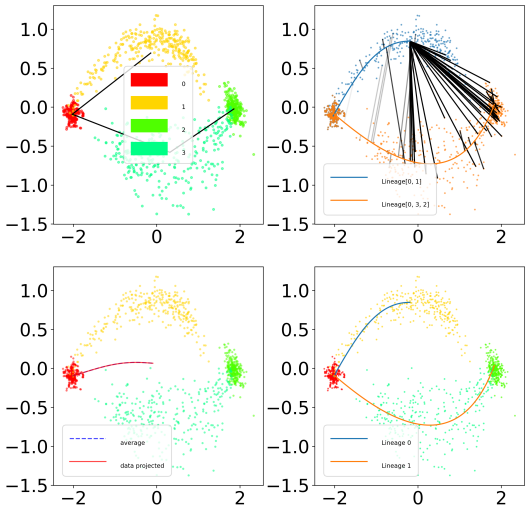
For perturbation study 2, we observed the following results in pseudotime calculation performance using Slingshot (Figure 7):

- **(a) No modification**: Slingshot was unable to capture the correct branching cell trajectory. We hypothesize that this is due to the lack of cells in some areas of the reduced space for the bottom trajectory that makes it challenging for Slingshot to detect two branches (Figure 8)

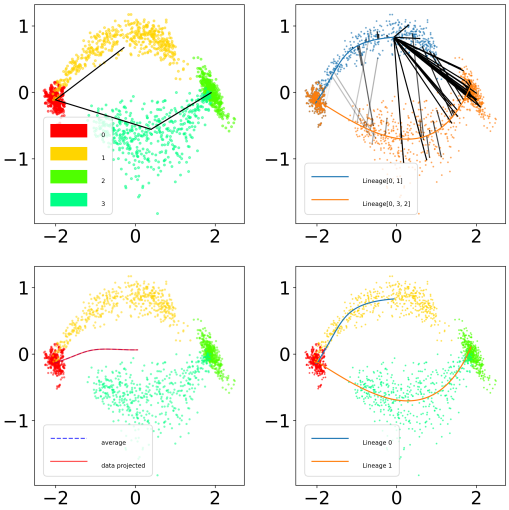
- **(b) Decreasing Principal Components to 6**: When the number of principal components used in the PCA projection decreased to 6, Slingshot was able to identify the branching cell lineages. This makes sense because reduced dimensionality focuses the algorithm on the most informative variance and filters out noise so that Slingshot can fit its curves to the true branching geometry.
- **(c) Increasing Number of Simulated Cells to 500**: Similar to perturbation study 1, when we increase the number of simulated cells to 500 per cell we found that Slingshot is able to capture the cell lineage more accurately.
- **(d) Modifying MR Production Rates**: Unfortunately, this perturbation resulted in cell clusters that overlap with each other, forming a cyclic pattern that is not reflected in biology nor the ground truth cell lineage. Somehow, Slingshot still predicts that there are two cell differentiation trajectories, though they are not fit to the cell clusters.



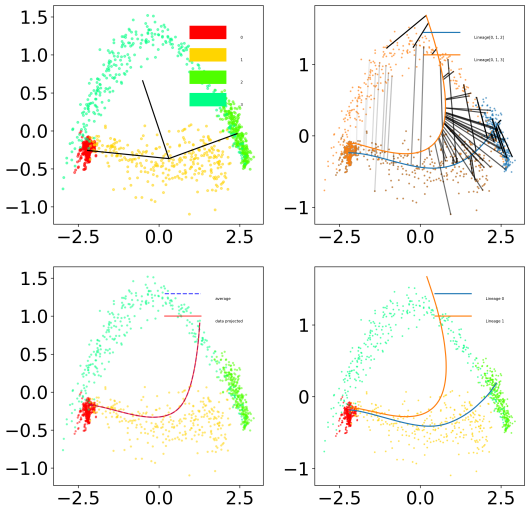
(a) Synthetic GRN with 10 genes and 2 master regulators where cells differentiate in a bifurcation pattern, no modifications to data



(b) Synthetic GRN with 10 genes and 2 master regulators where cells differentiate in a bifurcation pattern, plotted using PCA with 6 principal components

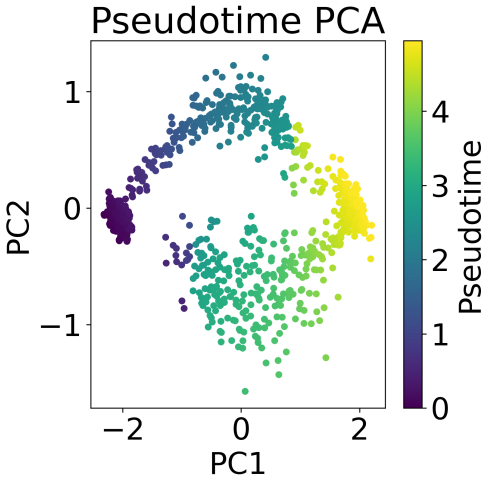


(c) Synthetic GRN with 10 genes and 2 master regulators where cells differentiate in a bifurcation pattern, simulated with 500 cells per cell type

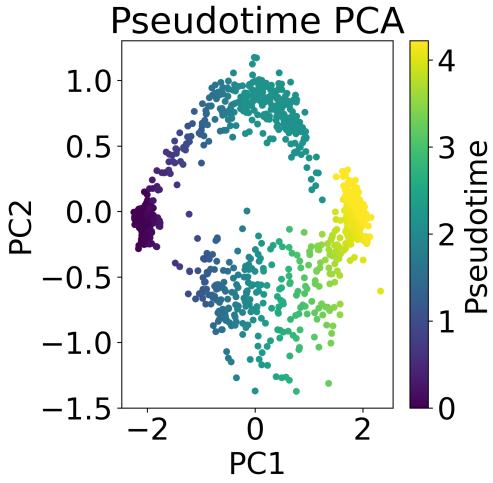


(d) Synthetic GRN with 10 genes and 2 master regulators where cells differentiate in a bifurcation pattern with altered migration rates

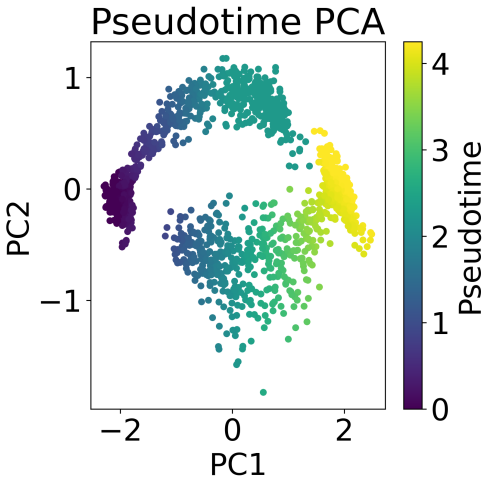
Figure 7: Pseudotime simulation from perturbation study 2



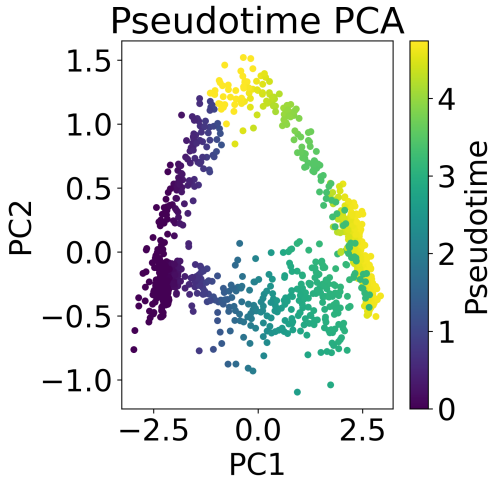
(a) Synthetic GRN with 10 genes and 2 master regulators where cells differentiate in a bifurcation pattern without manual adjustment



(b) Synthetic GRN with 10 genes and 2 master regulators where cells differentiate in a bifurcation pattern, plotted using PCA with 6 principal components



(c) Synthetic GRN with 10 genes and 2 master regulators where cells differentiate in a bifurcation pattern, simulated with 500 cells per cell type



(d) Synthetic GRN with 10 genes and 2 master regulators where cells differentiate in a bifurcation pattern with altered migration rates compared to (a). Clusters assigned a later pseudotime from the bottom branch overlap with the top branch in the reduced space.

Figure 8: Pseudotimes from perturbation study 2

Discussion

Limitations in Simulating Cycles with SERGIO and Slingshot

Both SERGIO and Slingshot, as well as other GRN simulators, continue to have a difficult time modeling cycles in GRNs and cyclic cell differentiation trajectories, such as those seen in feedback-driven developmental loops [3, 6, 8]. This limitation reduces the realism of the synthetic model and users should anticipate that experiments ran with synthetic data from this pipeline may fail to capture critical aspects of biological systems. With SERGIO, multiple genes are allows to regulate a target gene in the GRN, but the simulation is restricted to cell differentiation patterns where a single parent gives rise to one or more children, but not where multiple lineages converge into a shared fate. Thus, some biologically relevant processes such as lineage convergence cannot be accurately modeled.

The Role of Master Regulators in Driving Gene Expression Dynamics

MR, as described before, are key gene regulators that heavily influence the identity of a cell by orchestrating gene regulation and expression. In the perturbation study, manually altering the migration rates of MR to create a greater dichotomy between the rate of each MR across cell types changed the PCA projection of the cells significantly, suggesting that the change in MR rates heavily impacted gene expression simulation and cell clustering. In Perturbation Study 1, making this change decreased the sparsity between cell clusters which most likely enabled a more accurate pseudotime fitting. This observation underscores the dominant influence of MRs in shaping the gene regulatory landscape and guiding cell differentiation.

Importance of Cell Number in Accurate Pseudotime Inference

Pseudotime inference methods rely on the resolution of single-cell sampling to reconstruct developmental trajectories. We hypothesize that the reason Slingshot did not identify two branching cell lineages in Figure 7 setup *a* was because there were too few cells simulated, resulting in poor pseudotime estimates due to sparse or uneven coverage of the expression space. Key transitional cell states may be under-sampled or missed entirely, leading to errors in trajectory reconstruction or incorrect branching structures. Figure 5 example *c* demonstrates how increased cells simulated per cell type results in fuller cell clusters and less sparsity and empty spaces in dimensionality reduction plots. This resulted in a more accurate cell trajectory prediction in Figure 5.

Significance of Optimal Principal Component Selection in Pseudotime Inference

Exploring optimal component selection enforces a balance between information retention and noise suppression. We observed that in both perturbation studies, decreasing the number of principal components used in the PCA pseudotime projections by filtering out low-variance noise and concentrating on the dominant biological signals, which led to clearer branch separation (Figure 8, example *b*) and smoother, more interpretable trajectories.

Future Work

Future directions for this work involve expanding the current perturbation studies to systematically investigate how varying the number of simulated cells, MRs, and principal components impacts pseudotime inference, lineage reconstruction, and GRN inference accuracy. Understanding the relationship between these parameters and accurate pseudotime prediction could provide valuable insights into the optimal conditions for simulator performance and support the generation of accurate, high-quality synthetic data.

Moreover, extending this pipeline to include functionality for simulating cyclic developmental trajectories and gene networks would be critical to generating more robust and realistic synthetic data. However, this development requires further work on the GRN and Pseudotime simulators themselves, which is slightly out of scope of the synthetical data pipeline development.

Furthermore, it is important that we design experiments to validate the realism of the synthetic GRNs and cell trajectories. We hope to simulate real datasets and explore real cell lineage patterns to compare to our synthetic data.

We also hope to produce results of using the synthetic data on GRN reconstruction methods and attaching the inference methods validation step to the pipeline for more streamlined use.

Acknowledgments

This thesis work would not have been possible without generous support from the Brown community. I thank my readers and advisors Dr. Ritambhara Singh and Dr. Bjorn Sandstede for their continuous guidance, knowledge, and compassionate mentorship. I also thank Victoria Antonetti for working diligently with me to create the pipeline and always answering my questions. I also want to thank Dr. Wenjun Zhao for her helpful insights and encouraging positivity which are unparalleled and have driven this project forward more times than I can count.

Thank you also to Dr. Eric Ewing, for always patiently helping me with my questions and sharing his understanding of computer science with me. I am so grateful for his mentorship and kind guidance, both in academics and in my student leadership.

I would also love to thank the many staff members of Brown that help me everyday to study and learn as much as I am able to do. I am grateful for the dining hall workers, custodians, shuttle bus drivers, and administrative staff for their constant support that keeps our community running.

Thank you to Helen, Seehanah, Julia, Jane, Maddy, Sarah, Jenny, Faith, Christine for being my friends throughout my college journey, for bearing with me and believing in me. Anything I have accomplished has been because you have lifted me upon your shoulders.

Lastly, thank you to God and to the Christian community at Brown for their companionship thoughout college. Thank you to my amazing family–my younger sister Loretta and my little brother Ben, and my supportive parents. Thank you for instilling in me hope, values, and a love for learning, as well as feeding me really great food and challenging me to always look forward.

Bibliography

[1] P. Badia-i-Mompel, L. Wessels, S. Müller-Dott, et al. "Gene regulatory network inference in the era of single-cell multi-omics". In: *Nature Reviews Genetics* 24 (2023), pp. 739–754. DOI: 10.1038/s41576-023-00618-5. URL: https://doi.org/10.1038/s41576-023-00618-5.

[2] Samuel S Chan and Michael Kyba. "What is a Master Regulator?" In: *Journal of Stem Cell Research & Therapy* 3 (2013), p. 114. DOI: 10.4172/2157-7633.1000e114.

[3] Payam Dibaeinia and Saurabh Sinha. "SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks". In: *Cell Systems* 11.3 (2020), 252–271.e11. ISSN: 2405-4712. DOI: 10.1016/j.cels.2020.08.003. URL: https://www.sciencedirect.com/science/article/pii/S2405471220302878.

[4] Cansu Ergen, Guannan Xing, Chen Xu, et al. "Consensus prediction of cell type labels in single-cell data with popV". In: *Nature Genetics* 56 (2024), pp. 2731–2738. DOI: 10.1038/s41588-024-01993-3.

[5] Malte D Luecken and Fabian J Theis. "Current best practices in single-cell RNA-seq analysis: a tutorial". In: *Nature Reviews Genetics* 20.8 (2019), pp. 435–450. DOI: 10.1038/s41576-018-0088-9.

[6] Kelly Street et al. "Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics". In: *BMC Genomics* 19.1 (2018), p. 477. DOI: 10.1186/s12864-018-4772-0.

[7] W. Zhao et al. "Optimal transport reveals dynamic gene regulatory networks via gene velocity estimation". In: *bioRxiv [Preprint]* (Sept. 2024). 2024.09.12.612590. DOI: 10.1101/2024.09.12.612590.

[8] Y. Zinati, A. Takiddeen, and A. Emad. "GRouNdGAN: GRN-guided simulation of single-cell RNA-seq data using causal generative adversarial networks". In: *Nature Communications* 15 (2024), p. 4055. DOI: 10.1038/s41467-024-48516-6. URL: https://doi.org/10.1038/s41467-024-48516-6.