Deep Learning Models of Cellular Decision-Making Using Single-Cell Genomic Data

by

Mehrshad Sadria

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Applied Mathematics

Waterloo, Ontario, Canada, 2024

© Mehrshad Sadria 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:

- **Christina Leslie**
  Professor, Computational and Systems Biology Program, Memorial Sloan Kettering Cancer Center, New York, NY, USA

Supervisor:

- **Anita Layton**
  Professor, Department of Applied Mathematics Faculty of Mathematics, University of Waterloo

Member:

- **Gary Bader**
  Professor, Department of Professor of Molecular Genetics and Computer Science, The Donnelly Centre, University of Toronto

Internal Member:

- **Brian Ingalls**
  Professor, Department of Applied Mathematics Faculty of Mathematics, University of Waterloo

Internal-External Member

- **Yaoliang Yu**
  Associate Professor, David R. Cheriton School of Computer Science, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapter 3 introduces Fatecode, the first algorithm developed during my PhD. Fatecode is a computational method designed to predict cell fate regulators solely from scRNA-seq data. By learning a latent representation through a deep learning-based classification-supervised autoencoder, it enables in silico perturbation experiments. These experiments identify genes that, when perturbed, can shift cell type distributions, either increasing or decreasing the population of specific cell types. The research titled *Fatecode Enables Cell Fate Regulator Prediction Using Classification-Supervised Autoencoder Perturbation* was conducted with Prof. Anita T. Layton, Prof. Sidhartha Goyal, and Prof. Gary Bader and has been published in Cell Reports Methods (1).

Chapter 4 presents CLERA, a novel computational framework for uncovering dynamic models and identifying active gene programs from single-cell RNA sequencing data. By combining a supervised autoencoder with Sparse Identification of Nonlinear Dynamics, CLERA uses prior knowledge to extract low-dimensional representations and reveal the driving forces behind cellular processes. It pinpoints central genes, reconstructs gene expression dynamics, and captures key regulatory genes and temporal patterns across various cell types using personalized page rank. The research titled *Discovering Governing Equations of Biological Systems through Representation Learning and Sparse Model Discovery* was conducted in collaboration with Vasu Swaroop (an undergrad student in our lab at the time of the study), and is currently under review in *Nucleic Acids Research* journal (2).

Chapter 5 introduces FateNet, a computational approach that integrates dynamical systems theory with deep learning to explore the timing of cell decision-making using scRNA-seq data. By leveraging information about normal forms and scaling behavior near bifurcations FateNet predicts when cell decisions occur with greater accuracy than conventional methods, while offering a

qualitative understanding into the specific type of bifurcation the system is likely to experience. The work titled *FateNet: An Integration of Dynamical Systems and Deep Learning for Cell Fate Prediction* was conducted with Dr. Thomas Bury and published in Bioinformatics in 2024 (3).

Chapter 6 delves into scVAEDer's capabilities, the first scalable deep-learning model that integrates autoencoders and deep diffusion models using scRNA-seq data. This approach learns a latent representation of data, which captures both global patterns and local variations. The chapter demonstrates scVAEDer's accuracy by showing its ability to generate new high quality scRNA-seq data, predict perturbation effects across cell types, track gene expression changes during dedifferentiation, and identify key regulators in biological processes. The work titled *The Power of Two: Integrating Deep Diffusion Models and Variational Autoencoders for Single-Cell Transcriptomics Analysis* was conducted with Prof. Antia Layton and is currently under review at the *Genome Biology* journal (4).

Chapter 7 highlights Deep Lineage, a novel deep-learning method for analyzing time-series scRNA-seq data with matched lineage-tracing data. Our method accurately predicts early cell fate biases and gene expression profiles at different time points within a clone, surpassing current state-of-the-art methods in fate prediction accuracy. Also, through in silico perturbations in cellular reprogramming and hematopoiesis data, we show that Deep Lineage accurately models dynamic multicellular responses while identifying key genes and pathways associated with cell fate determination. This research was conducted in collaboration with Allen Zhang (an undergrad student in our lab) and Prof. Gary Bader. The work titled *Deep Lineage: Single-Cell Lineage Tracing and Fate Inference Using Deep Learning* is currently under review at *Cell Systems* (5).

Abstract

Cellular decision-making is a complex process that governs how cells respond to their environment, differentiate, or commit to specific fates. Understanding the mechanisms that drive these decisions is critical for advancing fields such as regenerative medicine, cancer research, and developmental biology. The ability to investigate cellular decision-making at the single-cell level, particularly through high-resolution technologies like single-cell RNA sequencing (scRNA-seq), opens new doors for dissecting the intricate regulatory networks that guide cell fate. With this information, we can better predict how cells transition between states, respond to external stimuli, or contribute to disease progression.

In this thesis, the objective is to develop different deep learning methods for analyzing cellular decision-making processes using scRNA-seq data. These computational approaches aim to reveal the molecular programs governing cell fate and to predict how gene perturbations can alter cellular outcomes. Central to this effort is the finding of the accurate latent representation of high-dimensional data. By learning meaningful and compressed representations, we can improve the interpretability and accuracy of downstream analyses, including clustering, trajectory inference, gene regulatory network reconstruction, and the identification of key gene regulators. Also, these representations can be used further to generate high quality gene expression data.

We developed several methods to capture the dynamics of cellular decision making. **Fatecode**, a classification-supervised autoencoder, predicts cell fate regulators by learning latent representations from scRNA-seq data. It enables in silico perturbation experiments to identify genes that can shift cell type distributions, offering insights into key regulators of differentiation. **CLERA** combines a supervised autoencoder with Sparse Identification of Nonlinear Dynamics (SINDy) to uncover dynamic models and active gene programs. By leveraging prior knowledge, CLERA extracts low-dimensional representations, reconstructs gene expression dynamics, and identifies central regulatory genes in various cell types. **FateNet** integrates dynamical systems theory with deep learning to predict the timing of cell decisions. By analyzing scaling behavior near bifurcations, FateNet provides insights into key decision points in cellular processes,

improving our understanding of these transitions. **scVAEDer** is a generative deep-learning model that integrates autoencoders with deep diffusion models. It captures both global patterns and local variations in scRNA-seq data, enabling high-quality synthetic data generation, prediction of perturbation effects, and tracking of gene expression changes over time. Finally, **Deep Lineage** analyzes time-series scRNA-seq data alongside lineage-tracing information, predicting early cell fate biases and gene expression profiles with high accuracy. Through in silico perturbations, Deep Lineage reveals dynamic multicellular responses over time and identifies key genes and pathways involved in cell fate decision making.

I hope the methods presented in this thesis can advance the integration of deep learning into the study of cellular dynamics, helping to identify important gene programs and regulators involved in decision-making processes. These computational tools provide valuable resources for advancing research in regenerative medicine, developmental biology, and disease modeling. By improving our ability to model and predict cellular behavior, these methods can contribute to more effective therapies and a deeper understanding of the molecular processes driving development and disease.

Acknowledgments

First and foremost, I want to express my deepest gratitude to my advisors, Prof. Anita Layton and Prof. Gary Bader, for their amazing support and guidance throughout my PhD. Anita, it feels like just yesterday we had our first video call, where you asked why I wanted to go to grad school, and I confidently replied, “To work on fundamental problems and win a Nobel Prize!” Since then, I’ve been incredibly fortunate to have you as my mentor. You’ve not only inspired me to tackle important questions but have also been a true friend, supporting me both academically and personally in life. Thank you for trusting me and giving me the freedom to pursue my research interests and for all your support.

Gary, your invaluable mentorship has profoundly shaped this thesis. I am deeply thankful for your belief in me, your guidance in conceptualizing and designing various parts of my projects, and for always being willing to dedicate personal time even on weekends to work with me. You were there to listen when I needed advice, both scientifically and in life. The impact both of my advisors have had on my scientific journey and personal development is immeasurable. They’ve helped me stay committed to high-quality research and have played a key role in shaping my identity as a researcher while offering incredible support and friendship along the way.

I am also grateful to my committee members, Prof. Brian Ingalls, Prof. Yaoliang Yu, and Prof. Christina Leslie. Thank you for your insightful questions, your continuous support, and for dedicating your time to engage with my work and guide me along this path.

To my amazing collaborators: Dr. Tom Bury, Prof. Sid Goyal, Prof. Maryam Faiz, Vasu, Allen, and Justine (and many others). Working with you has been both inspiring and rewarding. I truly enjoyed spending time with each of you, and your contributions have significantly shaped this thesis. I would also like to acknowledge Dr. Hamidreza Sepangi and Dr. Jafari, my undergraduate advisors, for being among my first mentors and laying the foundation of my scientific growth. A special thanks to Prof. Laurence Pelletier at the Lunenfeld-Tanenbaum Research Institute and Prof. Jan Krumsiek at Cornell University for graciously hosting me in their labs and allowing me to collaborate with their research teams. Your scientific insights were

invaluable to my work. I would also like to extend my gratitude to the members of the Layton and Bader groups for the many stimulating discussions, shared experiences, and unwavering support through the ups and downs of graduate life.

I also want to express my appreciation to the staff of the Applied Mathematics department for their help in navigating the administrative challenges of academia. A particular thanks to Lopez from the cleaning unit—our Friday evening chats always lifted my spirits. I'd also like to thank the SJU crew for their kindness and the joy they brought to daily lunches (SJU forever).

Over the years, I've crossed paths with many remarkable individuals who have made my life richer and brighter. To my dear friends Dimcho, Russel, Nima, Farbod, Aidin, Mohammad, Panos, Charitini, Yusuf, Ekin, Amir, Baris, Sepehr, Antonina, Moji, William, Mahtab, Niloo, Sahand and Saman, as well as many others whose names I cannot list here but who remain close to my heart. Thank you for the laughter, support, and unforgettable memories. And my partner-in-crime, Shervin Hakimi... From skating sessions and late-night workouts to chasing solar eclipses, we've done so much together. Your brilliant mind and adventurous soul have made every moment together awesome. I'm incredibly lucky to have you as a friend, and your presence has meant a lot to me. I am also grateful to Alireza Vafa, who has been like an older brother to me. Alireza, you've taught me so much—not just about celestial mechanics during high school but also about kindness and how to be a better person. Few people bring me as much joy as you do during our endless conversations about astronomy, photography, and everything else. To Pat, Gord, and your wonderful family, thank you for your kindness, support, and love over the years. You have become family to me, and I am thankful to have you in my life. I also want to thank Marybel for your endless support and continued encouragement. I am incredibly thankful to have you by my side.

Lastly, I want to express my deepest appreciation to my family—my mom, dad, uncles, aunts, and cousins. You are all amazing and I wouldn't have been here without your support.

Dedication

To my mom, with deep gratitude for her unwavering love, support, and encouragement throughout this journey.

Table of Contents

Examining Committee Membership ii
Author's Declaration iii
Statement of Contributions iv
Abstract vi
Acknowledgments viii
Dedication x
List of Figures xviii
List of Tables xxxviii
List of Abbreviations xli
Chapter 1 Introduction 1
Chapter 2 Computational methods 14
2.1 Machine Learning 14
2.2 Supervised Learning 14
2.2.1 Classification 14
2.2.2 Regression 15
2.3 Unsupervised Learning 15
2.3.1 Modeling the Data Distribution and Sampling 16
2.3.2 Revealing the Underlying Structure of the Data 17
2.3.3 Clustering 17

2.3.4 Dimensionality Reduction 18
2.4 Self-Supervised Learning 19
2.4.1 Representation Learning 21
2.5 The Challenge of Interpretability in Deep Neural Networks 26
2.5.1 Koopman Theory 27
2.5.2 Sparse Identification of Nonlinear Dynamics 28
2.6 Dynamical Systems and the Theory of Bifurcation 29
2.6.1 Bifurcation Theory 29
2.6.2 Core Bifurcation Types in Dynamical Systems 30
2.7 Single-cell genomics 32
Chapter 3 Fatecode 36
3.1 Overview 37
3.2 Introduction 37
3.3 Results 38
3.3.1 Fatecode method overview 38
3.3.2 Optimizing model architecture and hyperparameters 41
3.3.3 Fatecode accurately detects known regulators from simulated scRNA-seq data 42
3.3.4 Fatecode identifies known cell fate regulator genes in mouse hematopoiesis 45
3.3.5 Fatecode detects important regulators in cell differentiation and lineage commitment in zebrafish 48
3.3.6 Fatecode identifies cell fate regulators in mouse hippocampus development 50
3.4 Discussion 53

3.5 Method 55
3.5.1 Deep representation learning 55
3.5.2 VAE 55
3.5.3 CVAE 56
3.5.4 Overall network architecture of Fatecode 56
3.5.5 Classification 57
3.5.6 Identifying key regulators in cell differentiation 57
3.5.7 Data visualization 60
3.5.8 Differential expression analysis 60
3.5.9 Data preprocessing 60
3.5.10 Data availability 60
3.5.11 Code availability 61
Chapter 4 Discovering Governing Equations of Biological Systems through Representation Learning and Sparse Model Discovery 62
4.1 Overview 62
4.2 Introduction 63
4.3 Result 65
4.3.1 Discovery of Dynamical Systems and Gene Programs from Simulated Data 65
4.3.2 CLERA Uncovers Dynamics and Gene Programs in Pancreatic Development 68
4.3.3 CLERA Reveals Central Genes and Dynamics in Hematopoietic Differentiation 71
4.4 Discussion 74
4.5 Methods 77

4.5.1 Autoencoders 77
4.5.2 SINDy and SINDy autoencoder 77
4.5.3 Overview of CLERA 78
4.5.4 Data Preprocessing 80
4.5.5 Components of CLERA Objective function 81
4.5.6 Modelling choices and training 82
4.5.7 Personalized Pagerank 84
4.5.8 Simulated data 86
4.5.9 Data availability 87
4.5.10 Code availability 87
4.5.11 Competing interests 87
Chapter 5 FateNet: an integration of dynamical systems and deep learning for cell fate prediction 88
5.1 Overview 88
5.2 Introduction 89
5.3 Results 93
5.3.1 Bifurcation prediction in a simple gene regulatory network 93
5.3.2 Bifurcation prediction in a large simulated gene regulatory network 94
5.3.3 Bifurcation prediction in biological data 97
5.3.4 Effect of gene knockout / over-expression 102
5.4 Discussion 104
5.5 Methods 106

5.5.1 Generation of training data for FateNet 106
5.5.2 FateNet architecture and performance 107
5.5.3 Applying the model to pseudotime data 110
5.5.4 Alternative methods for bifurcation prediction 111
5.5.5 Simple model for a gene regulatory network 111
5.5.6 SERGIO simulation 112
5.5.7 Analysis of scRNA-seq data 112
5.5.8 Data availability 113
5.5.9 Code availability 113
Chapter 6 The Power of Two: integrating deep diffusion models and variational autoencoders for single-cell transcriptomics analysis 114
6.1 Overview 114
6.2 Introduction 115
6.3 Results 117
6.3.1 Method outline 117
6.3.2 scVAEDer generates novel realistic scRNA-seq data 119
6.3.3 scVAEDer models the transition between cells 120
6.3.4 scVAEDer predicts perturbation responses using feature manipulation 122
6.3.5 Disease progression and master regulators 124
6.4 Discussion 127
6.5 Methods 128
6.5.1 Variational autoencoder (VAE) 128

6.5.2 Deep Diffusion Model (DDM) 129
6.5.3 Interpolation 131
6.5.4 Model architecture and training process 132
6.5.5 Exponential Moving Average (EMA) 133
6.5.6 Variance Scheduler 134
6.5.7 For visualization, reduced dimension data 134
6.5.8 Data preprocessing 134
6.5.9 Comparison with other methods 134
6.5.10 Data availability 135
6.5.11 Code availability 135
6.5.12 Competing interests 135
Chapter 7 Deep Lineage: Single-Cell Lineage Tracing and Fate Inference Using Deep Learning 136
7.1 Overview 136
7.2 Introduction 137
7.3 Results 138
7.3.1 Model Overview 138
7.3.2 Deep Lineage accurately predicts gene expression and early cell fate bias in mouse hematopoiesis 140
7.3.3 Deep Lineage accurately predicts gene expression and early cell fate bias in regeneration 145
7.3.4 Deep Lineage can identify key genes in cellular reprogramming and generate realistic in silico perturbation experiments 148

7.4 Methods 151
7.4.1 Autoencoder for latent embedding of scRNA-seq data 151
7.4.2 RNN models for gene expression and early cell fate prediction 151
7.4.3 Data preprocessing 154
7.4.4 Modelling choices and training 155
7.4.5 Comparison with existing methods 156
7.4.6 Simulating and Evaluating In Silico Perturbations 157
7.4.7 Hardware used 157
7.4.8 Data availability 158
7.4.9 Code availability 158
Chapter 8 Conclusions and future directions 159
8.1 Summary of Contributions 159
8.2 Future direction 161
References 165
APPENDICES 191
A. Appendix to chapter 3 191
B. Appendix to chapter 4 197
C. Appendix to chapter 5 211
D. Appendix to chapter 6 220
E. Appendix to chapter 7 229

List of Figures

Figure 1-1: The natural world showcases remarkable examples of longevity and potential immortality. (a) The bowhead whale, capable of living over 200 years, thrives in Arctic waters. (b) The Aldabra giant tortoise, known for its lifespan of up to 250 years. (c) The Greenland shark, one of the longest-living vertebrates, can live up to 500 years in the deep ocean [Photo credit: Hemming1952]. (d) The Immortal Jellyfish (Turritopsis dohrnii) uniquely reverts to its juvenile form through transdifferentiation, effectively restarting its life cycle and achieving a form of biological immortality [Photo credit: Chen Yiming]..... 2

Figure 1-2: Throughout history, the quest for immortality has captivated human imagination, as depicted in various myths and legends. (a) The Epic of Gilgamesh illustrates an early tale of seeking eternal life. (b) Qin Shi Huang, the First Emperor of China, pursued the elixir of life. (c) Persian mythology tells of King Jamshid's long reign and eventual downfall due to arrogance. (d) The mythical elixir of life symbolizes humanity's enduring hope for immortality. (e) The Fountain of Youth represents a legendary source of rejuvenation. (f) In Greek mythology, Tithonus was granted eternal life without eternal youth, highlighting the potential curse of immortality..... 4

Figure 1-3: Two complementary strategies to combat aging: regenerative medicine and enhancing cellular resistance. Regenerative medicine involves replacing or repairing damaged and aged cells, akin to replacing worn-out parts in a car. This process, such as regenerating tissues through stem cell therapy or partial reprogramming, relies heavily on the concept of cellular decision-making, where cells assess environmental signals and gene programs to choose between repair or regeneration. Enhancing cellular resistance, on the other hand, fortifies the resilience of existing cells, similar to reinforcing a car to endure tough conditions. This strategy includes the use of metabolic interventions to optimize cellular health and performance. Both approaches work

together to address the hallmarks of aging, promoting extended lifespan and improved healthspan. [Hallmark figure from Ref. (13) and other figs generated by Dall-E 3]..... 9

Figure 2-1: Renormalization in the Ising Model and JPEG Image Compression. (a) Renormalization in the Ising model, where spins on a fine lattice are grouped into blocks to form a coarser representation of the system's magnetization. (b) JPEG image compression of the image, showing the progressive reduction in detail through increasing levels of compression. Both processes highlight the principle of discarding fine-scale details to emphasize larger-scale structures. 22

Figure 2-2: Identifiability of ICA and its application on blind source separation illustrated. Demonstration of ICA Identifiability and Source Separation. The top row displays the original signals, which are then combined either linearly or nonlinearly, shown in the middle and bottom rows, respectively. Linear ICA and nonlinear ICA are applied to these mixtures, successfully recovering the original signals in the rightmost column. In contrast, when PCA and its nonlinear variant, VAE (without further constraints), are applied to the same mixtures (middle column), separation is not achieved. Figure from Ref. (40). 25

Figure 2-3: Bifurcation diagrams for the saddle–node, transcritical, pitchfork co–dimension 1 bifurcation. The plot shows the stability properties of dynamical systems depending on the bifurcation parameter α and the initial condition of the system IC. Figure from Ref (50)..... 32

Figure 3-1: Fatecode workflow for in silico perturbation experiments and cell fate regulator detection. The 3D model (top) represents a Waddington-like landscape depicting cellular reprogramming processes. We seek to identify genes (question marks) that regulate paths on this landscape (wavy lines), by transitioning them to another path (red arrows). A classification-supervised autoencoder learns a latent space representing the original data, optimized for both

input reconstruction and cell type classification. The latent layer is systematically perturbed and by investigating all resulting perturbation-generated cell type distributions, distributions with an increase or decrease in a cell type of interest are identified. Perturbation output is simulated by subtracting the perturbed from unperturbed latent layers and feeding it to the decoder to identify a cell by gene matrix of prioritization scores that can help us to prioritize genes predicted to be important for achieving a desired cell population distribution. An average of the cell fate regulator prioritization scores across cells in each cell type is computed. By sorting these genes based on their prioritization scores for a cell type of interest, the model predicts genes that are important for regulating the levels of a given cell type. 40

Figure 3-2: Comparison of autoencoder architectures for analyzing data for hematopoiesis regulation in zebrafish blood a, Comparison of correlation between input and output of AE, variational autoencoder (VAE), and conditional variational autoencoder (CVAE). b, Mean square error between input and output of the three autoencoder architectures showing that AE produces the lowest error rate for this data set. c, UMAP visualization of the latent layer of the under-complete autoencoder (AE). d, Confusion matrix for the classifier connected to the latent layer of AE demonstrating excellent classification performance..... 42

Figure 3-3 Fatecode detects known regulators using simulated data generated by SERGIO. a, The schematic structure of the gene regulatory network to generate scRNA-seq. Red nodes are known regulators and green nodes are non-regulators whose production rates are determined by their associated regulators. Our goal is to identify known regulators from the generated scRNA-seq data using Fatecode. b, Benchmark comparisons of the detection rate of predefined regulators generated by SERGIO using Fatecode compared with a naive differential gene expression (DGE) baseline. The red and green areas represent the performance of Fatecode and DGE, respectively, on the simulated data with 400 cells. c, Benchmark comparisons of the detection rate of known regulators using Fatecode, scFates, and DGE on simulated data with 2700 cells. d, Venn diagram

showing the similarity between the number of known regulators uncovered by Fatecode across various latent layer sizes. 44

Figure 3-4: Fatecode accurately detects regulators and predicts the effect of single-cell perturbations. a, Hematopoiesis data from Paul et al. (80) visualized as a UMAP and clustered into 9 cell types. b,d, The results of in silico perturbations that change the initial cell frequency to the desired distribution (for b our objective is to promote monocytes while reducing the number of erythrocytes. As for d, we aim for an increase in the erythroid population and a decline in MEP and megakaryocytes). c,e, Gene prioritization scores per cell type for *Irf8* and *Cebpa*. f, Pathway enrichment analysis results. Gene ontology (GO) biological processes show significant processes related to cell development and hematopoiesis. 47

Figure 3-5: In silico experiments to induce hematopoietic stem/progenitor cells using hematopoiesis in zebrafish. a, A series of latent layer perturbations and their effect on cell distribution. b, Cells that switch from their initial cell type to HSPCs are highlighted..... 49

Figure 3-6 :Fatecode identifies key genes in mouse neurogenesis. a, UMAP embedding of fourteen major cell types. b, latent layer node perturbation leads to an increase in mature granule cells while a decrease in immature granule cells. c pathway enrichment analysis shows the relevant biological process using the top 200 genes selected based on their prioritization scores for mature granule cells. 52

Figure 4-1. CLERA discovers dynamical systems and gene programs from simulated data. a, Schematic of a two-gene regulatory network (G_1 and G_2) with discovered governing equations and parameters shown. b, Comparison of generated gene expression data (top) and solutions from equations discovered by SINDy (bottom) for the two-gene system over time. Gene 1 and Gene 2 expression levels are plotted against time. c, Gene interaction networks for cell type 7 derived from

SERGIO ground truth (top) and CLERA (bottom). Nodes represent genes, coloured by gene programs identified through clustering. d, Heatmaps showing Jaccard similarity between SERGIO and CLERA-derived gene program clusters across nine cell types (CellType0 to CellType8). Colour intensity indicates the degree of similarity, with lighter colours representing higher similarity and darker colours lower similarity. 68

Figure 4-2: CLERA uncovers dynamics and gene programs in pancreatic development. a, Discovered differential equations governing mouse pancreas development data from scRNA-seq, showing sparse and interpretable models and connections between latent variables. b, Temporal dynamics of latent variables, which illustrate distinct patterns across cell types. c, Interaction graphs for various stages of pancreatic development, show dynamic gene interactions over time for different cell types. d, Clustering results showing gene program similarities across cell types, with shared genes in cluster 1 among Ductal, Ngn3 low EP (endocrine progenitor), and Ngn3 high EP cell types. e, Gene Set Enrichment Analysis results indicating pathways related to pancreatic development and neurogenesis. f, Degree centrality analysis identifying key genes for each cell type, including Spp1, Chgb, Neurog3, Ins1, Ins2, Clu, and Sox9. 71

Figure 4-3. CLERA reveals central genes and dynamics in hematopoietic differentiation. a, Differential equations discovered for bone marrow development data, showing connections between latent variables. b, Temporal dynamics of latent variables, with distinct patterns across cell types. c, Interaction graphs for different stages of bone marrow development, which capture dynamic gene interactions for each cell type. d, Clustering analysis of co-regulated gene groups at each developmental stage, with significant similarities between precursors, monocytes, DCs, and among Ery-1 and Ery-2, Mono-1 and Mono-2 subpopulations. e, Degree centrality analysis identifies key genes driving cellular differentiation, including Mpo, HOPX, Malat1, FOS, CD52, FAM30A, and CD74. 73

Figure 5-1: Schematic of workflow with FateNet. Input data, in the form of a gene expression matrix, is reduced in dimension using PCA, clusters are obtained and a pseudotime series is constructed. The pseudotime series leading up to a cell fate transition is passed into FateNet, which outputs a probability distribution over different bifurcations. FateNet is pre-trained using many simulations from stochastic differential equations (SDEs) going through different types of bifurcation. The output of FateNet consists of probabilities for the occurrence of different bifurcations. A spike in the probability for a bifurcation is an indication that the bifurcation is about to occur. The cells associated with the timing of the spike can be examined to identify the key genes responsible for initiating the bifurcation process. 92

Figure 5-2: Simulations and predictions in the simple gene regulatory network model going through a fold, pitchfork, and no bifurcation. (a-c) Bifurcation diagrams showing the stable (solid) and unstable (dashed) states of the model as a parameter is varied. (d-f) Model simulation (gray) with the bifurcation parameter varying linearly with time (Methods), and smoothing (black) with a Lowess filter with span 0.2. The model reaches the bifurcation at pseudotime 500. (g-i) Probabilities assigned by FateNet for each class of bifurcation as progressively more of the data becomes available. The arrow shows the time window where there is insufficient data for FateNet to make a prediction. FateNet uses the data after smoothing (i.e. not the trend) when making its predictions. The vertical dashed line indicates the time when the bifurcation is crossed. PF: pitchfork; TC: transcritical. 94

Figure 5-3: Bifurcation predictions in a simulation of SERGIO. (a) Uniform Manifold Approximation and Projection (UMAP) visualization of scRNA-seq data generated by SERGIO, with distinct clusters, color-coded based on cell type. (b) PAGA network graph representing the interconnectivity and relationships between cell types. (c) Bifurcation and null trajectories of cells organized in pseudotime (top) and the predictions of FateNet (bottom). The first principal component of the gene expression data is used to make predictions. The bifurcation trajectory shows a cell-fate transition between cell type 5 and cell types 3 and 6. The vertical dashed line

indicates the time when the bifurcation is crossed. Data is smoothed using a Lowess filter with span 0.2 and the detrended data are passed to our model. The null trajectory is generated by taking a random sampling from the first 20% of the detrended data and adding it to the original trend. DL probabilities are the probabilities assigned by our model for each event among Null, Fold, Transcritical (TC) and Pitchfork (PF). 96

Figure 5-4: Predictions in data of mouse hematopoietic stem cell differentiation from undifferentiated cells (gray) to neutrophils (orange). (a) UMAP plot of mouse hematopoiesis data, emphasizing the transition (arrows) from progenitor cells (gray) to neutrophils (orange), elucidating the dynamic differentiation process. (b) Bifurcation and null trajectories with model predictions. The bifurcation trajectory (left) is the first principal component against pseudotime downsampled by a factor of 100. The dashed line shows the transition. The data is detrended using a Lowess filter with a span 0.2 and used as input to the model. The model outputs probabilities for each event among Null, Fold, Transcritical (TC) and Pitchfork (PF). The yellow box highlights the initial spike in bifurcation probabilities between pseudotime 0.28 and 0.32. The null trajectory (right) is generated by random sampling from the first 20% of the detrended bifurcation trajectory and adding them to the trend. (c) ROC curves for predictions of any bifurcation using variance (Var), lag-1 autocorrelation (AC), sample entropy (SE), Kolmogorov complexity (KC) and FateNet (FN). Predictions are made at evenly spaced time points between 0.3 and 0.6 for 100 unique downsampled bifurcation trajectories and corresponding nulls, resulting in a total of 1400 predictions. The inset shows the probabilities assigned to each bifurcation between pseudotime 0.5 and 0.6. Boxes show the median and interquartile range, and whiskers show the full range. 99

Figure 5-5: Bifurcation predictions in murine pancreatic development focusing on differentiation of Fev+ to Alpha, Beta and Delta cells. (a) Uniform Manifold Approximation and Projection (UMAP) visualization of Pancreas development data, with distinct clusters, color-coded based on cell type. (b) Bifurcation and null trajectories with model predictions. The bifurcation trajectory (left) is the first principal component against pseudotime. The dashed line shows the

transition point. The data is detrended using a Lowess filter with a span of 0.2 and used as input to the model. The model outputs probabilities for each event among Null, Fold, Transcritical (TC) and Pitchfork (PF). The null trajectory (right) is generated by random sampling from the first 20% of the detrended bifurcation trajectory and adding them to the trend. 101

Figure 5-6: Exploring system response to various in silico perturbations. (a) UMAP visualization shows in silico perturbations, with green lightning indicating overexpression and red lightning denoting knockout perturbations. Each perturbation is individually implemented to observe how the system experiences shifts in the bifurcation dynamics. Model predictions for stem cell differentiation to neutrophils in mouse hematopoiesis after knocking out (b) and over-expressing (c) a few numbers of the most significant genes. Genes are knocked out by setting their expression to zero. Genes are overexpressed by multiplying their expression by a factor of two. In each case, ten equally-spaced predictions are made between pseudotime 0.45 and 0.6..... 103

Figure 6-1: scVAEDer overview. scVAEDer integrates VAE and DDM. First, a VAE is trained using the gene expression data. Then the VAE latent embedding is used to train the DDM through the processes of latent space diffusion and denoising. Combining together the model is able to decode back the gene space with high accuracy. scVAEDer can be used for different downstream analysis tasks such as generating novel high-quality scRNA-seq data, understanding changes in gene expression during cellular differentiation, predicting the effect of perturbations on new cell types when expression data is available for multiple conditions, detecting master regulators by interpolating between different cellular states and ranking fast responder genes based on their velocity values. 118

Figure 6-2: scVAEDer accurately learns the latent representation and generates new high-quality scRNA-seq data. a, Red, forward diffusion process with 1000 steps on hematopoiesis in zebrafish as we add noise to the data; blue, reverse process as the model learns how to denoise. b, UMAP visualization of the real data embedding. c, Samples generated from DDM prior. d, samples

generated from the VAE. e, Total Variation Distance (TVD) between latent embedding of data and samples generated from the DDM as well as VAE prior distributions. 120

Figure 6-3: scVAEDer can be used to understand cellular dedifferentiation. a, Mapping HSPC and monocyte cells into the latent prior of DDM. b, Using DDM (1000 diffusion steps) and performing latent linear interpolation with 2000 equidistant samples (red dots). The absence of any sample generated in the empty region suggests that the model has learned an accurate embedding. c, Heatmap showing the similarity between gene expression of generated states and the real average expression of HSPCs and monocytes (using 100 DDM steps before interpolation). d, Expression of selected marker genes along the interpolation path. 122

Figure 6-4: scVAEDer is more accurate than the SOTA methods in predicting perturbation responses. a, Data generation using latent prior of DDM with 1000 diffusion steps. b, Comparison of the correlation values of average gene expression between real and predicted cells of various cell types obtained using scGen, scPreGAN, and scVAEDer. c, Violin plots for selected key genes across control, real stimulated, and stimulation predicted by scVAEDer and scGen in Dendritic cells (DC). 124

Figure 6-5: scVAEDer accurately detects master regulators during cellular reprogramming. a, The reverse process of DDM allows the generation of high-quality reprogramming samples from random Gaussian noise (the quality of generated samples is critical for downstream analysis). b, UMAP visualization of the data latent embedding, colored based on their state (red: failed, blue: reprogrammed). c, Data generated by interpolating between reprogrammed and failed states, represented by red dots. Remarkably, none of the interpolated samples are found outside the real representation of data. d, Correlation values between the new interpolated samples and the average gene expression of reprogrammed and failed cells, which demonstrates the shift in gene expression between failed and reprogrammed states. e, Ranking of genes based on high expression differences between t=0 and t=3000. f, Illustration showing the process of computing gene velocities along

the interpolation path to detect master regulators. g, Gene set enrichment analysis using 400 genes with the highest velocities (fast responders), which reveals pathways that are crucial during the cell reprogramming process. 126

Figure 7-1: Predicting Gene Expression and Early Cell Fate Bias via Combined scRNA-seq and Lineage Tracing. A visual depiction of the Deep Lineage is presented. On the left, the Waddington landscape illustrates the developmental trajectory, showing how cells within each clone (represented by different colours) differentiate into distinct mature cell types originating from stem cells. On the right, by combining scRNA-seq data and lineage tracing information the gene expression profiles of cells within each clone are used as input for the autoencoder. The resulting latent embeddings of cells in a clone are subsequently used as inputs for the LSTM, Bi-LSTM, or the GRU model. This integrated architecture enables the accurate prediction of both early cell fate bias and gene expression profile of unseen days within a clone. Notably, Deep Lineage offers the flexibility to adapt different preprocessing steps, diverse barcoding techniques, various dimensionality reduction methods, and a range of various deep learning models (left black box). 139

Figure 7-2: Deep Lineage accurately predicts single-cell gene expression of unseen cells in hematopoiesis a, UMAP plot of the hematopoiesis data from Weinreb et al. Each point represents an individual cell, color-coded by cell type. b, Schematic representation of the gene expression prediction process using two regression models: one for predicting day 4 gene expression (trained on data from days 2 and 6), and the other for predicting day 6 gene expression (trained on data from days 2 and 4) within a clone. c, Correlation of gene expression between Deep Lineage predictions and cells excluded during training, categorized by cell types and time points. d,e, Comparison of average gene expression values of 2000 genes between Deep Lineage predicted and real cells for Monocytes and Neutrophils. Gene expression at day 4 was averaged across predicted cells per clone, and then correlation was computed with actual data for each gene ($R2$ indicates the squared Pearson correlation coefficient between the predicted and ground truth

values). f,g, Violin plots show gene expression distributions between predicted and real cells for randomly selected genes both Monocytes and Neutrophils. 143

Figure 7-3: Exploring progenitor bias prediction in hematopoiesis with Deep lineage and comparison to state-of-the-art methods. a, Schematic UMAP diagram illustrating the trajectory of stem cell differentiation and the possible cellular outcomes with cell colors indicating distinct clonal lineages. Grey points represent other mature cell types (not neutrophils and monocytes). b, Performance evaluation of the classifier using accuracy and cross-entropy loss metrics to predict early cell fate bias by employing gene expression of cells on just day 2, using cells on days 2 and 4, and including all three days within a clone. c, Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) values for the model using cells from days 2 and 4 or all three days with a clone, showing our model's performance in early cell fate bias prediction. Day 2 performance is very close to day 2&4 performance, thus not plotted. d, Comparative analysis with state-of-the-art methods in predicting early cell fate bias. Deep Lineage outperforms existing methods showing a higher classification accuracy. 144

Figure 7-4: Deep Lineage accurately predicts single-cell gene expression of unseen cells in reprogramming. a, UMAP visualization demonstrates fibroblast cell reprogramming into iEPs, dots represent cells and are color-coded by time point. b, Correlation values between gene expression of ground truth cells and Deep Lineage predictions for cells excluded during training across different stages of the reprogramming process. c, Violin plots show gene expression distributions between real cells and Deep Lineage predictions for randomly selected genes for both Failed and Reprogrammed cells. d,e. Comparative analysis of the top 2000 differentially expressed genes between predicted and real cells. High correlation values are observed for both successful and failed reprogramming cells, highlighting the accuracy of the model for different cell types ($\boldsymbol{R2}$ indicates the squared Pearson correlation coefficient between the predicted and ground truth values). 147

Figure 7-5: Early cell fate prediction in reprogramming with Deep Lineage and comparison to state-of-the-art method. a, UMAP visualization illustrates progenitor bias toward either successful or failed reprogramming outcomes. b, The model's accuracy in detecting successful or failed outcomes of progenitors is examined using gene expression data of cells up to days 12, 15, 21, and all days within a clone. c, Benchmarking comparison between Deep Lineage and CellRank in predicting fate outcome when using all time points up to a given time point (e.g. day 12) to infer time point “day 28”. Using ROC curves and AUC values, the graph shows the predictive power of both models in predicting fate bias. Deep Lineage consistently outperforms CellRank..... 148

Figure 7-6: In silico perturbations of early cells in hematopoiesis and reprogramming data successfully modify cell fate at later stages. a, UMAP representation showing the procedure for in silico perturbation of cells on day 15 within failed reprogrammed clones, aiming to switch them into successfully reprogrammed states. b, Correlation values between in silico perturbed data and actual successfully and unsuccessfully reprogrammed cells on Day 28, showing the power of Deep Lineage in performing in silico experiments. c, Gene set enrichment analysis reveals the biological processes associated with the top 200 genes identified by the SHAP method (40 genes selected from each time point) d, Identification of the top 10 gene candidates using SHAP analysis at each time point during fibroblast cell reprogramming. e, UMAP visualization depicting the in silico perturbation experiment targeting progenitor cells (Dark orange) in clones with a Neutrophil fate (blue), resulting in a transformation to Monocytes (green). f, The top 10 gene candidates for hematopoiesis detected by SHAP analysis across all time points. g, Gene set enrichment analysis of the top 200 genes identified by SHAP analysis..... 150

Figure 9-1 Supplementary Fig. 1: Gene expression perturbations to change the cell-type distribution. This figure demonstrates the transition of an initial cell type distribution to a desired target distribution through gene-level adjustments. The circles represent the system's state, with the frequencies of cell types indicated below. 191

Figure 9-2 Supplementary Fig. 2: Comparison of autoencoder architectures for analyzing data for endocrine development in the mouse pancreas. a, comparison of the input-output correlation for the AE, the variational autoencoder (VAE), and the conditional variational autoencoder (CVAE). b, the mean square error of the three autoencoder architectures' input and output. VAE performs better than other architectures for this data. 192

Figure 9-3 Supplementary Fig. 3: Gene prioritization score for the mouse hematopoiesis data. a, b, c, Fatecode accurately determines gene prioritization scores for various genes, including *Klf1*, *Runx1*, and *Fli1*, across different cell types. 193

Figure 9-4 Supplementary Fig. 4: Fatecode analysis of hematopoiesis data identifies master regulators governing cell switching dynamics. a, visualization of the hematopoiesis dataset from Weinreb et al. hematopoietic progenitors differentiate into different cell types such as mast cell (Ma), basophil (Ba), eosinophil (Eos), megakaryocyte (Mk), lymphoid precursor (Ly), migratory dendritic cell (mDC) and plasmacytoid dendritic cell (pDC), erythrocyte (Er), neutrophil (Neu), monocyte (Mo). b, The effect of different perturbation sizes of a node in the latent layer on the cell distribution. c, gene set enrichment analysis results. Gene ontology (GO) biological processes enrichment analysis shows significant process terms related to mouse hematopoiesis, stem cell development, and mesenchymal cell differentiation. 194

Figure 9-5 Supplementary Fig. 5: Gene Regulatory Network of Ybx1 and its downstream target genes along with gene prioritization scores. The GRN was constructed using SCENIC, by filtering the top 2000 interactions with the highest SCENIC Importance Measure (IM) scores. Additionally, the top 400 predicted master regulators from Fatecode were mapped onto the GRN, and the resulting network is presented here using a network bar chart in which each value of a bar plot shows the Fatecode gene prioritization score of the gene for that cell type. 195

Figure 9-6 Supplementary Figure 1: Training and validation loss curves for training CLERA on the data generated using SERGIO. The low losses in (a)-(e) indicate a high accuracy and strong generalization. Refer to the Methods section for the description of each loss term. 197

Figure 9-7 Supplementary Figure 2: Training and validation loss curves for training CLERA on the pancreas data. The low losses in (a)-(e) indicate that CLERA successfully builds the latent layer with the desired characteristics. Refer to the Methods section for the description of each loss term. 198

Figure 9-8 Supplementary Figure 3: Training and validation loss curves for the bone marrow data in the transfer learning setting. Due to transfer learning, the model converges in fewer epochs as compared to the training of the Pancreas data. 199

Figure 9-9 Supplementary Figure 4: Training comparison on the bone marrow dataset using transfer learning (SINDy threshold frequency- 500) and vanilla training with two different thresholds (SINDy threshold frequency- 150 and 500). (a) shows the combined loss plotted against the number of epochs, illustrating the slower convergence and signs of overfitting in the vanilla models, particularly with the lower threshold (red). (b) presents the data with normalized epochs, highlighting the efficiency of transfer learning, which reaches stability in a fraction of the time compared to vanilla training. While the vanilla models require most of the training period to converge (especially with a lower threshold), the transfer learning model generalizes well much earlier in the training process, as shown by the stabilized validation loss. 200

Figure 9-10 Supplementary Figure 5: Classification results of (a) SERGIO generated data (b) Pancreas data and (c) Bone marrows data. After training CLERA on different datasets. the prediction of the cell type for all the preprocessed input (validation and training) by the classification network is shown, The classifier achieves high accuracy with very few

misclassifications. Even for cell types with fewer instances, the model can accurately predict the cell type from the latent space. This strong classification performance suggests that the latent space captures cell-type-specific features. Accurate classification ensures that the predicted ODEs reliably infer relationships and can predict cellular behavior across different conditions. 201

Figure 9-11 Supplementary Figure 6: Mean intersection percentage between top-k genes of different latent variables present at various time distances for (a) SERGIO generated dataset (b) Pancreas dataset (c) Bone marrows dataset. The mean intersection percentage between the top-k genes for various time distances in the SERGIO-generated dataset with k=30 (left), the Pancreas dataset with k=300 (middle), and Bone marrow data with k=300 (right) datasets. The intersection percentage, plotted for each latent variable, is higher for closer time steps and gradually decreases as the time distance grows. This shows the model's ability to capture temporal coherence in gene expression patterns, where similar gene sets are shared between adjacent time points, reflecting the gradual changes in biological processes. The consistency decreases with larger time distances, indicating the differences in active gene programs..... 202

Figure 9-12 Supplementary Figure 7: The mean and standard deviation of SHAP values between each cell type and latent variable for the pancreas data. The SHAP values calculated using the classifier network are plotted which show that each cell type is primarily driven by one or more latent variables for classification. This demonstrates that certain latent variables are key drivers in predicting particular cell types, suggesting a cell type-level abstraction within the latent space..... 203

Figure 9-13 Supplementary Figure 8: SHAP Value Analysis of Latent Variables in hematopoietic differentiation. This figure illustrates the mean and standard deviation of SHAP values for each cell type and latent variable. The values, derived from the classifier network, demonstrate that specific latent variables significantly influence cell type classification, indicating a cell type abstraction in the latent space..... 204

Figure 9-14 Supplementary Figure 9: Latent Variable Time Series discovered using CLERA for (a) pancreas dataset and (b) Bone marrows dataset. Colors have been assigned as the cell type labels. The latent space demonstrates clear cell type differentiation, with distinct cell types following unique trends and values over time. This pattern indicates that the learned latent space successfully captures cell type-specific dynamics, ensuring that the model reflects latent qualities at a cell type level granularity. 205

Figure 9-15 Supplementary Figure 10: Results of centrality measures on (a) Pancreas dataset and (b) Bone marrows dataset. The figure shows the distribution of three centrality measures—Closeness, Eigenvector, and PageRank—for the most central genes across different cell types in (a) the pancreas dataset and (b) the bone marrow dataset. In both datasets, some genes exhibit consistently high centrality values across multiple cell types, indicating their importance in the gene regulatory networks. 206

Figure 9-16 Supplementary Figure 1*:* Simulations and predictions in the simple gene regulatory network model using a smaller noise amplitude of 0.01. Refer to the caption of Figure 1 for further details..... 211

Figure 9-17 Supplementary Figure 2*:* Simulations and predictions in the simple gene regulatory network model using a larger noise amplitude of 0.1. Refer to the caption of Figure 1 for further details..... 212

Figure 9-18 Supplementary Figure 3: Gene Ontology (GO) term and Reactome pathway enrichment analysis for the top 250 significant genes. a) GO: Biological Process (BP) showing the enriched biological processes such as different metabolic processes, cellular response to different stimulus, and cell apoptosis indicating relevant biological activities associated with these genes. b) GO: Molecular Function (MF) highlights the enriched molecular functions including protein

binding and enzyme binding. c) GO: Cellular Component (CC) displays the enriched cellular components such as cytoplasm and intracellular organelles, suggesting the subcellular localizations where these genes are active. d) The REAC panel represents the enriched pathways from the Reactome database, including immune system processes and cellular responses to stress, which are related to the process of neutrophil development..... 213

Figure 9-19 Supplementary Figure 4: Enrichment map analysis of biological pathways. The figure shows the complex connections and interactions between different active biological processes. Blue edges of varying thickness connect the nodes, indicating the strength of the connection between pathways by the number of mutual genes. To simplify the network, only edges and nodes meeting the cutoff (Node Cutoff P-value > 0.01, Edge Cutoff > 0.35) are shown.... 214

Figure 9-20 Supplementary Figure 5: Architecture for the neural networks in FateNet. Each network consists of 2 convolutional and 2 long-short-term memory (LSTM) layers. Hyperparameters are shown for Network 1. The input to the network is a time series of length 500. This is passed through a 1D convolution, with a kernel size of 12, 50 filters, and the ReLU activation function. Padding is applied to the ends of the time series to maintain the input dimension. This is then passed through a max pooling operation with pool size 2, reducing the dimension by a factor of 2. This process repeats through another convolution and max pooling layer. This is then passed to the first LSTM layer with 100 memory cells. For this layer, the cells return the entire sequence. This then enters the second LSTM layer with 20 memory cells, which returns only the end value of the sequence. Each memory cell is then connected with the 4 output nodes via a dense layer. The output is passed through a softmax filter to obtain a probability distribution over the 4 possible outcomes..... 215

Figure 9-21 Supplementary Figure 6: training and validation accuracy vs. epoch for Network 1 and Network 2 during the training phase. The accuracy on the validation set peaks at epoch 71 and 154, respectively, at which point the models are selected..... 216

Figure 9-22 Supplementary Figure 7: Confusion matrices for Network 1 and Network 2 when evaluated on their test sets from the library of generated dynamical systems. We show performance on the multi-class classification problem, where the model must predict the specific event, and the binary classification problem, where the model only needs to predict whether or not a bifurcation is approaching. Cell values show (row-)normalized classification rates for each class. (A) Network 1 on the multi-class classification problem obtains an F1 score of 0.64. (B) Network 2 on the multi-class classification problem obtains an F1 score of 0.95. (C) Network 1 on the binary classification problem obtains an F1 score of 0.80. (D) Network 2 on the binary classification problem obtains an F1 score of 0.98. TC: transcritical. PF: pitchfork..... 217

Figure 9-23 Supplementary Figure 1: Gene expression changes as a result of interpolation. The expression of selected maker genes is upregulated or downregulated as we interpolate between monocytes to HSPCs with 2000 equidistant steps. 220

Figure 9-24 Supplementary Figure 2: The number of diffusion steps changes the level of detail in generated data Samples with different levels of granularity are generated by adjusting the number of diffusion steps before interpolation. If a large number of diffusion steps are used, more structure in the initial data will be destroyed, causing a gradual loss of information, resulting in coarser interpolations, and potentially causing abrupt changes in the correlation values between successive steps..... 221

Figure 9-25 Supplementary Figure 3: scVAEDer can correctly predict the distribution of genes in the stimulated condition. a, Violin plots showing the distribution of specific key response genes across control, real stimulated condition, and prediction of stimulation using scVAEDer and scGen in Dendritic cells (DC). scVAEDer more accurately predicts the effect of the perturbation on genes. b, Comparison of the correlation values of various cell types obtained using scGen, scPreGAN and scVAEDer for intestinal epithelial cells after Salmonella infections data..... 222

Figure 9-26 Supplementary Figure 4: DDM training and variance scheduling. A comparison between the effect of different variance scheduling methods on and values. The user can choose between one of these 3 methods prior to training the model. 223

Figure 9-27 Supplementary Figure 1: Deep Lineage is compatible with both single and cumulative barcoding data. (a) Illustrates single barcoding, while (b) presents a cumulative barcoding strategy. 229

Figure 9-28 Supplementary Figure 2: The impact of different preprocessing steps and model types on the performance of Deep Lineage for hematopoiesis data. a, The effect of different preprocessing on the performance of LSTM and GRU models. b, The effect of preprocessing steps and model architecture on the LSTM performance. The first architecture includes: AE_first_hidden_layer:500, AE_second hidden_layer:100, AE_latent:75, Alpha:0.3, Num_Layers_BRNN:4, Reg_Dropout:0.3. The second architecture includes: AE_first_hidden_layer:500, AE_second_hidden_layer:200, AE_latent:50, Alpha:0.1, Num_Layers_BRNN: 2, Reg_Dropout: 0.3. As expected, using only raw counts or just scaling without log transformation results in poor performance. 230

Figure 9-29 Supplementary Figure 3: Correlation values show the predictive power of Deep Lineage in estimating gene expression profiles of unseen cells. a, The distribution of correlation values for monocytes, neutrophils, and combination of cells on day 6 in hematopoiesis. b, The distribution of correlation values for failed, successfully reprogrammed cells, and the combination of cells on day 28 of the reprogramming process. 231

Figure 9-30 Supplementary Figure 4: Expression Prediction Accuracy in fibroblast reprogramming. The dot plot illustrates the average expression of various genes between real cells and Deep Lineage predictions, for successful and failed reprogrammed cells. 232

Figure 9-31 Supplementary Figure 5: Ablation studies were performed on the hematopoiesis and reprogramming data. Heatmaps display the accuracy of Deep Lineage in predicting early cell fate in control and ablation conditions for both hematopoiesis (a) and reprogramming (b). The bar plot compares these conditions using accuracy, precision, recall, and F1 score, emphasizing the impact of clonal structure ablation on model prediction..... 233

List of Tables

Table 3-1: List of zebrafish hematopoiesis regulator genes predicted by Fatecode with literature evidence for involvement in this process. 50

Table 9-1 Supplementary Table 1: The hyperparameters used to train CLERA on bone marrow and pancreas development data. Several hyperparameters are tested for training the models. To evaluate the performance of these choices, we used metrics such as the training and validation curves, the number of active terms in the discovered ODE, the structure of the identified ODE, and the Gene Set Enrichment Analysis (GSEA). *Each consecutive pair in this list corresponds to a linear layer followed by an activation function. Activation is not applied after the last layer. 210

Table 9-2 Supplementary Table 1. Significant genes identified during the critical period of cell decision-making (highlighted in the yellow box in Figure 3). The table includes columns for gene names, corresponding references to previous studies, and a description of their roles..... 219

Table 9-3 Supplementary Table 1: Top 15 genes ranked based on their velocity values, along with references that demonstrate their association with stem cell function and cellular reprogramming. 225

Table 9-4 Supplementary Table 2: Detailed architecture of VAE part of scVAEDer. ... 227

Table 9-5 Supplementary Table 3: Detailed architecture of DDM part of scVAEDer. .. 228

Table 9-6 Supplementary Table 1: Key genes identified in reprogramming and hematopoiesis with their associated functions. This table presents a curated list of top genes

identified by Deep Lineage as significant in both cellular reprogramming and hematopoiesis. Each row includes the gene symbol and a brief description of its known function or role, supported by relevant literature. The genes listed play diverse roles in stem cell maintenance, differentiation, signaling pathways, and tissue homeostasis. 235

Table 9-7 Supplementary table 2: Preprocessing Steps. This table shows the different preprocessing steps and hyperparameters applied to both datasets used in the study. 236

Table 9-8 Supplementary table 3: Regression Architecture for Hematopoiesis Data. This table presents various architectures and parameters evaluated for the regression part of the hematopoiesis data analysis. 237

Table 9-9 Supplementary table 4: Regression Architecture for Reprogramming Data. This table presents various architectures and parameters evaluated for the regression part of the reprogramming data analysis. 238

Table 9-10 Supplementary table 5: Cell Fate Prediction Architecture and Parameter Search for Reprogramming Dataset. This table describes different architectures and parameters tested for the fate prediction component in the reprogramming dataset, specifically for the task where all days except day 28 of clones are given to predict day 28. 238

Table 9-11 Supplementary table 6 : Cell Fate Prediction Architecture and Parameter Search for Reprogramming Dataset. This table describes different architectures and parameters tested for the fate prediction component in the reprogramming dataset. The task involves using data from all days except day 21 and day 28 of clones to predict day 28. 239

Table 9-12 Supplementary table 7: Comparison of Deep Learning Architectures for Cell Fate Classification in Reprogramming. This table describes different architectures and parameters tested for the fate prediction component in the reprogramming dataset. The task involves using data from all days except day 15, 21, and day 28 of clones to predict day 28. 239

Table 9-13 Supplementary table 8: Ablation Study for Hematopoiesis Dataset. This table provides results from ablation studies conducted on the hematopoiesis dataset, showing the importance of clonal data and also how different parameters contribute to overall performance. 240

Table 9-14 Supplementary table 9: Ablation Study for Reprogramming Dataset. This table provides results from ablation studies conducted on the hematopoiesis dataset, showing the importance of clonal data and also how different parameters contribute to overall performance. 240

List of Abbreviations

Machine learning: ML

Artificial intelligence: AI

Long Short-Term Memory: LSTM

Gated Recurrent Unit: GRU

Principal Component Analysis: PCA

Stochastic differential equations: SD

Gene Set Enrichment Analysis: GSEA

Cellular Latent Equation and Representation Analysis: CLERA

Independent Component Analysis: ICA

Mammalian target of rapamycin complex: mTORC1

Hematopoietic Stem/Progenitor Cell: HSPC

Gene regulatory network: GRN

Differential gene expression: DGE

Under-complete autoencoder: AE

Variational autoencoder: VAE

Conditional variational autoencoder: CVAE

Ordinary differential equations: ODEs

Partial differential equations: PDEs

Sparse Identification of Nonlinear Dynamics: SINDy

Uniform Manifold Approximation and Projection: UMAP

Machine learning: ML

Artificial intelligence: AI

Long Short-Term Memory: LSTM

Gated Recurrent Unit: GRU

Single-cell RNA sequencing: scRNA-seq

Denoising Diffusion Models: DDMs

Gene Expression Omnibus: GEO

Variance: Var

Lag-1 autocorrelation: AC

Sample entropy: SE

Kolmogorov complexity: KC

FateNet: FN

Transcritical: TC

Pitchfork: PF

Chapter 1
Introduction

“Death smiles at us all, but all a man can do is smile back." These words by Marcus Aurelius reflect an acceptance of the inevitable. But must we simply smile back? Could we dare to challenge death, or even evade it entirely? The questions of death, life, and immortality have captivated the minds of philosophers, scientists, and scholars for millennia. For some, death is an inescapable fate, a natural end to life’s cycle. Others see it as a threshold to something beyond. Yet, a few bold individuals have refused to accept death as part of life's natural order, instead dedicating themselves to the search for ways to extend life—or even achieve immortality.

Humanity has long pursued the dream of extending life, whether by lengthening our natural lifespan or achieving something as radical as immortality. However, should we ever succeed, we would not be the first. Nature offers extraordinary examples of longevity. The bowhead whale, for instance, can live for over 200 years, while the Aldabra giant tortoise can survive for up to 250 years(6) (Fig1-1. a,b). Most remarkably, the Greenland shark can live up to 500 years, making it one of the longest-living vertebrates(7) (Fig1-1. c). Yet none of these creatures can claim to be truly immortal except for *Turritopsis dohrnii*, commonly known as the "Immortal Jellyfish"(8) (Fig1-1. d). When faced with injury or stress, it undergoes transdifferentiation—a rare process that allows it to transform its cells and revert to a juvenile state, effectively restarting its life cycle. While this unique process is remarkable, the mechanisms of extreme longevity or immortality found in other species are typically highly specialized, making them difficult to apply to humans.

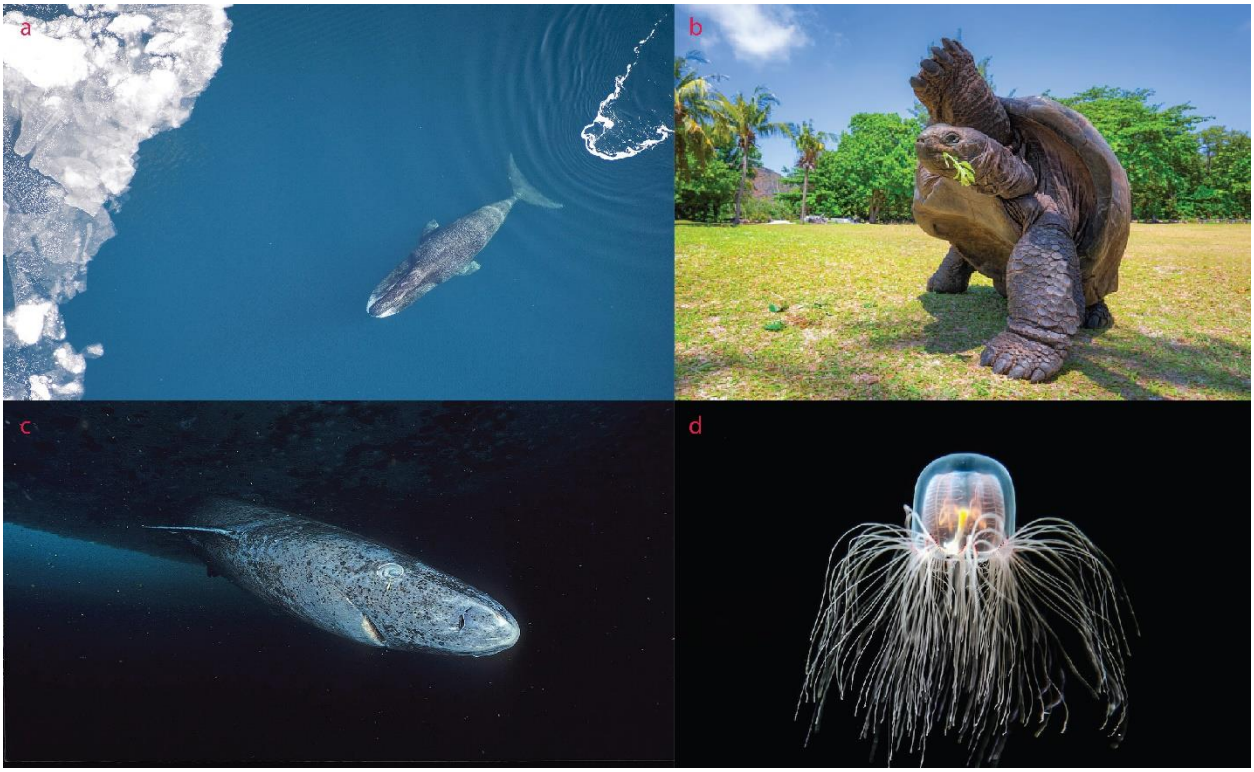


Figure 1-1: The natural world showcases remarkable examples of longevity and potential immortality. (a) The bowhead whale, capable of living over 200 years, thrives in Arctic waters. (b) The Aldabra giant tortoise, known for its lifespan of up to 250 years. (c) The Greenland shark, one of the longest-living vertebrates, can live up to 500 years in the deep ocean [Photo credit: Hemming1952]. (d) The Immortal Jellyfish (Turritopsis dohrnii) uniquely reverts to its juvenile form through transdifferentiation, effectively restarting its life cycle and achieving a form of biological immortality [Photo credit: Chen Yiming].

Despite these challenges, the quest for immortality is a light in the darkness for those who dream of transcending our natural lifespan. This fascination with eternal life has been a recurring theme in mythology and folklore for as long as humans have recorded their thoughts. From the ancient myths of the Fountain of Youth (Fig1-2. e) to the worship of undying deities, the desire to escape death has spanned time and cultures. One of the earliest recorded tales of the search for immortality is found in the *Epic of Gilgamesh*(9) (Fig1-2. a). In this ancient Sumerian story, the legendary king Gilgamesh embarks on a quest to defeat death after the tragic loss of his companion, Enkidu. Gilgamesh seeks immortality to avoid his friend's fate, but ultimately fails in his pursuit. Instead, he comes to the realization that immortality lies not in eternal life, but in the enduring legacy of one's actions and contributions to civilization. This profound narrative is one of the oldest in human

history. In a later era, the First Emperor of China, Qin Shi Huang, also sought to escape death (10) (Fig1-2. b). Obsessed with finding the mythical elixir of life (Fig1-2. d), he dispatched vast expeditions to locate the fabled Mount Penglai, believed to hold the secret to immortality. Despite his efforts, Qin Shi Huang's quest ended in failure, as he tragically died from mercury poisoning, ironically caused by the very pills prescribed to him in his pursuit of eternal life. Mythological tales from around the world echo this same yearning. In Persian mythology, the legendary king Jamshid is one of the most well-known figures associated with immortality(11) (Fig1-2. c). Jamshid was granted divine favor, allowing him to rule for hundreds of years. His reign marked a golden age, a time of prosperity and innovation for humanity. According to legend, Jamshid discovered the secret to immortality and ruled for centuries, growing more powerful and wise as time passed. However, his story also serves as a warning: Jamshid's arrogance grew as he lived beyond the natural span of life. He began to see himself as divine, eventually losing the favor of the gods and bringing about his downfall. In Greek mythology, several stories address the theme of immortality, often with a cautionary twist. One notable example is the story of Tithonus, a mortal man who was granted eternal life by the gods at the request of Eos, the goddess of dawn (12) (Fig1-2. f). However, Eos forgot to ask for eternal youth, and as a result, Tithonus aged endlessly, growing frail and withered. His immortality became a curse, as he was doomed to an eternity of decrepitude. Regardless of their basis in myth or reality, these stories underscore the significance of the quest for immortality throughout human history. They reflect our deep-seated fear of death and our enduring desire to transcend our natural limitations. From ancient alchemists seeking the elixir of life to heroic figures embarking on legendary quests, this pursuit has spurred cultural narratives and philosophical exploration, showing how the dream of eternal life has shaped human thought.



Figure 1-2: Throughout history, the quest for immortality has captivated human imagination, as depicted in various myths and legends. (a) The Epic of Gilgamesh illustrates an early tale of seeking eternal life. (b) Qin Shi Huang, the First Emperor of China, pursued the elixir of life. (c) Persian mythology tells of King Jamshid's long reign and eventual downfall due to arrogance. (d) The mythical elixir of life symbolizes humanity's enduring hope for immortality. (e) The Fountain of Youth represents a legendary source of rejuvenation. (f) In Greek mythology, Tithonus was granted eternal life without eternal youth, highlighting the potential curse of immortality.

Today, while the root causes of aging still remain elusive, various computational and experimental studies are beginning to illuminate these underlying mechanisms. This modern exploration bridges the gap between ancient aspirations and current scientific understanding, can offer hope for extending human lifespan and improving the quality of life. Central to this understanding is the recognition of the hallmarks of aging (13), which describe the key biological processes that contribute to the aging phenotype:

1. Genomic Instability: This hallmark refers to the accumulation of DNA damage over time due to environmental stressors (like UV radiation and chemicals) and inherent replication errors. As cells divide, they may fail to accurately replicate their DNA, leading to mutations

and chromosomal abnormalities. This instability contributes to cellular dysfunction and increases the risk of age-related diseases, including cancer.

2. Telomere Attrition: Telomeres are repetitive DNA sequences at the ends of chromosomes that protect them from deterioration. With each cell division, telomeres shorten. When they reach a critically short length, cells can no longer divide, leading to cellular senescence or apoptosis (programmed cell death). Telomere attrition is associated with age-related decline in tissue regeneration and function.

3. Epigenetic Alterations: Epigenetics involves changes in gene expression that do not alter the DNA sequence itself. Factors like aging, lifestyle, and environmental influences can lead to modifications of the epigenome, such as DNA methylation and histone modification. These changes can disrupt normal cellular processes, leading to altered cell behavior and contributing to aging and disease.

4. Loss of Proteostasis: Proteostasis refers to the cellular mechanisms that maintain the proper folding, trafficking, and degradation of proteins. Aging disrupts these mechanisms, resulting in the accumulation of misfolded or damaged proteins. This proteotoxic stress can impair cellular function and lead to age-related diseases, including neurodegenerative disorders.

5. Deregulated Nutrient Sensing: Cells have pathways that sense and respond to nutrient availability (such as insulin signaling and mTOR pathways). Aging can impair these signaling pathways, leading to metabolic dysfunction. This deregulation can contribute to obesity, diabetes, and other metabolic disorders, affecting overall health and longevity.

6. Mitochondrial Dysfunction: Mitochondria are vital for energy production through oxidative phosphorylation. With age, mitochondrial function declines due to accumulated damage and mutations in mitochondrial DNA. This dysfunction leads to decreased energy

production and increased generation of reactive oxygen species, contributing to oxidative stress and cellular damage.

7. Cellular Senescence: Cellular senescence is a state in which cells stop dividing but remain metabolically active. Senescent cells secrete pro-inflammatory factors and other molecules that can disrupt tissue function and promote chronic inflammation. The accumulation of senescent cells is linked to various age-related diseases and contributes to the aging process.

8. Stem Cell Exhaustion: Stem cells play a crucial role in tissue regeneration and repair. As we age, the pool of functional stem cells diminishes, leading to reduced regenerative capacity. This exhaustion can impair the body's ability to heal and replace damaged tissues, contributing to age-related decline in function.

9. Altered Intercellular Communication: Aging affects the way cells communicate with each other through signaling pathways. This alteration can lead to chronic inflammation and changes in tissue microenvironments. The resulting dysregulation can contribute to a range of age-related diseases, including neurodegenerative conditions and cardiovascular diseases.

10. Loss of Tissue Integrity: As we age, tissues can become less organized and functional due to changes in cellular composition and structure. This loss of integrity can impair organ function and increase susceptibility to injury and disease.

11. Increased Inflammation: Chronic, low-grade inflammation often referred to as "inflammaging," is common in aging. This persistent inflammation can result from various factors, including the accumulation of senescent cells and changes in immune function. Chronic inflammation contributes to the development of many age-related diseases, such as cardiovascular disease, diabetes, and neurodegenerative disorders.

12. Decline in Immune Function: Aging leads to a gradual decline in the immune system's ability to respond effectively to pathogens, known as immunosenescence. This decline results in increased susceptibility to infections, reduced vaccine efficacy, and a higher incidence of autoimmune diseases. Maintaining a robust immune system is crucial for healthy aging.

To combat aging, two main strategies can be broadly categorized: regenerative medicine (14) and enhancing cellular resilience (15) (Fig1-3).

1. **Regenerative medicine**: This strategy focuses on regenerating and replenishing damaged or senescent cells to restore tissue function. Techniques such as cellular reprogramming and stem cell therapy aim to generate new, healthy cells that can replace those lost to aging or injury. Induced pluripotent stem cells and methods to rejuvenate aged cells are integral to this approach, as they offer the potential for improved tissue repair and regeneration.
2. **Enhancing Cellular Resistance**: This approach emphasizes increasing the resilience of existing cells to combat age-related damage. By targeting mechanisms that promote cellular health, we can enhance the ability of cells to withstand stressors and external perturbations. For example, compounds that improve mitochondrial function can significantly bolster cellular resistance. Also, metabolic interventions, such as caloric restriction mimetics and nutrient-sensing enhancement, contribute to this effort by optimizing cellular responses to various challenges.

To better explain these strategies, we can draw a parallel example to maintaining a car. Replacing old cells is similar to swapping out worn or malfunctioning parts of a vehicle; just as old brake pads or a failing engine can hinder a car's performance, aging or damaged cells can impair bodily functions. Techniques in regenerative medicine like stem cell therapy, and partial reprogramming are similar to installing a new engine or critical parts, revitalizing the vehicle's performance. Enhancing cellular resistance, on the other hand, resembles protecting the car to withstand various conditions, such as reinforcing the body for rough terrain, using better tires for improved traction, or using better materials to make the car parts. By enhancing cellular resilience through methods

like metabolic interventions and regular physical activity, cells can better cope with external stresses, much like a well-equipped car that handles challenging road conditions.

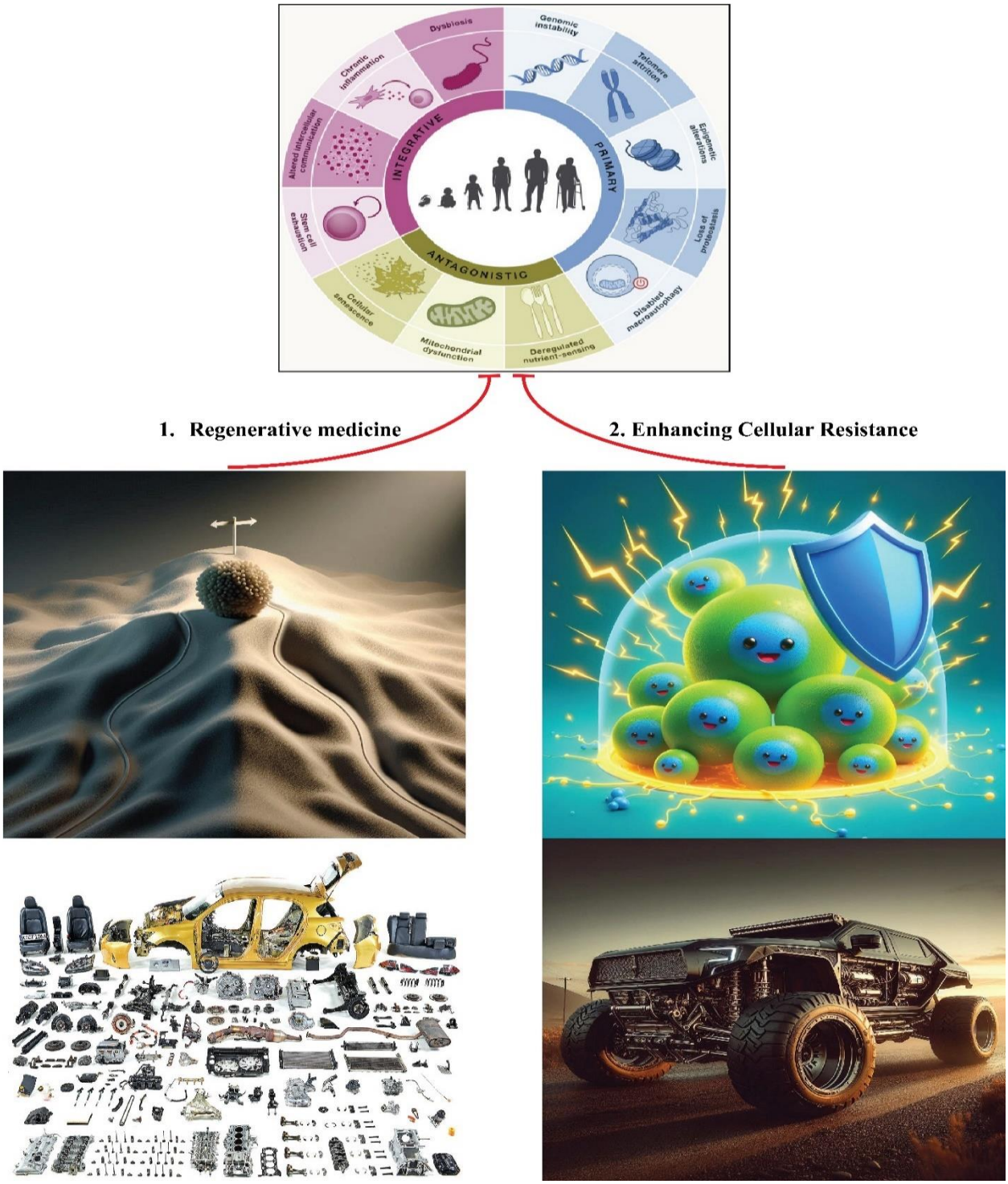


Figure 1-3: Two complementary strategies to combat aging: regenerative medicine and enhancing cellular resistance. Regenerative medicine involves replacing or repairing damaged and aged cells, akin to replacing worn-out parts in a car. This process, such as regenerating tissues through stem cell therapy or partial

reprogramming, relies heavily on the concept of cellular decision-making, where cells assess environmental signals and gene programs to choose between repair or regeneration. Enhancing cellular resistance, on the other hand, fortifies the resilience of existing cells, similar to reinforcing a car to endure tough conditions. This strategy includes the use of metabolic interventions to optimize cellular health and performance. Both approaches work together to address the hallmarks of aging, promoting extended lifespan and improved healthspan. [Hallmark figure from Ref. **(13)** and other figs generated by Dall-E 3**]**

For my master's thesis, we focused on modeling and analyzing different processes (16–18) especially pathways and regulators that can affect the extension of lifespan and healthspan in various model organisms, such as mTORC1, AMPK, and NAD(19,20). These pathways have been shown to play a major role in understanding how biological processes can be manipulated to promote longevity. We also delved into the complex connections between these master regulators and the circadian clock, which affects the rate of aging. By integrating the circadian rhythm's effect on metabolic processes, we aimed to uncover how synchronizing these pathways could potentially lead to enhanced health and longevity (21). To achieve this, we built these biological signaling pathways in silico using mechanistic models specifically ordinary differential equations, which allowed us to simulate and analyze their interactions. We also focused on modeling the effects of different drugs, such as metformin, resveratrol, and NMN, on this system. These compounds have gained attention for their potential to target key aging pathways and improve healthspan. By incorporating these drugs into our models, we sought to explore their mechanisms of action, the timing of the medicine for different age groups, and their effects on the dynamics of cellular aging. Overall, for my master's thesis, we mainly aimed to understand ways to increase the resilience of cells (second approach: Enhancing Cellular Resistance) by modeling the key pathways in aging and identifying potential interventions to promote healthy longevity.

For my PhD, we concentrate on the first approach: using computational and primarily data-driven models in regenerative medicine. Our main questions revolve around how to restore cells to their functional and healthy states. Specifically, we seek to answer: Which genes can be targeted to achieve the desired cellular state and type? When does cell decision-making happen, and what are the consequences of these decisions on the dynamics of the overall system? How accurately can we predict the effects of various perturbations, such as gene knocked out, diseases or drugs,

on cellular gene expression profiles? In biological processes like development or differentiation, which gene programs are active? Also, can we predict how perturbing early cells within a clone affects the cell type and gene expression profiles in subsequent cells of the clone?

As these developed methods have already undergone thorough rounds of peer review, they have been largely preserved in their original form, with only minor modifications to enhance clarity, integrate relevant background, and improve the overall flow of the thesis. The remainder of this manuscript is structured as follows:

Chapter 2 offers an overview of the core computational and mathematical approaches used throughout the thesis, including unsupervised and self-supervised learning, representation learning, dynamic model decomposition, bifurcation theory, and single-cell data modalities.

Chapter 3 introduces Fatecode, the first algorithm developed during my PhD. Fatecode is a computational method designed to predict cell fate regulators solely from single scRNA-seq data. By learning a latent representation through a deep learning-based classification-supervised autoencoder, it enables in silico perturbation experiments. These experiments identify genes that, when perturbed, can shift cell type distributions, either increasing or decreasing the population of specific cell types. The research titled *Fatecode Enables Cell Fate Regulator Prediction Using Classification-Supervised Autoencoder Perturbation* was conducted with Prof. Anita T. Layton, Prof. Sidhartha Goyal, and Prof. Gary Bader and has been published in Cell Reports Methods (1). We also experimentally validated Fatecode by reprogramming astrocytes into oligodendrocytes, in collaboration with researchers at the University of Toronto, in the study titled *Direct lineage conversion of postnatal mouse cortical astrocytes to oligodendrocyte lineage cells*, which is currently under review in *eLife* journal (22).

Chapter 4 presents CLERA, a novel computational framework for uncovering dynamic models and identifying active gene programs from single-cell RNA sequencing data. By combining a supervised autoencoder with Sparse Identification of Nonlinear Dynamics, CLERA uses prior knowledge to extract low-dimensional representations and reveal the driving forces

behind cellular processes. It pinpoints central genes, reconstructs gene expression dynamics, and captures key regulatory genes and temporal patterns across various cell types using personalized page rank. The research titled *Discovering Governing Equations of Biological Systems through Representation Learning and Sparse Model Discovery* was conducted in collaboration with Vasu Swaroop (coop student in our lab) and is currently under review in *Nucleic Acids Research* journal (2).

Chapter 5 introduces FateNet, a computational approach that integrates dynamical systems theory with deep learning to explore the timing of cell decision-making using scRNA-seq data. By leveraging information about normal forms and scaling behavior near bifurcations FateNet predicts when cell decisions occur with greater accuracy than conventional methods, while offering a qualitative understanding into the specific type of bifurcation the system is likely to experience. The work titled *FateNet: An Integration of Dynamical Systems and Deep Learning for Cell Fate Prediction* was conducted with Dr. Thomas Bury and published in Bioinformatics in 2024 (3).

Chapter 6 delves into scVAEDer's capabilities, the first generative deep-learning model that integrates autoencoders and deep diffusion models to model biological processes using scRNA-seq data. This approach learns a latent representation of the data, which captures both global patterns and local variations. scVAEDer's accuracy is shown by showing its ability to generate new high quality scRNA-seq data, predict perturbation effects across cell types, track gene expression changes during dedifferentiation, and identify key regulators in biological processes. The work titled *The Power of Two: Integrating Deep Diffusion Models and Variational Autoencoders for Single-Cell Transcriptomics Analysis* was conducted with Prof. Antia Layton and is currently under review at the Genome Biology journal (4).

Chapter 7 highlights Deep Lineage, a novel deep-learning method for analyzing time-series scRNA-seq data with matched lineage-tracing data. Our method accurately predicts early cell fate biases and gene expression profiles at different time points within a clone, surpassing current state-of-the-art methods in fate prediction accuracy. Also, through in silico perturbations in cellular reprogramming and hematopoiesis data, we show that Deep Lineage accurately models dynamic multicellular responses while identifying key genes and pathways associated with cell

fate determination. This research was conducted in collaboration with Allen Zhang and Prof. Gary Bader. The work titled *Deep Lineage: Single-Cell Lineage Tracing and Fate Inference Using Deep Learning* and is currently under review at *Cell Systems* (5).

Chapter 8 summarizes the key findings of my thesis and offers conclusions and future directions for research across various areas, including computational methods, biology, and social science.

Chapter 2
Computational methods

2.1 Machine Learning

Machine learning (ML) is a subfield of artificial intelligence (AI) focused on designing algorithms that learn from and make predictions based on data. Unlike traditional algorithms, which follow explicit instructions, machine learning models identify patterns in data autonomously and adapt as they are exposed to new information. At its core, machine learning involves finding an optimal function f that maps input data X to an output Y (23). In supervised learning, this function can be represented mathematically as:

$$f: X \rightarrow Y$$

Where X is the set of input features and Y is the output (target variable or label). ML methods are broadly and commonly categorized into supervised, unsupervised, and self-supervised learning, each serving distinct roles in data analysis (23). In the sections that follow, I will describe the machine learning methodologies used in my PhD research, focusing on supervised and unsupervised learning.

2.2 Supervised Learning

Supervised learning involves learning a function from labeled data, where each training sample consists of input-output pairs. The model's goal is to predict the correct output for unseen data based on what it learned from the labeled examples. This approach is widely used in tasks such as classification and regression.

2.2.1 Classification

Classification is the task of predicting discrete labels from input data. In my work, I used classification models to predict cell types from gene expression data (1,24), forecasting the fate of clones(5), and classifying patients according to their disease status(25). Mathematically,

classification involves learning a mapping $f: X \rightarrow \{1,2, \ldots, k\}$ where (k) represents the number of classes. For example, in predicting monocyte versus neutrophil fate, (X) would represent gene expression data, and $(k = 2)$. The classifier is trained by minimizing a loss function such as cross-entropy loss, which is given by:

$$L = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k} \log(\hat{y}_{i,k})$$

where $y_{i,k}$ is the true label, $\hat{y}_{i,k}$ is the predicted probability for class k, and N is the number of samples.

2.2.2 Regression

In regression, the model predicts continuous outputs rather than discrete labels(23). In my research, I used regression models to predict gene expression levels at future time points, such as Day 21 and Day 28, using data from days 6, 9, 12, and 15. The regression task can be described as learning a function $f: X \rightarrow R$, where the output is a continuous variable. The loss function typically used for regression is the mean squared error (MSE), which is expressed as:

$$L = \left(\frac{1}{N}\right)\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where y_i is the true output, $\hat{y}_i$ is the predicted output, and N is the number of samples. In both classification and regression, the primary challenge is to ensure that the model generalizes well to unseen data, which is typically addressed using regularization techniques.

2.3 Unsupervised Learning

Unsupervised learning techniques play a pivotal role in the analysis of large-scale datasets, especially in biological and cellular studies. These methods aim to uncover hidden patterns, relationships, and structures in the data without the use of labeled outputs. The core techniques revolve around modeling the data distribution, discovering its underlying structure, clustering

similar data points, and dimensionality reduction (26). Below, I outline these concepts in detail, focusing on their mathematical foundations.

2.3.1 Modeling the Data Distribution and Sampling

One of the primary goals in unsupervised learning is to model the probability distribution $P(X)$ of the data X. This helps in understanding the likelihood of certain data points and allows for the generation of synthetic samples that follow the same distribution as the original data. A common approach to modeling data distribution is through Gaussian Mixture Models, which assume that the data is generated from a mixture of several Gaussian distributions. The probability density function of a Gaussian Mixture Models can be expressed as:

$$P(X) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X \mid \mu_k, \Sigma_k)$$

where K is the number of Gaussian components, π_k is the weight of the k-th Gaussian, and $\mathcal{N}(X \mid \mu_k, \Sigma_k)$ represents the Gaussian distribution with mean μ_k and covariance matrix Σ_k. Maximizing the likelihood function for the Gaussian Mixture Models can be done using the Expectation-Maximization algorithm, iteratively refining the estimates of the parameters μ_k, Σ_k, π_k for each Gaussian component (23). Modeling the data distribution in this way enables us to infer underlying structures and generate new samples for simulations or data augmentation. This allows GMMs to capture complex data structures by modeling overlapping or non-spherical clusters. Moreover, GMMs provide practical applications such as inferring latent structures, creating synthetic datasets for augmentation, or identifying anomalies in data by detecting points that deviate significantly from the learned distribution. By leveraging this flexibility, GMMs become an essential tool for probabilistic modeling in unsupervised learning.

2.3.2 Revealing the Underlying Structure of the Data

In biological data, the relationships between cells or genes often lie in a non-obvious manifold or latent space. Revealing this latent structure requires non-linear approaches that can map the high-dimensional data into lower-dimensional manifolds while preserving the essential properties of the original dataset. One such approach is Manifold Learning, which assumes that the data lies in a lower-dimensional, non-linear subspace. One such method is Isomap, a technique that extends classical Multidimensional Scaling by incorporating geodesic distances. Isomap seeks to preserve the global geometry of the data by minimizing the following error:

$$\min_{Y} \sum_{i,j} \left(d_G(X_i, X_j) - |Y_i - Y_j|\right)^2$$

where $d_G(X_i, X_j)$ is the geodesic distance between points X_i and X_j in the high-dimensional space, and $|Y_i - Y_j|$ is the Euclidean distance between their low-dimensional representations. The geodesic distance d_G is computed by approximating the shortest paths along a neighborhood graph, capturing the structure of the data. This method is useful in biological contexts, such as tracing differentiation pathways or revealing gene expression landscapes, providing insights into complex cellular processes. It is worth noting that UMAP and t-SNE are among the most commonly used methods for visualizing scRNA-seq data structures(27,28).

2.3.3 Clustering

Clustering is a fundamental unsupervised technique used to group similar data points based on their feature similarities. The goal is to assign data points X_i to clusters C_k, such that points within the same cluster are more similar to each other than to those in other clusters. One of the most widely used clustering algorithms is K-Means. The algorithm iteratively assigns data points to K clusters based on the Euclidean distance to the cluster centroids. Mathematically, K-Means aims to minimize the following objective function:

$$J = \sum_{k=1}^{K} \sum_{X_i \in C_k} |X_i - \mu_k|^2$$

where μ_k is the centroid of the k-th cluster, and $X_i \in C_k$ denotes that the data point X_i belongs to cluster C_k. The algorithm alternates between assigning points to the nearest cluster and updating the centroids, leading to compact and well-separated clusters. Another notable technique is Density-Based Spatial Clustering of Applications with Noise (DBSCAN), which identifies clusters based on the density of data points, making it particularly effective for datasets with irregular shapes and noisy data (29). DBSCAN assigns points to clusters if they are densely packed within a neighborhood radius ϵ, and labels points as outliers if they do not meet this criterion.

2.3.4 Dimensionality Reduction

High-dimensional data, such as scRNA-seq, can be challenging to interpret directly due to the "curse of dimensionality." Dimensionality reduction techniques aim to project this data into a lower-dimensional space while retaining the most relevant information. Principal Component Analysis (PCA) is a commonly used method that projects data into a subspace spanned by the directions of maximum variance. The objective is to find a set of orthogonal vectors (principal components) that maximize the variance of the projected data:

$$\max \quad v^T \Sigma v$$

where Σ is the covariance matrix of the data, and v is the vector corresponding to the principal components. For non-linear data, methods such as Uniform Manifold Approximation and Projection (UMAP) have proven effective(28). UMAP optimizes the cross-entropy between the high-dimensional and low-dimensional representations of the data, preserving both global and local structures. The algorithm constructs a weighted graph where points are connected based on a similarity metric, such as cosine or Euclidean distance, and optimizes the embedding to reflect this graph in the lower-dimensional space.

$$\min_Y \sum_{i,j} P_{ij} \log \frac{P_{ij}}{Q_{ij}} + \left(1 - P_{ij}\right) \log \frac{1 - P_{ij}}{1 - Q_{ij}}$$

where P_{ij} and Q_{ij} are the probabilities of points i and j being connected in the high-dimensional and low-dimensional spaces, respectively. By reducing the dimensionality of the data, we can

visualize complex relationships and trajectories that were otherwise obscured in the high-dimensional space, aiding in the interpretation of biological processes and cell differentiation.

2.4 Self-Supervised Learning

Self-supervised learning has emerged as a pivotal paradigm within the field of machine learning, bridging the gap between supervised and unsupervised learning (23). Unlike traditional supervised learning, which relies on labeled data, and unsupervised learning, which seeks to identify inherent structures without labels, self-supervised learning leverages the data itself to generate supervisory signals(30). This approach enables models to learn meaningful representations by solving pretext tasks derived from the input data, thereby effectively using large amounts of unlabeled data.

At the core of self-supervised learning is the concept of creating auxiliary tasks where the labels are intrinsically derived from the data. These tasks train the model to understand and capture the underlying structures and patterns within the data, facilitating the learning of robust and generalizable features. Mathematically, let X denote the input data. In self-supervised learning, a defined task is defined to generate a pseudo-label Y from X (30). The model is then trained to predict Y given X, thereby learning representations that are useful for downstream tasks. This can be expressed as:

$$Y = f(X)$$

$$\min_{\theta} L(g_{\theta}(X), Y)$$

where f is the function defining the defined task, g_{θ} represents the model with parameters θ, and L is the loss function measuring the discrepancy between the model's predictions and the pseudo-labels.

One of the most recent techniques in self-supervised learning is Masked Language Modeling (MLM), which was popularized by models like BERT in the realm of natural language processing (31). MLM involves masking certain parts of the input and training the model to predict the masked elements. For a given input sequence $X = (x_1, x_2, \ldots, x_n)$, a subset of tokens

$\{x_{m_1}, x_{m_2}, \ldots, x_{m_k}\}$ are masked, resulting in $\tilde{X}$. The model is then trained to predict the original tokens based on the masked input:

$$\mathcal{L} = -\sum_{m \in \mathcal{M}} \log P\left(x_m \mid \tilde{X}\right)$$

Here, $\mathcal{M}$ is the set of masked positions, and $P\left(x_m \mid \tilde{X}\right)$ is the probability of the model predicting the original token x_m given the masked input $\tilde{X}$.

Generative Models also play a crucial role in self-supervised learning by aiming to model the underlying data distribution through data generation. Autoencoders can be seen as a self-supervised learning method, leading to the modeling of underlying data distributions through their encoding capabilities(23). An autoencoder consists of an encoder f_θ and a decoder g_ϕ, where the encoder maps the input X to a latent representation Z, and the decoder reconstructs X from Z:

$$Z = f_\theta(X)$$

$$\hat{X} = g_\phi(Z)$$

The objective is to minimize the reconstruction loss:

$$\mathcal{L} = |X - \hat{X}|^2$$

The optimization process in self-supervised learning typically involves minimizing a loss function that captures the discrepancy between the model's predictions and the pseudo-labels generated from the data. Optimization techniques such as stochastic gradient descent (SGD) are employed to iteratively update the model parameters θ to minimize the loss:

$$\theta^{t+1} = \theta^t - \eta \nabla_\theta L(\theta^{(t)})$$

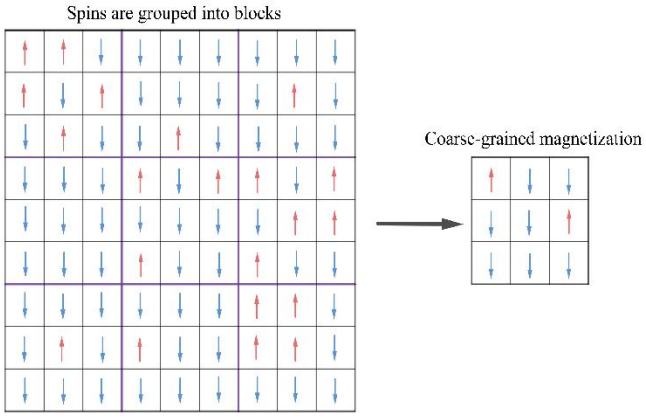
where η is the learning rate, and $\nabla_\theta L(\theta^{(t)})$ denotes the gradient of the loss with respect to the parameters. Through such optimization, self-supervised learning models progressively refine their representations, capturing intricate patterns and dependencies within the data.

2.4.1 Representation Learning

In statistical physics and quantum field theory renormalization group theory provides a powerful framework for understanding how systems behave across different scales (32). At its core, renormalization involves transforming a system by systematically zooming out to focus on large-scale behavior while discarding microscopic details that are irrelevant at those scales. This transformation often is done manually, by defining rules or mappings that preserve essential features of the system while simplifying its description (model or data). For example, in the Ising model of magnetism, a transformation reduces a detailed lattice of spins to a coarser representation, allowing us to study phase transitions and critical phenomena without getting lost in irrelevant details (32) (Fig 2-1).

In data science and image processing, similar manual transformations can be found in techniques like JPEG image compression (33) (Fig 2-1). The JPEG algorithm uses the Discrete Cosine Transform to convert an image from the spatial domain where each pixel is a value into the frequency domain where patterns of varying frequencies are represented. High-frequency components, which often correspond to noise or less important details, are discarded, while low-frequency components, which capture the essential structure of the image, are kept. This transformation is manually defined, with explicit rules for how to process the data. While effective for specific tasks like image compression, manual transformations require human expertise and may not generalize well to other domains.

a



b



Figure 2-1: Renormalization in the Ising Model and JPEG Image Compression. (a) Renormalization in the Ising model, where spins on a fine lattice are grouped into blocks to form a coarser representation of the system's magnetization. (b) JPEG image compression of the image, showing the progressive reduction in detail through increasing levels of compression. Both processes highlight the principle of discarding fine-scale details to emphasize larger-scale structures.

Representation learning automates this process of transformation, allowing models to discover relevant transformations directly from the data without human intervention (26). In contrast to manual approaches where specific rules are set (feature engineering), representation learning models can learn to extract features from raw data in a more flexible and adaptive manner. This is particularly advantageous when dealing with complex, high-dimensional data, such as gene expression profiles or natural images, where the best transformation is often unknown or difficult to handcraft (34). Representations learned by neural networks are now widely used in different ML areas, such as speech recognition (35), natural language processing and video representation (36), and domain adaptation (37).

At the heart of representation learning is the idea that a model can learn to transform data into a latent representation that makes it easier to solve downstream tasks. In the supervised learning setting, this process is tightly coupled with the task at hand. The model is trained on labeled data, and as it learns to perform the task (e.g., classification or regression), it simultaneously learns to extract useful features from the input data. Usually, this is done end-to-end, where the model is not only optimizing the final task but also learning intermediate representations through layers of abstraction. For example, in a deep convolutional neural network used for image recognition, the lower layers may learn to detect edges or textures, while higher layers capture more complex patterns like shapes or objects. Mathematically, we can represent this transformation as learning a function $f: X \to Z$, where X is the input data and Z is the learned representation. The goal is to find a representation Z that simplifies the problem, allowing the model to efficiently solve a task $g: Z \to Y$, where Y is the output. In supervised learning, the objective is to minimize a loss function $L\big(g\big(f(X)\big), Y\big)$, adjusting the model parameters such that the learned representation $f(X) = Z$ is optimal for the specific task.

The process of labeling data is both expensive and time-consuming, and most datasets remain largely unlabeled, so unsupervised methods offer a more practical approach to representation learning. Instead of generating task-specific representations that require labels, unsupervised learning focuses on uncovering more general-purpose representations that can capture the core structure of the data(38). The key challenge in this approach is finding transformations that reveal the hidden patterns, correlations, and variations in the data, without the benefit of labeled examples. The transformation encodes the underlying factors of variation in a way that can be useful across different tasks, without being tied to any specific one. Representation learning methods like autoencoders aim to learn latent variables that summarize the data in a compact form while preserving its most important characteristics. These latent variables, or representations, capture key patterns and dependencies that can be generalized across a range of tasks. The power of these methods lies in their ability to represent data in a lower-dimensional space while retaining the most relevant information(39). However, even though these techniques are powerful, they face a significant challenge related to identifiability.

Identifiability is important in learning meaningful representations, as it refers to the ability to uniquely recover the true latent structure from the observed data. In the linear case, methods like independent component analysis (ICA) are often used to separate mixed observations into independent latent sources, relying on statistical assumptions such as non-Gaussianity(40) (Fig 2-2). While ICA is well-studied and effective in linear scenarios, its ability to handle nonlinear transformations is severely limited. Nonlinear transformations introduce too many degrees of freedom, making the problem of recovering the original latent components ill-posed(40) (Fig 2-2). The fundamental difficulty of nonlinear identifiability is captured in the following theorem which shows that in a nonlinear setting, it is possible to find a transformation that results in independent components, even if those components do not correspond to the true sources of variation in the data(39):

Theorem 1: Let x be a random vector of any distribution (x should be continuous). Then, there exists a transformation $g: \mathbb{R}^d \to [0,1]^d$ such that $z = g(x)$ has a uniform distribution, with its

components being independent. Moreover, the function **g** can be chosen so that the first variable is simply transformed by a scalar function: $z_1 = g_1(x_1)$.

To overcome this limitation, one strategy is to introduce auxiliary variables or leverage temporal dynamics to provide additional structure for learning identifiable representations(40). For instance, in time series data, temporal dependencies can act as a source of information that helps disentangle the underlying factors of variation. The time evolution of data points can offer insights into the relationships between latent components, making it easier to identify the true structure.

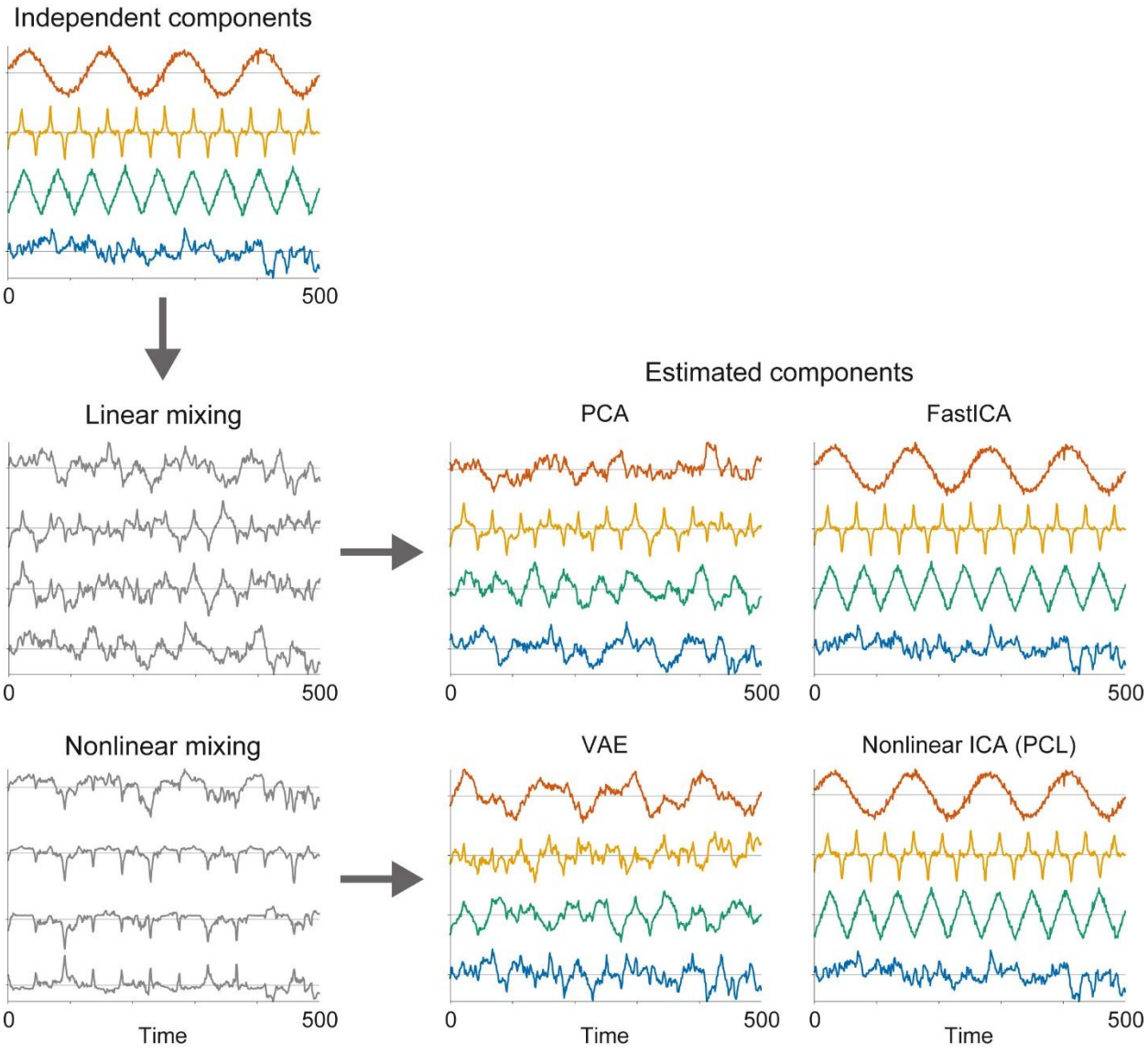


Figure 2-2: Identifiability of ICA and its application on blind source separation illustrated. Demonstration of ICA Identifiability and Source Separation. The top row displays the original signals, which are then combined either linearly or nonlinearly, shown in the middle and bottom rows, respectively. Linear ICA and nonlinear ICA are applied to these mixtures, successfully recovering the original signals in the rightmost column. In contrast, when PCA and its nonlinear variant, VAE (without further constraints), are applied to the same mixtures (middle column), separation is not achieved. Figure from Ref. **(40).**

Another approach is to incorporate task-specific knowledge or constraints that provide guidance for learning more identifiable models. By integrating prior knowledge about the domain or imposing constraints on the learned representations, we can reduce the ambiguity inherent in nonlinear transformations. This can be done through regularization techniques or auxiliary tasks that help the model focus on meaningful variations in the data (40). For example, auxiliary tasks like cell type classification or trajectory prediction can help guide the learning process by enforcing consistency between the learned representations and known biological processes.

In projects like Fatecode, CLERA, and Deep Lineage, we adopted this approach by combining the nonlinear transformations of autoencoders with auxiliary tasks and domain-specific regularization. For example, in Fatecode, we combined deep learning-based methods with auxiliary tasks related (cell type classification), while in CLERA, we used auxiliary information such as cell type classification and dynamic regularization through methods like Sparse Identification of Nonlinear Dynamics (SINDy) (41). This combination of non-linear autoencoders with auxiliary tasks enabled the learning of more identifiable and biologically meaningful representations that captured the underlying structure of the data, which allows for more accurate predictions and insights into cellular dynamics.

2.5 The Challenge of Interpretability in Deep Neural Networks

Although mentioned machine learning approaches have revolutionized many fields, they often face limitations in scientific domains where interpretability and generalization are important. ML models, especially deep learning, excel at fitting complex datasets but tend to act as "black boxes" providing little insight into the underlying physical processes. This lack of interpretability can hinder the ability to uncover the governing laws or equations that describe a system, as the relationships between variables remain obscured(42). Also, ML models often struggle to generalize beyond the specific data they were trained on, which limits their ability to predict unobserved phenomena or extend to different regimes of a system’s behavior. In contrast, approaches rooted in physics and applied mathematics aim to discover governing equations that are interpretable, generalizable, and aligned with the constraints of the system. Efforts have been

made to combine the traditional mechanistic modeling strategies (common in physics and mathematics) with data driven approaches. This combination is particularly valuable when the governing equations of a system are unknown, and we aim to infer them directly from observational data(43). Linear methods, such as Koopman theory, offer a foundation for understanding dynamical systems through linear representations, while more advanced techniques like the SINDy method extend this framework to nonlinear systems (44).

2.5.1 Koopman Theory

In the linear case, methods like Koopman theory offer a valuable framework for analyzing complex systems by converting nonlinear dynamics into linear representations in a higher-dimensional space(44). The key idea behind Koopman theory is that even though a system may behave nonlinearly in its natural state, it is possible to describe this system using linear operators when viewed through a set of observables. These observables represent different aspects of the system's state, and by expanding the system into this higher-dimensional space, the dynamics become linear(45). This transformation simplifies the analysis of complex behaviors, allowing scientists to leverage traditional linear techniques such as eigenvalue decomposition to study nonlinear phenomena. Mathematically, the Koopman operator K acts on a function $f(x)$, where x is the system's state, transforming it according to the system's time evolution $T(x)$:

$$K\, f(x) = \, f\big(T(x)\big)$$

By working in this space of observables, researchers can decompose the system's evolution into Koopman modes, which represent different components of the dynamics. This method is particularly useful for studying time-series data and extracting global features of a system's behavior over time. However, while Koopman theory can reveal important aspects of a system, it also has limitations. One of the main challenges is that accurately representing a nonlinear system in a linear framework often requires expanding the space into a very high (or even infinite) dimensional setting. This not only complicates the model but can also hinder interpretability, making it difficult to draw clear conclusions about the underlying mechanisms driving the system.

2.5.2 Sparse Identification of Nonlinear Dynamics

In contrast to linear methods like Koopman theory, SINDy is specifically designed to identify governing equations for systems that exhibit nonlinear behavior (41). The core idea behind SINDy is that many physical systems, despite their complexity, can be described by a small number of dominant interactions or governing laws. These interactions are often nonlinear, and SINDy's strength lies in its ability to uncover these underlying equations directly from data, while promoting sparsity to keep the model interpretable.

The starting point for SINDy is a set of time-series data that describe the state of the system at different points in time. From this data, the goal is to discover a system of equations that describe the rate of change of the system's state over time. Formally, consider a system whose dynamics are governed by the following differential equation:

$$\frac{dx}{dt} = f(x)$$

where $x = [x_1, x_2, \ldots, x_n]$ is a vector representing the state of the system, and $f(x)$ is an unknown function that describes how the state evolves over time. The objective of SINDy is to approximate this function $f(x)$ by a sparse combination of candidate functions that can explain the system's dynamics.

To achieve this, SINDy constructs a library of candidate functions $\Theta(x)$, which may include polynomials, trigonometric functions, or any other basis functions that are relevant to the system being studied. The library matrix $\Theta(x)$ is structured as follows:

$$\Theta(\mathrm{x}) = [1 \quad x_1 \quad x_2 \quad x_1^2 \quad x_2^2 \quad x_1 x_2 \quad \ldots]$$

Each row corresponds to a time step in the observed data, while each column represents a possible nonlinear function of the state variables. The task is now to find a sparse vector ξ such that:

$$\frac{dx}{dt} = \Theta(x)\xi$$

Here, ξ is a vector of coefficients that selects the relevant terms from the library, promoting sparsity through techniques such as Lasso regression or sequential thresholding (46). The sparsity constraint ensures that only a few terms from the candidate library are used to model the system's dynamics, resulting in a simplified and interpretable model.

2.6 Dynamical Systems and the Theory of Bifurcation

Dynamical systems are mathematical models used to describe the time-dependent behavior of complex systems in various fields, including biology, physics, and engineering. These systems can be represented by ordinary differential equations (ODEs) or partial differential equations (PDEs) that govern the evolution of the system's state over time. In the context of biology, for example, dynamical systems can model processes like gene regulation, cellular growth, or population dynamics(47).

One of the most critical aspects of studying dynamical systems is understanding how their behavior changes as parameters within the system are varied. A bifurcation refers to a qualitative change in the system's long-term behavior as a parameter crosses a critical threshold (47). At this point, the system's equilibrium or periodic solution changes its stability, leading to phenomena such as sudden transitions between different states or the emergence of new patterns.

2.6.1 Bifurcation Theory

Bifurcation theory provides a mathematical framework for studying these qualitative changes. It helps identify the points, called bifurcation points, where the system's behavior shifts and allows us to classify the types of bifurcations based on the nature of these transitions (48). Mathematically, bifurcations occur when the Jacobian matrix of a system evaluated at its fixed points has eigenvalues crossing the imaginary axis of the complex plane, signaling a change in stability. Consider a system described by the differential equation:

$$\frac{dx}{dt} = f(x, \alpha)$$

where x represents the state of the system, and μ is a bifurcation parameter. A bifurcation occurs when small changes in μ cause significant changes in the nature of the solutions to this equation.

2.6.2 Core Bifurcation Types in Dynamical Systems

2.6.2.1 **Saddle-Node Bifurcation**

A saddle-node bifurcation occurs when two equilibrium points (one stable and one unstable) collide and annihilate each other as the bifurcation parameter is varied(49) (Fig 2-3). This type of bifurcation is common in systems where the stability of a state can abruptly disappear, leading to sudden transitions. Saddle-node bifurcations are characterized by the following equation:

$$\frac{dx}{dt} = \alpha - x^2$$

In this equation, as the parameter μ passes through zero, two fixed points $x = \pm\sqrt{\alpha}$ merge and vanish. For $\alpha > 0$, there are two fixed points: one stable and one unstable. For $\alpha < 0$, no fixed points exist, leading to a qualitative change in the system's behavior.

2.6.2.2 **Hopf Bifurcation**

A Hopf bifurcation occurs when a pair of complex-conjugate eigenvalues of the Jacobian cross the imaginary axis as the bifurcation parameter is varied, leading to the emergence of periodic solutions (limit cycles)(49) (Fig 2-3). This transition marks a change from a stable equilibrium to an oscillatory state, often seen in systems that exhibit periodic behavior, such as biological rhythms or chemical reactions. The normal form of a Hopf bifurcation can be written as:

$$\frac{\mathrm{dz}}{\mathrm{dt}} = z((\alpha + \mathrm{i}\omega) - |\mathrm{z}|^2)$$

Here, z is a complex variable, α is the bifurcation parameter, and ω is the natural frequency of oscillation. When α crosses zero, the system undergoes a bifurcation from a stable equilibrium to a limit cycle with frequency ω.

2.6.2.3 **Pitchfork Bifurcation**

A pitchfork bifurcation occurs when a symmetric system exhibits a transition from a single stable equilibrium to multiple equilibria as the bifurcation parameter is varied(49) (Fig 2-3). This bifurcation comes in two forms: supercritical and subcritical. In the supercritical case, a stable equilibrium splits into two stable equilibria and one unstable equilibrium, while in the subcritical case, the bifurcation produces two unstable equilibria. The standard form of a pitchfork bifurcation is:

$$\frac{dx}{dt} = \alpha x - x^3$$

For $\alpha < 0$, the system has a single stable equilibrium at $x = 0$. As $\alpha > 0$, the system bifurcates into two stable equilibria at $x = \pm\sqrt{\alpha}$, with $x = 0$ becoming unstable.

2.6.2.4 **Transcritical Bifurcation**

In a transcritical bifurcation, two equilibrium points exchange their stability as the bifurcation parameter changes (49) (Fig 2-3). This type of bifurcation is often seen in population dynamics, where one equilibrium represents extinction, and another represents a stable population. The normal form for a transcritical bifurcation is:

$$\frac{dx}{dt} = \alpha x - x^2$$

As α crosses zero, the equilibria at $x = 0$ and $x = \alpha$ exchange stability, causing a shift in the system's dynamics.

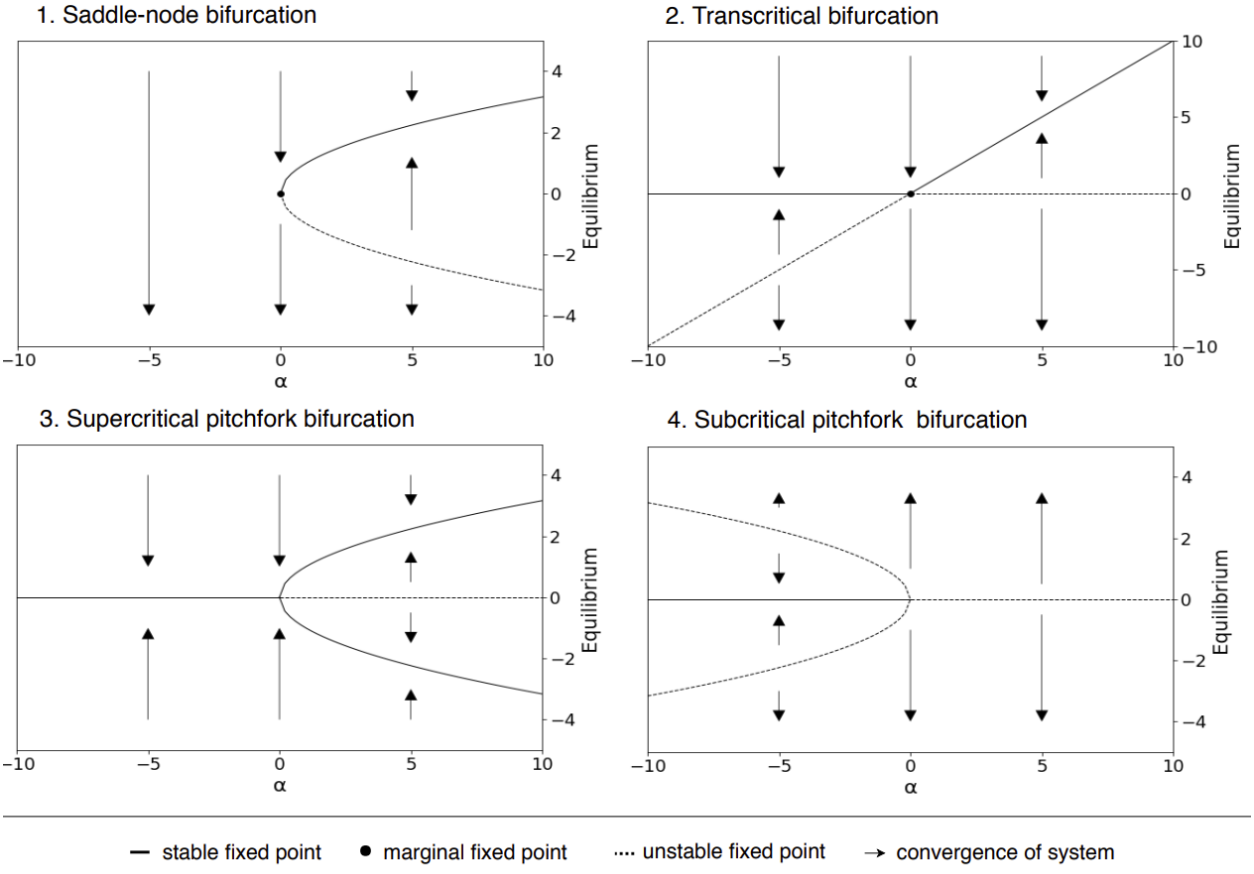


Figure 2-3: Bifurcation diagrams for the saddle–node, transcritical, pitchfork co–dimension 1 bifurcation. The plot shows the stability properties of dynamical systems depending on the bifurcation parameter α and the initial condition of the system IC. Figure from Ref **(50).**

2.7 Single-cell genomics

scRNA-seq has revolutionized modern biology by enabling the high-resolution analysis of individual cells. Initially constrained to experiments with a few hundred cells, scRNA-seq technologies have rapidly evolved to accommodate datasets with millions of cells spanning various tissues and organs (51). This vast increase in scale has allowed for an unprecedented view of cellular heterogeneity, showing the diversity within a seemingly uniform cell population that was masked by traditional bulk assays. With this level of granularity, scRNA-seq provides critical information into the processes that underlie cellular decision-making whether a cell commits to a

specific lineage, responds to external stimuli, or deviates from normal functioning in pathological states. Understanding how cells make decisions is a central challenge in biology because these decisions ultimately dictate the functional organization of tissues and organs(52). Cellular decision-making is a complex and dynamic process driven by intrinsic factors, such as gene expression profiles and chromatin accessibility, as well as extrinsic signals from the microenvironment. Moreover, the stochastic nature of molecular interactions introduces another layer of complexity, making it essential to investigate decision-making at the single-cell level. Investigating the mechanisms by which cells transition between states during normal development, tumor progression, or tissue regeneration is crucial, as this knowledge can inform the development of targeted therapies and interventions in cancer treatment, regenerative medicine, and immune system modulation(53).

From a computational perspective, scRNA-seq data is often represented as a matrix X with dimensions $n \times m$, where n is the number of cells and m is the number of genes. This structure facilitates a variety of downstream analytical tasks, including clustering cells into distinct subpopulations, identifying differentially expressed genes, and reconstructing developmental trajectories(54). These tasks enable us to explore how gene expression changes across cell types, how cells progress through different states over time, and how rare cell types can be identified within a heterogeneous population. Furthermore, dimensionality reduction techniques, such as PCA or UMAP(28), are commonly used to visualize high-dimensional data in two or three dimensions, providing interpretable visualization of cellular diversity.

The continuous advancement of single-cell technologies has also introduced new modalities, expanding beyond RNA sequencing to include measurements of chromatin accessibility (scATAC-seq), protein expression (CITE-seq), and even multi-omic approaches that capture multiple layers of cellular information simultaneously(55). These technologies offer a more comprehensive view of cellular states by integrating data on transcriptional activity, epigenetic regulation, and protein expression. Multimodal data can provide a richer context for understanding the regulatory networks that drive cellular decision-making. For instance, scATAC-seq can reveal the regions of the genome that are accessible due to open chromatin conformation,

while CITE-seq allows for the quantification of surface proteins in conjunction with RNA expression, giving a more complete view of cell identity and function (55). Furthermore, advancements in single-cell methodologies have paved the way for other techniques like Perturb-seq and lineage tracing. These methods offer powerful tools to study not only the static gene expression profiles of cells, but also the temporal and causal relationships between gene regulation, cellular identity, and fate.

Lineage tracing is a method that allows researchers to track the developmental trajectories of cells over time, which provides a direct view into how individual cells evolve into distinct lineages(56). This method has been invaluable in studies of development, regeneration, and disease, as it enables the mapping of how a single progenitor cell can give rise to diverse cell types. Traditionally, lineage tracing involved genetic markers introduced into cells to permanently label their progeny. However, with the advent of high-throughput sequencing, more sophisticated barcoding strategies such as CRISPR-based lineage tracing have been developed. In these approaches, cells are engineered to express unique DNA sequences (barcodes) that can be read out via sequencing, to reconstruct their lineage relationships. When combined with scRNA-seq, lineage tracing provides a detailed view of how cells transition through various states. This integration allows us to map not only where a cell came from but also how its gene expression changes as it commits to a specific fate. Lineage tracing can reveal how a stem cell differentiates into various cell types or how cells evolve during cancer progression. By integrating lineage data with transcriptional profiles, we can identify the molecular mechanisms and key regulators driving these transitions within a biological process which are essential in developmental biology and regenerative medicine, leading to new strategies for tissue engineering and repair.

Perturb-seq is another transformative technique that combines CRISPR-mediated gene perturbation with scRNA-seq, enabling the systematic study of gene function at high resolution (57). This method allows us to simultaneously target multiple genes and assess the downstream consequences on cellular behavior and leads to a high-resolution view of gene regulatory networks (GRNs). The strength of Perturb-seq lies in its ability to link perturbations with cellular phenotypes in a highly multiplexed manner. By perturbing dozens or even hundreds of genes concurrently,

researchers can measure how each perturbation affects the transcriptional landscape of individual cells. This is particularly powerful for causal discovery in GRNs, as it overcomes the limitations of relying solely on observational data (58). Perturbation data generated by Perturb-seq can provide the necessary interventions to enable causal inference and the identification of true causal relationships between variables which facilitates a more accurate reconstruction of GRNs. Perturb-seq has been successfully applied across various fields, including immunology and cancer research. In immunology, it has uncovered key regulators of immune responses that could be targeted for therapeutic intervention (59). In cancer research, it has revealed how different cell types within tumor microenvironments respond to perturbations, highlighting the variability in cancer progression and resistance mechanisms (60). By perturbing transcription factors or signaling pathways, we can learn how molecular circuits shape cellular decision-making processes, which reveals crucial mechanisms underlying cell identity, differentiation, and responses to external stimuli. Integrating Perturb-seq with computational models enables prediction of perturbation effects across different cell types, advancing precision medicine strategies that tailor treatments based on cell-specific gene regulatory profiles.

Throughout my PhD research, I focused on developing various computational methods that leverage scRNA-seq data to model and explore the complexities of cellular decision-making processes. My work addressed various aspects of this complex process, including unraveling the temporal dynamics of cell fate decisions within biological systems, aiming to uncover the timing of these pivotal moments and their consequences on the system. I also developed methods to identify key genes and active gene programs driving these decision-making processes and examined how different perturbations such as genetic manipulations, disease states, and drug interventions affect gene expression and cell states. These methods revealed fundamental mechanisms underlying cellular decision-making and established a robust and precise framework for predicting cellular responses to different perturbations. I hope my research advances our ability to understand and manipulate cellular decision making processes in both health and disease contexts.

Chapter 3
Fatecode

This chapter introduces Fatecode, the first algorithm developed during my PhD. Fatecode is a computational method designed to predict cell fate regulators solely from scRNA-seq data. By learning a latent representation through a deep learning-based classification-supervised autoencoder, it enables in silico perturbation experiments. These experiments identify genes that, when perturbed, can shift cell type distributions, either increasing or decreasing the population of specific cell types. The research titled *Fatecode Enables Cell Fate Regulator Prediction Using Classification-Supervised Autoencoder Perturbation* was conducted with Prof. Anita T. Layton, Prof. Sidhartha Goyal, and Prof. Gary Bader and has been published in Cell Reports Methods (1). The author list on the publication is below:

Mehrshad Sadria[1], Anita Layton[1,2,3,4], Sidharta Goyal[5], Gary D. Bader[6,7,8,9,10]
[1] Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada.
[2] Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.
[3] Department of Biology, University of Waterloo, Waterloo, Ontario, Canada.
[4] School of Pharmacy, University of Waterloo, Waterloo, Ontario, Canada.
[5] Department of Physics, University of Toronto, Toronto, Ontario, Canada.
[6] Department of Molecular Genetics, University of Toronto, Toronto, Ontario, Canada.
[7] The Donnelly Centre, University of Toronto, Toronto, Ontario, Canada.
[8] Department of Computer Science, University of Toronto, Toronto, Ontario, Canada.
[9] The Lunenfeld-Tanenbaum Research Institute, Sinai Health System, Toronto, Ontario, Canada.
[10] Princess Margaret Cancer Centre, University Health Network, Toronto, Ontario, Canada.

We also experimentally validated Fatecode by reprogramming astrocytes into oligodendrocytes, in collaboration with researchers at the University of Toronto, in the study titled *Direct lineage conversion of postnatal mouse cortical astrocytes to oligodendrocyte lineage cells*, which is currently under review in *eLife* journal (22).

3.1 Overview

Cell reprogramming, which guides the conversion between cell states, is a promising technology for tissue repair and regeneration, with the ultimate goal of accelerating recovery from diseases or injuries. To accomplish this, regulators must be identified and manipulated to control cell fate. The ability to accurately identify cell fate regulators from single-cell transcriptomics data would help accomplish this goal. We propose Fatecode, a computational method that predicts cell fate regulators based only on scRNA-seq data. Fatecode learns a latent representation of the scRNA-seq data using a deep learning-based classification-supervised autoencoder and then performs in silico perturbation experiments on the latent representation to predict genes that when perturbed would alter the original cell type distribution to increase or decrease the population size of a cell type of interest. We assessed Fatecode's performance using simulations from a mechanistic gene regulatory network model and scRNA-seq data mapping blood and brain development of different organisms. Our results suggest that Fatecode can detect known cell fate regulators from single-cell transcriptomics datasets. We hope this method will accelerate the discovery of novel cell fate regulators that can be used to engineer and grow cells for therapeutic use in regenerative medicine applications.

3.2 Introduction

In tissue development, specific regulator genes control how cells change state and type to form a complete tissue (61). These regulators are also important because they can be used to control cell fate for multiple applications, including in regenerative medicine and cancer (62). However, it remains a challenge to identify these regulators within complex and dynamic tissue systems (61).

Cell fate regulators can be identified using experimental methods such as high-throughput genetic perturbation screens (e.g. CRISPR-based) with single-cell gene expression (scRNA-seq) readouts (63,64). However, these methods are challenging to run on arbitrary biological systems. Computational methods have been developed to predict gene expression programs that explain the difference between perturbed and unperturbed states (4,65–67) or to predict the linear effect of

perturbing a particular transcription factor (68). Also, computational methods which determine the ordering of cell states along a trajectory, based on their gene expression profiles using a pseudotime or actual time approach (69–73), have been used to examine the cell decision-making process by identifying genes that are differentially expressed between trajectory branches. However, these latter methods often have trouble identifying accurate trajectories and branch points (74). Furthermore, none of the above methods are designed to identify cell fate regulators in normal developmental processes.

We develop Fatecode, a computational method to predict important cell fate regulator genes for cell types of interest. Fatecode predicts cell fate regulators based only on scRNA-seq data covering a given range of cell types to be analyzed. Fatecode learns a latent representation of the scRNA-seq data using a deep learning-based classification-supervised autoencoder (75,76) and then performs in silico perturbation experiments on the latent representation to predict genes that when perturbed would alter the original cell type distribution to increase or decrease the population size of a cell type of interest. Fatecode can be thought of as an in silico CRISPR perturbation screen that identifies genes that influence cell fate, based on a cell type readout. These genes can be traditional (e.g. transcription factors) or non-traditional regulators (any other genes). We assessed Fatecode's performance using simulated data produced by a mechanistic model based on a pre-defined gene regulatory network with known cell fate regulators (77) and tested it on scRNA-seq maps of blood and developing brain from zebrafish and mouse (78–81).

3.3 Results

3.3.1 Fatecode method overview

Fatecode uses a classification-supervised autoencoder to detect key genes that can shift the cell type frequencies in an input scRNA-seq data set towards a desired distribution of cell types. Taking single-cell gene expression profiles as input, the autoencoder learns a latent space with reduced dimensions capturing the input information (reduce gene dimension x cell matrix). A supervised cell type classifier is included as part of the loss function to create a latent space composed of

features that support optimal cell type classification in addition to input data reconstruction. Known cell type annotations in the input data are used to train the classifier. This ensures that the latent space is relevant for cell type classification used in later stages. Each latent layer node of the autoencoder, which represents a reduced dimension of the input, is systematically perturbed to simulate altering key gene expression programs (sets of genes that are correlated with each other that are represented by individual learned latent layer dimension). Cell types are then reclassified to characterize the effect of the perturbation and the autoencoder's decoder uses the perturbed and unperturbed latent embeddings to generate a gene-by-cell matrix of gene prioritization scores. This matrix is used to identify genes important for the perturbation effect (Method section, Fig. 1, Supplementary Fig. 1). Resulting cell type distributions are generated for each possible perturbation and then manually evaluated to identify those that increase or decrease proportions of desired cell types. In this way, regulator genes are identified to increase or decrease a given cell type proportion relative to all other cell types and these are predicted to be cell fate regulators for the given cell type. An average of the cell fate regulator prioritization scores across cells in each cell type is computed to produce a final regulator list for each cell type.

Our latent layer perturbation approach is inspired by latent vector operations used in natural language processing and computer vision applications to generate novel text and images **(82–84)**. In those applications, perturbation operations performed on the latent layer generally yield superior results compared to operations performed directly in the input space. The classification component of Fatecode is used to exclude possible latent space regions that do not conform to the overall structure of the data. This helps in learning a model which is more representative of the underlying data distribution.

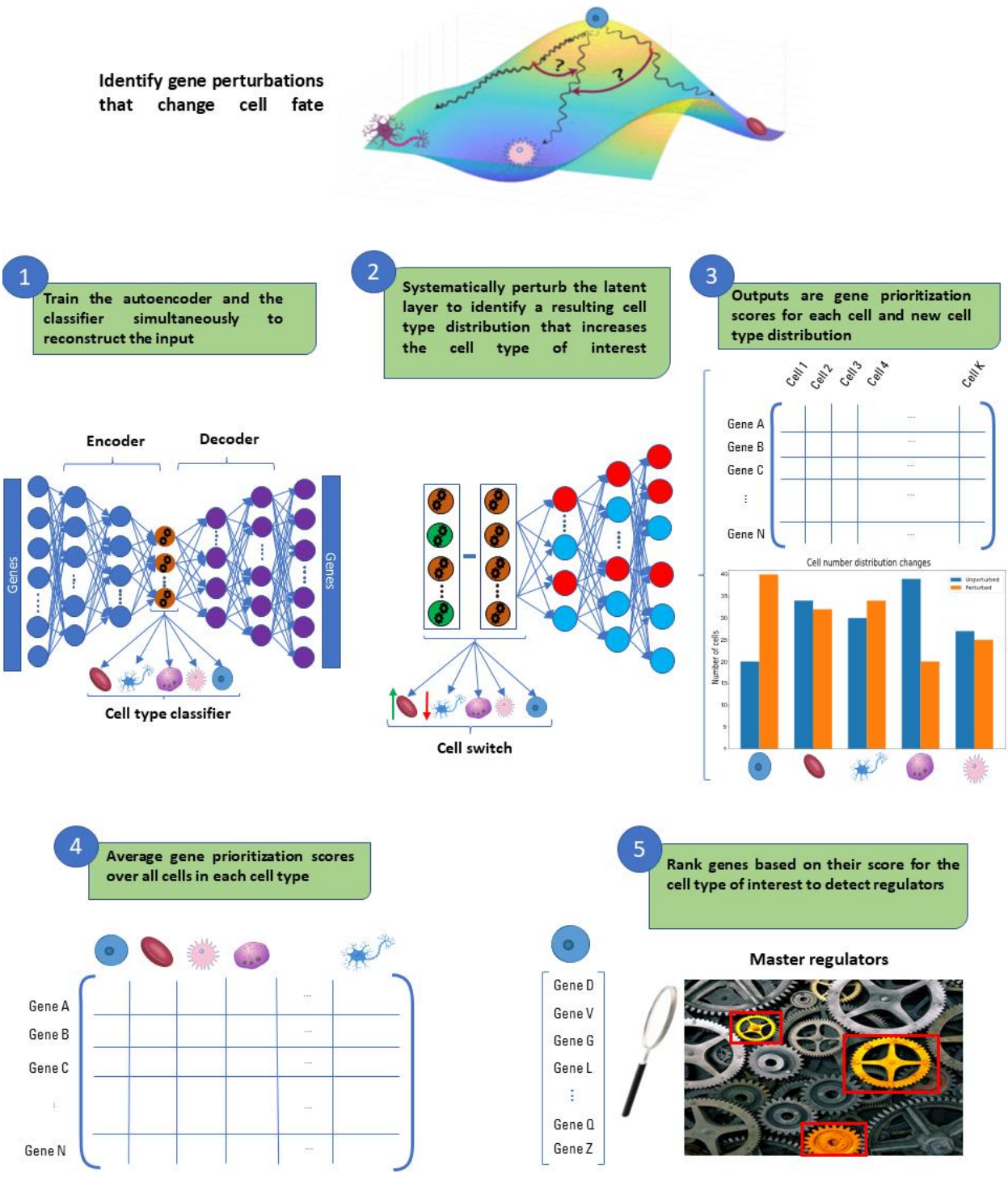


Figure 3-1: Fatecode workflow for in silico perturbation experiments and cell fate regulator detection. The 3D model (top) represents a Waddington-like landscape depicting cellular reprogramming processes. We

seek to identify genes (question marks) that regulate paths on this landscape (wavy lines), by transitioning them to another path (red arrows). A classification-supervised autoencoder learns a latent space representing the original data, optimized for both input reconstruction and cell type classification. The latent layer is systematically perturbed and by investigating all resulting perturbation-generated cell type distributions, distributions with an increase or decrease in a cell type of interest are identified. Perturbation output is simulated by subtracting the perturbed from unperturbed latent layers and feeding it to the decoder to identify a cell by gene matrix of prioritization scores that can help us to prioritize genes predicted to be important for achieving a desired cell population distribution. An average of the cell fate regulator prioritization scores across cells in each cell type is computed. By sorting these genes based on their prioritization scores for a cell type of interest, the model predicts genes that are important for regulating the levels of a given cell type.

3.3.2 Optimizing model architecture and hyperparameters

Fatecode relies on the latent embedding of an autoencoder, but different types of autoencoders may produce different results, depending on the input data (see Supplement) (66,85–87). To investigate this in our problem context, we evaluated the performance of three common autoencoder architectures: under-complete autoencoder (AE), variational autoencoder (VAE), and conditional variational autoencoder (CVAE) (88). The first step of Fatecode evaluates these three autoencoder architectures, and other hyperparameters (Supplementary Note 1), to find the ones that reconstruct the input data best, measured by mean squared error (MSE) for reconstruction and cross-entropy for cell type classification. To illustrate the importance of this step, we compared how the choice of autoencoder affects learning the underlying representation for two single-cell gene expression data sets in adult zebrafish blood (78) and murine pancreatic development (89). AE produced the lowest reconstruction error for the zebrafish data (averaged over cell types) (Figs. 2a,b). AE also produced a latent layer that successfully reduces the dimension and cleanly separates the five known cell types in the data (Fig. 2c), and its cell type classifier yields a high accuracy (Fig. 2d). However, for the mouse data, VAE achieved a higher accuracy compared to the other autoencoders (Supplementary Fig. 2).

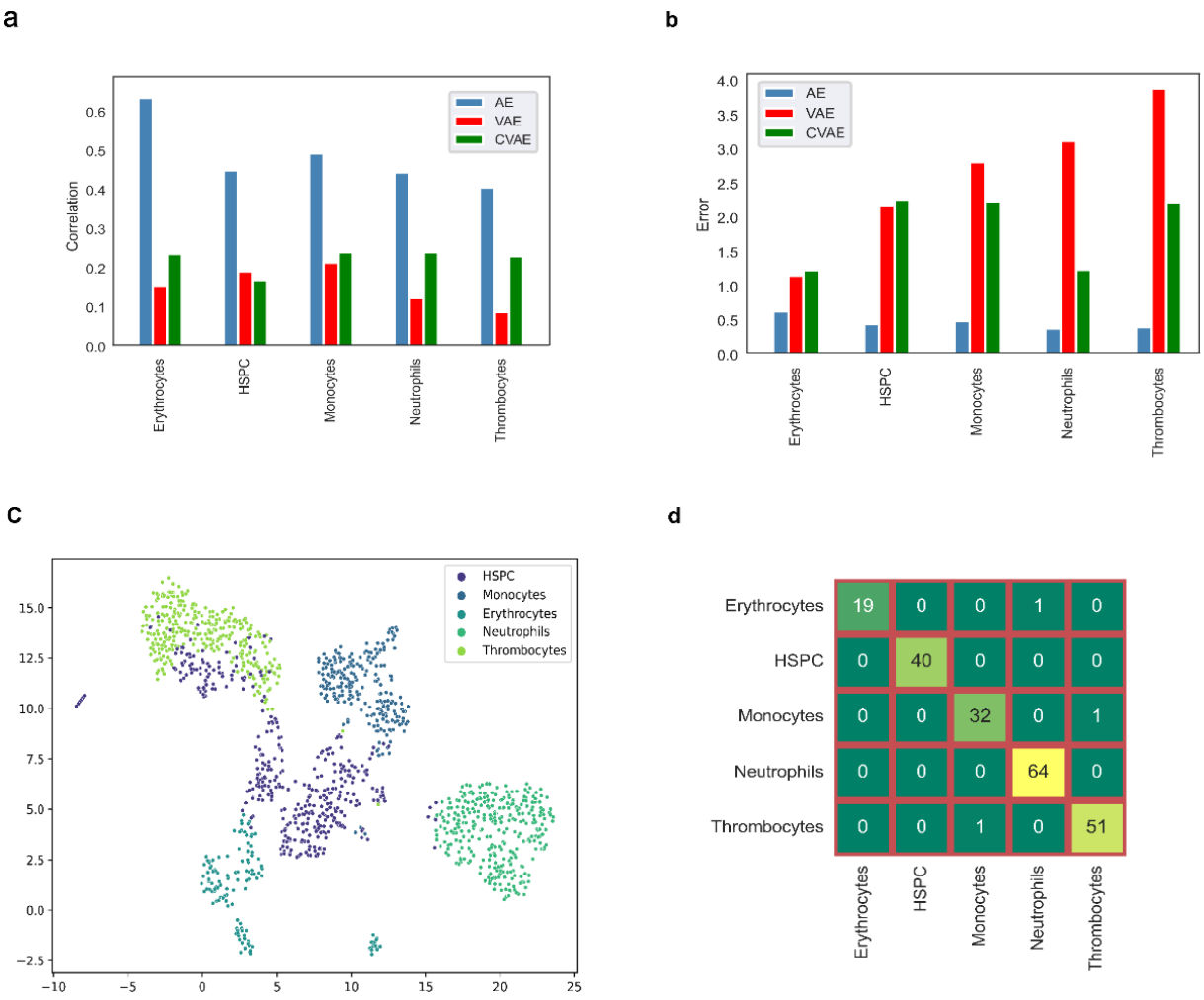


Figure 3-2: Comparison of autoencoder architectures for analyzing data for hematopoiesis regulation in zebrafish blood a, Comparison of correlation between input and output of AE, variational autoencoder (VAE), and conditional variational autoencoder (CVAE). b, Mean square error between input and output of the three autoencoder architectures showing that AE produces the lowest error rate for this data set. c, UMAP visualization of the latent layer of the under-complete autoencoder (AE). d, Confusion matrix for the classifier connected to the latent layer of AE demonstrating excellent classification performance.

3.3.3 Fatecode accurately detects known regulators from simulated scRNA-seq data

To assess the accuracy by which Fatecode identifies cell fate regulators using gene expression profiles, we applied the method to simulated single-cell RNA-seq data generated from known gene

regulatory network (GRN) structures using SERGIO (77). SERGIO allows users to specify the number of cell types and key regulators in the simulated GRN (Fig. 3a). While Fatecode is not specific to gene regulatory networks (i.e. it can identify a list of genes of any type, not just transcription factors), a GRN-based simulation is expected to provide a good benchmark for our method. A matrix of 400 cells and 2700 genes, with 20 known regulators and 9 cell types was generated and run through Fatecode. Predicted cell fate regulator genes and their prioritization scores were compared to the known SERGIO regulator list. The number of known regulator genes identified increases as more genes are prioritized (Fig. 3b). Almost all of the known regulator genes (18 out of 20) were identified when 150 genes were prioritized (out of 2700). To compare with a naive baseline, we identified cell type markers (top 20 genes) using differential gene expression (DGE) analysis on the same data using Seurat's non-parametric Wilcoxon rank-sum test (90). Fatecode identifies a greater proportion of known regulators than DGE analysis over up to 150 prioritized genes (Fig. 3b). As SERGIO is a stochastic method, we analyzed five additional simulated datasets of the same size, all of which yielded similar results (plotted as shading in Fig. 3b). We repeated this analysis on a larger dataset consisting of 2700 cells, 1,200 genes, with 65 predefined regulators, and 9 distinct cell types. We used Fatecode to identify the top 180 key genes of this data, and DGE analysis to identify the top 25 differentially expressed genes from each cell type. Also, for comparison, we included scFates, a method specifically designed for trajectory-based differential gene expression analysis. Fatecode consistently outperformed both DGE methods in detecting known regulators. We further evaluated performance by varying the top k gene threshold of DGE, and Fatecode consistently outperformed DGE across all tested thresholds, demonstrating its robustness while varying the number of genes considered (Fig. 3c, Supplementary Fig 3). Thus, Fatecode performs well at identifying known regulators in simulated single-cell RNA-seq data.

We also examined the sensitivity of our model by the size of the latent layer in the autoencoder, by training Fatecode with different latent layer sizes (n=50, 75, and 100 dimensions) using the 2700 cell simulated data (Fig. 3d). Our results show general consistency across the different latent layer sizes, indicating that Fatecode exhibits robustness across a range of latent layer sizes.

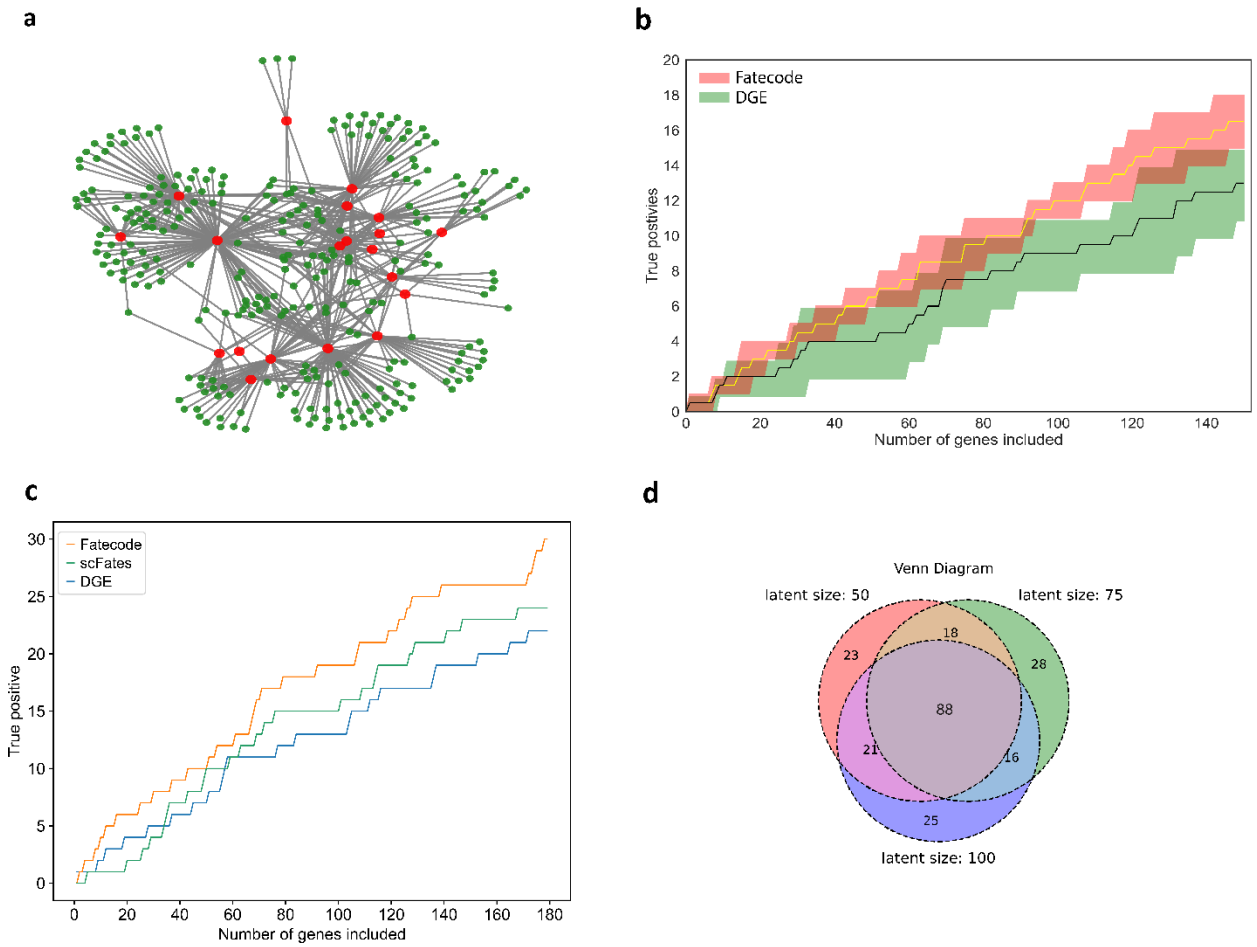


Figure 3-3 Fatecode detects known regulators using simulated data generated by SERGIO. a, The schematic structure of the gene regulatory network to generate scRNA-seq. Red nodes are known regulators and green nodes are non-regulators whose production rates are determined by their associated regulators. Our goal is to identify known regulators from the generated scRNA-seq data using Fatecode. b, Benchmark comparisons of the detection rate of predefined regulators generated by SERGIO using Fatecode compared with a naive differential gene expression (DGE) baseline. The red and green areas represent the performance of Fatecode and DGE, respectively, on the simulated data with 400 cells. c, Benchmark comparisons of the detection rate of known regulators using Fatecode, scFates, and DGE on simulated data with 2700 cells. d, Venn diagram showing the similarity between the number of known regulators uncovered by Fatecode across various latent layer sizes.

3.3.4 Fatecode identifies known cell fate regulator genes in mouse hematopoiesis

Hematopoiesis is a cell differentiation process by which the body produces mature blood cells from hematopoietic stem cells. We applied Fatecode to a published mouse hematopoiesis single-cell differentiation dataset which involves the differentiation of myeloid progenitors into 9 cell types (Fig. 4a)(80). We then examined Fatecode's accuracy in predicting cell fate regulators that lead to the desired cell type distribution by comparing the results with ground truth experimental perturbation data and known regulator genes (68,80,91,92). Fatecode learned a latent node that, when perturbed, simultaneously increases the monocyte population and decreases erythrocytes and granulocytes (Fig. 4b). Previous studies have demonstrated that *Irf8* is important in promoting the differentiation of the GM (Granulocyte-Monocyte) lineage, particularly monocytes, and functions as a key regulator in determining the fate between granulocytes and monocytes. Fatecode accurately predicted *Irf8* as an important cell fate regulator in the monocyte differentiation process. It correctly assigned a high positive score for monocytes and late_GMP (Granulocyte-Macrophage Progenitor) and negative scores for granulocytes and MEP (Megakaryocyte-Erythroid Progenitor) lineages, consistent with previous studies (Fig. 4c). Next, we investigated the prediction results for *Cebpa*, the knock-out of which leads to a decline in the population of differentiated myeloid cells, while concurrently increasing the number of erythrocytes. Fatecode accurately assigned a high positive score to *Cebpa* for monocytes and granulocytes and a negative score to erythrocytes and MEPs (Figs. 4d,e). In another example, *Klf1* is a key regulator in driving differentiation towards the ME (Megakaryocyte-Erythroid) lineage, specifically promoting the development of erythrocytes, while simultaneously inhibiting the GMP lineage. Fatecode correctly assigned a set of positive scores to *Klf1* for erythrocytes and MEP, indicating its ability to capture a key regulator in ME lineage differentiation (Supplementary Fig. 3a). We also tested Fatecode's ability to detect genes which are known to be important in maintaining stemness and inhibiting differentiation. Fatecode correctly predicted *Runx1* as a candidate that has negative scores for perturbations that increase all mature cell types (all cell types expect MEP and GMP) (Supplementary Fig. 3b). Lastly, we examined the prediction results for *Fli1*, which exhibits diverse effects on

differentiation. Fatecode accurately gives positive scores for the association between *Fli1* and megakaryocytes, monocytes, and granulocytes and also assigns a notable negative score to erythrocytes in agreement with the literature (68,93) (Supplementary Fig. 3c). These simulations show Fatecode accurately identifies known cell fate regulators that have been reported in previous perturbation-based experimental studies.

Furthermore, to evaluate the role of the top 200 genes detected by Fatecode for monocytes, we performed pathway enrichment analysis. Pathways that are significantly enriched in these 200 genes include those related to the immune system, hemopoiesis, cell development, and cell differentiation, which agrees with their Fatecode-predicted role in monocyte development (Fig. 4f).

We extended our analysis to a larger hematopoiesis single-cell differentiation data that involves differentiation into twelve cell types (Supplementary Fig. 4a) (81). We applied Fatecode to detect genes that can increase the pool of undifferentiated cells in this system (Supplementary Fig. 4b). One candidate detected by Fatecode in this process is *Entpd8*, the deletion of which in mice elevates the neutrophil and monocyte population (94). Fatecode predictions are consistent with this experimental result. Fatecode also predicted *Nlrp6* as a regulator of neutrophil and monocyte differentiation. Cai et al. showed that the number of hematopoietic stem cells and granulocyte-monocyte progenitors is reduced in Kp-infected *Nlrp6*$^{-/-}$ mice, while the survival of mature neutrophils in bone marrow is increased (95). We repeated gene set enrichment analysis using the top 200 genes detected by Fatecode. Biological processes related to mouse hematopoiesis, stem cell development, and metabolic signaling were enriched, showing that Fatecode can again capture relevant pathways for this biological process (Supplementary Fig. 4c).

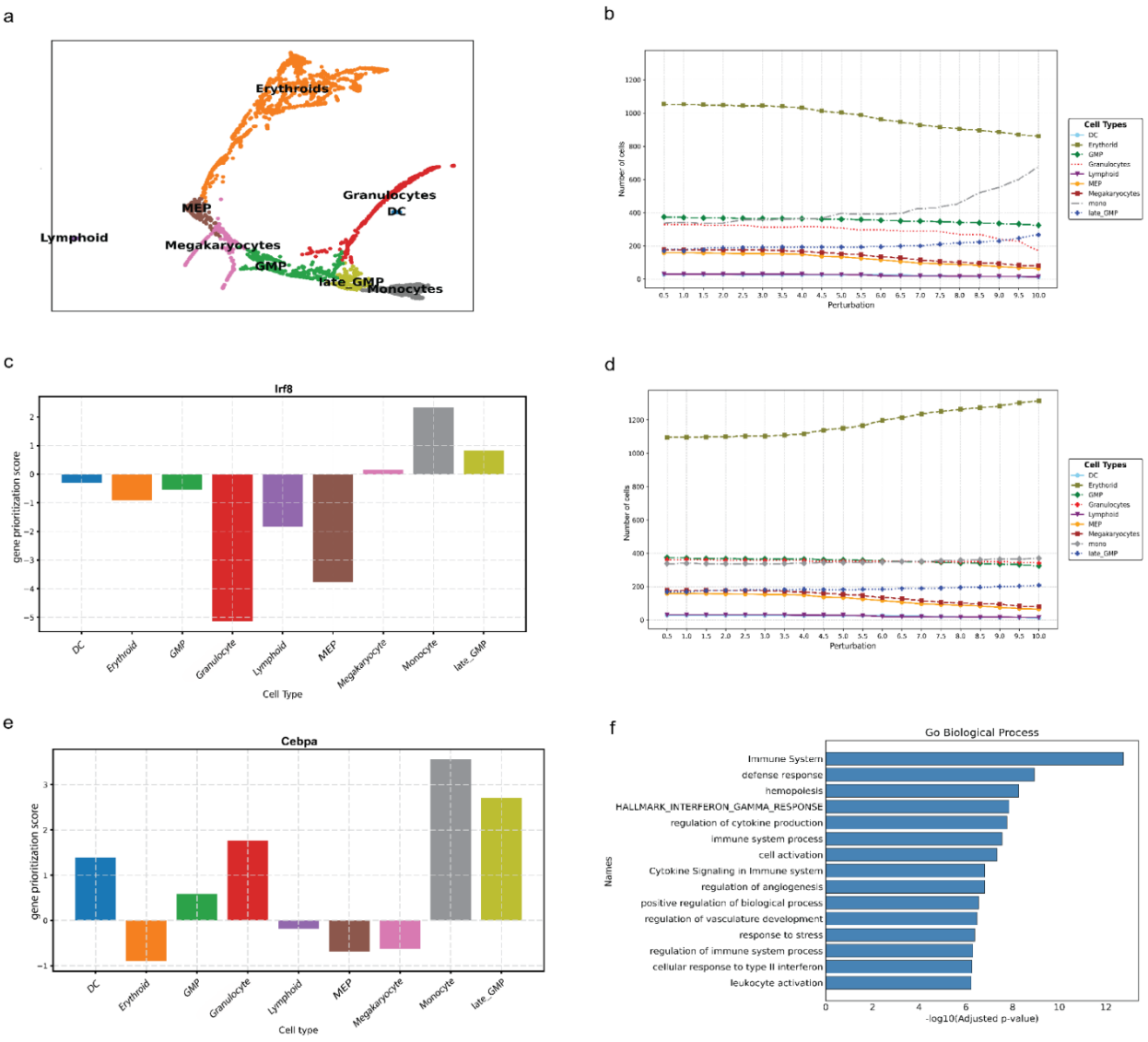


Figure 3-4: Fatecode accurately detects regulators and predicts the effect of single-cell perturbations. a, Hematopoiesis data from Paul et al. **(80)** visualized as a UMAP and clustered into 9 cell types. b,d, The results of in silico perturbations that change the initial cell frequency to the desired distribution (for b our objective is to promote monocytes while reducing the number of erythrocytes. As for d, we aim for an increase in the erythroid population and a decline in MEP and megakaryocytes). c,e, Gene prioritization scores per cell type for *Irf8* and *Cebpa*. f, Pathway enrichment analysis results. Gene ontology (GO) biological processes show significant processes related to cell development and hematopoiesis.

3.3.5 Fatecode detects important regulators in cell differentiation and lineage commitment in zebrafish

We applied Fatecode to zebrafish hematopoiesis data (78) as an additional demonstration and test. From all possible perturbations on the latent layer performed by Fatecode, we selected ones that resulted in the greatest predicted relative increase in Hematopoietic Stem and Progenitor Cells (HSPCs) (Fig. 5a). As shown in Fig. 5b, following the perturbation, some cells (mostly monocytes) are predicted to switch to HSPCs (Fig. 5b). Fatecode gives a significant score to Signal Transducer And Activator Of Transcription 5A (*stat5a*) as one of the most important genes for HSPCs. *Stat5a* is a key regulator of normal hematopoiesis with pleiotropic roles in hematopoietic stem cells (96). Also, knockout studies have shown that the deletion of *stat5a* led to an increase in HSPC cycling, gradually reduced survival, and depleted the HSPC pool (97). Next Fatecode gives *irf8* a high positive score for monocytes. Irf8 is a key regulator of monocyte development and it has been known to be important for myelopoiesis in different model organisms (98,99). It functions at an early step of the transcriptional program that governs differentiation from myeloid progenitors to monocytes/macrophages and plays a key role in stem cell renewal and maintenance (99,100). Fatecode also identified a strong negative connection between *foxo3* and myeloid cell differentiation, consistent with *foxo3* knockout studies, which show a significant increase in granulocyte/monocyte progenitors in the spleen, bone marrow, and blood and enhance short-term hematopoietic stem cell proliferation (101–103). Fatecode found an important role played by the *otud* gene family, a subgroup of deubiquitination enzymes, by assigning a high positive score between HSPCs and the *otud* gene family. Consistent with our prediction, knockout of *otud* genes in Xenopus results in developmental impairments (104). Also, elevated expression of *otud* genes leads to the acquisition of stem cell properties (105). Fatecode also predicted the negative score between *thbs1* and HSPCs, where *thbs1* has been previously shown to limit the expression of essential self-renewal transcription factors, including *oct3* and *oct4*, *sox2*, *klf4*, and *c-myc*, within cells (106). Other key gene candidates identified by Fatecode for this perturbation are also known to be involved in hematopoiesis (Table 1).

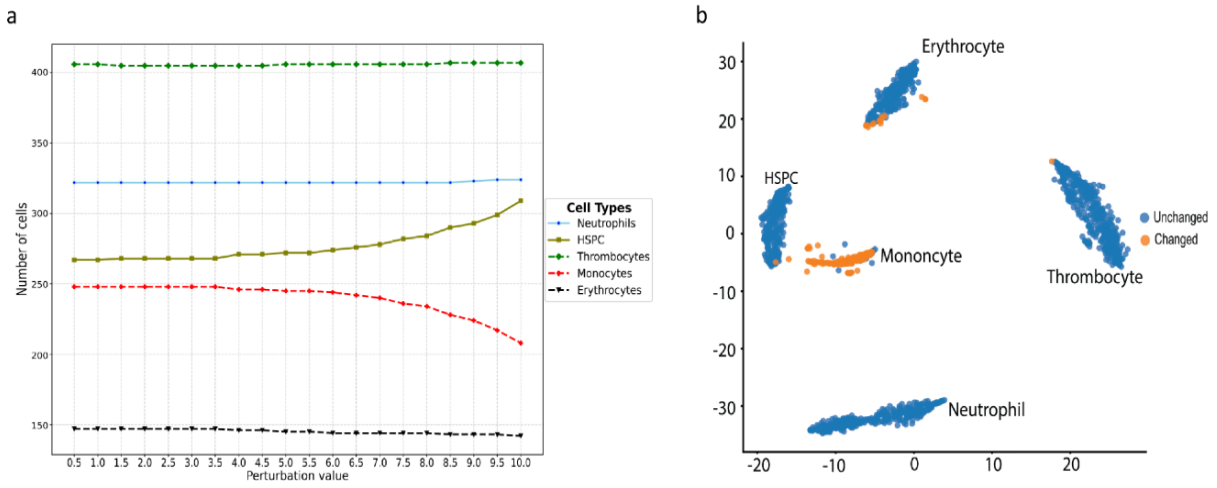


Figure 3-5: In silico experiments to induce hematopoietic stem/progenitor cells using hematopoiesis in zebrafish. a, A series of latent layer perturbations and their effect on cell distribution. b, Cells that switch from their initial cell type to HSPCs are highlighted.

Gene	**Roles**	**References**
cdk1	Plays an important role in the maintenance of pluripotency and genomic stability in human pluripotent stem cells.	(107)
top2a	Controls the survival of human pluripotent stem cells.	(108)
hmgb2	Regulates hematopoietic stem cell maintenance.	(109)
ube2c	Highly expressed in hESCs and is a biomarker of cancer stemness.	(110,111)

fbxo11	Depletion leads to the hematopoietic population with stem cell characteristics	(112)
hmgn2	Facilitates the maintenance of active chromatin states required for stem cell identity in a pluripotent stem cell model.	(113)
aspm	Regulates symmetric stem cell division by tuning Cyclin E ubiquitination.	(114)
myb	Regulates hematopoietic stem cell and myeloid progenitor cell development.	(115)
kpna2	Exhibits strong interactions with oct4 in embryonic stem cells.	(116)

Table 3-1: List of zebrafish hematopoiesis regulator genes predicted by Fatecode with literature evidence for involvement in this process.

3.3.6 Fatecode identifies cell fate regulators in mouse hippocampus development

To demonstrate Fatecode on a larger biological dataset, we applied it to developing mouse hippocampus cell scRNA-seq data (79), composed of 18,213 cells and 3,001 genes. The data is clustered in 14 annotated cell types (Fig. 6a). We first sought to identify regulators in the differentiation process that preferentially increase mature granule cells (Fig. 6b). Fatecode predicts the ZFP gene family (*Zfp94, Zfp189, Zfp706*) as positively important in granule cell differentiation. The Zfp family is a definitive marker for the cerebellar granule neuron lineage and plays a critical role in granule cell specification within the developing cerebellum (117). For example, the lack of

Zfp521 results in a significant reduction in the number of granule cells (118). *Id2* and *Id3* are important in maintaining the size and cellular structure of the brains of adult mice. It also has been shown that the absence of $Id2$ leads to a decrease in the number of granule neurons (119,120). In line with this earlier research, Fatecode assigns a high positive score between both *Id2* and *Id3* for mature granule cells. These two transcriptional regulators have also been found to determine the fate of differentiating CD8+ T cells (121).

Next, we applied Fatecode to determine regulators that mediate the differentiation process which preferentially increases oligodendrocyte progenitor cells (OPC), and decreases granulocytes (both mature and immature) and oligodendrocytes. Fatecode predicted *Igfbpl1* as having an impact on OPC to oligodendrocyte differentiation, which is consistent with published experimental studies (122,123). Furthermore, we considered *Fth1*, which provides neuroprotection and is enriched in oligodendrocytes. Mice lacking *Fth1* have more microglia cells compared to the control and a significant reduction in neurons and oligodendrocytes (124). Fatecode accurately assigned a high positive score linking *Fth1* to oligodendrocytes and mature granule cells and a negative score for *Fth1* and microglia cells showing knocking out of *Fth1* leads to an increase in microglia cells consistent with the experimental studies. Thymosin beta 4 (*Tmsb4x*) is a key candidate in the context of neurogenesis during brain development (125). Its expression is linked to neurogenic processes and exerts regulatory control over the expansion of the stem cell pool within the early neuroepithelium. The *Tmsb4x* gene knock-out elicits a pronounced effect on the differentiation process in vitro. Specifically, it significantly promotes the differentiation of stem cells, further emphasizing its role in orchestrating cellular fate determination (126). Our method correctly assigns a negative score for *Tmsb4x* and all cells except Neuroblasts and Radial Glia-like cells. To further validate the performance of Fatecode in detecting key genes, we performed pathway enrichment analysis on the top 200 Fatecode-predicted regulators. This analysis showed that pathways related to brain development, synaptic signaling, and protein synthesis were significantly enriched in these genes (Fig. 6c).

To illustrate further downstream analysis that is possible based on Fatecode results, we applied SCENIC on the mouse hippocampus development dataset to construct a GRN consisting

of the top 2000 interactions based on their SCENIC Importance Measure (IM) scores which shows the significance of the input gene (referred to as the "TF") in determining the prediction outcome for the target (127). We then mapped the top 400 Fatecode-predicted regulators to the SCENIC-inferred GRN. The resulting networks can be used as a guide for identifying specific GRN mechanisms to target in follow-up experiments (*Ybx1* example, Supplementary Fig. 5) to test the regulatory relationships and potential roles of regulators in cellular reprogramming. While SCENIC predicts useful additional information to support experiment planning, it only considers transcription factor regulators. Other types of genes in Fatecode's output can be identified as cell fate regulators and should also be examined.

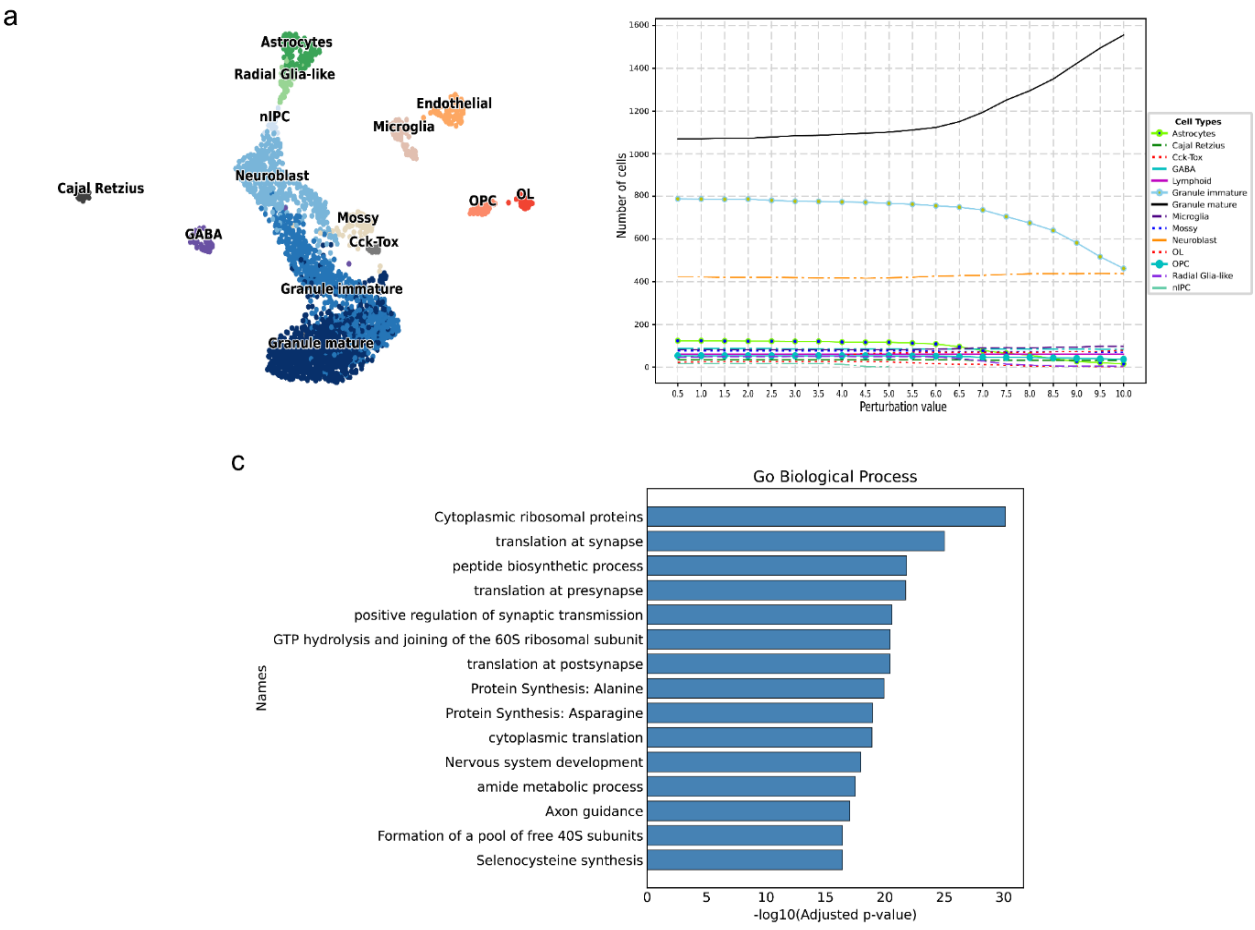


Figure 3-6 :Fatecode identifies key genes in mouse neurogenesis. a, UMAP embedding of fourteen major cell types. b, latent layer node perturbation leads to an increase in mature granule cells while

a decrease in immature granule cells. c pathway enrichment analysis shows the relevant biological process using the top 200 genes selected based on their prioritization scores for mature granule cells.

3.4 Discussion

Cell reprogramming is a promising technology for tissue repair and regeneration, with the ultimate goal of accelerating recovery from diseases or injuries, as well as the development of novel therapies (128). An important component in successful cell reprogramming is to correctly identify the regulators and trajectories from single-cell transcriptomics data. However, the number of genes in these datasets is large and the number of underlying regulatory interactions is much larger. Recent studies have demonstrated that the expression of a single regulator is insufficient to produce an end-point phenotype (129). Instead, a group of control networks acts together across a variety of biological processes and pathways to induce a complete lineage conversion (130). To efficiently and accurately map these control networks, we have developed a deep learning method, Fatecode, which we have successfully applied to analyze diverse datasets. First, our method discovers an efficient architecture and latent layer for an input single-cell dataset. Then by performing operations on the latent layer, it is able to predict perturbations for cell fate reprogramming. Fatecode was validated using simulated scRNA-seq data with predefined regulators and by predicting regulators in a variety of scRNA-seq data and manually comparing the results to the literature.

The fundamental idea in Fatecode is similar to the minimum Hamiltonian in physics and the potential energy landscape concept (131). The authors have shown that the most common autoencoders are naturally associated with an energy function, independent of the training procedure. This reasoning suggests that regulators can be seen as genes that allow the system to achieve a target cell type distribution via the most efficient path through the energy landscape. Fatecode uses the latent layer as a guide to determine what node in the latent layer must be perturbed to achieve the desired reprogramming effect. Then the decoder maps the modified latent layer to gene space for gene identification. It's also useful to understand if regulators are cell type-

specific or not. For example, the mammalian target of rapamycin complex (mTORC1) is widely important in cell fate decision-making, and also important in the regulation of T-cell fate (19,20,132,133). Running Fatecode for different cell conversions can help identify cell-type specific and non-specific regulators.

Fatecode can be thought of as an in silico CRISPR perturbation screen that identifies genes that may influence cell fate. Unfortunately, we were not able to find a published genome-wide CRISPR perturbation screen of an appropriate cell line and with a cell fate readout. Most genome-wide CRISPR-screens use standard cell lines that are not naturally multi-potent and thus are not expected to generate multiple cell fates. CRISPR has been used to evaluate cell fate regulators, but only examining one or a few candidate genes in a single paper. We used these latter small-scale results to verify that Fatecode results agree with these experiments (cited publications in text). Because we couldn't find genome-wide CRISPR screens with a cell fate readout, we used GRN simulations and small-scale CRISPR experiments to validate our findings. In the future, we hope genome-scale CRISPR screens for cell fate regulators will be published for us to compare to.

Despite offering a useful input data representation, how the autoencoder latent layer represents the input data may be difficult to understand. Future work will need to better understand how the input data is represented and learned in the latent layer given diverse input data. However, our results showed that Fatecode predictions are relatively stable when changing the size of the latent layer, indicating that latent information is likely captured consistently.

In conclusion, we developed an effective computational framework for predicting key players in cell fate control and the consequences of perturbations on cell type frequencies. Fatecode's modular design enables users to select an autoencoder architecture that produces an accurate model for their data. By leveraging the power of classification-supervised autoencoders and the associated energy manifold learning process, Fatecode generates useful hypotheses about genes that could be manipulated to achieve desired cell transitions.

3.5 Method

3.5.1 Deep representation learning

Autoencoders are a class of neural networks with a latent layer capable of learning nonlinear representations of the input data in an unsupervised manner. An autoencoder consists of an encoder that maps the input to the latent space and a decoder which transfers the latent space back to the original space. It can be used for denoising, reducing dimensionality, or learning the representation (or manifold) of the data. We implemented three autoencoder architectures: under-complete AutoEncoder (AE), Variational AutoEncoder (VAE), and Conditional VAriational Encoder (CVAE) (88) (Fig. 1). AE has a single latent layer. VAE constrains the latent layer by modeling the latent space as a multivariate Gaussian distribution with a mean and a standard deviation. CVAE conditions the latent space on class labels and thus can generate data based on a given class label. The biological task for our autoencoder is to learn a reduced dimension representation of a cell by gene matrix capturing measurements of a single-cell transcriptomics experiment mapping cellular trajectories. Only the gene dimension is reduced, so the latent space describes a reduced representation of each input cell transcriptome. To make the latent layer more specific for our biological task, we added a cell type classification task to the standard regression tasks. The classification task, described in more detail below, predicts the type of each latent cell and compares it to a known input cell type. The training process works to optimize both classification and regression performance simultaneously. This reduces the space of latent layer candidates since not all possible latent layers are useful for the classification task.

3.5.2 VAE

VAE is a type of autoencoder that estimates a latent set of probability density functions that model the input data. Unlike AE, which learns an unconstrained representation of the data, VAE assumes a Gaussian distribution for the prior. An input gene by cell matrix X is run through an encoder, which generates parameters for the set of distributions $Q(z \mid X)$. Then, from Q, a latent k-vector z

is sampled, and the decoder transforms *z* into an output, with the condition that the output is similar to the input, where k equals the number of components (or distributions) in the VAE. The VAE total loss consists of the reconstruction loss (first term) and the KL-divergence loss (second term):

$$E[\log P(X \mid z)] - D_{KL}[Q(z \mid X) \| P(z)]$$

$$D_{KL}[Q(z|X)||P(z)] = -\frac{1}{2}\sum_{k=1}^{K}[1 + log\sigma_k{}^2 - \mu_k{}^2 - \sigma_k{}^2]$$

where μ_k and σ_k are the k-th components of output vectors μ_k (X) and σ_k(X), respectively.

3.5.3 CVAE

CVAE is distinguished from VAE by its embedding of conditional information in the objective function. CVAE relies on two inputs: the features and the class labels, *c*, instead of using only the features, as is done with a VAE and AE. The CVAE architecture allows the encoder and the decoder to be conditioned by *c*. Hence, the variational lower bound objective is changed to the following form:

$$\mathrm{E}[\log P(X \mid z, c)] - D_{KL}[Q(z \mid X, c) \| P(z \mid c)],$$

3.5.4 Overall network architecture of Fatecode

The Fatecode autoencoder architecture was chosen for each of the datasets analyzed in this study using a hyperparameter search (More details in Supplementary Note 1). Encoder and decoder architectures are constrained to have the same number of outer and inner layer nodes. For the analysis of hematopoiesis regulation in zebrafish, Fatecode consists of a fully connected encoder and decoder. The encoder and decoder are both two-layer networks of 92 (outer layer) and 48 (inner layer) nodes with the LeakyReLU activation function and the latent layer has 18 nodes. For the analysis of hematopoiesis in mouse data by Weinreb et al. (81), the encoder/decoder has a 506-node outer layer and a 253-node inner layer, and the latent layer has 125 nodes. For the mouse

hematopoiesis data by Paul et al. (80) the encoder/decoder has a 100-node outer layer and a 40-node and the latent layer has 20 nodes. For the developing mouse hippocampus data, we used a two-layer encoder/decoder of 50 (outer), 26 (inner), and a latent layer of 15 nodes. Our model was built using software packages and libraries, including TensorFlow V2.10.0, scikit-learn V1.1.3, scanpy V1.9.1, numpy V1.23.4, and pandas V1.5.1. Differential gene expression analysis was performed using the Wilcoxon rank-sum test. To account for multiple testing, we applied the Benjamini–Hochberg correction to the calculated P-values obtained from the DEG analysis. Genes with a corrected p-value below 0.05 were considered statistically significant. For the identification of enriched gene ontology terms in our study, we used the GSEApy package V1.0.4 with its default parameter settings.

3.5.5 Classification

The classifier determines cell types using the latent layer as input to a single hidden layer and then an output layer (with one node per cell type), all fully connected. ReLu and softmax activation functions are used for the hidden and output layers, respectively. The number of nodes in the hidden layer is varied during the hyperparameter optimization. For adult zebrafish blood data (78), we use 15 and 5 nodes for the hidden and output layers, respectively. We use 25 and 12 nodes for classifying hematopoiesis in mouse data by Weinreb et al (81), 20 and 9 nodes for data from Paul et al (80), and 22 and 14 for the developing mouse hippocampus data (79). All cell labels are assigned by using the predefined cell type labels of the original studies.

3.5.6 Identifying key regulators in cell differentiation

Consider adjustments (e.g. one or more gene knock-outs or over-expressions) that will transition a baseline cell type distribution (“A”) to a given desired target distribution (“B”). For example, in the target cell distribution, our objective is to increase the number of cell type N while decreasing

the number of cell type P (Supplementary Figure 1). To detect genes that are important in a given transition, Fatecode analyzes the effects of perturbations on cell fate by systematically perturbing individual autoencoder latent nodes learned from a single-cell transcriptomics data set capturing cellular trajectories. Each latent variable perturbation results in a single-cell transcriptome through the decoding process and a corresponding cell type distribution, proceeding as follows after training Fatecode:

1. The gene expression data, denoted as E, corresponding to a mixture of cells with cell type distribution A, undergoes encoding to produce a matrix of latent variables represented as X ($X = encoder(E)$). Each column of X is associated with a cell in E; each row corresponds to a latent variable).

2. In a series of simulations, finite perturbations of different sizes $\boldsymbol{K}$ (e.g., from a 50% reduction to a 10-fold increase) are applied to each row j (number of latent variables) in X sequentially. For each perturbed latent layer row, ${X_j}^*$

$$ {X_j}^* = kX_j $$

3. We then run the cell type classifier trained within Fatecode on the perturbed latent layer to predict the cell type distribution for each across all perturbation conditions.

$$ New_{cell\ type\ distribution} = classifier({X_j}^*) $$

4. Then, we can identify a perturbed latent layer row, ${X_j}^*$, and its associated perturbation size, k, that is closest to the desired target distribution B.

5. To identify genes important for the transition from cell type distributions A to B, we compute the difference between the selected X^* and the X latent layers. For instance, if increasing latent node #9 5-fold can best approximate the desired distribution B, then the difference between the selected X^* and X latent layers is a latent node by cell matrix with all zero entries, except for the 9th row, which is 5 times X_9.

6. With this selected perturbation matrix $(X^* - X)$, the decoder produces a gene-by-cell matrix. Then the average gene expression profile of all cells in each cell type is computed, resulting in a gene by cell_type matrix *M*. The (*i*,*j*)-th entry of *M* is the prioritization score for the *i*-th gene in cell_type *j*.

7. To identify the regulators predicted to be important for transitioning initial cell type distribution A to target B, we rank the genes based on their prioritization scores for a cell type of interest.

$$Regulators = sort\left(M_{desired\ cell\ type}\right)$$

We note that *M* does not directly specify how much each gene should be perturbed to yield target B. Nonetheless, *M* contains information about genes that are important in transitioning cell type distribution from initial state A to the desired state B. This idea is similar to the minimum Hamiltonian and potential energy in physics or the optimal path with the least action (131).

We also examined the model's performance in detecting regulators when operating on the output of the decoder compared to the latent layer. To achieve this, we fed the perturbed vector to the decoder and subtracted the result from the unperturbed condition. We then investigated the genes that showed significant changes. Our results indicate that working on the latent layer led to better outcomes in detecting regulators than operating on the output of the decoder. This observation is in line with previous research in computer vision and natural language processing, where using the latent space consistently yielded superior results compared to the original data space (83,134). We assume this is true in general when using an autoencoder with a non-linear activation function with reasonably complex data, as we have in biology (in contrast to the linear activation function case where $Decoder\left(X_{perturbed}\right) - Decoder(X) = Decoder\left(X_{perturbed} - X\right)$).

3.5.7 Data visualization

Python package “UMAP” was used to visualize the latent layer as a reduced dimensionality space. Network visualizations used Cytoscape (135).

3.5.8 Differential expression analysis

Differential gene expression analysis was performed using the Wilcoxon rank-sum test. To account for multiple testing, we applied the Benjamini–Hochberg correction to the calculated P-values obtained from the DEG analysis. Genes with a corrected p-value below 0.05 were considered statistically significant. For scFates we used the default parameters. For the identification of enriched gene ontology terms in our study, we used the GSEApy package V1.0.4 with its default parameter settings.

3.5.9 Data preprocessing

The scRNA-Seq gene expression data is log normalized, scaled, and centered. In the training process, 80% of the data is allocated for training the classification autoencoder, while the remaining 20% is utilized for testing purposes.

3.5.10 Data availability

The datasets used in the present study are openly accessible in public repositories. The zebrafish hematopoiesis data can be found under the accession number E-MTAB-5530 on ArrayExpress. We downloaded a preprocessed version of the “Dentate Gyrus neurogenesis” data (under accession number GSE95753) from https://scvelo.readthedocs.io/en/stable/. The hematopoiesis Paul et al. data can be downloaded from the GEO under accession code GSE72859 and the preprocessed version was downloaded from https://celloracle.org/. To generate simulated data we used the same parameters for the differential equations as in https://github.com/PayamDiba/SERGIO (77). The

hematopoiesis Weinreb et al. data can be downloaded from GEO under accession number GSE140802 and the preprocessed version was downloaded from https://cospar.readthedocs.io/en/latest/.

3.5.11 Code availability

Code supporting this study is available on: https://github.com/MehrshadSD/Fatecode

Chapter 4
Discovering Governing Equations of Biological Systems through Representation Learning and Sparse Model Discovery

This chapter presents CLERA, a novel computational framework for uncovering dynamic models and identifying active gene programs from single-cell RNA sequencing data. By combining a supervised autoencoder with Sparse Identification of Nonlinear Dynamics, CLERA uses prior knowledge to extract low-dimensional representations and reveal the driving forces behind cellular processes. It pinpoints central genes, reconstructs gene expression dynamics, and captures key regulatory genes and temporal patterns across various cell types using personalized page rank. The research titled *Discovering Governing Equations of Biological Systems through Representation Learning and Sparse Model Discovery* was conducted in collaboration with Vasu Swaroop (an undergrad student in our lab at the time of the study), and is currently under review in *Nucleic Acids Research* journal (2). The author list on the publication is below:

Mehrshad Sadria[1], Vasu Swaroop[2]

[1] Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

[2] Department of Computer Science Information Systems, BITS-Pilani, Pilani Campus, Pilani, 333031, India

4.1 Overview

Understanding the governing rules of complex biological systems remains a significant challenge due to the nonlinear, high-dimensional nature of biological data. In this study, we present CLERA, a novel end-to-end computational framework designed to uncover parsimonious dynamical models and identify active gene programs from single-cell RNA sequencing data. By integrating a supervised autoencoder architecture with Sparse Identification of Nonlinear Dynamics, CLERA

leverages prior knowledge to simultaneously extract related low-dimensional representation and uncovers the underlying dynamical systems that drive the processes. Through the analysis of both synthetic and biological data, CLERA demonstrates robust performance in reconstructing gene expression dynamics, identifying key regulatory genes, and capturing temporal patterns across distinct cell types. CLERA's ability to generate dynamic interaction networks, combined with network rewiring using Personalized PageRank to highlight central genes and active gene programs, offers new insights into the complex regulatory mechanisms underlying cellular processes.

4.2 Introduction

Across many scientific disciplines, discovering governing equations has traditionally served as the cornerstone of understanding systems (46). Derived from mathematical and physical laws, these equations provide interpretable and generalizable frameworks for explaining and predicting various phenomena. In areas such as biology (20), epidemiology (18), and finance (136), mathematical models are used to model signalling pathways, population dynamics, and disease spread, and market fluctuations, respectively. However, for complex systems with high dimensionality and nonlinearity, including biological processes, traditional approaches often fall short (137). Discovering the main equations governing these systems can be challenging, and even when partial knowledge exists, relying solely on first principles becomes impractical (138).

The modern era, with its abundance of data and computational power, has facilitated the emergence of data-driven model discovery as a powerful paradigm in scientific exploration (41). This approach directly leverages data to uncover the hidden principles that govern complex systems. In the context of cellular biology, single-cell RNA sequencing (scRNA-seq) provides an unprecedented window into individual cells, which offers insights into gene expression variation across diverse cellular populations (139). By analyzing this data, researchers can investigate the molecular machinery underlying development, disease, and response to external perturbation (4,24). The noisy, nonlinear, and high-dimensional characteristics of scRNA-seq data and the

biological processes it captures pose significant challenges for analysis and interpretation (140). These complexities make it difficult to uncover the underlying principles of biological processes and pinpoint their key drivers (141). While previous methods have achieved success in specific tasks, limitations remain. For instance, the correlative nature of most methods prevents them from capturing causal features and true representations, thus limiting their generalizability (142). Furthermore, these models struggle to discover governing relationships among underlying variables in a parsimonious manner, similar to classical physics settings, which further hinders true interpretability. Therefore, a crucial step in understanding any biological process lies in developing models that not only can accurately predict but also reveal the underlying connections between features in an interpretable and parsimonious manner (42). This ensures the models can be applied across diverse environments and provides clearer insights into the mechanisms governing the process (46).

In the realm of high-dimensional biological data like scRNA-seq, the ability to capture causal representations of the data becomes particularly valuable. This approach goes beyond identifying correlations and allows us to understand the true relationships between variables (143). In this context, identifiability, the ability to uniquely recover the underlying causal structure from observed data, becomes a crucial aspect of representation learning. Traditional methods like Independent Component Analysis (ICA) have achieved success in many areas of linear representation learning (40). In fact, if all latent components are non-Gaussian and independent, ICA can be identifiable. However, ICA struggles with the inherent nonlinearities and complex interactions present in biological data (40). While perfect identifiability, especially in non-linear settings, remains a challenge, incorporating temporal structure, employing additional tasks, or using auxiliary information can facilitate the way to attain identifiability (144,145). Notably, autoencoders can offer a promising avenue for achieving identifiability in non-linear settings (146). By carefully designing their architecture and loss function, autoencoders can help extract meaningful representations from complex biological data (1,5,66,147).

In this work, we present CLERA (Cellular Latent Equation and Representation Analysis), a novel end-to-end computational framework that combines the power of data-driven model discovery, specifically Sparse Identification of Nonlinear Dynamics (SINDy), and representation learning. Leveraging a supervised autoencoder architecture, CLERA simultaneously extracts a compact and relevant representation from high-dimensional data and uses it to discover the underlying low-dimensional, non-linear dynamical model governing the system. This learned embedding further allows us to not only identify active gene programs and key genes but also track their transitions over time across cell types, providing insights into the complex dynamic regulatory mechanisms of biological systems. We validate CLERA's performance on both simulated data (with known active gene programs) with different sizes and real-world biological datasets.

4.3 Result

4.3.1 Discovery of Dynamical Systems and Gene Programs from Simulated Data

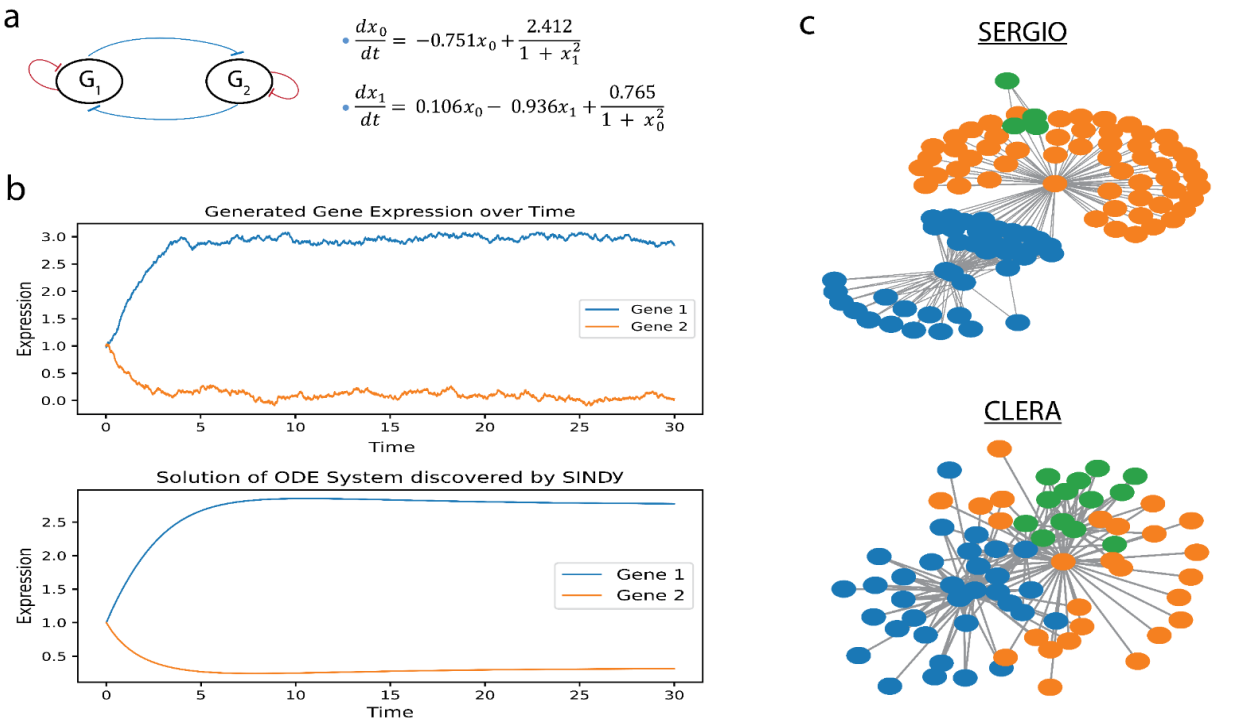
We first investigate the performance of the SINDy part of CLERA in discovering the underlying governing equations of a simple simulated biological system with two driver genes (Fig. 1a). The dynamics of this system are described by a well-established set of differential equations commonly used in various biological contexts such as the lac operon, metabolic signalling pathways, and the cell cycle (148). Synthetic data is generated using this system of equations with varying noise levels. We then apply SINDy, to recover the equations. Notably, the governing equations are discovered with high accuracy. Figure 1a shows this successful reconstruction, with the recovered parameters closely mirroring the original values (Method section). Also, the results from solving the discovered differential equation closely match the generated data, further validating the accuracy of the equations discovered (Fig. 1b).

Then to evaluate the performance of CLERA in a more realistic scenario with a larger dataset, we apply it to simulated data generated by SERGIO, which incorporates various types of noise for realistic data generation (77). We train CLERA on a simulated dataset, with 6300 cells

across 100 genes and nine distinct cell types. We hypothesize that an optimal representation learned by the model should not only achieve high accuracy in data reconstruction (autoencoder loss) but also should perform well in tasks such as cell type classification and sparse dynamical model discovery.

To address stochasticity and ensure robustness in finding the optimal latent embedding, we run CLERA multiple times (50 for this data) using various initial conditions. We select the model with the lowest combined loss (method section) while also prioritizing parsimony in the discovered model. In our analysis, we observe that CLERA successfully identified a latent embedding with high accuracy in both reconstruction and classification (Supplementary Figure 1, Supplementary Figure 5a).

We then leverage the representation learned by CLERA to identify active gene programs and their dynamics over time. To uncover the connection between latent nodes and genes we compute SHAP (149) values between each node in the autoencoder's latent layer and genes and rank the identified genes based on their SHAP values (choosing the top 30 genes for each node). Using the results from the SHAP method and the discovered differential equations, we construct a network of interactions between latent nodes and genes. We then apply Personalized PageRank (PPR) to this network, starting from each gene, to identify the most relevant genes for the selected gene (150). This approach enables us to refine the network by selecting only the top connected genes with the highest PPR scores while filtering out the latent nodes. A clustering algorithm is applied to this graph to detect the gene programs. Given that CLERA can uniquely incorporate a time component, this process can be done for different stages of the trajectory and cell types. To assess CLERA's performance in capturing active gene programs, we perform the same clustering analysis on the SERGIO ground truth network and compare the resulting clusters obtained (Fig. 1c for celltype7). We also observed a high degree of similarity between the gene programs of the SERGIO predefined network and the identified gene interaction networks for each cell type, as measured by the Jaccard similarity. This suggests that the latent embedding learned by CLERA can effectively capture active gene programs (Fig. 1d).


a
G1
G2
$\frac{dx_0}{dt} = -0.751x_0 + \frac{2.412}{1 + x_1^2}$
$\frac{dx_1}{dt} = 0.106x_0 - 0.936x_1 + \frac{0.765}{1 + x_0^2}$
b
Generated Gene Expression over Time
Expression
Time
Gene 1
Gene 2
Solution of ODE System discovered by SINDY
Expression
Time
Gene 1
Gene 2
c
SERGIO
CLERA

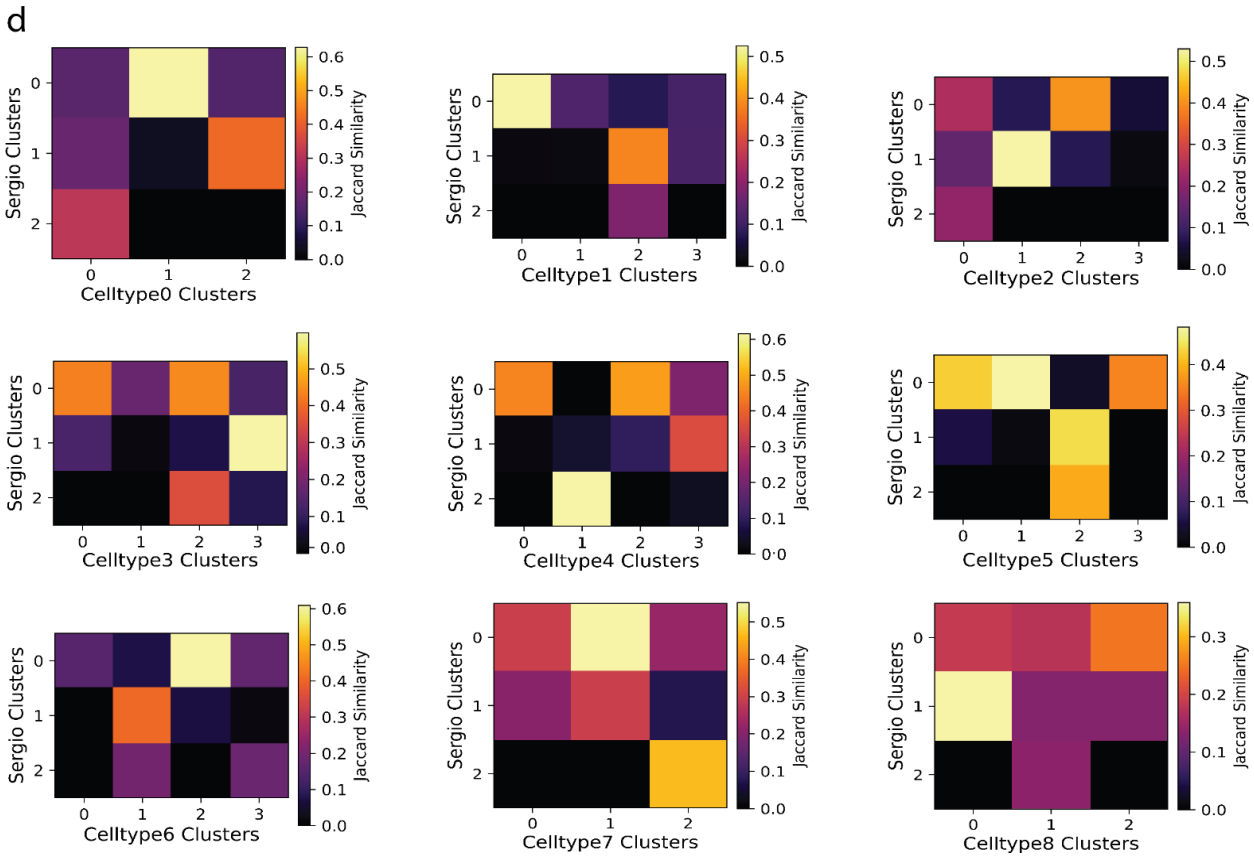

d
Sergio Clusters
Celltype0 Clusters
Celltype1 Clusters
Celltype2 Clusters
Celltype3 Clusters
Celltype4 Clusters
Celltype5 Clusters
Celltype6 Clusters
Celltype7 Clusters
Celltype8 Clusters
Jaccard Similarity

Figure 4-1. CLERA discovers dynamical systems and gene programs from simulated data. a, Schematic of a two-gene regulatory network (G_1 and G_2) with discovered governing equations and parameters shown. b, Comparison of generated gene expression data (top) and solutions from equations discovered by SINDy (bottom) for the two-gene system over time. Gene 1 and Gene 2 expression levels are plotted against time. c, Gene interaction networks for cell type 7 derived from SERGIO ground truth (top) and CLERA (bottom). Nodes represent genes, coloured by gene programs identified through clustering. d, Heatmaps showing Jaccard similarity between SERGIO and CLERA-derived gene program clusters across nine cell types (CellType0 to CellType8). Colour intensity indicates the degree of similarity, with lighter colours representing higher similarity and darker colours lower similarity.

4.3.2 CLERA Uncovers Dynamics and Gene Programs in Pancreatic Development

We further evaluate CLERA on biological scRNA-seq data from mouse pancreas during embryonic development. This dataset comprises 3696 cells, with 27998 genes clustered into eight distinct cell types (89). Following hyperparameter optimization and preprocessing, we trained CLERA several times with varying initializations (Methods), using the gene expression and computed pseudotime (cell ordering) information. CLERA successfully identifies a set of sparse and interpretable differential equations with all individual loss terms in our total loss function decreasing (Fig 2a, Supplementary Figure 2). Also, we observe the temporal dynamic of different latent variables captures distinct patterns for each cell type (Fig. 2b, Supplementary Figure 9a). CLERA also achieves a high classification accuracy using the latent variables where certain latent variables emerge as dominant predictors for individual cell types (Supplementary Figure 5b, Supplementary Figure 7).

To explore the connection between latent nodes and genes, we calculate the SHAP values (151) for each gene-latent node pair and identify the top "K" genes (K=300 for this data) connected to each latent node, ranked by their absolute values. Using the discovered equations (latent node-latent node interaction) and SHAP values (gene-latent node interaction), we generate a series of interaction graphs for various stages of pancreas development (Fig. 2c). Unlike traditional network inference methods, which only produce a static graph for the whole process, our approach captures dynamic graphs over time. Next, we apply a clustering algorithm to the interaction graphs to

identify groups of interconnected and potentially co-regulated genes. These graphs, representing different stages of pancreatic development, allowed us to observe changes in gene interactions over time. To understand the similarity of active gene programs across different cell types, we analyze the clustering results for a specific cell type and transfer the identified gene colours to the analysis of other cell types (Fig. 2d). We observe a high degree of similarity in shared genes for cluster 1 among Ductal, Ngn3 low EP, and Ngn3 high EP cell types. Analyzing these shared genes reveals several previously known key genes, such as Sox9, Neurog3, Hes1, Foxa3, and Nfib, as well as important signalling pathways like Wnt, Notch, and TGF-β (152). Furthermore, Gene Set Enrichment Analysis reveals several pathways related to pancreatic development, demonstrating the biological relevance of the gene programs discovered by CLERA (Fig. 2e). Interestingly, CLERA also captures pathways involved in neurogenesis and neural development, which aligns with previous studies and highlights the molecular and cellular similarities between pancreatic and neural cell differentiation (89,153).

To identify key and central genes for each cell type, we restructure the interaction network using the PPR technique to remove latent nodes. This network rewiring allows us to focus directly on gene interactions. We then apply centrality measures to the restructured network to identify the most influential genes for each cell type (Fig. 2f, Supplementary Figure 10a). As a result of the centrality analysis, several key genes are identified, including Spp1 (154), a regulator of the epithelial-mesenchymal transitory axis and duct cell de-differentiation; Chgb (155), a neuroendocrine cell marker; and Neurog3 (156,157), crucial for endocrine cell differentiation. The analysis also confirms the central roles of Ins1 and Ins2 in beta cell function, along with Clu (158) and Sox9 (159), both critical for progenitor cell maintenance and differentiation. CLERA correctly captures these key genes, aligning with prior studies that emphasize their importance in pancreatic development.

a

$$\frac{dz_0}{dt} = 0.22\ exp(z_5) - 0.12(z_3^3) \qquad \frac{dz_1}{dt} = 0.31\ exp(z_5) - 0.31(z_3^3) - 0.27(z_3 z_5)$$

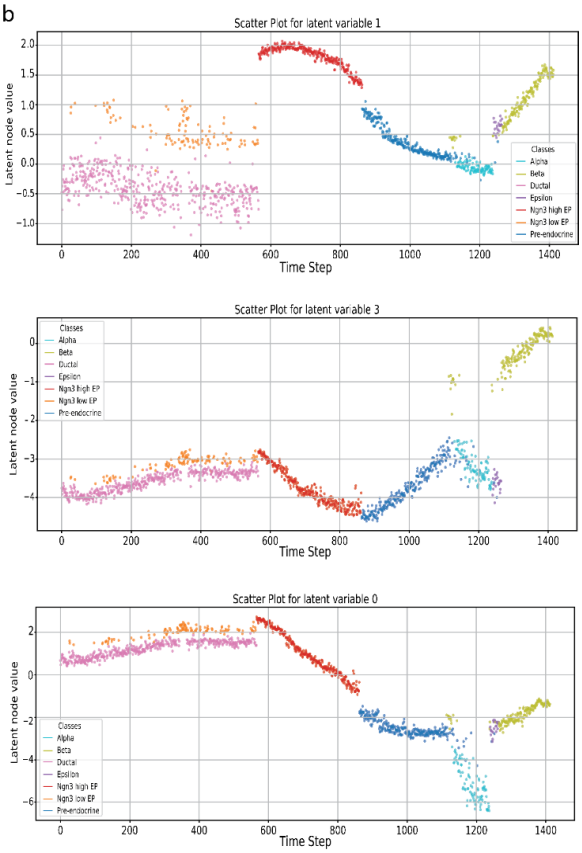
$$\frac{dz_2}{dt} = 0.32(z_4) - 0.52\ exp(z_5) + 0.27(z_3^3) + 0.27(z_3 z_4) + 0.33(z_4 z_5)$$

$$\frac{dz_3}{dt} = 0.18(z_3) + 0.12(z_4) + 0.45\ exp(z_4) - 0.18(z_0^2) - 0.14(z_0 z_2) + 0.08(z_1 z_2) + 0.15(z_3^3) + 0.15(z_3 z_4) + 0.84(z_4 z_5)$$

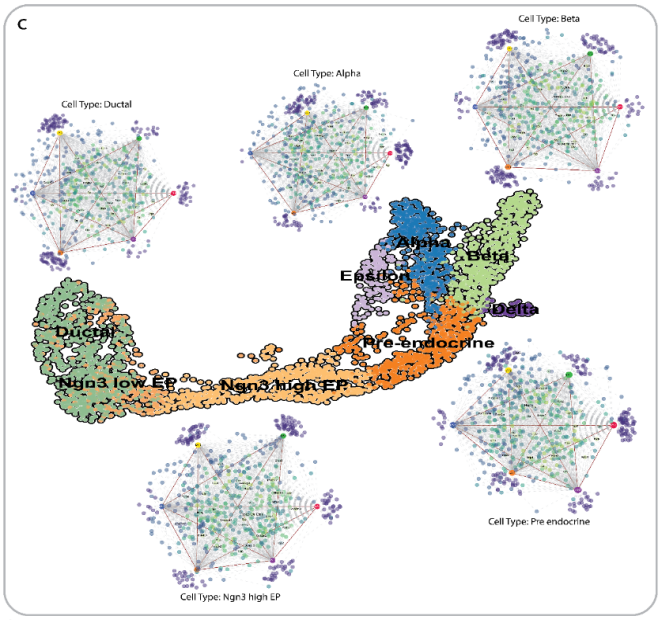
$$\frac{dz_4}{dt} = -0.37(z_3) - 0.34\ exp(z_1) - 0.11\ exp(z_2) - 0.39(z_0 z_4) - 0.45(z_0 z_5) - 0.23(z_1 z_2) + 0.52(z_3 z_4) + 0.43(z_3 z_5) - 0.45(z_4^2) - 0.2(z_4 z_5) + 0.43(z_5^2)$$

$$\frac{dz_5}{dt} = 0.21\ exp(z_5) - 0.11(z_3^3)$$

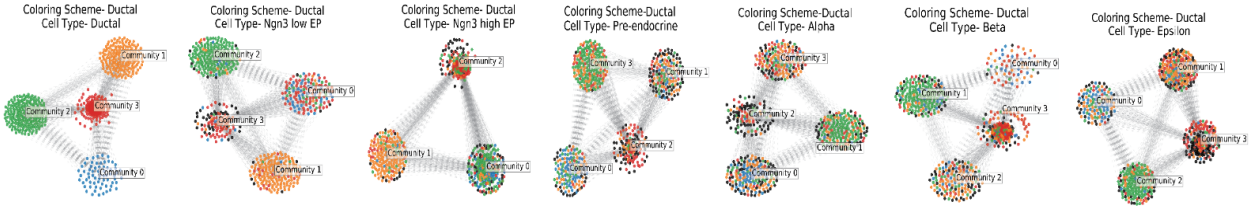
b



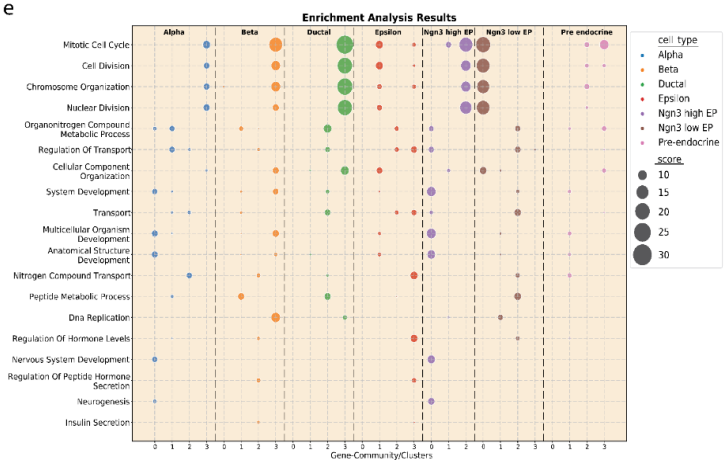
c



d



e



f

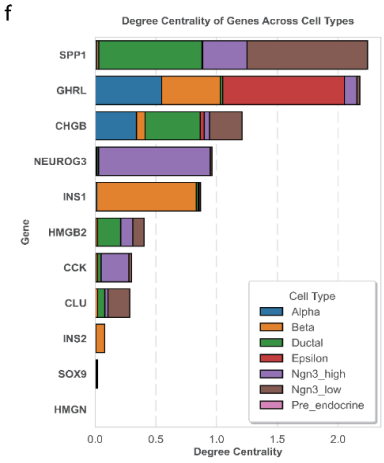


Figure 4-2: CLERA uncovers dynamics and gene programs in pancreatic development. a, Discovered differential equations governing mouse pancreas development data from scRNA-seq, showing sparse and interpretable models and connections between latent variables. b, Temporal dynamics of latent variables, which illustrate distinct patterns across cell types. c, Interaction graphs for various stages of pancreatic development, show dynamic gene interactions over time for different cell types. d, Clustering results showing gene program similarities across cell types, with shared genes in cluster 1 among Ductal, Ngn3 low EP (endocrine progenitor), and Ngn3 high EP cell types. e, Gene Set Enrichment Analysis results indicating pathways related to pancreatic development and neurogenesis. f, Degree centrality analysis identifying key genes for each cell type, including Spp1, Chgb, Neurog3, Ins1, Ins2, Clu, and Sox9.

4.3.3 CLERA Reveals Central Genes and Dynamics in Hematopoietic Differentiation

Next, we investigate bone marrow development, examining the differentiation of hematopoietic stem cells (HSPCs) into erythroids, monocytes, and dendritic cells (DCs). This dataset comprises 5780 cells and 14319 genes clustered into 10 distinct cell types (73).

To enhance CLERA's performance on this data, we apply transfer learning from our previous pancreas study (method section). By initializing CLERA with pre-trained weights, we leverage the knowledge and relationships obtained from the previous part, which results in a faster optimization and more accurate representation of the data. Also, we observe a decrease across all components of the loss function, showing that all loss terms were effectively optimized and also parsimonious discovered equations (Fig. 3a, Supplementary Figure 3). Furthermore, investigating the latent space shows that the temporal dynamics of different latent variables capture distinct patterns for each cell type, which shows that the embedding learnt by CLERA can identify and characterize unique behavioural signatures for each cell type (Fig. 3b, Supplementary Figure 9b). We observe that the latent embedding discovered by CLERA achieves high classification accuracy and also shows distinctive cell type-level differentiation, where specific latent variables drive the classification of particular cell types (Supplementary Figure 5c, Supplementary Figure 8).

Then, by identifying top genes using the SHAP method and leveraging the discovered equations, we generate a series of graphs representing different stages of bone marrow development (Fig. 3c). Through clustering analysis on these graphs, we identify groups of co-

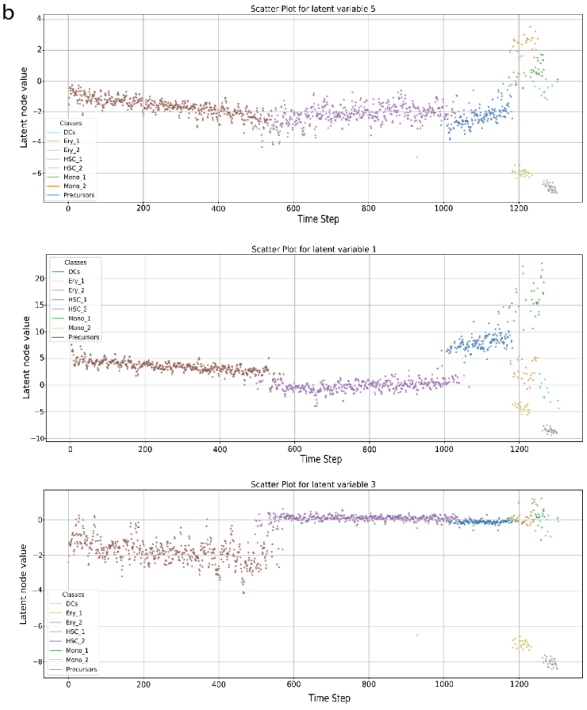
regulated genes at each developmental stage (Fig. 3d). Moreover, using label transfer techniques, we identify a significant similarity in co-regulated genes between precursors (cluster 1), monocytes (cluster 3 in Mono_1 and cluster 2 in Mono_2) and DCs (cluster 0). Some of the key genes discovered have been shown to be crucial for monocyte development, including ID2, TYROBP, FLT3, PDE4B, and GLIPR1(160–162). We also observe a strong similarity between the two erythroid subpopulations Ery_1 and Ery_2, particularly between clusters 1 and 3, and between clusters 2 and 1. Similarly, the monocyte subpopulations Mono_1 and Mono_2 show considerable overlap, with cluster 3 in Mono_1 closely aligning with cluster 1 in Mono_2, cluster 2 in Mono_1 resembling cluster 0 in Mono_2, and cluster 0 in Mono_1 closely matching cluster 2 in Mono_2. This suggests that these subpopulations have many common genes, which shows similarities in their developmental pathways and active gene programs.

To identify the critical genes within these networks, we apply PPR for network rewiring, which allows us to remove latent nodes and focus on direct gene interactions. Centrality measures then pinpoint key genes driving cellular differentiation during hematopoiesis (Fig. 3e, Supplementary Figure 10b). MPO (163), crucial for neutrophil differentiation, is identified as a key myeloid marker, while HOPX (163) emerges as a regulator of primitive hematopoiesis, guiding early progenitor cell fate. Malat1 (164), known for regulating gene expression in hematopoietic stem cells, and FOS (165,166), linked to cell proliferation and differentiation under cytokine signalling, are also highlighted. Also, CD52 (167), a marker of mature lymphocytes, FAm30A, which has shown links to immune response regulation and other hematopoietic lineages, and CD74 (168), essential for antigen presentation in immune cells, are captured. CLERA effectively identified these genes, which align with their known roles in hematopoiesis.

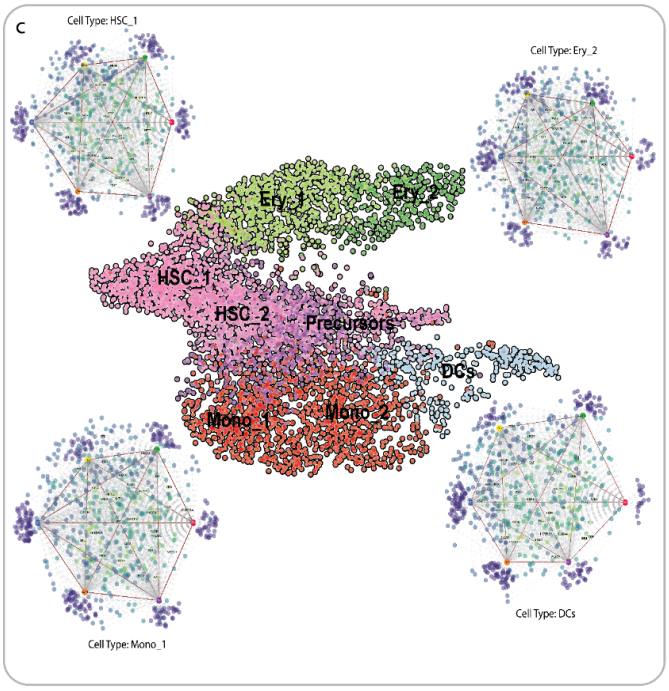
a

- $\frac{dz_0}{dt} = 0.29(z_0 z_5) - 0.10(z_3 z_5) - 0.22(z_5^2)$
- $\frac{dz_1}{dt} = 0.13(z_0 z_2) - 0.11(z_0 z_3) - 0.36(z_0 z_5) - 0.15(z_1 z_3) + 0.17(z_1 z_5) - 0.18(z_3^2) + 0.20(z_3 z_4) + 0.18(z_3 z_5) + 0.37(z_5^2)$
- $\frac{dz_2}{dt} = 0.12(z_0^2) + 0.29(z_0 z_5) + 0.10(z_2 z_3) - 0.27(z_3^2) + 0.13(z_3 z_4) - 0.27(z_5^2)$
- $\frac{dz_3}{dt} = -0.35(z_0^2) + 0.34(z_0 z_1) + 0.42(z_0 z_5) - 0.20(z_1 z_3) - 0.20(z_1 z_5) + 0.12(z_3^2) - 0.15(z_5^2)$
- $\frac{dz_4}{dt} = -0.11(z_3^2) + 0.13(z_3 z_5)$
- $\frac{dz_5}{dt} = -0.23(z_0^2) + 0.26(z_0 z_5)$

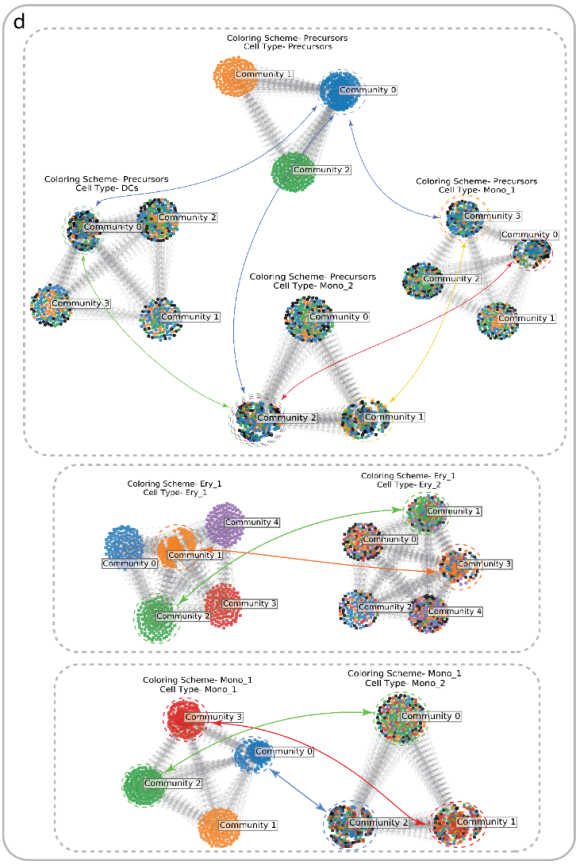
b



c



d



e

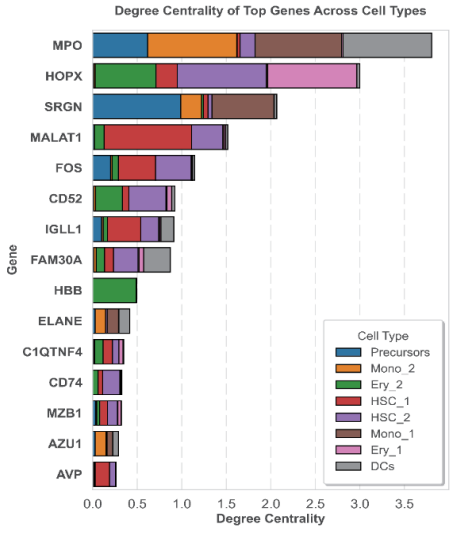


Figure 4-3. CLERA reveals central genes and dynamics in hematopoietic differentiation. a, Differential equations discovered for bone marrow development data, showing connections between latent variables. b, Temporal dynamics of latent variables, with distinct patterns across cell types. c, Interaction graphs for different stages of bone marrow development, which capture dynamic gene interactions for each cell type. d, Clustering analysis of co-regulated gene groups at each developmental stage, with significant similarities between precursors, monocytes, DCs, and among Ery-1 and Ery-2, Mono-1 and Mono-2 subpopulations. e, Degree centrality analysis identifies key genes driving cellular differentiation, including Mpo, HOPX, Malat1, FOS, CD52, FAM30A, and CD74.

4.4 Discussion

Here we introduce a novel method, CLERA which represents a significant advancement in our ability to uncover the underlying principles governing complex biological systems. CLERA integrates data-driven model discovery and representation learning, to provide a robust way for uncovering interpretable and parsimonious dynamical models from high-dimensional scRNA-seq data. The ability to simultaneously learn coordinates and governing equations is perhaps the most complementary approach to traditional physics modelling. While classical methods rely on predefined equations based on known physical laws, CLERA also allows the data to guide the discovery of both the relevant coordinates (latent representation) and the equations that govern their dynamics. This approach is particularly valuable in biological systems where the underlying mechanisms are often complex and not fully understood.

Our results demonstrate CLERA's ability to accurately recover governing equations from both simulated and real-world biological data. In the simple simulated biological system, CLERA precisely predicts the governing equations, even under varying noise levels. This performance is further validated with a larger simulated dataset generated by SERGIO, which incorporates realistic noise patterns, showing CLERA's accuracy in modelling complex, high-dimensional data. When applied to real-world datasets from the pancreas and bone marrow development, CLERA leads to biologically relevant insights. The model effectively identifies key genes through network rewiring and centrality measures, as well as detecting active gene programs involved in these processes aligned with previous experimental studies. Another key strength of CLERA lies in its ability to capture the temporal dynamics of gene programs. Unlike traditional network inference

methods that produce static graphs, CLERA generates dynamic interaction networks that evolve over time across cell types. This temporal aspect is crucial for understanding differences in complex regulatory mechanisms of cellular processes such as differentiation and development. The identification of potentially novel key genes and gene programs, along with their temporal progression, can present new opportunities for further experimental exploration of dynamic cellular processes.

CLERA integrates multiple components in its loss function to address the challenges of unsupervised learning of identifiable nonlinear representations. The combination of reconstruction loss, classification loss, and SINDy losses introduces inductive biases that guide the model toward learning more meaningful and interpretable embeddings. This comprehensive loss function imposes constraints on latent dynamics, to ensure consistency with known biological principles while maintaining the flexibility to discover new relationships from the data. By balancing these different loss components, CLERA achieves a robust and parsimonious framework for uncovering interpretable dynamical models. Also, the use of transfer learning from the pancreas to the bone marrow dataset demonstrates the model's adaptability and potential for generalization across different biological contexts. This approach not only improves performance but also suggests that certain learned representations may be conserved across different developmental processes.

In CLERA, we intentionally avoid assuming independence between latent variables, diverging from some recent methods (169). In biological systems, the strict assumption of independence can be problematic, as many genes participate in multiple programs simultaneously. This interconnectedness makes it challenging to capture the true nature of these systems with models that enforce strict independence. By allowing for overlapping gene programs and complex interactions, CLERA's approach is better suited to reflect the complexity and reality we have in biology. We believe by not imposing independence between latent variables, CLERA can more accurately model the temporal dynamics and complex interactions captured by ODEs, leading to a better understanding of the underlying biological processes.

A critical aspect of using SINDy for Ordinary Differential Equation (ODE) discovery in biological systems is the question of whether there is a single unique way to represent (model) a

given process or if multiple valid representations exist. While CLERA effectively recovers governing equations from data, it's important to recognize that the method may yield different but equally valid solutions due to its consideration of linear transformations and rotations of the underlying dynamical systems. This inherent flexibility reflects the possibility that processes might not have a singular unique representation, but rather several that are mathematically equivalent yet distinct in form. This raises an important challenge in interpreting the biological significance of the discovered ODEs, as it suggests that multiple models could potentially describe the same underlying process. Especially for systems with limited prior knowledge, further investigation is needed to determine whether this flexibility captures biological variability or whether it points to the need for additional constraints to narrow down the solution space to the most biologically relevant model.

Despite its strengths, CLERA has some limitations that should be addressed in future work. First, the model's performance is sensitive to hyperparameter choices, particularly the dimension of the latent space and library of candidate functions. Developing more robust methods for automatic hyperparameter tuning could improve the model's consistency. Second, while CLERA can handle noise in the data to some extent, high noise levels can still impact its performance. Further improvements in denoising and data preprocessing techniques could enhance the model's robustness. Third, the interpretation of the learned latent space and governing equations requires careful consideration. While we have shown that the model captures biologically relevant information, additional work is needed to develop standardized methods for interpreting these results in the context of specific biological questions.

In conclusion, CLERA represents a significant step forward in our ability to uncover the governing principles of complex biological systems from high-dimensional scRNA-seq data in a parsimonious and generalizable manner. By bridging the gap between data-driven discovery and mechanistic understanding, this approach has the potential to deepen our understanding of cellular processes and the interactions between active gene programs.

4.5 Methods

4.5.1 Autoencoders

An autoencoder is a neural network architecture consisting of an encoder and a decoder (170). The encoder compresses high-dimensional input data into a lower-dimensional latent representation reducing dimensionality while preserving essential information. The decoder reconstructs the original data from this latent representation. Regularizations and architectural choices can be applied to constrain the latent embeddings, influencing their quality and utility. This flexibility allows autoencoders to be used for dimensionality reduction, feature learning, and data denoising. Training involves minimizing the reconstruction error, typically measured by mean squared error, to align the input with its reconstruction. For an input X, the encoder E produces latent embeddings z = E(x), which the decoder D uses to reconstruct the input as $\hat{X}$ = D(z). The reconstruction loss guides the optimization process so that the latent layer accurately captures the input data. The reconstruction loss for inputs is given by:

$$Reconstruction\ loss = \frac{1}{n}\sum_{i=1}^{n}(X_i - \widehat{X_i})^2$$

4.5.2 SINDy and SINDy autoencoder

SINDy is a regression technique used to discover a best-fit dynamical system from training data (41). It takes input data $x_t \in \mathbb{R}^n$ and predicts a system of ODEs as $\frac{dx_t}{dt} = f(x_t)$, where f is a library of candidate functions composed of basis functions that we want SINDy to model. SINDy selects a subset of active terms from these candidates, promoting parsimony in the model, resulting in a simpler and more interpretable model that captures the essential dynamics. For m input snapshots, we stack them to form $X = [x_1, x_2, ..., x_m]^T$ and compute the corresponding derivatives $\dot{X} = [\dot{x}_1, \dot{x}_2, ..., \dot{x}_m]^T$. The library is constructed as $\Theta(X) = [1, \theta_1(X), \theta_2(X), ..., \theta_p(X)] \in \mathbb{R}^{m\times p}$, where p is the length of the candidate library, and θ_i are the candidate model terms. The coefficient matrix can be represented as

$\Xi = [\xi_1, \xi_2, ..., \xi_n] \in \mathbb{R}^{p \times n}$, where ξ_i contains the coefficients for the library functions of differential equation i.

To address the challenge of high-dimensional data, SINDy autoencoders leverage an autoencoder to find a low-dimensional state $z = \phi(x)$, where x is the time-series input, z is the latent embedding, and ϕ is the encoder (46). We incorporate sparse regression into the autoencoder training to facilitate the simultaneous discovery of latent embeddings and the corresponding ODE. The SINDy Autoencoder finds a latent dimension $z_t = \phi(x_t) \in \mathbb{R}^d$, where d is the size of the latent embedding, and an associated dynamic model $\frac{dz_t}{dt} = f(z_t) = \Theta(z_t)\Xi$, where f has few active terms.

We train the autoencoder with additional SINDy losses as regularizations to ensure that the latent embeddings have a parsimonious model representation. The reconstruction loss, $L_{recon} = ||x - \psi(\phi(x))||_2^2$ is used which ensures accurate embeddings, while the SINDy weight regularization, $L_{SINDyReg} = ||\Xi||_1$ promotes sparsity of the coefficients. We also define two SINDy losses: $L_{dz/dt} = ||\nabla_x \phi(x_t)\dot{x}_t - \Theta(z_t)\Xi||_2^2$, which enforces accurate prediction of dynamics, and $L_{dx/dt} = ||\dot{x}_t - \nabla_z \psi(z_t)\Theta(z_t)\Xi||_2^2$, which reduces the error between the real-time derivatives of the input and the predicted derivatives using SINDy. We threshold the coefficients below a certain magnitude at regular intervals to promote parsimony. Finally, we conduct an additional refinement phase without L_1 loss and thresholding for a few epochs to refine the coefficients.

4.5.3 Overview of CLERA

CLERA is a deep learning-based model discovery method designed to uncover parsimonious and interpretable dynamical systems and active gene programs. It employs a specialized loss function to facilitate the joint discovery of a low-dimensional, interpretable latent space. The method's architecture integrates multiple key components to analyze gene expression data. At its core,

CLERA processes gene expression data to generate a cell-by-latent embeddings matrix. This latent representation serves as a crucial intermediate step, enabling three primary functions:

1. Data Reconstruction: A decoder network uses latent embeddings to reconstruct the original count matrix.

2. Cell-Type Classification: The latent space is used to classify distinct cell types within the dataset.

3. Dynamical System Discovery: CLERA incorporates SINDy to uncover a system of ODEs. These ODEs describe the temporal dynamics and interactions of the latent embeddings over the underlying biological processes.

This integration allows CLERA to simultaneously capture cell identities, gene expression patterns, and temporal dynamics.

Unsupervised learning of identifiable nonlinear representations has long been recognized as theoretically impossible without incorporating inductive biases or suitable constraints on the model (40). This fundamental challenge in representation learning has led researchers to explore various approaches to learn meaningful and causal representations of complex data. Autoencoders have emerged as a powerful tool for unsupervised representation learning, capable of discovering useful coordinate transformations and dimensionality reductions. However, while autoencoders can be trained in isolation, there is no guarantee that the learnt coordinates will be associated with sparse and physical dynamical models. To address this, CLERA integrates multiple components to learn interpretable and dynamically meaningful representations. By simultaneously training an autoencoder and a SINDy model, we constrain the network to learn coordinates associated with parsimonious dynamics. Also, CLERA incorporates a classifier connected to the latent embedding, leading to joint optimization of reconstruction, dynamics modelling, and classification tasks. By adding the classification task, the model can leverage additional information during training, to improve its ability to effectively capture the relevant dynamics of cell-specific behaviours. This integration introduces the inductive biases, potentially resulting in more meaningful and

interpretable embeddings of the system, while outputs a system of ODEs that describe the interactions between these latent variables.

4.5.4 Data Preprocessing

To preprocess the cell-by-gene matrix, the top 2,000 highly expressed genes are selected to focus on the most relevant features. The data is then normalized and log-transformed to stabilize variance and make the distributions of gene expression levels more comparable. The resulting matrix is analyzed using pseudotime (73) to establish a temporal sequence in the data and enable the conversion of static gene expression profiles into a dynamic time series which is essential for SINDy modelling. Given the sensitivity of SINDy to noise and the inherently noisy nature of scRNA-seq data, the time series is constructed by averaging gene expression counts over a sliding window of four-time steps (pseudo cell) with a stride of two steps. We experimented with multiple sliding window lengths but found that smaller window sizes would result in less effective denoising, while larger window sizes would lead to insufficient training data. Ultimately, a window length of 4 was chosen as the optimal balance between effective denoising and maintaining sufficient training data (Supplementary Note 1). This averaging process helps to smooth the data, reduce noise, and increase power. During this process, the cell type within each window is determined and assigned to ensure that cell identity is preserved. Derivatives of gene expression are calculated under the assumption of unit time steps between cells, a reasonable assumption given the continuous nature of pseudotime. This allows for consistent derivative estimation across the trajectory. Constructing the time series requires careful selection of cell types based on biological relevance, as some cell types may not contribute meaningfully to the dynamics being studied. This selection ensures that the resulting time-by-gene matrix accurately reflects the biological processes of interest

4.5.5 Components of CLERA Objective function

The preprocessed time series results in a pseudo-cell by gene matrix. CLERA takes each step as input to the autoencoder and generates reconstructions of the time-averaged gene expression for that box. These steps form a batch, consisting of 1292 samples in the Pancreas dataset and 1172 samples in the hematopoietic differentiation dataset. During training, the derivatives calculated during preprocessing are used to calculate the SINDy losses while the cell type serves as the label for the classification network.

During classification, the latent representation is passed through a classification network, which outputs the probability of the embedding belonging to each cell type. The cell types are represented as labels $l = [l_1\, l_2 \ldots l_c]$ where c is the total number of cell types. The classification network C produces the probability of the input belonging to each cell type, $y = C(z)$ where $y = [\widehat{y_1}\widehat{y_2} \ldots \widehat{y_c}]$ and $\widehat{y_j}$ is the probability of the label being j. Multi-class cross-entropy loss $L_{class} = -\sum_{j=1}^{c} y_j \log(\widehat{y_j})$ is used for this task. To improve stability during training, L1 regularization $L_{NetReg} = ||\varphi||_1 + ||\psi||_1 + ||C||_1$ is applied to the network weights. This regularization encourages sparsity in the weights, reducing the risk of overfitting and improving the model's generalization capabilities.

The loss function for CLERA is a weighted sum of 6 losses: Reconstruction Loss, Classification Loss, SINDy loss over the latent embedding, SINDy loss over the reconstructed genes, SINDy weight regularization, and Network weight regularization which can be represented as

$$L = \lambda_{recon}L_{recon} + \lambda_{\frac{dz}{dt}}L_{\frac{dz}{dt}} + \lambda_{\frac{dx}{dt}}L_{\frac{dx}{dt}} + \lambda_{class}L_{class} + \lambda_{SINDyReg}L_{SINDyReg} + \lambda_{NetReg}L_{NetReg}$$

where different $\lambda_{loss\ type}$ represent the loss weights.

4.5.6 Modelling choices and training

The model hyperparameters significantly influence the nature of the resulting equations and require careful tuning to ensure effective training and accuracy of CLERA. Several conditions are essential for this process. First, we monitor the loss curves and the number of active SINDy coefficients closely, as they provide key insights that guide refinement. Second, the functional forms used in the model are designed to align with established prior knowledge of the process. Additionally, we make assumptions to maintain the integrity of the system's dynamics: the right-hand side of every equation in the ODE should be non-zero, as a zero value would indicate that SINDy was unable to model the temporal features for that latent variable. Moreover, the discovered equations should be parsimonious, balancing expressiveness and simplicity. The dimension of the autoencoder's latent layer plays a crucial role in this balance, which shows the trade-off between model complexity and parsimony. As the size of the latent dimensions increases, the SINDy library expands quadratically. Therefore, selecting the appropriate latent dimension is crucial, as a larger latent space can lead to more complex candidate functions, which complicates the model and increases the risk of overfitting. Experiments with higher-dimensional latent spaces (e.g., dimensions of 10 or 15) reveal that the model consistently produces ODEs where the right-hand side is zero for some of the latent variables. This indicates that higher latent dimensions produce variables that are less critical for capturing the dynamics of the system. Based on these findings, the final model uses a latent space of 6 dimensions, which provides a balance between capturing essential dynamics and avoiding unnecessary complexity (Supplementary Note 1). The autoencoder and classification weights are initialized with Xavier initialization which is known to promote stability and efficient learning. We train both the pancreas and bone marrow datasets each for 50 times and we select the best model and parameters considering the conditions and criteria above such as the lowest loss, parsimony, and model complexity.

In modelling complex biological systems, it is crucial to leverage the prior knowledge we have of the system to carefully select a library of candidate functions. This selection should include functions capable of capturing a wide range of dynamics. Studies have considered various terms, such as polynomials up to degree 3, Michaelis-Menten kinetics, fractional terms, and exponential

functions, to model nonlinear and dynamic processes (16,19,21,171,172). Building on these approaches, we carefully select our SINDy library to include linear, second-order polynomial terms along with exponential, sinusoidal and fractional function forms. The sinusoidal terms are crucial for capturing oscillatory behaviors, while the exponential terms effectively model processes involving growth or decay. The fractional function helps with modelling nonlinear behaviors typically observed with biological processes, such as Michaelis-Menten kinetics. The SINDy model coefficients are randomly initialized with 1 and -1 which ensures that there is no bias towards one particular direction (positive or negative coefficients). This makes the model better at discovering the true underlying relationships in the data. Refer to Supplementary Table 1 for different Hyperparameters that are chosen while training CLERA on pancreas and bone marrow datasets.

For the bone marrow dataset, the autoencoder weight and SINDy coefficient initialization are done in a transfer learning framework. The SINDy library and autoencoder architecture are consistent across both the pancreas and bone marrow, ensuring that the trained weights and SINDy coefficients of the pancreas can be used as initialization for bone marrow. CLERA is initially trained on the Pancreas dataset using 15 different random initializations, resulting in 15 distinct instances of trained Pancreas weights. These trained instances are then evaluated on the bone marrow data. The top 6 experiments that exhibit the lowest losses when tested on the bone marrow data are selected. From these 6 experiments, an average of the Autoencoder weights and SINDy coefficients is computed. This averaged set of weights and coefficients is used as the initialization for subsequent bone marrow training. By leveraging transfer learning in this manner, the model benefits from faster convergence and improved performance, as the pre-trained weights provide a more informed starting point (Supplementary Fig 4).

Another key difference between the pancreas and bone marrow training lies in the SINDy thresholding interval. For the bone marrow dataset, the thresholding interval at 500 epochs is set higher than that for the pancreas dataset which is kept at 150. This adjustment is necessary as the SINDy coefficients derived from the pancreas experiments exhibit magnitudes much lower than

1, due to the application of L1 regularization. By setting a higher threshold for the bone marrow training, the network is better at adjusting and correcting the coefficient magnitudes, which ensures that key SINDy terms do not undergo early thresholding.

In the unsupervised training setting, where there is an absence of explicit target ODE terms and structure, an inductive bias is introduced to determine the appropriate number of training steps. A maximum acceptable term threshold is established, defining the maximum number of active SINDy coefficients permitted on the right-hand side of the ODE. During training, once the number of active coefficients falls below this threshold, the refinement phase is initiated, consisting of an additional 150 epochs. This threshold is kept at 30 to ensure that the model remains parsimonious.

4.5.7 Personalized Pagerank

PageRank is an algorithm that ranks nodes in a graph by evaluating the number and weight of links between them. Nodes that are connected to highly-ranked nodes are deemed more important and thus receive a higher PageRank score. The standard PageRank algorithm calculates a global importance score, treating all nodes equally in the process.

Personalized PageRank, on the other hand, is an extension of this algorithm that computes node rankings relative to a specific node or set of nodes within the graph. Unlike traditional PageRank, which provides a uniform importance score across the entire graph, PPR introduces a bias toward the selected nodes. This bias allows for the calculation of personalized importance scores, emphasizing the relevance of nodes in relation to the chosen node(s) of interest.

The interaction graph is constructed by including the gene-latent variable interaction using the top “K” SHAP for each latent variables and inter-latent-variable interaction via the ODE. As cells progress further in pseudotime and become more dissimilar, the mean intersection percentage between top-k genes of different latent variables decreases for all latent variables (Supplementary Fig 6, Supplementary Note 2). To rewire the original interaction graph and capture direct gene interactions, we use PPR to quantify the probability of connectivity between genes. This approach

models the likelihood that a random walk starting from one gene will visit another, thereby capturing both direct and indirect relationships in the network. For each gene i, we initialize a personalized vector v_i where the ith entry is set to 1 and all other entries are 0. This biases the random walk to originate from gene i. We then compute the PPR vector π_i by recursively solving the equation:

$$\pi_i = \alpha v_i + (1 - \alpha) A \pi_i$$

where A is the column-normalized adjacency matrix of the initial network and α is the teleportation probability (typically set to 0.85).

The resulting α_i vector contains PageRank scores for all genes relative to gene i. We interpret these scores as probabilities of connectivity, with higher values indicating stronger relationships. To construct the refined network, we select the top "E" edges for each gene based on these probabilities. In our implementation, we set E=600 to balance network sparsity and connectivity (Supplementary Note 2). This process is repeated for all genes, resulting in a directed, weighted graph that emphasizes the most significant regulatory relationships in the network. Also, the PPR-based rewiring captures higher-order network topology beyond direct interactions, which can reveal important indirect regulatory connections.

Calculating various centrality metrics on the obtained connectivity graph helps us identify the most influential genes, determine hubs of activity, and uncover critical pathways. Each centrality metric offers a unique perspective on the role and significance of nodes in the network. The centrality metrics used include:

- **Degree Centrality**: Degree centrality measures the number of direct connections a node has in a network. Nodes with a high degree of centrality are typically more active or prominent within the network.
- **Closeness Centrality**: Closeness centrality assesses the average length of the shortest paths from a node to all other nodes in the network. Lower scores indicate that the node occupies a more central and important position in the network.

- **Eigenvector Centrality**: Eigenvector centrality evaluates a node's influence based on both the quality and quantity of its connections. A node is more central if it is connected to other central nodes, highlighting its importance within the network.
- **PageRank**: PageRank assigns importance to a node based on the principle that connections from high-ranking nodes contribute more to a node's score, making it a more influential node in the network

By analyzing these centrality metrics, we identify the key genes in the interaction network, gene markers and hubs that can play a key role in cell function and gene interaction.

4.5.8 Simulated data

To evaluate SINDy's ability to detect governing equations, we design a simple two-gene regulatory network. This system is governed by stochastic differential equations, incorporating both deterministic dynamics and stochastic fluctuations. The network features two mutually inhibiting genes, described by G_1 and G_2:

$$dG_1 = \left(\frac{m_1}{\left(1 + {G_2}^2\right)} - k_D G_1\right) dt + \sigma_1 \sqrt{dt}\, dW_1$$

$$dG_2 = \left(\frac{m_2}{\left(1 + {G_1}^2\right)} - k_D G_2\right) dt + \sigma_2 \sqrt{dt}\, dW_2$$

In this model k_D represents the degradation rate, m_1 and m_2 determine the synthesis rates of the genes, σ_1 and σ_2 are noise intensities, dt is the time step, dw_1 and dw_2 are increments of independent Wiener processes modelling Gaussian noise. By adjusting these parameters, the system can exhibit saddle-node and pitchfork bifurcations. To generate training data for SINDy, we set m_1=3, m_2=1, k_D =1 and σ_1= σ_2= 0.2. The Euler–Maruyama method was used to simulate these stochastic differential equations, to capture the system's dynamics while accounting for noise that scales with time steps.

4.5.9 Data availability

The datasets used in this study are publicly available in public repositories. Simulated scRNA data was generated using SERGIO, which is accessible at https://github.com/PayamDiba/SERGIO. Preprocessed versions of both the pancreas development data (accession number GSE132188) and the human hematopoiesis data (accessed through the Human Cell Atlas data portal under the Human Hematopoietic Profiling project) can be downloaded from https://scvelo.readthedocs.io/en/stable/.

4.5.10 Code availability

Our Python implementation of CLERA can be found at: https://github.com/vasu-swaroop/CLERA

4.5.11 Competing interests

The authors declare no competing interest.

Chapter 5
FateNet: an integration of dynamical systems and deep learning for cell fate prediction

This chapter introduces FateNet, a computational approach that integrates dynamical systems theory with deep learning to explore the timing of cell decision-making using scRNA-seq data. By leveraging information about normal forms and scaling behavior near bifurcations FateNet predicts when cell decisions occur with greater accuracy than conventional methods, while offering a qualitative understanding into the specific type of bifurcation the system is likely to experience. The work titled *FateNet: An Integration of Dynamical Systems and Deep Learning for Cell Fate Prediction* was conducted with Dr. Thomas Bury and published in Bioinformatics in 2024 (3). The author list on the publication is below:

Mehrshad Sadria[1], Thomas M. Bury[2]
[1] Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.
[2] Department of Physiology, McGill University, 3655 Promenade Sir William Osler, Montreal, Quebec H3G 1Y6, Canada

5.1 Overview

Understanding cellular decision-making, particularly its timing and impact on the biological system such as tissue health and function, is a fundamental challenge in biology and medicine. Existing methods for inferring fate decisions and cellular state dynamics from single-cell RNA sequencing data lack precision regarding decision points and broader tissue implications. Addressing this gap, we present FateNet, a computational approach integrating dynamical systems theory and deep learning to probe the cell decision-making process using scRNA-seq data. By leveraging information about normal forms and scaling behavior near bifurcations common to many dynamical systems, FateNet predicts cell decision occurrence with higher accuracy than conventional methods and offers qualitative insights into the new state of the biological system. Also, through in-silico perturbation experiments, FateNet identifies key genes and pathways

governing the differentiation process in hematopoiesis. Validated using different scRNA-seq data, FateNet emerges as a user-friendly and valuable tool for predicting critical points in biological processes, providing insights into complex trajectories.

5.2 Introduction

Complex dynamical systems can experience sudden shifts between states when they reach a critical threshold known as a tipping point or critical transition (173). These transitions have been extensively studied in fields such as ecology, climate science, finance, and epidemiology (174–178). Tipping points are also pertinent in the field of medicine, as seen in diseases such as diabetes and epileptic seizures (179,180). These tipping points are characterized by a sudden shift from a healthy state to a diseased condition, signaling a critical change in the system's dynamics (181). Recognizing early warning signs of these transitions during disease progression would enable the identification of pre-disease conditions and facilitate timely medical intervention (182). Fortunately, there are universal properties of tipping points that can present themselves before a tipping point occurs (183,184). One such example is critical slowing down, characterized by a decrease in local stability and systematic changes in properties of time series data such as variance, autocorrelation and the power spectrum (185,186). These universal properties suggest the possibility for early warning signals across a wide range of scientific domains (187–190).

In developmental biology, cells undergo a variety of transitions as they differentiate. The Waddington landscape is a fundamental concept for understanding these critical transitions, as it explains how cells experience alterations in their transcriptome and epigenome while transitioning into unique cell types (191). Predicting early cell fate bias and understanding the mechanisms underlying cell decision-making are crucial for advancing cellular reprogramming (4). By deciphering the regulatory mechanisms governing cell fate decisions, we can reprogram cells into specific cell types for regenerative medicine applications, such as replacing damaged or lost cells in various tissues and organs (128,192).

In recent years, single-cell technologies have provided us with a high level of precision in studying individual cells, allowing us to observe and understand cellular changes (193). However, they only provide a snapshot of the cellular state, limiting our ability to capture the dynamic changes that occur over time (194). To overcome this limitation, computational methods based on pseudotime analysis have been developed to reconstruct the temporal progression of cells (195). These methods infer the order of cells along a trajectory based on the similarity of their gene expressions. Various techniques are employed to perform pseudotime analysis, such as using the distance to the root cell or computing entropy for each cell to position them in the trajectory (196). While RNA velocity can be integrated into pseudotime analysis and provide information on the direction of cell state changes, it doesn't always provide accurate directions due to various factors (197). Despite the promising results of all these methods, they are not yet able to detect critical transitions with high accuracy and cannot provide specific details about the biological system changes that occur during and after the transition.

Cell fate transitions can be viewed as a bifurcation in a high-dimensional dynamical system (191,198). It has been suggested that universal properties of bifurcations such as critical slowing down could be harnessed to provide early warning signals for their arrival (183). In a system subject to environmental and/or intrinsic noise, critical slowing down can be detected via an increase in variance and lag-1 autocorrelation, thereby serving as an early warning signal (186,190). Other early warning signals for bifurcations include an increase in entropy (199) and a decrease in Kolmogorov complexity (200). These measures are currently used to predict bifurcations across a range of scientific domains (201), including recent studies in cellular biology (148,202,203). However, they have had mixed success and are unable to predict the type of bifurcation.

In recent years, deep learning has emerged as a powerful tool for predicting changes in complex dynamical systems (204,205). In particular, a neural network can learn to predict bifurcations by training it on a massive corpus of simulation data from dynamical systems with noise (206,207). However, these current methods cannot be applied directly to pseudotime series of scRNA-seq data for two reasons: (i) scRNA-seq data is very high-dimensional, typically on the

order of thousands of genes; and (ii) pseudotime series do not contain temporal correlations typical of dynamical systems with noise, since each data point is a snapshot from a unique cell, not a single cell evolving over time.

In this study, we introduce FateNet, a novel computational model that combines the theory of dynamical systems and deep learning to predict cell fate decision-making using scRNA-seq data. By leveraging universal properties of bifurcations such as scaling behavior and normal forms (48), FateNet learns to predict and distinguish different bifurcations in pseudotime simulations of a 'universe' of different dynamical systems. The universality of these properties allows FateNet to generalize to high-dimensional gene regulatory network models and biological data. This approach not only provides an understanding of when cells undergo state changes but also captures the type of these transitions, identifying the characteristics of the system's new state. By using FateNet we demonstrate how perturbing specific sets of genes can alter the type of transition a system undergoes. Notably, FateNet eliminates the need for training a model on the specific system under study and allows us to overcome the limitations of most deep learning models, which are typically restricted to the systems they were originally trained on. We test FateNet using simulated and biological scRNA-seq data of various sizes and compare its performance to current methods for bifurcation prediction. Our results underscore FateNet's ability to detect the process of cell fate decision-making, offering insights into the ongoing transitions within the system and providing information on manipulating gene sets to modify transition types.

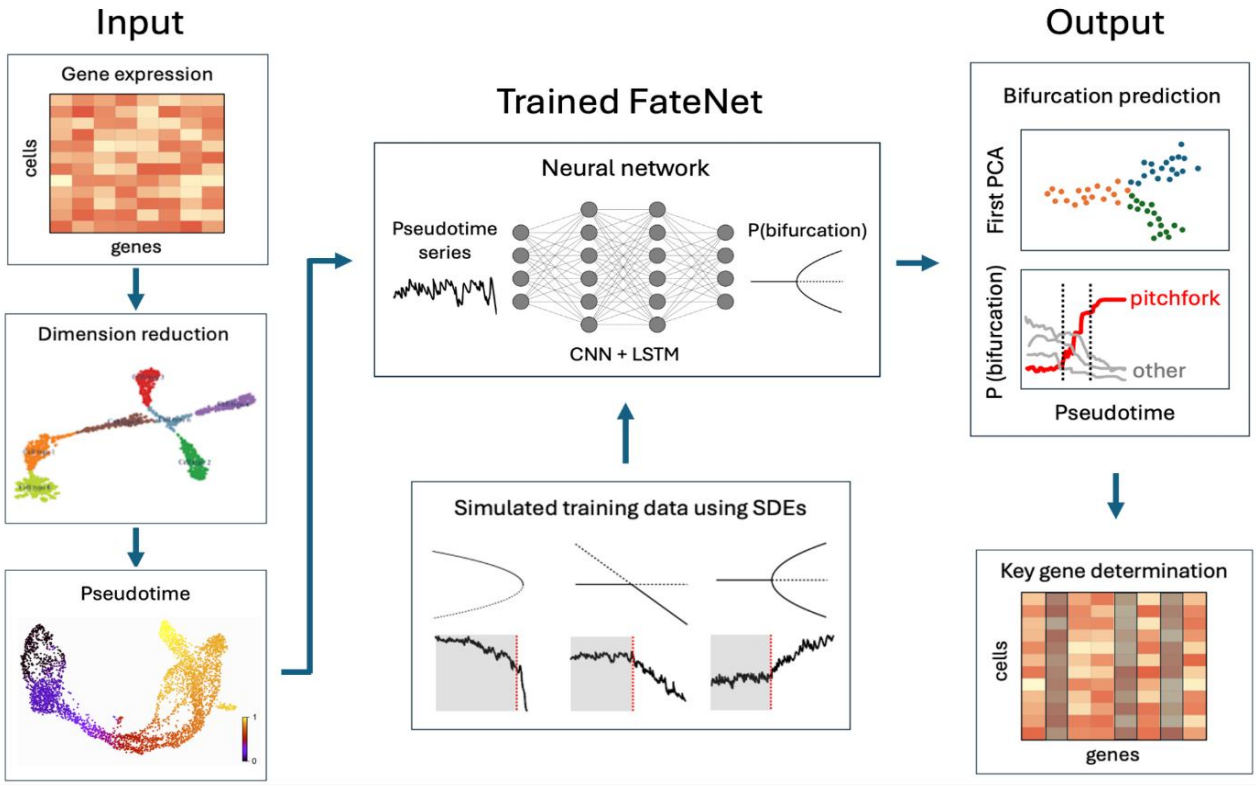


Figure 5-1: Schematic of workflow with FateNet. Input data, in the form of a gene expression matrix, is reduced in dimension using PCA, clusters are obtained and a pseudotime series is constructed. The pseudotime series leading up to a cell fate transition is passed into FateNet, which outputs a probability distribution over different bifurcations. FateNet is pre-trained using many simulations from stochastic differential equations (SDEs) going through different types of bifurcation. The output of FateNet consists of probabilities for the occurrence of different bifurcations. A spike in the probability for a bifurcation is an indication that the bifurcation is about to occur. The cells associated with the timing of the spike can be examined to identify the key genes responsible for initiating the bifurcation process.

5.3 Results

5.3.1 Bifurcation prediction in a simple gene regulatory network

To demonstrate FateNet, we use data generated from a simple model of a gene regulatory network that can undergo different types of bifurcation as different parameters are varied (Methods). The model undergoes a fold bifurcation as the synthesis rate of the first gene increases (m_1), and a pitchfork bifurcation as the degradation rate of the genes (k_D) decreases (148). We simulate the model with additive white noise and a linearly changing parameter that leads to (i) a fold bifurcation, (ii) a pitchfork bifurcation, and (iii) no bifurcation (Fig. 1). In the bifurcation scenarios, the bifurcation is reached at pseudotime 500. At a given point in time, FateNet takes in all preceding data and assigns a probability for a fold, transcritical and pitchfork bifurcation, and a probability for no bifurcation (null). A heightening in a bifurcation probability is taken as a signal that this bifurcation is approaching. FateNet successfully signals the approach of a fold and a pitchfork bifurcation in the gene regulatory network. In the case of the null trajectory, no significant probability is assigned to any of the bifurcations. It also signals the correct bifurcations in the cases of larger and smaller noise (Supplementary Figures 5,6).

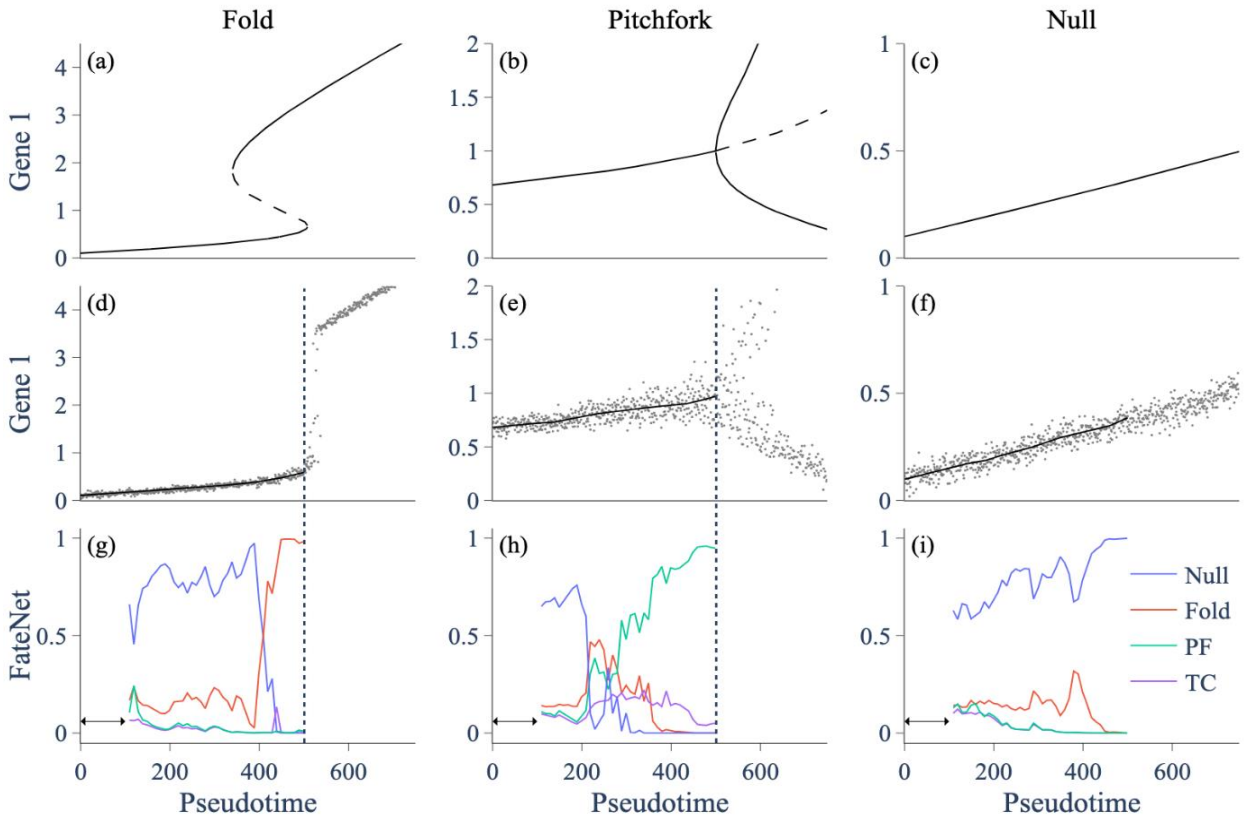


Figure 5-2: Simulations and predictions in the simple gene regulatory network model going through a fold, pitchfork, and no bifurcation. (a-c) Bifurcation diagrams showing the stable (solid) and unstable (dashed) states of the model as a parameter is varied. (d-f) Model simulation (gray) with the bifurcation parameter varying linearly with time (Methods), and smoothing (black) with a Lowess filter with span 0.2. The model reaches the bifurcation at pseudotime 500. (g-i) Probabilities assigned by FateNet for each class of bifurcation as progressively more of the data becomes available. The arrow shows the time window where there is insufficient data for FateNet to make a prediction. FateNet uses the data after smoothing (i.e. not the trend) when making its predictions. The vertical dashed line indicates the time when the bifurcation is crossed. PF: pitchfork; TC: transcritical.

5.3.2 Bifurcation prediction in a large simulated gene regulatory network

To assess the performance of FateNet on larger datasets we use SERGIO, a simulator based on stochastic differential equations that can generate both steady-state and dynamic scRNA-seq data (77). It incorporates different noise types for realistic dataset creation. We generate differentiation simulation data with 1800 cells, 100 genes and 7 cell types (Fig. 2a). After the preprocessing steps

in the 100-dimensional gene space, we apply Principal Component Analysis (PCA) to reduce the dimensionality of the data and obtain the principal components. To establish a cell ordering and understand the pseudotime dynamics within this dataset, we use Partition-based graph abstraction (PAGA) (Fig. 2b) (208). These data are then detrended and used as input to FateNet, leveraging the known underlying differentiation trajectory.

Focusing on one of the bifurcation points, our objective is to test whether our model can predict this bifurcation in advance and identify the specific type of bifurcation occurring. We find that FateNet not only provides an early signal for the upcoming change in cell state but also successfully identifies a pitchfork bifurcation in advance, consistent with the observed change in state at the cell-fate transition (Fig. 2c, Bifurcation trajectory panel). We then test our model on a scenario where the system does not undergo a cell-fate transition (Fig. 2c, Null trajectory panel). To generate such a trajectory, we sample points randomly (with replacement) from the first 20% of the detrended data and add it to the trend of the original data. In this case, our model correctly predicts the absence of any critical transition (Null), indicating that it has learned distinct features associated with the presence/absence of an upcoming cell-fate transition (Fig. 2c, Null trajectory panel).

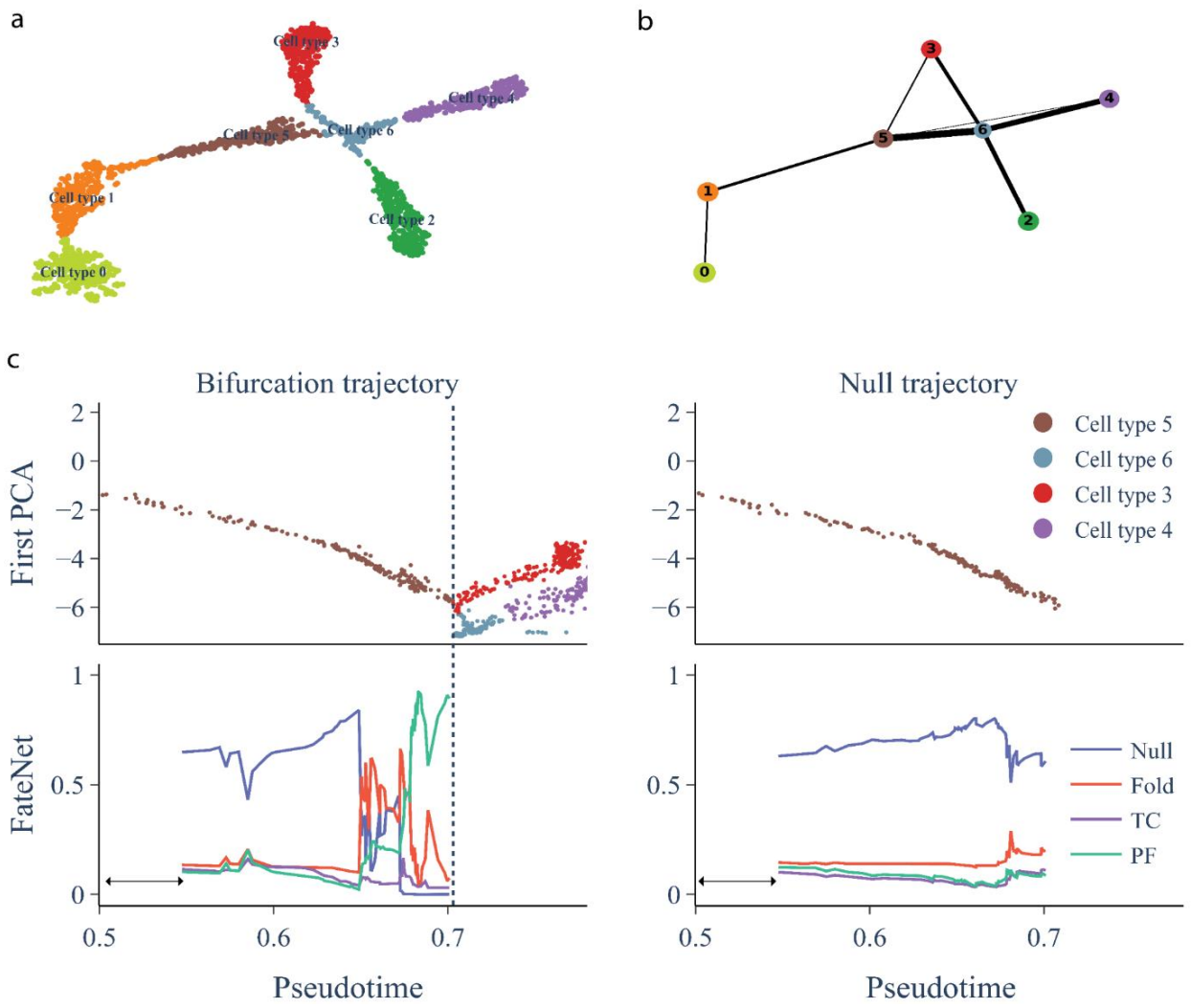


Figure 5-3: Bifurcation predictions in a simulation of SERGIO. (a) Uniform Manifold Approximation and Projection (UMAP) visualization of scRNA-seq data generated by SERGIO, with distinct clusters, color-coded based on cell type. (b) PAGA network graph representing the interconnectivity and relationships between cell types. (c) Bifurcation and null trajectories of cells organized in pseudotime (top) and the predictions of FateNet (bottom). The first principal component of the gene expression data is used to make predictions. The bifurcation trajectory shows a cell-fate transition between cell type 5 and cell types 3 and 6. The vertical dashed line indicates the time when the bifurcation is crossed. Data is smoothed using a Lowess filter with span 0.2 and the detrended data are passed to our model. The null trajectory is generated by taking a random sampling from the first 20% of the detrended data and adding it to the original trend. DL probabilities are the probabilities assigned by our model for each event among Null, Fold, Transcritical (TC) and Pitchfork (PF).

5.3.3 Bifurcation prediction in biological data

To test FateNet on biological data, we use temporal scRNA-seq data of mouse hematopoietic stem cell differentiation with 130,887 cells and 25,289 genes (81). Our emphasis is on the differentiation of progenitor cells, specifically exploring the decision-making process of neutrophil fate (Fig. 4a). In this context, we aim to understand both the timing of cell fate decision-making and the specific type of differentiation occurring within the system. Therefore, we use cells that are classified as undifferentiated or neutrophils and extract the top 3000 most variable genes. On this, PCA is conducted and the first principal component is used to make predictions of a bifurcation (Fig. 4b, Bifurcation trajectory panel). We find that FateNet predicts a pitchfork bifurcation before the transition from an undifferentiated cell to a neutrophil (Fig. 3b, Bifurcation trajectory panel). The transition is also preceded by an increase in variance, which is consistent with the phenomenon of critical slowing down that accompanies bifurcations. We compare this with a null time series that is generated by taking a random sample from the first 20% of the detrended bifurcation trajectory and adding this to the trend. This way, we demonstrate that the model is not making predictions based on the trend, but rather on the dynamics around the trend, which provide information about an approaching bifurcation. On the null trajectories, our model correctly predicts 'Null', i.e. no bifurcation (Fig. 4b, Null trajectory panel).

To assess the performance of FateNet and compare it to conventional methods for bifurcation prediction, we make predictions on 100 unique downsampled bifurcation trajectories and corresponding nulls from the biological data. For each trajectory, we make seven equally-spaced predictions between pseudotime 0.3 and 0.6, resulting in a total of 1400 predictions. The receiver operating characteristic (ROC) curve (Fig. 4c) illustrates performance on the binary classification problem of whether or not a bifurcation is approaching. An area under the curve (AUC) of 1 corresponds to a perfect performance, whereas an AUC of 0.5 (dashed line) is no better than random. FateNet achieves the highest performance (AUC=0.93), followed by variance (AUC=0.85), sample entropy (AUC=0.63), Kolmogorov complexity (AUC=0.53) and lag-1 autocorrelation (AUC=0.46). In addition, FateNet uniquely provides a prediction about the bifurcation type, which becomes more evident closer to the bifurcation. We show the specific

bifurcation probabilities from pseudotime 0.5 onwards, demonstrating that pitchfork is the favored bifurcation across the 100 downsampled trajectories.

To understand the underlying biological mechanisms governing cell fate decision-making, we focus on a critical segment of the pseudotime trajectory between 0.28 and 0.32, where a notable increase in bifurcation probability occurs (Fig. 4b, yellow box). By conducting differential gene expression analysis on cells within this specific temporal window, we identify key genes such as Myc, Ybx1, S100a8 and S100a9, Set, and H2afy whose expression showed a significant change compared to other parts of the trajectory. Remarkably, several of these genes have been shown by previous studies to play a key role in fundamental cellular processes, including stem cell differentiation, regulation of neutrophil differentiation, chromatin remodeling, and cellular metabolism (Supplementary Table 1 for a detailed list of genes, their functions and reference to previous studies). Furthermore, we leverage the top 250 genes with significant expression changes, from our results to scrutinize cellular pathways, components and functions involved in cell decision-making. Our analysis reveals enrichment in pathways linked to metabolic processes (organonitrogen compound metabolism, catabolic process, superoxide anion generation, protein metabolic processes), cell death, protein localization, and leukocyte activation (Supplementary Figure 1). These findings align with existing literature showing hematopoietic stem cells navigate a complex array of developmental pathways, including not only self-renewal and differentiation but also apoptosis and metabolism. The ultimate fate of dividing stem cells is shaped by the combination of signals from various regulators. Additionally, we use the enrichment map analysis to show the network of enriched pathways, illustrating the complex relationships and communication between the identified biological processes (Supplementary Figure 2). This result not only can help us understand the connections between active pathways within the cell's environment but also emphasizes the dynamic interactions influencing different regulatory mechanisms of cell fate decision making.

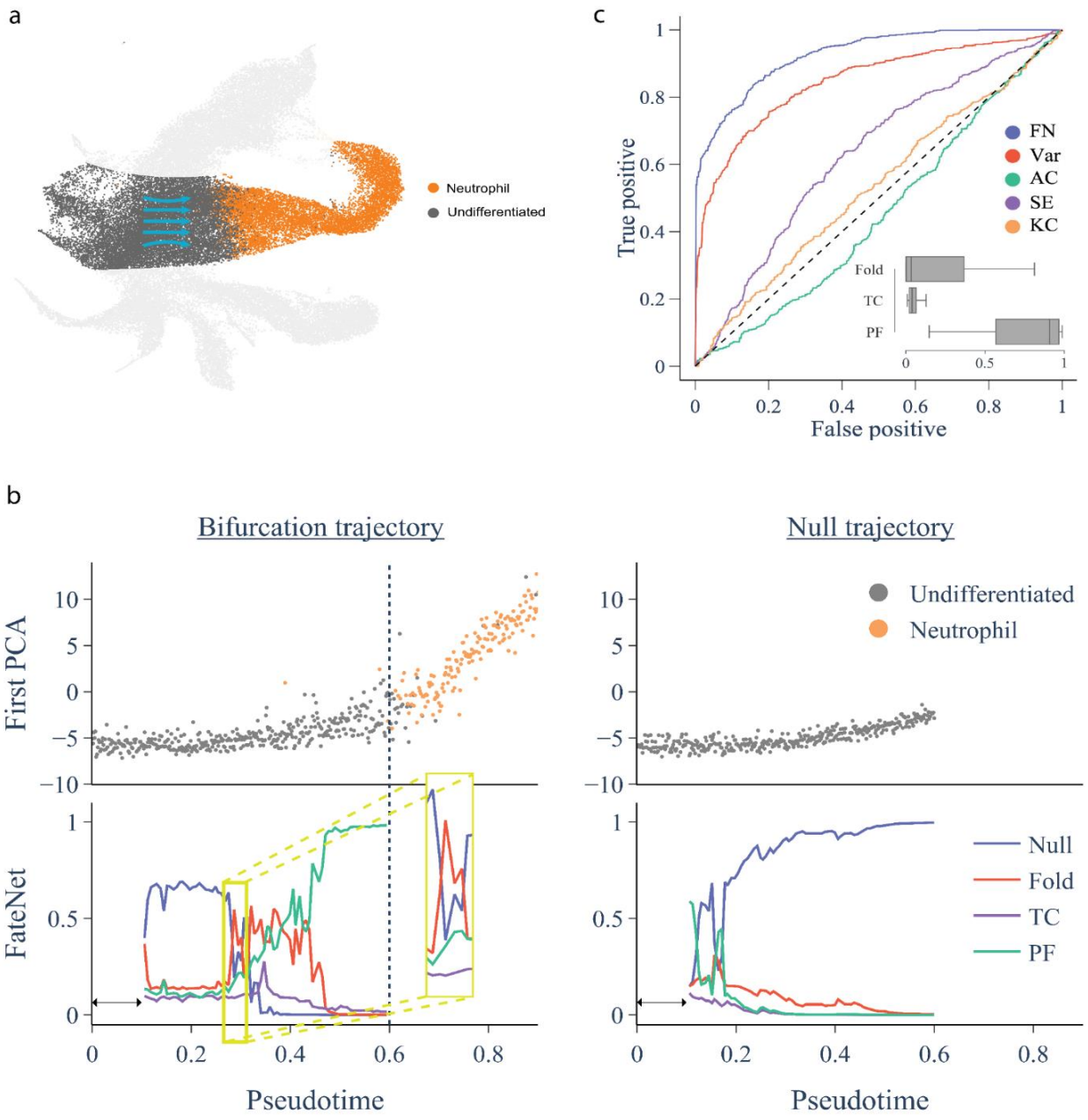


Figure 5-4: Predictions in data of mouse hematopoietic stem cell differentiation from undifferentiated cells (gray) to neutrophils (orange). (a) UMAP plot of mouse hematopoiesis data, emphasizing the transition (arrows) from progenitor cells (gray) to neutrophils (orange), elucidating the dynamic differentiation process. (b) Bifurcation and null trajectories with model predictions. The bifurcation trajectory (left) is the first principal component against pseudotime downsampled by a factor of 100. The dashed line shows the transition. The data is detrended using a Lowess filter with a span 0.2 and used as input to the model. The model outputs probabilities for each event among Null, Fold, Transcritical (TC) and Pitchfork (PF). The yellow box highlights the initial spike in bifurcation probabilities between

pseudotime 0.28 and 0.32. The null trajectory (right) is generated by random sampling from the first 20% of the detrended bifurcation trajectory and adding them to the trend. (c) ROC curves for predictions of any bifurcation using variance (Var), lag-1 autocorrelation (AC), sample entropy (SE), Kolmogorov complexity (KC) and FateNet (FN). Predictions are made at evenly spaced time points between 0.3 and 0.6 for 100 unique downsampled bifurcation trajectories and corresponding nulls, resulting in a total of 1400 predictions. The inset shows the probabilities assigned to each bifurcation between pseudotime 0.5 and 0.6. Boxes show the median and interquartile range, and whiskers show the full range.

To further validate our findings, we conduct an analysis on a mouse pancreas scRNA-seq dataset obtained from embryonic day 15.5 (Bastidas-Ponce et al., 2019), comprising 2,531 cells categorized into seven distinct cell types (Fig. 5a). Our primary objective is to focus on the differentiation process among endocrine cells. Through pseudotime ordering, we delineate a trajectory originating from Fev+ cells, which are subsequently bifurcated into multiple main branches. The termini of these branches include differentiated Alpha, Beta and Delta cells, indicating that the branches represent a transition toward fully differentiated cell states.

FateNet predicts a fold bifurcation for the transition from Fev+ cells to Alpha, Beta and Delta cells, identifying this bifurcation in advance at around 0.89 (Fig. 5b). To further test FateNet's accuracy, we compare it with a null time series generated by taking a random sample from the first 20% of the detrended bifurcation trajectory and adding this to the trend. FateNet correctly predicts 'Null,' indicating no bifurcation in this control scenario (Fig. 5b).

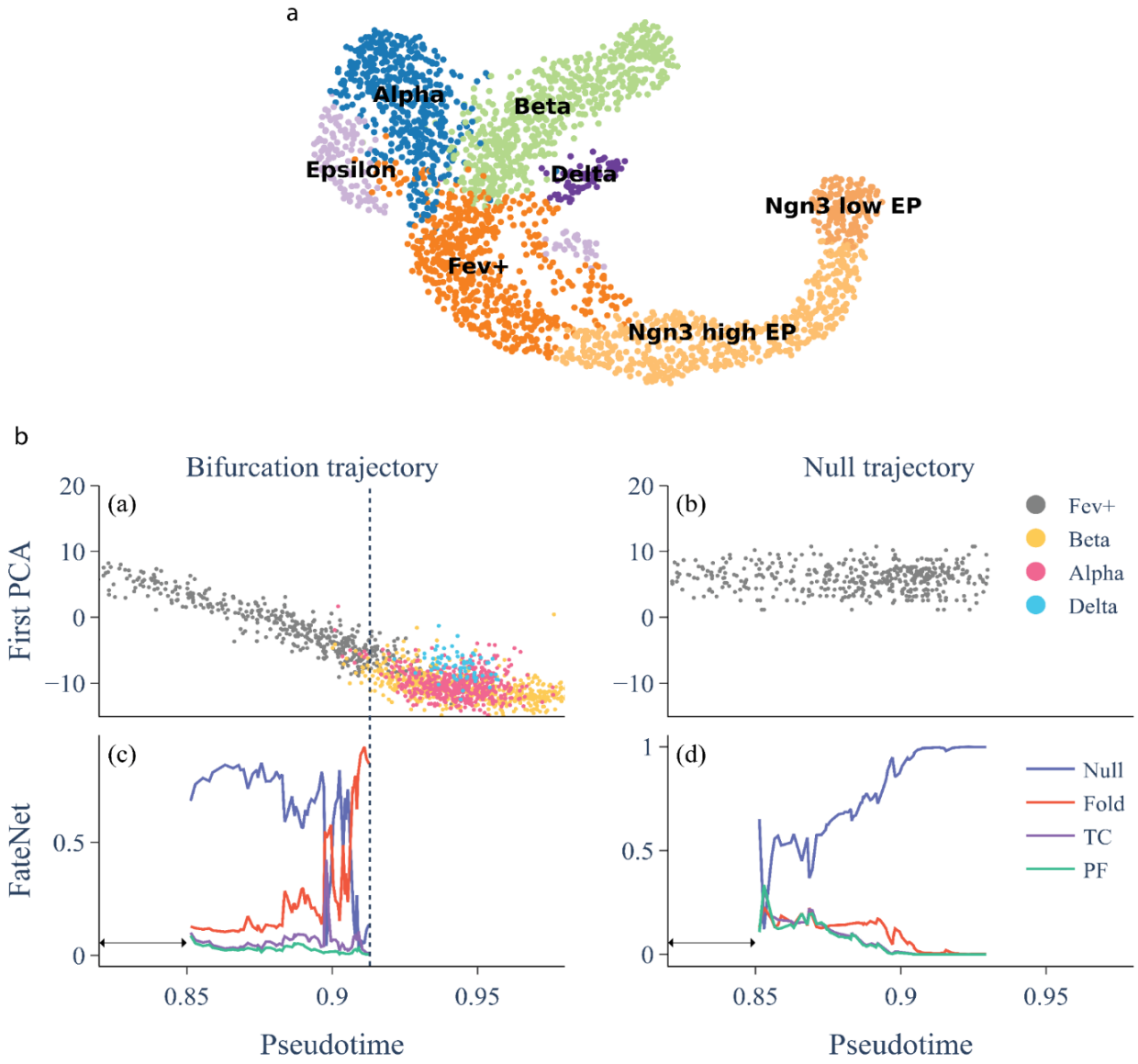


Figure 5-5: Bifurcation predictions in murine pancreatic development focusing on differentiation of Fev+ to Alpha, Beta and Delta cells. (a) Uniform Manifold Approximation and Projection (UMAP) visualization of Pancreas development data, with distinct clusters, color-coded based on cell type. (b) Bifurcation and null trajectories with model predictions. The bifurcation trajectory (left) is the first principal component against pseudotime. The dashed line shows the transition point. The data is detrended using a Lowess filter with a span of 0.2 and used as input to the model. The model outputs probabilities for each event among Null, Fold, Transcritical (TC) and Pitchfork (PF). The null trajectory (right) is generated by random sampling from the first 20% of the detrended bifurcation trajectory and adding them to the trend.

5.3.4 Effect of gene knockout / over-expression

We investigate the effect of in silico knocking out and overexpressing genes (hard and soft interventions) on the predictions made by our model (Fig. 6a). We knockout the most significant genes in the first PCA component of the data by setting their expression to zero. We find that knocking out as few as five of the top genes results in a change in the bifurcation prediction from a pitchfork bifurcation to a fold bifurcation (Fig. 6b). Continuing to knock out genes increases the prediction for no bifurcation (Null) until eventually, after knocking out the top 30 genes, no bifurcation is predicted at all. We also overexpress genes by multiplying their expression by a factor of two. We find that overexpressing a small number of the top genes (5-10) strengthens the prediction of a pitchfork bifurcation (Fig. 6c), whereas overexpressing a larger number of genes weakens the prediction of a pitchfork bifurcation. These results suggest that there are a few genes that are instrumental in the type of bifurcation that the system goes through. When these key genes are subjected to knockout, there is a substantial alteration in the bifurcation type of the system's dynamics (Fig. 6b). On the other hand, with the top genes overexpressed, the bifurcation type is predicted with greater probability (Fig. 6c). However, it is essential to highlight that overexpressing a broader set of genes can trigger additional regulatory mechanisms (Fig. 6a,c). This broader activation manifests in an increased probability of other types of bifurcations.

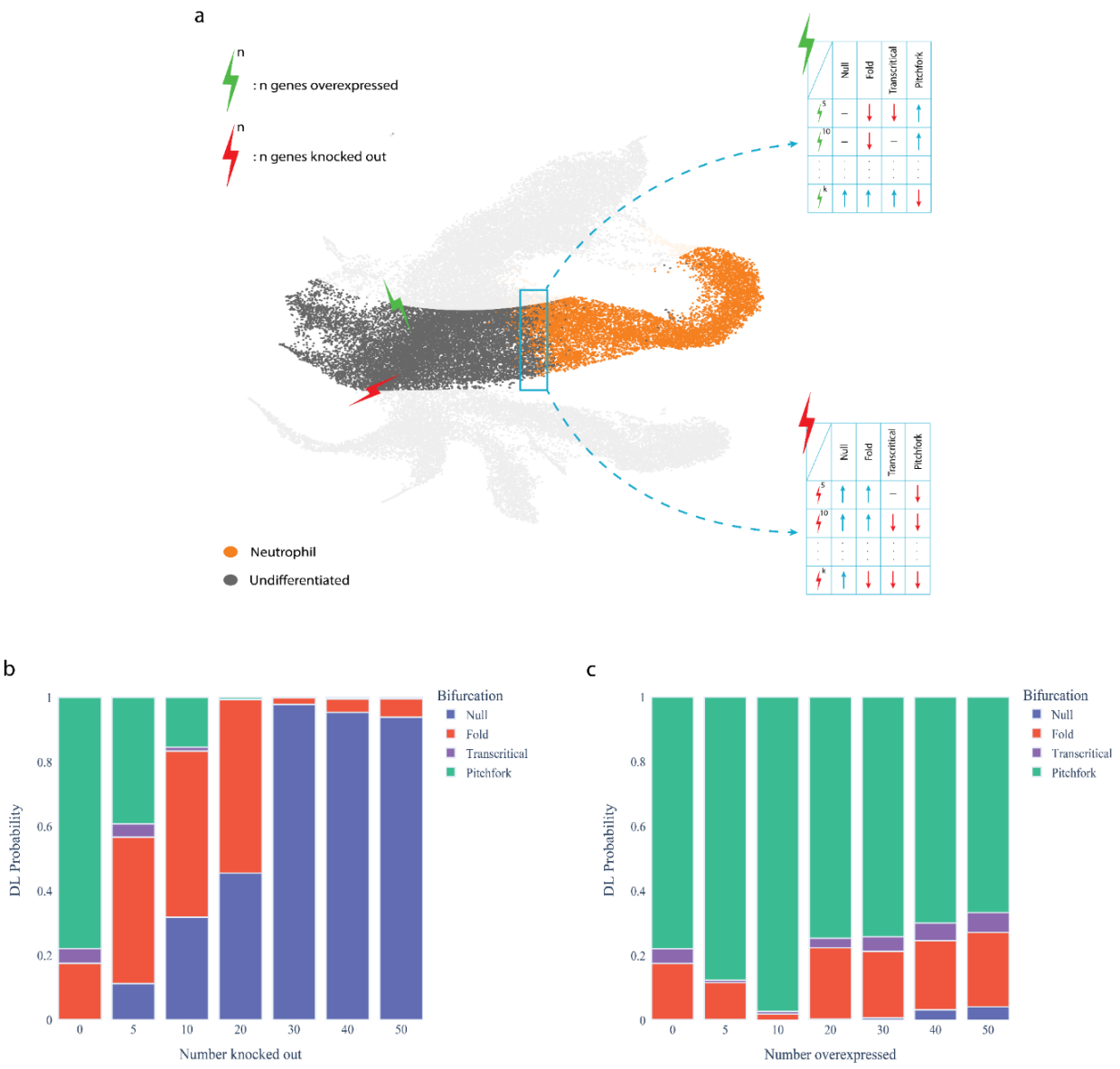


Figure 5-6: Exploring system response to various in silico perturbations. (a) UMAP visualization shows in silico perturbations, with green lightning indicating overexpression and red lightning denoting knockout perturbations. Each perturbation is individually implemented to observe how the system experiences shifts in the **bifurcation** dynamics. Model predictions for stem cell differentiation to neutrophils in mouse hematopoiesis after knocking out (b) and over-expressing (c) a few numbers of the most significant genes. Genes are knocked out by setting their expression to zero. Genes are overexpressed by multiplying their expression by a factor of two. In each case, ten equally-**spaced** predictions are made between pseudotime 0.45 and 0.6.

5.4 Discussion

In various domains such as ecology, climate, health, and finance, the identification of critical points, often referred to as bifurcation points, holds significant importance. Early detection is crucial for strategic decision-making and intervention, minimizing the potential for adverse consequences. Also understanding the type of these transitions is pivotal for preemptive actions in these diverse fields (209). For instance, in health, the heart can spontaneously transition from a normal rhythm to a dangerous one, known as a cardiac arrhythmia. Early detection of these critical transitions in cardiac activity can enable prompt medical intervention, significantly impacting patient outcomes and preventing life-threatening situations (210).

Biological processes exhibit similar critical points and can undergo different types of transitions which occur during both normal development and disease progression. In normal developmental trajectories, cells move through Waddington's landscape, experiencing bifurcations regulated by genetic and environmental cues. These events shape cell fate, determining whether a cell adopts a neuronal, muscular, or another specialized identity (211). Similarly, in disease trajectories, bifurcations can lead to different outcomes. For instance, consider a disease process where a cell can either recover or progress to a more severe state (212). Analyzing the paths cells take during differentiation and the process of cellular decision-making, including the precise timing of these events, is crucial for understanding development and unlocking the potential of stem cell therapies (213). However, the challenges in predicting the precise type and timing of these transitions persist due to the intricate and dynamic nature of biological systems, alongside the added complexity introduced by the high-dimensional nature of the data.

In response to these challenges, we introduce FateNet, a novel framework that integrates dynamical systems theory with deep learning to discern when cells make decisions and predict the type of transition the system is approaching. For a deep learning classifier to be effective and applicable across diverse scenarios, it necessitates training on a broad spectrum of data. We generate time series training data using simulations from a comprehensive library of different dynamical system models that possess various types of bifurcation. The universal properties of bifurcations, manifested in time series as a system approaches a bifurcation, facilitate this

generalizability. To validate our framework, we conduct extensive testing using both simulated and biological data, spanning different dataset sizes and varying noise levels. For the hematopoietic data, predictions were made on downsampled trajectories, showing that FateNet can work with missing data. Since FateNet predicts the type of bifurcation, it can identify genes that can prevent harmful bifurcations in a system and promote favorable transitions. Here, by performing both hard (knock-out) and soft (over-expression) interventions to a developmental system we have shown the possibility of targeting a specific set of genes to promote, prevent or modify the type of transition in hematopoiesis.

FateNet is the first attempt to combine dynamical systems theory with deep learning to predict the type of bifurcation resulting in a cell fate transition. It is trained to predict fold, pitchfork and transcritical bifurcations, which have been implicated in various cell fate transitions (191,214–216). However, there exist other types of bifurcations that may also be involved with cell fate transitions, such as the Hopf bifurcation (217) and global bifurcations (218). Future developments will involve a training library that includes a larger number of bifurcation types. FateNet is also currently restricted to univariate data, which is obtained from the first principal component. A boost in performance could be obtained by training the classifiers on multivariate data, allowing them to interpret multiple components of the PCA, where potentially more information about the bifurcation could be obtained. Recent success in predicting critical transitions using network measures (202,219,220) suggests that training classifiers in a similar way on gene interaction networks could be fruitful. Finally, it could be beneficial to determine the time series features that FateNet uses to determine the type of bifurcation using interpretability methods such as layer-wise relevance propagation (221). It seems plausible that it is learning features such as critical slowing down and distinguishable changes in the stability landscape to determine the type of bifurcation.

We believe future studies can benefit from the application of FateNet to compare bifurcation types in biological systems, particularly when comparing differences in the type and timing of system transitions in healthy and diseased conditions. Many disorders, including cancer and fibrosis, can be perceived as distinct types of transitions within a biological system (20,222–224). Consequently, FateNet provides valuable insights into identifying effective targets for

interventions and optimal timing in guiding and redirecting the system's evolution from an impaired to a healthy state.

5.5 Methods

5.5.1 Generation of training data for FateNet

We generate training data using simulations of a library of generated dynamical systems. Each dynamical system consists of the normal form for a bifurcation and higher-order polynomial terms with random coefficients and additive white noise. The higher-order terms add diversity to the training models, while still guaranteeing that they possess the desired bifurcation. We include fold, transcritical and pitchfork bifurcations in the library. The model framework for the fold bifurcation is:

$$x_{t+1} = -\mu + x_t - x_t^2 + \sum_{i=3}^{10} \alpha_i (x_t - \sqrt{-\mu})^i + \sigma\epsilon_t,$$

where x_t is the state variable, μ_t is the (potentially time-dependent) bifurcation parameter, α_i are drawn from the standard normal distribution, σ is the noise amplitude drawn from a uniform distribution between 0.005 and 0.015, and the noise process ε_t is drawn from a standard normal distribution. The initial value for the bifurcation parameter μ_0 is drawn from a uniform distribution with lower and upper bounds that make the dominant eigenvalue of Jacobian between 0 and 0.8 (the bifurcation occurs when this eigenvalue is 1). This results in bifurcation trajectories that start at different distances away from the bifurcation. Consequently, in the fold model, μ_0 can take values between -0.25 and -0.01. In all models, the bifurcation occurs at μ=0.

The model framework for the transcritical bifurcation is:

$$x_{t+1} = (1+\mu)x_t - x_t^2 + \sum_{i=3}^{10} \alpha_i x_t^i + \sigma\epsilon_t,$$

where μ_0 can take values between -1 and -0.2. The model framework for the pitchfork bifurcation is:

$$x_{t+1} = (1+\mu)x_t \pm x_t^3 + \sum_{i=4}^{10} \alpha_i x_t^i + \sigma\epsilon_t,$$

where μ_0 can take values between -1 and -0.2. For each model framework, we generate 1000 unique models. We run 20 simulations of each model going up to its bifurcation by incrementing μ_t linearly from μ_0 up to 0 over 600 time steps. If the model undergoes a noise-induced transition (defined as a deviation from equilibrium greater than 10 times the noise amplitude, σ), then the point prior to the transition is taken as the end of the time series. The last 500 data points are kept. If the model transitions before 500 points, it is discarded and replaced by a newly generated model. From these 20 simulations of 500 points, we construct 20 pseudotime series by placing the data in a 20x500 matrix and extracting elements in a diagonal fashion such that subsequent points in the pseudotime series come from different time simulations. Formally, denoting the data point from simulation i at time t as $x_{i,t}$, for i=0,...,20, t=0,...,499, then the ith pseudotime series is given by $y_{i,t} = x_{(i+t)mod20,t}$.

This process generates 20,000 'forced' pseudotime series for each bifurcation. We similarly generate 20,000 'null' pseudotime series where μ is kept fixed in each model. This gives a total of 80,000 time series that are labelled according to whether they are 'fold', 'transcritical', 'pitchfork' or 'null' trajectories. This set of time series is then shuffled and partitioned into a training, validation and test set according to the ratio 0.95:0.025:0.025. The validation and test sets were chosen as a small percentage because a set containing a few thousand time series is adequate to provide a representative estimate of the performance measures used to assess the algorithm.

5.5.2 FateNet architecture and performance

FateNet consists of two types of neural network that are trained to predict a bifurcation label given a portion of pseudotime series data. Network 1 is trained on time series censored at the beginning

and at the end by a randomly drawn length. This forces it to predict bifurcations based on the middle portions of the time series. Network 2 is trained on time series censored only at the beginning, allowing it to learn from data right up to the bifurcation. Formally, the length of the censored time series L is drawn from a uniform distribution with lower and upper bounds of 50 and 500, respectively. Then, for Network 1, the start time of the censored time series is drawn from a uniform distribution between 0 and 500-L, and for Network 2 the start time is set to 500-L. The censored time series are then normalized by their mean absolute value and prepended with zeros to make them 500 points in length—a requirement for the neural network. FateNet uses the average prediction from these two networks.

Each network has a CNN-LSTM (convolutional neural network—long short-term memory network) architecture (Supplementary Figure 3). The CNN layers capture local features of the time series and LSTM layers capture dependencies over time. The networks receive an input time series $x \in \Re^{500}$ through a convolutional layer to obtain the hidden units

$$h_i = \sigma\left(b + \sum_{j=1}^{12} w_j x_{i+j-6}\right)$$

where i runs from 0 to 500, σ is the ReLU activation function, and b and w_j are the bias and weights of the kernel, which are trainable parameters. We use a kernel size of 12, and pad the edges of the time series with zeros to maintain the input dimension. We apply this operation for 50 kernel filters for Network 1 and 100 filters for Network 2. We then apply a dropout of 10%, which randomly fixes this proportion of hidden units to zero at each iteration of the training process. The units are then passed through a max pooling layer, which takes the maximum value over a window that strides over the units. We use a pool size of 2 and a stride length of 2. This process is then repeated for a second convolutional and max pooling layer.

The output is then passed to an LSTM layer with 100 memory cells for Network 1 and 50 cells for Network 2, where each cell is capable of capturing both long and short-term dependencies in the data (225,226). A LSTM cell consists of several components including the input gate (i_t), the forget gate (f_t), the output gate (o_t), the cell state (c_t) and the hidden state (h_t), where t runs

from 0 to 250. Each LSTM cell is updated as follows. The forget, input, and output gates are update as

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

where x_t is the input from the previous hidden layer, W and U are weight matrices, b are bias vectors, σ is the sigmoid activation function

$$\sigma(x) = \frac{e^x}{1 + e^x}$$

and the initial value for the hidden state is $h_0 = 0$. Meanwhile, a cell input activation vector is computed as

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

which is used to determine the new cell state and hidden state

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

$$h_t = o_t * \tanh(c_t)$$

where * is element-wise multiplication of vectors. The initial value for the cell state is taken as $c_0 = 0$.

This is then passed to the second LSTM layer with 20 memory cells. Here, only the final value of each cell state sequence is stored. Finally, this is passed through a dense layer with softmax activation to obtain the probabilities

$$y_i = \text{softmax}\left(b_i + \sum_{j=1}^{10} w_{ij} x_j\right)$$

where i runs from 1 to 4, b_i are the biases, and w_{ij} are the weights, and x_j are the inputs from the previous hidden layer. In total, Network 1 contains 100,864 trainable parameters and Network 2 contains 157,364. To obtain optimal values, we performed a hyperparameter sweep over the

number of convolutional layers, the number of convolutional filters, and the number of memory cells in the LSTM layers.

The networks were initialized and trained in Tensorflow v2.10 using the Adam optimization algorithm with a learning rate of 0.0005, a batch size of 1024, and categorical cross entropy as a loss function, given by

$$L = -\sum_{i=1}^{4} t_i \log(y_i)$$

where the vector t is the one-hot encoded truth label and y is the probability vector output from the model. The networks reached peak performance on their validation sets after 71 and 154 epochs, respectively (Supplementary Figure 7). When evaluated on the test set, Network 1 obtained an F1 score of 0.64 on the multi-class prediction problem and 0.80 on the binary prediction problem. Network 2 obtained F1 scores of 0.95 and 0.98, respectively. The lower performance of Network 1 is due to it being trained and tested only on middle portions of time series that do not necessarily contain distinguishable information about the type of bifurcation. The confusion matrices (Supplementary Figure 4) show the performance of the networks at predicting each class in the test set. Out-of-sample predictions are made by taking the average prediction across an ensemble of 5 networks of type 1 and 5 networks of type 2.

5.5.3 Applying the model to pseudotime data

To obtain model predictions on pseudotime data, we first obtain the principal component from a PCA. We then detrend it using a Lowess filter with a span of 0.2. Predictions from our model at a given point in pseudotime are obtained by taking the preceding data, normalizing them, prepending them with zeros to make the input 500 points in length, and feeding it into the model. The model then outputs a probability vector, whose components indicate the probability of each event. These computations are performed with the Python package ewstools (227).

5.5.4 Alternative methods for bifurcation prediction

We compare the performance of FateNet with popular methods such as variance, lag-1 autocorrelation, sample entropy and Kolmogorov complexity. An increase in the first three and a decrease in the latter is expected prior to the bifurcation. We compute these measures on the detrended data over a rolling window of 0.25 according to standard methods (228). The sample entropy and Kolmogorov complexity are computed using EntropyHub (229). The Kendall tau value is used to assess the trend of the measures, and as a discrimination threshold for computing the ROC curves.

5.5.5 Simple model for a gene regulatory network

This model consists of two coupled differential equations (148) adapted to include additive Gaussian white noise and an additional parameter to modulate the nonlinear interaction between the genes. The model is given by

$$\dot{g}_1 = \frac{m_1}{(1+g_2^2)} - k_D g_1 + \sigma\xi_1(t)$$
$$\dot{g}_2 = \frac{m_2}{(1+cg_1^2)} - k_D g_2 + \sigma\xi_2(t)$$

where g_1 and g_2 are the level of gene expression for each gene, k_D is the degradation rate, m_1 and m_2 determine the scales of their synthesis, c governs the nonlinear response of gene 2 to gene 1, σ is the noise amplitude, and $\xi_1(t)$ and $\xi_2(t)$ are Gaussian white noise processes.

The model possesses a fold and a pitchfork bifurcation. We simulate trajectories going through both types of bifurcation, as well as a null trajectory where there is no bifurcation but still a time-dependent parameter. For the fold trajectory, we take m_1 increasing linearly from 1 to 4.75, m_2=3, k_D=1, and c=1. For the pitchfork trajectory, we take m_1=1, m_2=1, k_D decreasing linearly from 1 to 0.25, and c=1. For the null trajectory, we take the same parameters as the fold trajectory, except we set c=0.1, which removes the fold bifurcation. In each case, we simulate the model using the Euler-Maruyama method with a step size of 0.01 and a noise amplitude σ=0.05. We then downsample the data by a factor of 100. As described in the section on generating training data,

we then construct a pseudotime series using 20 simulations of the model and taking subsequent points from each simulation.

5.5.6 SERGIO simulation

SERGIO is a simulator of single-cell gene expression data based on a stochastic differential equation (SDE) framework that can simulate noise and variability of gene expression, as well as the effects of external stimuli and cell cycle progression. SERGIO can model the stochastic nature of transcription, the regulation of genes by multiple transcription factors, and the differentiation of cells along complex trajectories. SERGIO also allows users to specify a gene regulatory network and various parameters that control the simulation process. Using SERGIO, we generated a synthetic scRNA-seq dataset that mimics a dynamic differentiation program, similar to the DS11 dataset described by (77). Our dataset consists of 1800 single cells from 7 cell types, with a gene regulatory network of 100 genes.

5.5.7 Analysis of scRNA-seq data

As a preprocessing step, we scaled, centered, and log-normalized the scRNA-seq gene expression data, and extracted the top 3000 most variable genes. We used the top 40 principal components to map the highly variable genes onto a lower dimension for clustering, using a k-nearest-neighbor graph with K=30. Clusters were visualized using the Python package "UMAP". For the scRNA-seq data, partition-based graph abstraction (PAGA) was employed to sort cells accurately over time, allowing for a detailed mapping of the dynamic progression of the processes. PAGA is an extended version of diffusion pseudotime that considers disconnected graphs. By covering both aspects of clustering and pseudotemporal ordering it can model the underlying biological progression, assigning each cell a pseudotime value that reflects its position along the inferred trajectory. Also, Cytoscape and Enrichment Map are used to investigate and visualize the connection between the identified biological pathways (135,230).

In testing our model, we consider the bifurcation from undifferentiated cells to neutrophils. The bifurcation is crossed at approximately pseudotime 0.6. For making predictions, we consider the first PCA component as a function of pseudotime up to the bifurcation. This still contains a lot of data (61309 cells), so we downsample by a factor of 100 to obtain a shorter time series more appropriate for our model. We then detrend the data using a Lowess filter with a span of 0.2. We are able to obtain 100 unique bifurcation trajectories from the biological data by shifting the downsampling procedure one point at a time. We construct 100 null time series (that do not undergo a bifurcation) by sampling randomly from the first 20% of the detrended data and adding it to the trend of the original data.

To investigate the effect of gene knockout, we set the expression of a fixed number of genes to zero. The genes that are selected are those that are most highly represented in the top PCA component. Once these gene expressions have been set to zero, we recompute the top PCA component to obtain a new bifurcation trajectory. To investigate the effect of gene overexpression, we follow a similar procedure, except we multiply the expression of the most significant genes by a factor of two.

5.5.8 Data availability

The simulated scRNA-seq data can be generated by SERGIO and can be found at: https://github.com/PayamDiba/SERGIO. The hematopoiesis Weinreb et al. data can be downloaded from Gene Expression Omnibus (GEO) under accession number GSE140802. The data set of pancreatic development can be found from GEO under accession number GSE132188. The simulated scRNA-seq data generated by the simple gene regulatory network can be found in the GitHub repository https://github.com/ThomasMBury/fatenet.

5.5.9 Code availability

Our Python implementation can be found at: https://github.com/ThomasMBury/fatenet.

Chapter 6
The Power of Two: integrating deep diffusion models and variational autoencoders for single-cell transcriptomics analysis

This chapter delves into scVAEDer's capabilities, the first scalable deep-learning model that integrates autoencoders and deep diffusion models using scRNA-seq data. This approach learns a latent representation of data, which captures both global patterns and local variations. The chapter demonstrates scVAEDer's accuracy by showing its ability to generate new high quality scRNA-seq data, predict perturbation effects across cell types, track gene expression changes during dedifferentiation, and identify key regulators in biological processes. The work titled *The Power of Two: Integrating Deep Diffusion Models and Variational Autoencoders for Single-Cell Transcriptomics Analysis* was conducted with Prof. Antia Layton and is currently under review at the *Genome Biology* journal (4). The author list on the publication is below:

Mehrshad Sadria[1], Anita Layton[1,2,3,4]
[1] Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada.
[2] Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.
[3] Department of Biology, University of Waterloo, Waterloo, Ontario, Canada.
[4] School of Pharmacy, University of Waterloo, Waterloo, Ontario, Canada.

6.1 Overview

Discovering a lower-dimensional embedding of single-cell data can improve downstream analysis. The embedding should encapsulate both the high-level features and low-level variations. While existing generative models attempt to learn such low-dimensional representations, they have limitations. Here, we introduce scVAEDer, a scalable deep-learning model that combines the power of variational autoencoders and deep diffusion models to learn a meaningful representation that retains both global structure and local variations. Using the learned embeddings, scVAEDer can generate novel scRNA-seq data, predict perturbation response on various cell types, identify

changes in gene expression during dedifferentiation, and detect master regulators in biological processes.

6.2 Introduction

Recent advancements in single-cell sequencing technologies have presented an unparalleled opportunity to examine the characterization of cell types and states with unprecedented resolution and scale. Owing to this technology we can now generate vast amounts of data to explore intracellular and intercellular interactions and precisely quantify genomic, transcriptomic, and other multi-omics information at the cellular level (1). single-cell RNA sequencing (scRNA-seq) can also be used in conjunction with other technologies such as CRISPR-based perturbation (Perturb-seq) to uncover the function of genes and regulatory elements (2). Despite its potential, the technical complexity and high implementation costs of Perturb-seq have limited its widespread adoption (3).

Although scRNA-seq datasets have a high dimensional nature, their intrinsic dimensionality is often low, governed by cell regulatory networks (4). Machine learning (ML), particularly deep learning (DL) models like autoencoders (AEs), have emerged as powerful tools for analyzing this data (5,6). These models excel at various tasks, including dimensionality reduction, clustering, data denoising, and batch effect correction (7–9). In addition, there has been a surge of interest in using deep generative models, such as variational autoencoders (VAEs) and generative adversarial networks (GANs), for scRNA-seq analysis (5,6). By leveraging their ability to learn the underlying data distribution, these models have become valuable tools for generating realistic simulations of gene expression profiles, reconstructing cellular trajectories, and predicting cellular responses to different types of perturbations (7,8). While these generative models have demonstrated their potential, there are still issues that need to be addressed. Specifically, VAEs can experience posterior collapse (9), and GANs can struggle mode collapse (10), in which the learned mapping reduces significant portions of the input data to a single point in the output. Such problems are often caused by the disparity between the geometries of the prior and target data

distributions (11). The performance of these models is also affected by other issues such as training instability, vanishing gradients, and prior hole (12). All these challenges can eventually lead to biased or low-quality results, ultimately hindering the application and efficiency of using generative models in single-cell technology (13,14).

Recently, Denoising Diffusion Models (DDMs) have gained significant attention as potent generative models. They show their exceptional performance surpassing other generative models not only in image synthesis but also in domains such as natural language processing, music, and protein structure generation (15–17). In DDMs, the data undergoes a gradual perturbation through the diffusion process, then a deep neural network is trained to learn and remove the added noise in multiple steps. These methods can subsequently be employed to generate novel data in an iterative manner, starting from random noise without having issues associated with VAEs and GANs that were previously mentioned (18). However, traditional DDMs necessitate a costly, iterative sampling process and lack a low-dimensional latent representation that can capture both high level structure and fine-grained variations (19,20). They employ a Markov process to transform the data distribution through small perturbations, which demands a large number of diffusion steps in both the training and inference phases. As a result, employing DDMs for large datasets is often difficult or not feasible. Therefore, to enhance the quality and speed of these models, various attempts have been made. These include refining the forward process, implementing a more effective sampling method, integrating guidance from a classifier, or utilizing the low dimensional latent space of autoencoders for training (19,21–23).

In this study, we present scVAEDer, a new method for generating high-quality scRNA-seq data by combining the strengths of VAEs and DDMs (latent diffusion models). Instead of relying solely on VAE's prior, our method incorporates both VAE and DDM priors to more precisely capture the distribution of latent encodings in the data. This allows for effective estimation of intermediate cell states and precise prediction of gene expression changes during the transition process from one cell type to another. Furthermore, by using vector arithmetic in the DDM space scVAEDer outperforms state-of-the-art (SOTA) methods in predicting single-cell perturbation responses for novel conditions, which are not included in the training process. Finally, we show

that our method can successfully identify known and novel master regulatory genes involved in cellular reprogramming by computing the velocity of genes during the interpolation process. We evaluate the performance of our method by applying it to multiple datasets with diverse sizes and characteristics.

6.3 Results

6.3.1 Method outline

Generative models have shown great potential in analyzing single-cell data usually by finding a lower dimension which facilitates downstream tasks such as visualization, clustering, and data generation. However, they are not without limitations. For example, in VAEs one significant issue is the prior hole problem, where the distribution of all encodings of the training data does not perfectly form a Gaussian due to a mismatch between approximate posterior and prior distributions. This can negatively affect the quality of the generated data when interpolating or operating directly on the VAE's latent space.

We develop scVAEDer that combines the strengths of VAEs and DDMs to produce higher quality, and more reliable results without encountering VAE issues. First, it takes single-cell gene expression data as input to train a VAE. Then it uses the latent embedding generated by the VAE to train the DDM which models the data distribution by learning a denoising process iteratively (Methods). Once trained successfully, the reverse process of diffusion can be used as a generative model which maps an arbitrary Gaussian noise $N(0, I)$ to a new VAE latent representation after T successive steps (Fig. 1). Therefore, by sampling from the trained DDM and feeding it to the VAE's decoder, realistic gene expression profiles for different cell types can be generated. scVAEDer's ability to generate high-quality data is due to its access to both VAE and DDM latent spaces, which enable capturing high-level structures and low-level stochastic variations together, consistent with leveraging latent diffusion models in other domains (231–233). Using the trained embedding, we can also predict the effects of different perturbations and perform interpolations between cell types (Method section for more details). We apply scVAEDer to different biological

data and show that the method can accurately approximate the full distribution and trend of key genes during perturbation or cellular transition better than SOTA generative models (Fig. 1). Also, as biological systems need to respond quickly to external and internal perturbations, we believe master regulators can be discovered based on their rate of response. scVAEDer computes the velocities by looking at the changes during each interpolation step or considering the average velocity over N steps. Gene Set Enrichment Analysis (GSEA) on genes that are ranked based on high velocity reveals enrichment in several key pathways.

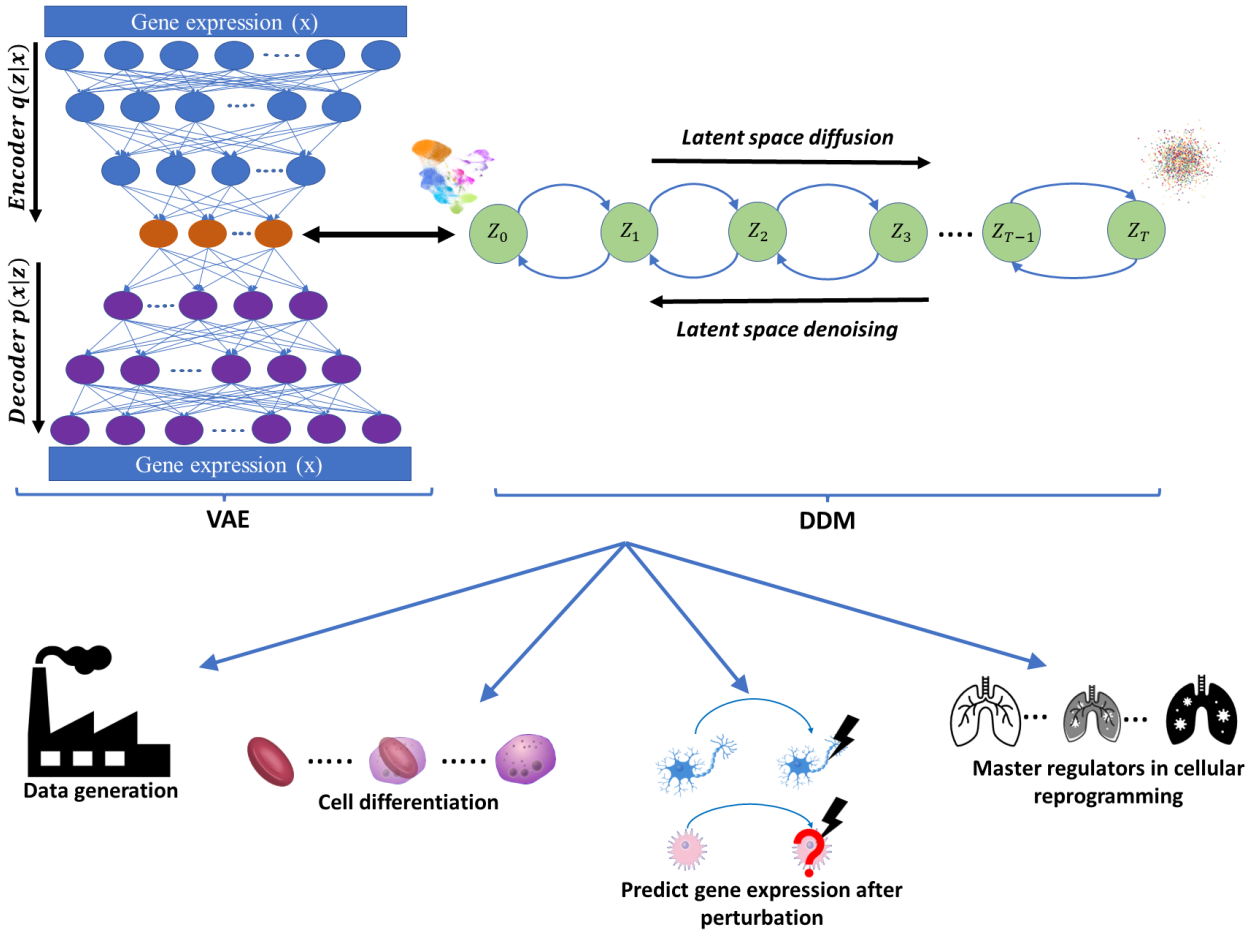


Figure 6-1: scVAEDer overview. scVAEDer integrates VAE and DDM. First, a VAE is trained using the gene expression data. Then the VAE latent embedding is used to train the DDM through the processes of latent space diffusion and denoising. Combining together the model is able to decode back the gene space with high accuracy. scVAEDer can be used for different downstream analysis tasks such as generating novel high-quality scRNA-seq data, understanding changes in gene expression during cellular differentiation, predicting the effect of perturbations on new cell types when expression data is available

for multiple conditions, detecting master regulators by interpolating between different cellular states and ranking fast responder genes based on their velocity values.

6.3.2 scVAEDer generates novel realistic scRNA-seq data

The application of DDMs in computer vision has yielded promising results for generating high-quality images. With this in mind, we investigate the potential of these models in generating single-cell gene expression data. By using scRNA-seq data of zebrafish hematopoiesis x_0 with 1390 cells and 1845 genes, we train a VAE and compute the latent code Z_{sem}= Encoder(x_0). Then, Z_{sem} is used to train a diffusion model. Figure 2a illustrates the transformation of input data into Gaussian noise through a gradual noise addition process using forward diffusion. Once DDM is trained, the reverse process can be employed to generate new samples (Fig. 2a, blue process). This involves an iterative denoising process, starting with random Gaussian noise. Increasing the number of steps in the training of DDMs results in higher-quality data generation, but also increases computational costs presenting a tradeoff between sample quality and cost (Methods for more details). To assess the performance of the data generation process, we use the fully trained diffusion model to produce new samples. Subsequently, we evaluate the quality of these generated samples. Our results show that scVAEDer generates high-quality samples with excellent fidelity to the original one, showing visual coherence and structural consistency that is important for scRNA-seq data (Fig. 2a). Furthermore, we demonstrate that the quality of data synthesis achieved by the diffusion model is superior to the ones generated by sampling from VAE's prior. Figure 2d provides clear evidence that the structure of the embedding is lost when just using the VAE prior, while the structure of the embedding generated by the diffusion model (Fig. 2c) is similar to the real data embedding (Fig. 2b). In addition, we compute the Total Variation Distance (TVD) between the real data embedding and those generated by sampling from VAE and scVAEDer to quantitatively determine which prior can generate samples closer to the true embedding. The samples produced by scVAEDer are significantly closer to the actual values and more precisely capture the distribution of latent embedding of data in comparison to the VAE model (Fig. 2e).

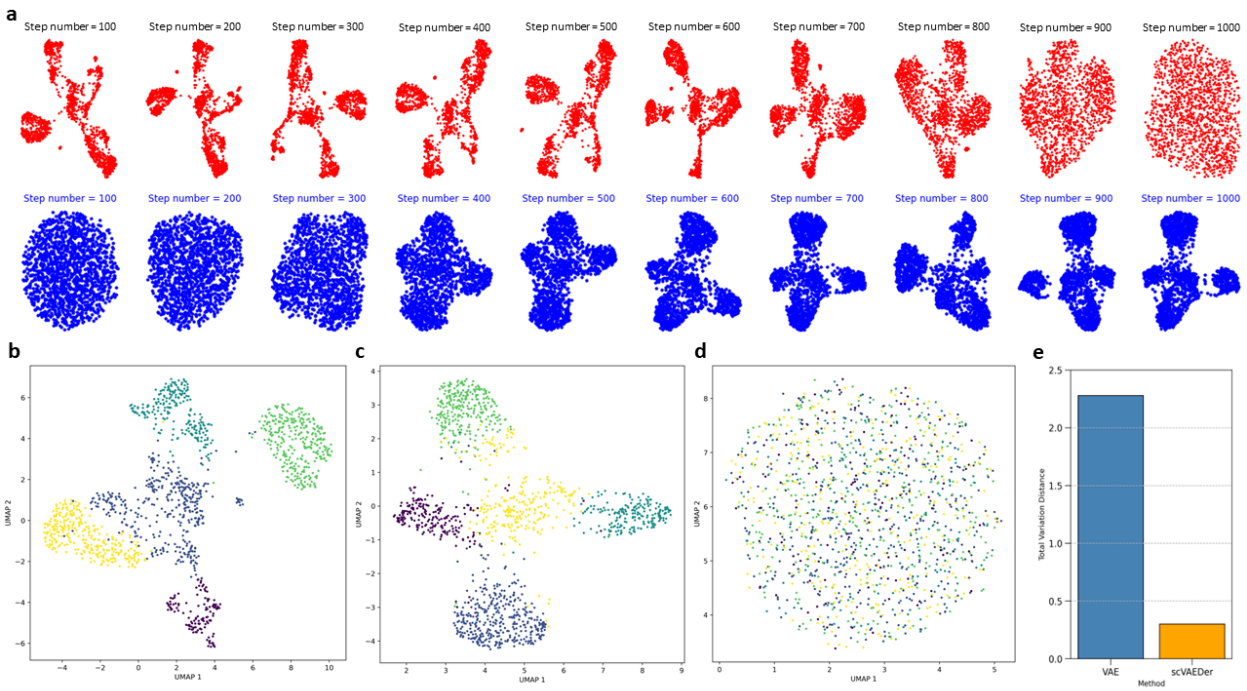


Figure 6-2: scVAEDer accurately learns the latent representation and generates new high-quality scRNA-seq data. a, Red, forward diffusion process with 1000 steps on hematopoiesis in zebrafish as we add noise to the data; blue, reverse process as the model learns how to denoise. b, UMAP visualization of the real data embedding. c, Samples generated from DDM prior. d, samples generated from the VAE. e, Total Variation Distance (TVD) between latent embedding of data and samples generated from the DDM as well as VAE prior distributions.

6.3.3 scVAEDer models the transition between cells

The ability of cells to dedifferentiate and transdifferentiate is crucial in regenerative medicine and disease treatments. Leveraging scVAEDer's access to the entire low-dimensional embeddings, we seek to investigate gene expression changes during the cellular transition. We use hematopoiesis data, specifically focusing on the transition from monocytes to Hematopoietic Stem/Progenitor Cells (HSPC). Monocytes are a central part of the host immune system and play an essential role in immune response and inflammation functions (234). However, the process of deriving pluripotent stem cells from monocytes is complicated and challenging (235). Therefore, we aim to leverage scVAEDer to further investigate this process and predict how the expression of key

genes changes as we interpolate between monocytes to HSPCs. To avoid the "prior hole" problem in VAEs, we interpolate in the prior space of the latent DDM (Methods). Therefore, we first compute the latent embedding for monocytes and HSPCs using the VAE encoder and use DDM forward pass to perfectly map the data to its prior (Figure 3a). As there are no prior holes in DDM latent space, we can safely move along the interpolation path using 2000 equidistant steps between monocytes and HSPCs, and then use the reverse process to achieve the interpolation results in VAE's latent space. These results can then be fed into the VAE's decoder to understand changes in expression patterns as we move along the interpolation trajectory in the gene space. Interestingly, there are no samples generated in the empty region, which suggests that scVAEDer learns the correct complex dynamic of dedifferentiation (red dots in Fig. 3b).

Additionally, we compute the correlation between the cells generated by interpolation and the average gene expression of monocytes and HSPCs. During the initial interpolation stages, the cells exhibit greater similarity to monocytes; but as the process proceeds, they become more similar to HSPCs (Fig. 3c). By investigating the expression pattern of key genes during interpolation, we can gain a deeper understanding of the fundamental mechanisms driving cell dedifferentiation. As can be seen in Figure 3d, there is a notable increase in the expression of HSPC markers in zebrafish, such as txnipa, sox4a, itga2b, and banf1 (236–239), which are critical regulators of stem cell maintenance and self-renewal. There is also a decrease in the expression of monocyte markers, such as cebpb, npsn, lyz, and fcer1gl (240–243), which are essential for the maturation and function of monocytes (Supplementary Figure 1 includes the trend of more genes). This suggests that scVAEDer can capture the shifts in the activity of key genes during dedifferentiation. Moreover, by adjusting the number of diffusion steps before the interpolation process, we can modify the level of detail in the generated cells and interpolate at both fine and coarse granularities (Supplementary Figure 2 and Methods). This finding is in agreement with the results reported by Ho et al (244).

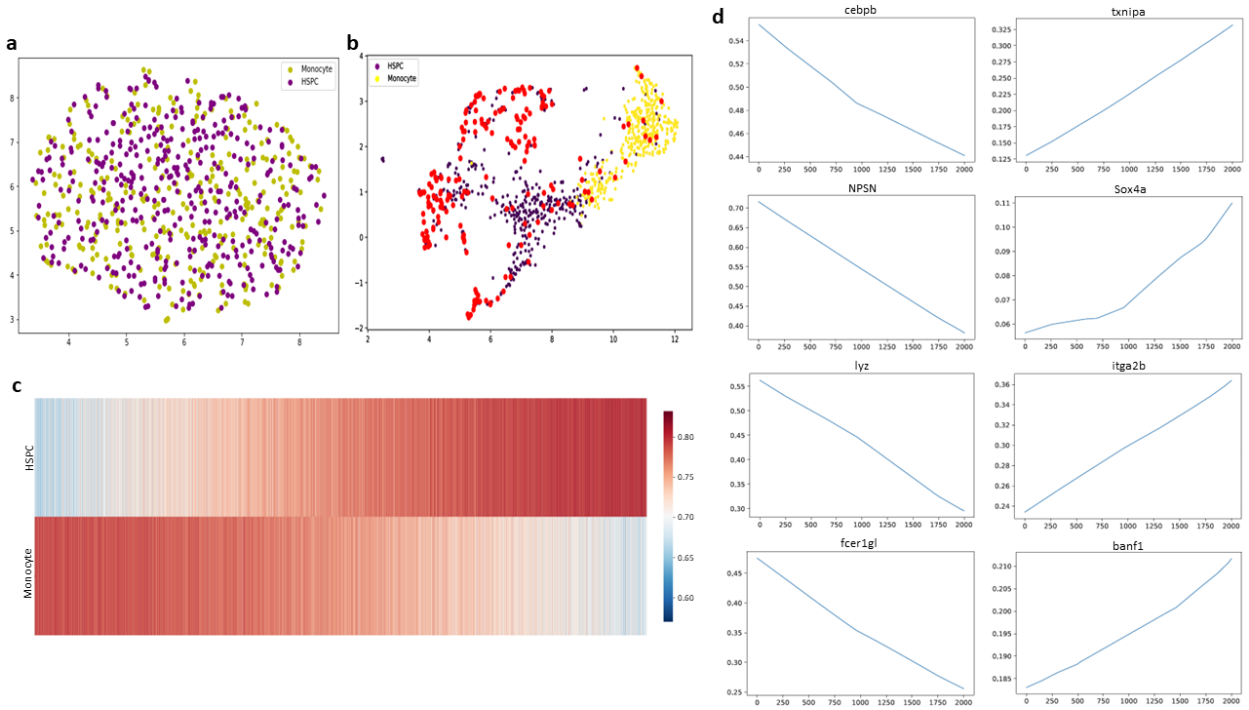


Figure 6-3: scVAEDer can be used to understand cellular dedifferentiation. a, Mapping HSPC and monocyte cells into the latent prior of DDM. b, Using DDM (1000 diffusion steps) and performing latent linear interpolation with 2000 equidistant samples (red dots). The absence of any sample generated in the empty region suggests that the model has learned an accurate embedding. c, Heatmap showing the similarity between gene expression of generated states and the real average expression of HSPCs and monocytes (using 100 DDM steps before interpolation). d, Expression of selected marker genes along the interpolation path.

6.3.4 scVAEDer predicts perturbation responses using feature manipulation

By studying cell responses to specific perturbations, we can understand the underlying mechanisms of a biological process, such as gene regulation and signal transduction, as well as the mechanisms of diseases. Different types of autoencoders have been used in the exploration of the effect of various types of perturbations on cells through vector arithmetic (66,192). scVAEDer latent can be used to predict the refined perturbed gene expression while avoiding VAE issues. To evaluate the performance of our model in predicting perturbation response, scRNA-seq data of human peripheral blood mononuclear cells (PBMCs) stimulated with interferon (IFN-β) (245) is

used for several training of scVAEDer (Fig. 4a). This dataset includes 6359 control and 7217 stimulated cells, with dramatic changes in the transcriptional profiles of immune cells induced by IFN-β stimulation. Following this initial assessment, we examine scVAEDer's performance in predicting perturbation responses across diverse cell types. This evaluation involves comparing scVAEDer against the state-of-the-art methods scGen (66) (based on VAE) and scPreGAN (67) (based on GAN) by calculating the correlation between the predicted and real average gene expression levels for various cell types (Fig. 4b, Methods). Notably, scVAEDer exhibits superior performance in predicting perturbation responses compared to the other models. To further investigate scVAEDer's performance, we conduct an extensive search to identify genes that are shown to be related and important in PBMC from patients with lupus pathogenesis. This includes genes such as ISG15, ISG20, FGL2, and ANXA5 (246–249). We then use these genes to assess scVAEDer's accuracy in predicting their expression distribution following perturbation for Dendritic cells (DC). Here, scVAEDer surpasses scGen in predicting the total distribution of these crucial genes, highlighting its high precision (Fig. 4c). Additional results for more key genes are provided in Supplementary Figure 3a.

Furthermore, to assess the overall robustness of scVAEDer, we incorporate an additional dataset of intestinal epithelial cells after Salmonella infections and re-evaluate scVAEDer's performance against scGen and scPreGAN (Supplementary Figure 3b). This broader evaluation confirms that scVAEDer consistently outperforms other methods across various cell types.

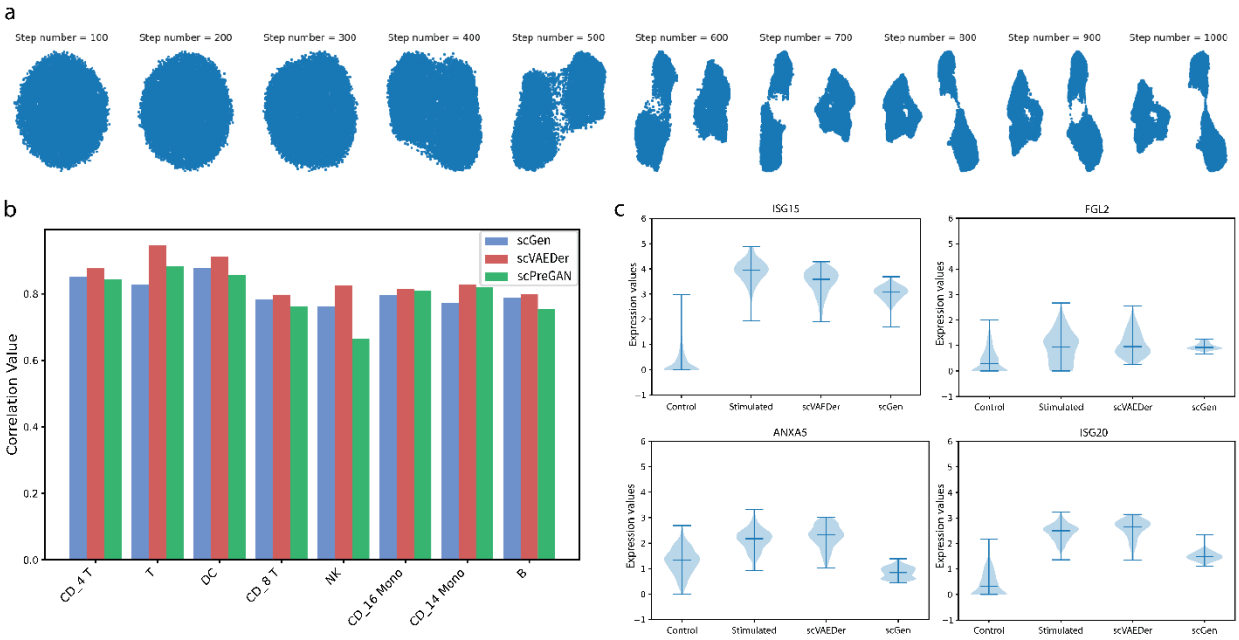


Figure 6-4: scVAEDer is more accurate than the SOTA methods in predicting perturbation responses. a, Data generation using latent prior of DDM with 1000 diffusion steps. b, Comparison of the correlation values of average gene expression between real and predicted cells of various cell types obtained using scGen, scPreGAN, and scVAEDer. c, Violin plots for selected key genes across control, real stimulated, and stimulation predicted by scVAEDer and scGen in Dendritic cells (DC).

6.3.5 Disease progression and master regulators

The study of specific genes known as "master regulators" is of paramount importance in the control of cell fate and differentiation, as they can be targeted to reprogram cells into desired cell types and states. As such, their detection is crucial for gaining insights into disease mechanisms and advancing the development of potential therapies. Here we investigate the changes in gene expression during the transition from failed to reprogrammed cell states and to identify master regulators involved in this process. To achieve this, we employ an approach that involves interpolating between failed and reprogrammed cell states using scVAEDer. Reprogramming data which has 18803 cells, and 28001 genes is used to train scVAEDer after the preprocessing steps. We first validate that scVAEDer is able to accurately generate high quality data from Gaussian noise (Fig. 5a). Then using the trained model, we obtain the latent embeddings of reprogrammed

and failed cells. The UMAP visualization of the cell states in the VAE space can be seen in Figure 5b. We use the average gene expression of failed and reprogrammed cells and interpolate in scVAEDer's diffusion prior. Then the reverse process maps the interpolation results back into VAE's space (Fig. 5c). The correlation values are computed between each interpolation step and the average gene expression of reprogrammed and failed cells. As we move along the interpolation path, the gene expression becomes more similar to the reprogrammed state and less like the failed state (Fig. 5d). We are also able to control the level of granularity during the interpolation process by adjusting the number of steps of DDM before interpolation (Methods). Additionally, to understand marker genes during this transitional process, we calculate the absolute difference between the gene expression at time t=0 and t=3000 (corresponds to 20% of the total 15,000 as it corresponds to the global structural changes in the data). The genes with the highest absolute scores are displayed in Figure. 5e. This allows us to identify both known and previously unknown gene markers linked to the reprogramming of fibroblasts (250–253).

Next, we use scVAEDer to detect the master regulators involved in this process, by computing the gene velocities, given by the rate of changes in gene expression during interpolation (Fig. 5f). We believe gene velocities are informative as they allow us to identify crucial genes that should undergo rapid changes in order to play their central role in a specific biological process. By ranking genes based on their absolute velocity values, we detect key genes in this process, including Wnt4, Ptma, and FoxA1, which have been shown to be important in reprogramming and regeneration processes (List of the top 15 genes and their corresponding functions can be found in Supplementary Table 1). As a biological system needs different genes and pathways to be activated in order to succeed in different parts of the biological process, our approach makes it possible to detect master regulators in each step. Furthermore, we perform gene set enrichment analysis by selecting the top 10 genes with the highest velocities at each step of the interpolation process, with a focus on the first 40 steps (400 genes in total). Pathways related to morphogenesis, developmental process, and differentiation are significantly enriched, which demonstrates the ability of scVAEDer to detect related master regulators and pathways of this specific biological process (Fig. 5g).

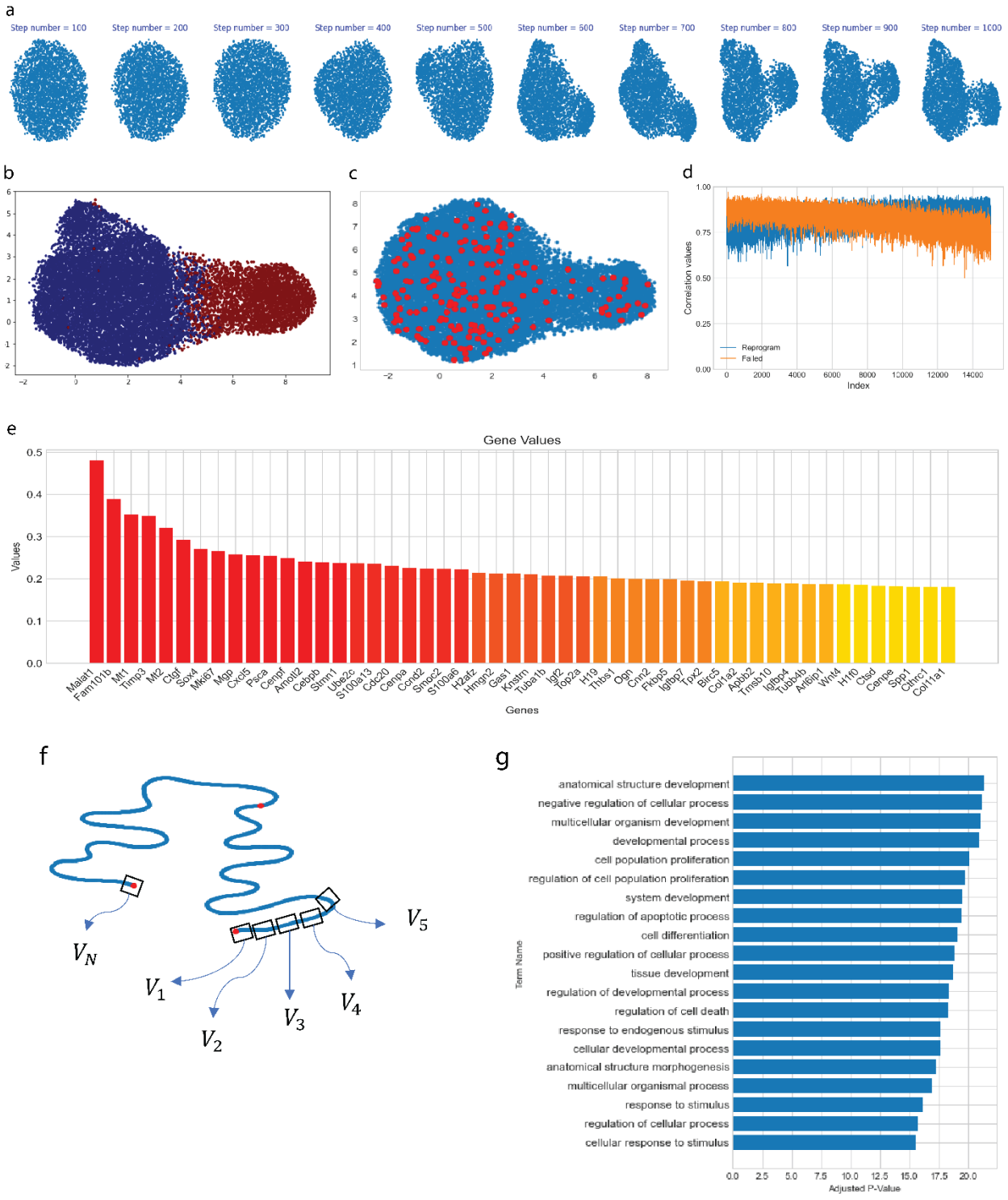


Figure 6-5: scVAEDer accurately detects master regulators during cellular reprogramming. a, The reverse process of DDM allows the generation of high-quality reprogramming samples from random

Gaussian noise (the quality of generated samples is critical for downstream analysis). b, UMAP visualization of the data latent embedding, colored based on their state (red: failed, blue: reprogrammed). c, Data generated by interpolating between reprogrammed and failed states, represented by red dots. Remarkably, none of the interpolated samples are found outside the real representation of data. d, Correlation values between the new interpolated samples and the average gene expression of reprogrammed and failed cells, which demonstrates the shift in gene expression between failed and reprogrammed states. e, Ranking of genes based on high expression differences between t=0 and t=3000. f, Illustration showing the process of computing gene velocities along the interpolation path to detect master regulators. g, Gene set enrichment analysis using 400 genes with the highest velocities (fast responders), which reveals pathways that are crucial during the cell reprogramming process.

6.4 Discussion

Analyzing scRNA-seq data is a challenging task which involves identifying the appropriate underlying data representation. The accuracy of this low-dimensional embedding is crucial for many downstream tasks, including cell clustering, perturbation response prediction, gene expression analysis, and visualization. By learning the correct representation of data, we are also able to generate novel samples, predict the effect of different perturbations on cells, and detect the master regulators of a GRN or marker genes of a biological process. To produce high-quality samples, the generative model should also be able to take into account and control both high-level and low-level variations between different cell types.

Here we propose scVAEDer, a novel method that unifies the power of VAE and DDM. scVAEDer can generate realistic gene expression data using the latent representations of VAE for high-level and DDM for low-level variations. scVAEDer has access to the lower dimension of data which enables us to predict the effect of different unseen perturbations on a variety of cell types. Using several datasets we show that scVAEDer outperforms the SOTA methods in predicting the effect of perturbations. Additionally, the DDM prior of our method can be used for interpolation between different cell types, allowing us to generate new cells along the differentiation trajectory and identify highly expressed genes in each step. We also demonstrate that scVAEDer's ability to compute gene velocity (fast responder genes) along the interpolation path can be useful in cellular reprogramming. Previous studies have demonstrated that master regulators are not typically the genes with the highest differential expression (254). To address

this, we focus on genes with high velocities instead and are able to discover both novel and previously identified master regulators and pathways that have been shown to play a vital role in cell fate and cellular reprogramming. It would be beneficial to conduct additional experiments to determine whether these identified key genes are specific to the biological process studied here or if they hold importance across multiple distinct decision-making processes. Also, investigating the relationship between high expression values of these genes and their status as master regulators could provide deeper insights into their regulatory functions in various biological contexts. Unlike some generative models that rely mainly on autoencoders, scVAEDer leverages the VAE primarily for its efficient representation learning capabilities and DDM as the generative model. This approach enables scVAEDer to excel in downstream tasks without the limitations typically associated with autoencoder-based generative models such as prior hole.

Future versions of scVAEDer can explore novel approaches to enhance its performance. One potential avenue lies in reformulating the DDM component as a continuous process. This can be achieved through the implementation of a stochastic diffusion equation (SDE) or a deterministic approach like probability flow ordinary differential equations (ODEs). Moreover, scVAEDer's architecture offers an opportunity for integration with other representation learning methods beyond VAE. Incorporating causal representation learning also holds promise, as it has the potential to not only capture the data's intrinsic structure but also unveil the underlying causal relationships between key variables. These modifications can improve scVAEDer's performance in future versions.

6.5 Methods

6.5.1 Variational autoencoder (VAE)

VAE is a generative model that is widely used in different deep learning applications. It can generate new samples by capturing the underlying data distribution. It is based on the idea of maximizing a lower bound of the log-likelihood of the data. This is achieved by minimizing the

KL divergence between the learned distribution and the true distribution. VAE models typically consist of two parts: an encoder that transfers input data to a latent space and a decoder network that maps latent space back to the original data space. The objective function can be written as:

$$L(\theta, \phi; x) = -D_{KL}(q_\phi(z|x)||p_\theta(z)) + E_{q_\phi(Z|X)}[logp_\theta(x|z)]$$

where θ and ϕ are the parameters of the decoder and encoder networks, x is the input data, z is the latent variable, $q_\phi(z|x)$ is the approximate posterior distribution over the latent variables, and $p_\theta(x|z)$ is the conditional distribution over the input data given the latent variables. In the equation above the first term is the KL divergence between the approximate posterior and the prior distribution $p_\theta(z)$. That encourages the learned distribution to be as close as possible to the prior distribution. The second term is reconstruction loss, which measures how well the model can reconstruct the input data from the latent embedding. By minimizing this objective function, the VAE model can learn to generate new samples similar to the input data.

6.5.2 Deep Diffusion Model (DDM)

Diffusion models belong to a category of generative models that have gained popularity in recent years. They can generate novel data from complex distributions, with higher quality compared to the existing models in this domain. To capture the underlying data distribution, DDMs follow a two-step process: a forward diffusion process and a reverse denoising process, working together in a coordinated manner. The forward process gradually destroys the structure of the data by adding noise, while the reverse process learns to recover the original data from the Gaussian noisy input.

It should be emphasized that the forward noising process is not trainable, but the reverse denoising process is parametrized and learned during training. In these processes, a first-order Markov chain with Gaussian transitions for the forward and backward processes is mostly used. The forward process can be written as:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)=\mathcal{N}\left(\mathbf{x}_t ; \sqrt{1-\beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$$

$$q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)=\prod_{t=1}^T q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$$

By choosing the correct values of the noise scheduler, we can assume $q(x_t \mid x_0)$ is isotropic Gaussian. By using the reparameterization trick, we get:

$$q\left(x_t \mid x_0\right)=\mathcal{N}\left(x_t ; \sqrt{\bar{\alpha}_t} x_0,\left(1-\bar{\alpha}_t\right) I\right)$$

where:

$$\alpha_t=1-\beta_t,$$

$$\bar{\alpha}_t=\prod_{s=0}^t \alpha_s$$

It's worth mentioning that besides the Gaussian distribution, there are other distributions such as the gamma distribution that can be used. These distributions share a common property, which is the fact that the sum of two or more distributions results in the same distribution as well (255).

The reverse process can be formulated as:

$$p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_{\boldsymbol{t}}\right)=\mathcal{N}\left(\boldsymbol{x}_{t-1} ; \boldsymbol{\mu}_\theta\left(\boldsymbol{x}_{\boldsymbol{t}}, t\right), \boldsymbol{\Sigma}_\theta\left(\boldsymbol{x}_{\boldsymbol{t}}, t\right)\right)$$

$$p_\theta\left(x_{0: T}\right)=p\left(x_T\right) \prod_{t=1}^T p_\theta\left(x_{t-1} \mid x_t\right)$$

which starts from a random Gaussian $x_T \coloneqq \mathcal{N}(x_T, 0, I)$, and in each step, the model learns $p_\theta(x_{0:T})$. To train the reverse process, it has been shown (244) that the loss function below works well as it leads to a better result:

$$L_{\text {simple }}(\theta):=\mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}, t\right)\right\|^2\right]$$

We can modify the above equation for our case as we train the diffusion model on the latent embedding of VAE. For the sake of simplicity, we ignore some of the weight terms (244). As a result, we express the DDM part of the scVAEDer objective as follows:

$$L_{\text{modified}}(\theta) := \mathbb{E}_{t,\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}),\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right]$$

where z is the encoded value of x using the VAE's encoder $q_\phi\,(z|x)$.

Once the training process is complete, it is possible to sample from the DDM prior iteratively using the sampling method below:

$$\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathbf{z}_t, t\right)\right) + \sigma_t\mathbf{J}$$

$$\mathbf{J}_T \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$$

6.5.3 Interpolation

A number of interpolation settings have been considered in previous generative models to understand the trajectory between two states (232,233,256). Here we have implemented two distinct interpolation approaches:

1. By leveraging the VAE, we encode x_1 and x_2 (gene values) to obtain z_1 and z_2 respectively in the VAE's latent embedding. Afterward, by using the diffusion forward process, we can obtain q_1 and q_2 in the prior space of DDMs. The interpolation can be performed in that space using $\hat{q}_{12}(\lambda) = (1-\lambda)\hat{q}_1 + \lambda\hat{q}_2$, where λ is between [0,1]. Then the reverse process map $\hat{q}_{12}$ back to the VAE's latent space and then by using the decoder of the VAE we can compute the interpolation values in the gene space., where λ is between [0,1]. Then the reverse process map $\hat{q}_{12}$ back to the VAE's latent space and then by using the decoder of the VAE we can compute the interpolation values in the gene space.

2. Following a similar approach as the previous part, we compute z_1 and z_2 using the VAE's encoder, and subsequently conduct interpolation using $\hat{z}_{12}(\lambda) = (1-\lambda)\hat{z}_1 + \lambda\hat{z}_2$. Then by adding Gaussian noise to $\hat{z}_{12}$https://www.codecogs.com/eqnedit.php?latex=\hat{\mathbf{z}}_{1 2}, and performing the reverse diffusion process, we are able to generate better quality results than those produced solely by VAE. As in the previous section, the output of the reverse process is fed to the VAE's decoder to map the embedding values from the VAE to the gene space.

The choice of interpolation approach is determined by the objective and the computational resources available. In this paper, we use the first approach for interpolation tasks as it provides greater flexibility and control over the number of diffusion steps and the amount of noise added. This level of control allows for a range of interpolation options, from coarse to fine adjustments, which can be advantageous in various scenarios. Previous studies have effectively demonstrated the benefits of this approach (244). Alternatively, for users with limited computational resources, the second method offers a more straightforward and computationally efficient solution.

Also, to achieve more consistent results, different methods can be used, such as an implicit instead of a probabilistic method (DDIM over DDPM) or the deterministic probability flow ODEs (256,257). Here we fix the stochasticity for all generated samples during interpolation to increase smoothness between samples. It is also essential to select the appropriate number of diffusion steps for each unique task, as this determines whether interpolation takes place at more refined or coarse levels of detail. When the diffusion step is zero, interpolation takes place in the VAE's latent space. As we increase the number of diffusion steps, the initial interpolation information gradually diminishes, resulting in interpolations that are less detailed, coarser, and more varied (Supplementary Figure 2).

6.5.4 Model architecture and training process

scVAEDer is trained on 4 Nvidia Tesla M60 machines with 32 GB GPUs, 448 GB system RAM, and 48 vCPUs. The training process involves two stages (detailed in Supplementary Tables. 2 and 3). The initial stage trains a VAE model, followed by the second stage where the diffusion model uses the latent embedding learned from the trained VAE. We choose a two-stage training process due to the higher memory and resource requirements of end-to-end training. Also, previous research has shown that end-to-end training can reduce both the quality of the generated data and the performance of the model (232). To ensure that the model can accurately learn the data embedding and generate high-quality samples, it is essential to select the suitable parameters, such

as the VAE and DDM network architectures, the number of diffusion steps, and the level of noise added during the forward process (Supplementary Tables 2 and 3). Therefore, a random hyperparameter search is performed on these parameters to minimize the loss functions, ultimately leading to better model performance. Notably, high-quality data generation in diffusion models depends on a crucial factor: the interplay between the number of steps and layers within the DDMs. While intuitively increasing these parameters seem beneficial, a critical trade-off exists. On the one hand, a larger number of steps facilitates a more gradual and controlled noise manipulation process. This finer control can lead to potentially higher fidelity in the generated data. However, this benefit comes at a significant cost. A substantial increase in computational resources is required as the number of steps and layers grows. Furthermore, excessively high step counts can lead to overfitting, resulting in unrealistic or overly smooth outputs that lack the natural characteristics of real data. Therefore, achieving a balance between computational complexity and quality of data generation is crucial. This can be achieved based on the available computational resources and checking the performance of the model with different hyperparameters to avoid overfitting and ensure optimal data generation. We observed that identifying the optimal DDM parameters, such as the number of diffusion steps, plays a key role in increasing model performance, as it directly influences the quality of generated data.

6.5.5 Exponential Moving Average (EMA)

EMA can be used to improve the performance of the training process. Unlike the conventional methods which directly adjust model weights, EMA calculates a running average of the model's weight parameters over the training process. This can lead to a more stable set of parameters. In our case, the weight parameters for the latent DDMs are tracked using EMA with a decay value of 0.999.

6.5.6 Variance Scheduler

The noise schedules β_1 to β_T are chosen such that the chain approximately converges to a standard Gaussian distribution after T steps. The variance parameter can be fixed to a constant or chosen as a schedule over the timesteps. To improve the performance, we incorporated three distinct variance schedulers, such as linear, quadratic, and sigmoidal. The user has the option to select any of these schedulers. The beta values are between $1e^{-5}$ and $0.5e^{-2}$ (Supplementary Figure 4).

6.5.7 For visualization, reduced dimension data

Clusters are visualized using the Python package "UMAP".

6.5.8 Data preprocessing

The scRNA-seq gene expression data are centered and log-normalized. 2000 highly variable genes were chosen. We also performed batch normalization on the latent layer of VAE before training DDM as we found it improves the convergence speed and accuracy of the model.

6.5.9 Comparison with other methods

In comparison with alternative methods, we obtain the scGen python package from the GitHub repository https://github.com/theislab/scgen and configure the requisite experimental environment accordingly. Subsequent modifications are applied to the scGen architecture to reduce computational complexity. To ensure a consistent evaluation, we maintain identical neural network architecture and hyperparameters for both scVAEDer and scGen. For scPreGAN, we utilize the python package available at https://github.com/JaneJiayiDong/scPreGAN.git and follow the default parameters outlined in their publication, with the exception of adjusting the model to accommodate 256 nodes instead of 512 and 128 nodes instead of 256 to facilitate a fair comparison and alleviate computational cost. We also evaluated scVAEDer's performance against VEGA(46). We excluded VEGA's performance from the comparison since its correlation values (ranging from

0.53 to 0.6 across cell types) were lower than those of other methods, such as scGen and scPreGAN. VEGA places a stronger emphasis on interpretability within its latent layer to provide insights into diverse biological experiments. The VEGA Python package was obtained from its GitHub repository: https://github.com/LucasESBS/vega.

6.5.10 Data availability

All datasets analyzed in the current study are publicly available and can be downloaded from their public repository. The zebrafish hematopoiesis (78) raw data can be found under the accession number E-MTAB-5530 on ArrayExpress. The raw PBMC of lupus patients (245) and Salmonella infection (258) datasets are available on the NCBI's Gene Expression Omnibus under accession numbers GSE96583 and GSE92332. The preprocessed ones can be downloaded from:https://github.com/theislab/scgen-reproducibility/blob/master/code/DataDownloader.py. Raw reprogramming data (259) is available from the Gene Expression Omnibus under accession number GSE99915. The preprocessed version can be downloaded from https://cospar.readthedocs.io/.

6.5.11 Code availability

Our Python implementation along with the versions of the packages used can be found at: https://github.com/MehrshadSD/scVAEDer

6.5.12 Competing interests

The authors declare that they have no competing interests.

Chapter 7
Deep Lineage: Single-Cell Lineage Tracing and Fate Inference Using Deep Learning

This chapter highlights Deep Lineage, a novel deep-learning method for analyzing time-series scRNA-seq data with matched lineage-tracing data. Our method accurately predicts early cell fate biases and gene expression profiles at different time points within a clone, surpassing current state-of-the-art methods in fate prediction accuracy. Also, through in silico perturbations in cellular reprogramming and hematopoiesis data, we show that Deep Lineage accurately models dynamic multicellular responses while identifying key genes and pathways associated with cell fate determination. This research was conducted in collaboration with Allen Zhang (an undergrad student in our lab) and Prof. Gary Bader. The work titled *Deep Lineage: Single-Cell Lineage Tracing and Fate Inference Using Deep Learning* is currently under review at *Cell Systems* (5). The author list on the publication is below:

Mehrshad Sadria[1], Allen Zhang[1], Gary D. Bader[2,3,4,5,6,7]
[1] Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada.
[2] Department of Molecular Genetics, University of Toronto, Toronto, Ontario, Canada.
[3] The Donnelly Centre, University of Toronto, Toronto, Ontario, Canada.
[4] Department of Computer Science, University of Toronto, Toronto, Ontario, Canada.
[5] The Lunenfeld-Tanenbaum Research Institute, Sinai Health System, Toronto, Ontario, Canada.
[6] Princess Margaret Cancer Centre, University Health Network, Toronto, Ontario, Canada.
[7] CIFAR Multiscale Human Program, CIFAR, Toronto, Ontario, Canada

7.1 Overview

Recent advances in single-cell RNA-sequencing and lineage tracing techniques have provided valuable insights into the temporal changes in gene expression during development, tumour progression, and disease onset. However, there are few computational methods available to analyze this information to help understand multicellular dynamics. We introduce Deep Lineage, a novel deep-learning method for analyzing time-series single-cell RNA sequencing with matched lineage-tracing data. Our method accurately predicts early cell fate biases and gene expression

profiles at different time points within a clone, surpassing current state-of-the-art methods in fate prediction accuracy. Additionally, through in silico perturbations in cellular reprogramming and hematopoiesis development data, we show that Deep Lineage can accurately model dynamic multicellular responses while identifying key genes and pathways associated with cell fate determination.

7.2 Introduction

Developmental biology aims to understand the processes by which single cells divide and differentiate into diverse cell types in different tissues (1). Single-cell RNA-sequencing (scRNA-seq) is a powerful tool for investigating developmental landscapes by profiling gene expression in individual cells at different stages (2). While scRNA-seq is valuable for investigating genetic mechanisms in cell differentiation, its destructive nature restricts it to capturing only a single time point snapshots of a developing system. This constraint poses challenges in tracing lineage relationships and tracking cell trajectories over time (3).

To overcome this limitation, a range of methods have been developed to study temporal relationships between cells. One class of computational methods, trajectory inference, predicts the order of cells along a developmental path based on the idea that cells that are developmentally related will exhibit similar gene expression profiles (4). However, trajectory inference methods are limited in their capacity to capture the complexity of developmental processes, such as lineage branching and cellular plasticity, and can be negatively affected by technical factors and noisy data (5,6). New experimental lineage tracing technologies that use CRISPR/Cas9 genome editing to record cell division events provide another avenue for studying cell development (7). Using data produced by these technologies, computational methods have been developed to infer developmental trajectories by considering both cell lineage and cell state information from multiple time points (8–11). Integration of these two types of information has made predictions more accurate (12). However, despite these advances, the existing methods still can not accurately predict cell fate (13).

Here, we present Deep Lineage, a novel computational model that integrates lineage tracing and scRNA-seq data to learn cellular trajectories, enabling the prediction of gene expression and cell type at any given time point along a trajectory. Deep Lineage uses Long Short-Term Memory (LSTM), Bi-directional Long Short-Term Memory (Bi-LSTM), or Gated Recurrent Units (GRUs) to model complex sequential dependencies and temporal dynamics of a cellular trajectory. An autoencoder-learned embedding captures essential features of the data to simplify input to the LSTM. To assess method performance, we analyzed data sets containing both lineage tracing and scRNA-seq data. Our results demonstrate that Deep Lineage surpasses state-of-the-art methods in detecting early fate bias within a developmental process and can accurately predict gene expression profiles at held-out time points within a clone during a biological process. Furthermore, we show Deep Lineage's ability to identify important genes and pathways involved in cell fate decision-making during a developmental process and accurately simulate genetic perturbation effects on gene expression of cell progenies. Such insights can aid in designing experiments to reprogram cells into specific cell types and potentially translate to clinical applications.

7.3 Results

7.3.1 Model Overview

Deep Lineage uses lineage tracing and multi-timepoint scRNA-seq data to learn a robust model of a cellular trajectory such that gene expression and cell type information at different time points within that trajectory can be predicted. It can be used to predict single-cell time points that have not been measured in a clone, including predicting cell fate at future time points from early time points (early cell fate bias). An autoencoder-learned embedding captures essential data features and a LSTM, Bi-LSTM (260) or GRU (261) is used to support cell type classification and gene expression regression tasks (Fig. 1). Deep Lineage treats cells and their progenies within a clone as interconnected entities. We conceptualize cellular relationships through "clones," which represent cells organized within a common lineage. These clones illustrate how cells are related

over time, similar to how components of a system are linked through shared functions or characteristics.

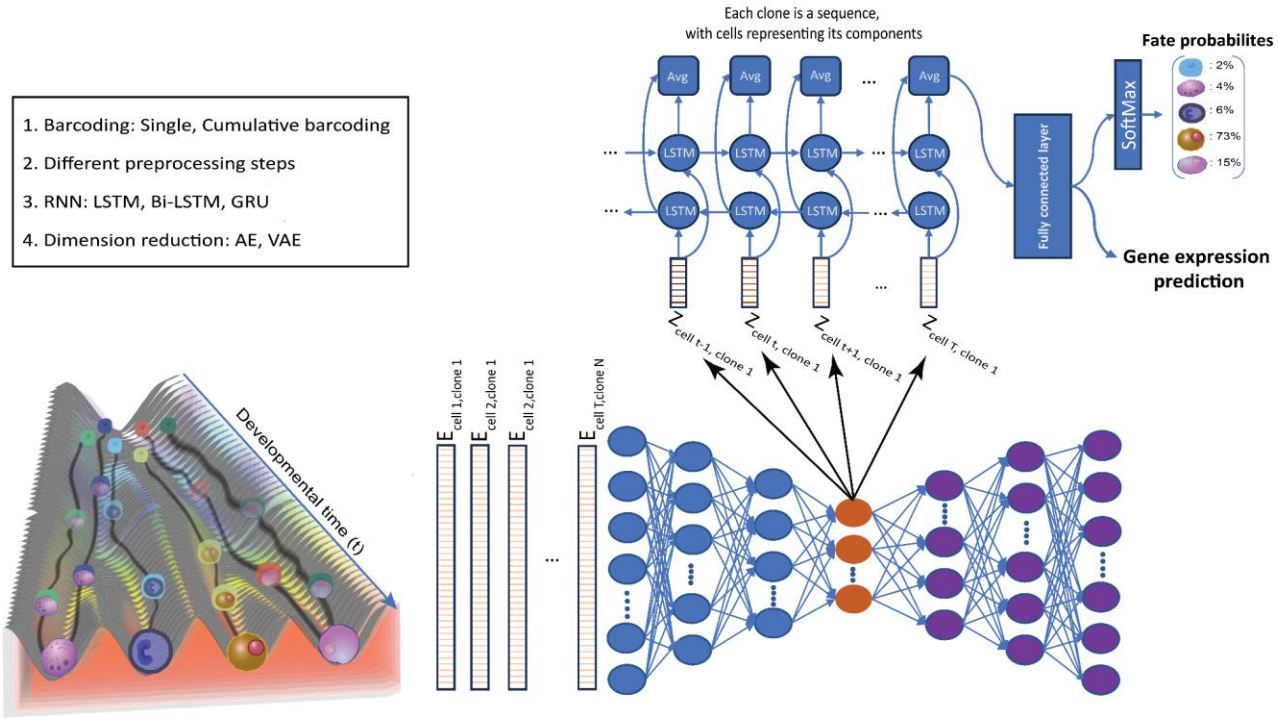


Figure 7-1: **Predicting Gene Expression and Early Cell Fate Bias via Combined scRNA-seq and Lineage Tracing.** A visual depiction of the Deep Lineage is presented. On the left, the Waddington landscape illustrates the developmental trajectory, showing how cells within each clone (represented by different colours) differentiate into distinct mature cell types originating from stem cells. On the right, by combining scRNA-seq data and lineage tracing information the gene expression profiles of cells within each clone are used as input for the autoencoder. The resulting latent embeddings of cells in a clone are subsequently used as inputs for the LSTM, Bi-LSTM, or the GRU model. This integrated architecture enables the accurate prediction of both early cell fate bias and gene expression profile of unseen days within a clone. Notably, Deep Lineage offers the flexibility to adapt different preprocessing steps, diverse barcoding techniques, various dimensionality reduction methods, and a range of various deep learning models (left black box).

We train the model, including an extensive hyperparameter search and data preprocessing to optimize the model's performance (lowest loss for reconstruction and classification), on predicting held-out time points. We then train the optimized model with all data. To use the model, single-cell gene expression profiles from one or more time points are encoded in the learned

autoencoder's low-dimensional representation. This encoding and the cell's clone label are fed into the LSTM or GRU network to predict cell fate bias for input cells, by examining cell types for each clone at a predicted future time point (Fig. 1). The model also predicts gene expression profiles for cells at different stages of the trajectory that can be used to impute missing time points. Deep Lineage can incorporate lineage tracing data using either single barcoding or cumulative barcoding methods (Supplementary Figure 1). Lineage tracing with cumulative barcoding can significantly boost the accuracy of the model compared to single barcoding resulting in more accurate predictions for both early cell fate bias and gene expression patterns at different stages of development (see below). By applying the SHapley Additive exPlanations (SHAP) method (149) to the trained model, Deep Lineage can accurately identify key genes and regulatory mechanisms linked to different fate outcomes at different developmental stages.

7.3.2 Deep Lineage accurately predicts gene expression and early cell fate bias in mouse hematopoiesis

We applied Deep Lineage to the published lineage tracing data of mouse hematopoiesis, which consists of gene expression of 130,887 cells, and 25,289 genes (81). These clones are sampled from three time points, on days 2, 4, and 6. We focused on the main cell lineage in this data set, specifically neutrophils, monocytes, and their progenitors (Fig. 2a). We trained models to predict the gene expression profile of held-out cells at two-time points within a clone (predict day 4 by training with days 2 and 6 and predict day 6 by training with days 2 and 4)(Fig. 2b), including optimizing preprocessing steps and hyperparameters (see methods). We calculated the correlation between the gene expression profile of the predicted and held-out cells. The results demonstrated a high degree of similarity (R = 0.81 for both experiments), indicating that Deep Lineage accurately predicts the gene expression of cells unseen by the model on both day 4 (trained on days 2 and 6) and day 6 (trained on days 2 and 4) (Fig. 2c and Supplementary Figure 3a). Also, the correlation between predicted gene expression and randomly sampled cells was lower, showing the specificity and accuracy of the model's predictions for the targeted cell types (Fig. 2c). We also observe a strong positive correlation between the average predicted expression of all genes and

actual expression in both monocytes and neutrophils on day 4 (Fig. 2d,e). Deep Lineage excels at predicting average gene expression and the full distribution of diverse genes, even for cells within a clone that were previously unseen by the model (Fig. 2f,g).

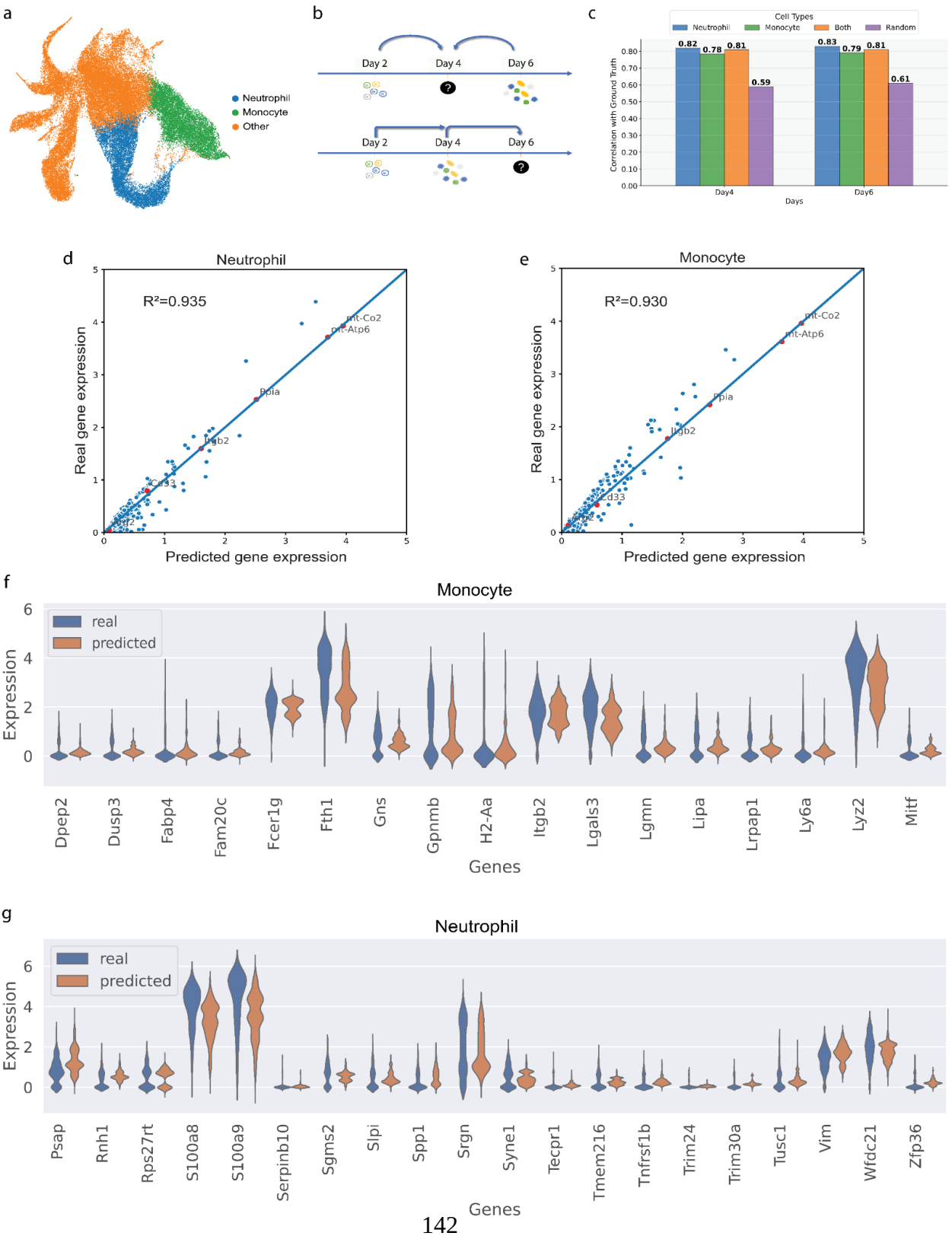

a
Neutrophil
Monocyte
Other
b
Day 2
Day 4
Day 6
Day 2
Day 4
Day 6
c
Cell Types
Neutrophil
Monocyte
Both
Random
Correlation with Ground Truth
0.82
0.78
0.81
0.59
0.83
0.79
0.81
0.61
Day4
Day6
Days
d
Neutrophil
R²=0.935
mt-Co2
mt-Atp6
Ppia
Itgb2
Cd33
Real gene expression
Predicted gene expression
e
Monocyte
R²=0.930
mt-Co2
mt-Atp6
Ppia
Itgb2
Cd33
Real gene expression
Predicted gene expression
f
Monocyte
real
predicted
Expression
Dpep2
Dusp3
Fabp4
Fam20c
Fcer1g
Fth1
Gns
Gpnmb
H2-Aa
Itgb2
Lgals3
Lgmn
Lipa
Lrpap1
Ly6a
Lyz2
Mitf
Genes
g
Neutrophil
real
predicted
Expression
Psap
Rnh1
Rps27rt
S100a8
S100a9
Serpinb10
Sgms2
Slpi
Spp1
Srgn
Syne1
Tecpr1
Tmem216
Tnfrsf1b
Trim24
Trim30a
Tusc1
Vim
Wfdc21
Zfp36
Genes

Figure 7-2: Deep Lineage accurately predicts single-cell gene expression of unseen cells in hematopoiesis a, UMAP plot of the hematopoiesis data from Weinreb et al. Each point represents an individual cell, color-coded by cell type. b, Schematic representation of the gene expression prediction process using two regression models: one for predicting day 4 gene expression (trained on data from days 2 and 6), and the other for predicting day 6 gene expression (trained on data from days 2 and 4) within a clone. c, Correlation of gene expression between Deep Lineage predictions and cells excluded during training, categorized by cell types and time points. d,e, Comparison of average gene expression values of 2000 genes between Deep Lineage predicted and real cells for Monocytes and Neutrophils. Gene expression at day 4 was averaged across predicted cells per clone, and then correlation was computed with actual data for each gene ($\boldsymbol{R^2}$ indicates the squared Pearson correlation coefficient between the predicted and ground truth values). f,g, Violin plots show gene expression distributions between predicted and real cells for randomly selected genes both Monocytes and Neutrophils.

To investigate early cell fate bias (Fig. 3a) in hematopoiesis, we trained a new bidirectional LSTM model using the cell type classification task and used it to predict late time point data from an early time point. Using just the gene expression of cells on day 2 of a clone, with the masking of data from days 4 and 6, the model attains a predictive accuracy of 76.8% in determining cell fate on day 6 (Fig. 3b). Providing gene expression of cells on days 2 and 4 of a clone improves the performance slightly and the model predicts the clonal fate on day 6 with 78.5% accuracy. If the model has access to the gene expression data from cells of all three days within a clone, it can almost perfectly predict the fate of the clone (99.2%) (Fig. 3b). This classification accuracy pertains to identifying the clone itself rather than predicting its fate. Subsequently, we generate the area under the curve (AUC) for cell fate prediction of this model by using gene expression of cells on days 2 and 4, and all 3 days (Fig. 3c). Thus, by using the trained classifier, we can predict the early progenitor bias from an early time point, accurately differentiating between monocyte and neutrophil outcomes. The model performance improves when trained with additional time points. Furthermore, to investigate the importance of lineage tracing and clonal information, we conducted an ablation study where we performed random sampling of cells at each time point, deliberately ignoring clonal data. For the experiment where we trained on day 2 and predicted outcomes for day 6, we constructed trajectories by focusing solely on the day of the cell and its final fate outcome while disregarding the clonal information. The results indicate that neglecting lineage tracing information substantially reduces Deep Lineage's prediction accuracy from 77% to 52%. Adjusting the model's architecture (LSTM vs. GRU, hyperparameter changes) results in minimal change in

accuracy (from 50% to 52%), highlighting the important role of clonal structure in fate prediction (Supplementary Figure 5a and Supplementary Tables 4).

Next, in a thorough comparison, we assessed our method alongside the state-of-the-art models: Cospar (262), Super OT (263), WOT (264), and GAN-Based OT (265), in predicting the fate of held-out clones during training. Our results indicate that Deep Lineage outperforms all other existing methods, demonstrating superior accuracy in predicting early fate bias (Fig. 3d). Specifically, the comparative analysis revealed that Deep Lineage achieved a high accuracy rate of 77%. In contrast, other methods such as Cospar attained a rate of 65%, Super OT exhibited a slightly higher accuracy at 67%, WOT maintained a similar accuracy of 65%, and GAN-Based OT achieved 62%.

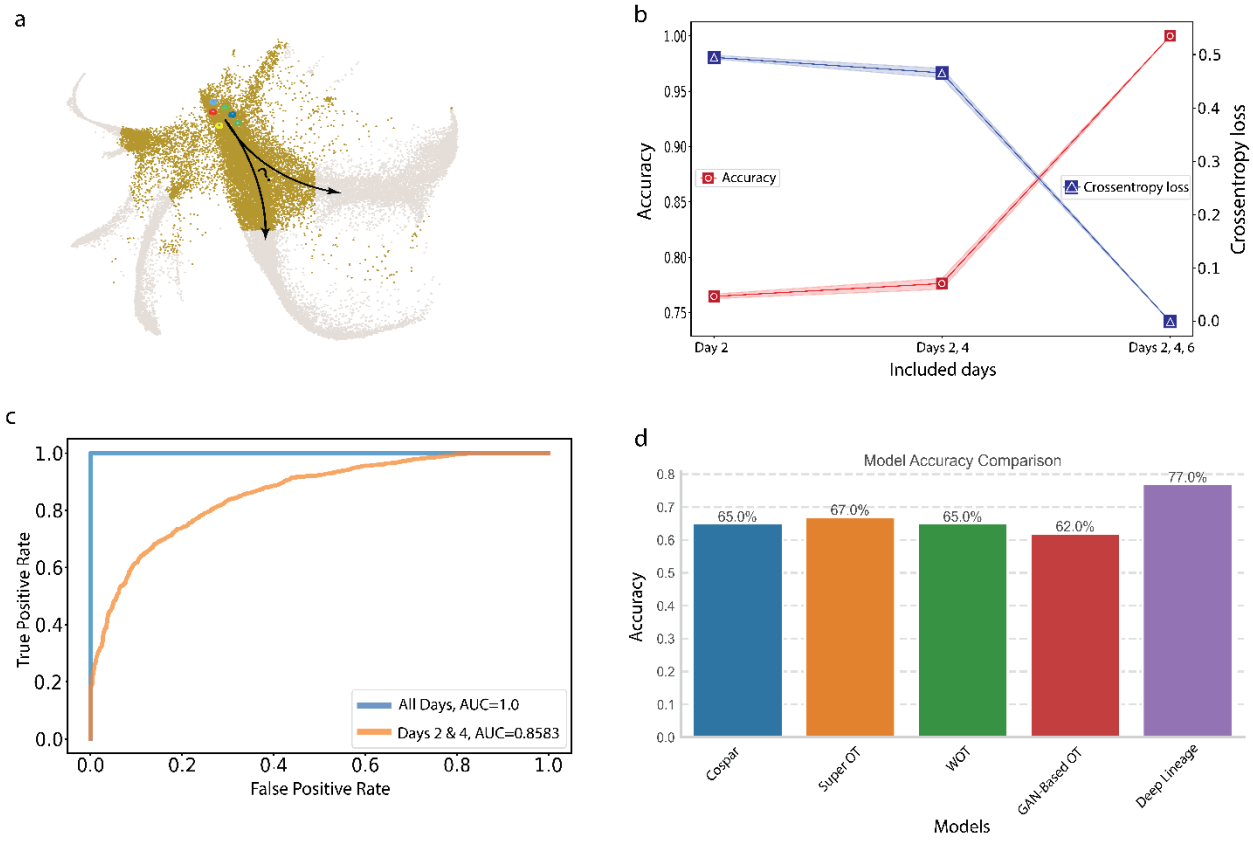


Figure 7-3: Exploring progenitor bias prediction in hematopoiesis with Deep lineage and comparison to state-of-the-art methods. a, Schematic UMAP diagram illustrating the trajectory of stem cell differentiation and the possible cellular outcomes with cell colors indicating distinct clonal lineages. Grey

points represent other mature cell types (not neutrophils and monocytes). b, Performance evaluation of the classifier using accuracy and cross-entropy loss metrics to predict early cell fate bias by employing gene expression of cells on just day 2, using cells on days 2 and 4, and including all three days within a clone. c, Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) values for the model using cells from days 2 and 4 or all three days with a clone, showing our model's performance in early cell fate bias prediction. Day 2 performance is very close to day 2&4 performance, thus not plotted. d, Comparative analysis with state-of-the-art methods in predicting early cell fate bias. Deep Lineage outperforms existing methods showing a higher classification accuracy.

7.3.3 Deep Lineage accurately predicts gene expression and early cell fate bias in regeneration

To evaluate the performance of our model on another dataset with more time points, we applied Deep Lineage to the reprogramming process of fibroblasts into induced endodermal progenitor (iEP) cells over 28 days (259). This data comprises gene expression data for 18803 cells and 28001 genes, across six time points (days 6, 9, 12, 15, 21, 28), with corresponding barcode-derived lineage data at three time points (days 0, 3, 13) (Fig. 4a). Following the same approach as above, we first perform an extensive hyperparameter search to select the parameters that yield the highest correlation between the held-out samples during training and those predicted by the model (See methods, Supplementary table 3). Then we assess the ability of our model to predict gene expression on different days during the fibroblast transition to iEP cells. Our method consistently has high performance in generating realistic gene expression profiles, regardless of the specific time point chosen for analysis (Fig. 4b, Supplementary Figure 3b). We also compared the gene expression distribution between the model's prediction and held-out cells at day 28 and showed that Deep Lineage can accurately predict the full gene expression distribution for both failed and reprogrammed cells (Fig 4c, and Supplementary Figure 4). Also, we observe a high similarity (R^2 = 0.987) between the average predicted gene expression and the actual expression levels for genes in both successfully and failed reprogrammed cells (Fig. 4d,e).

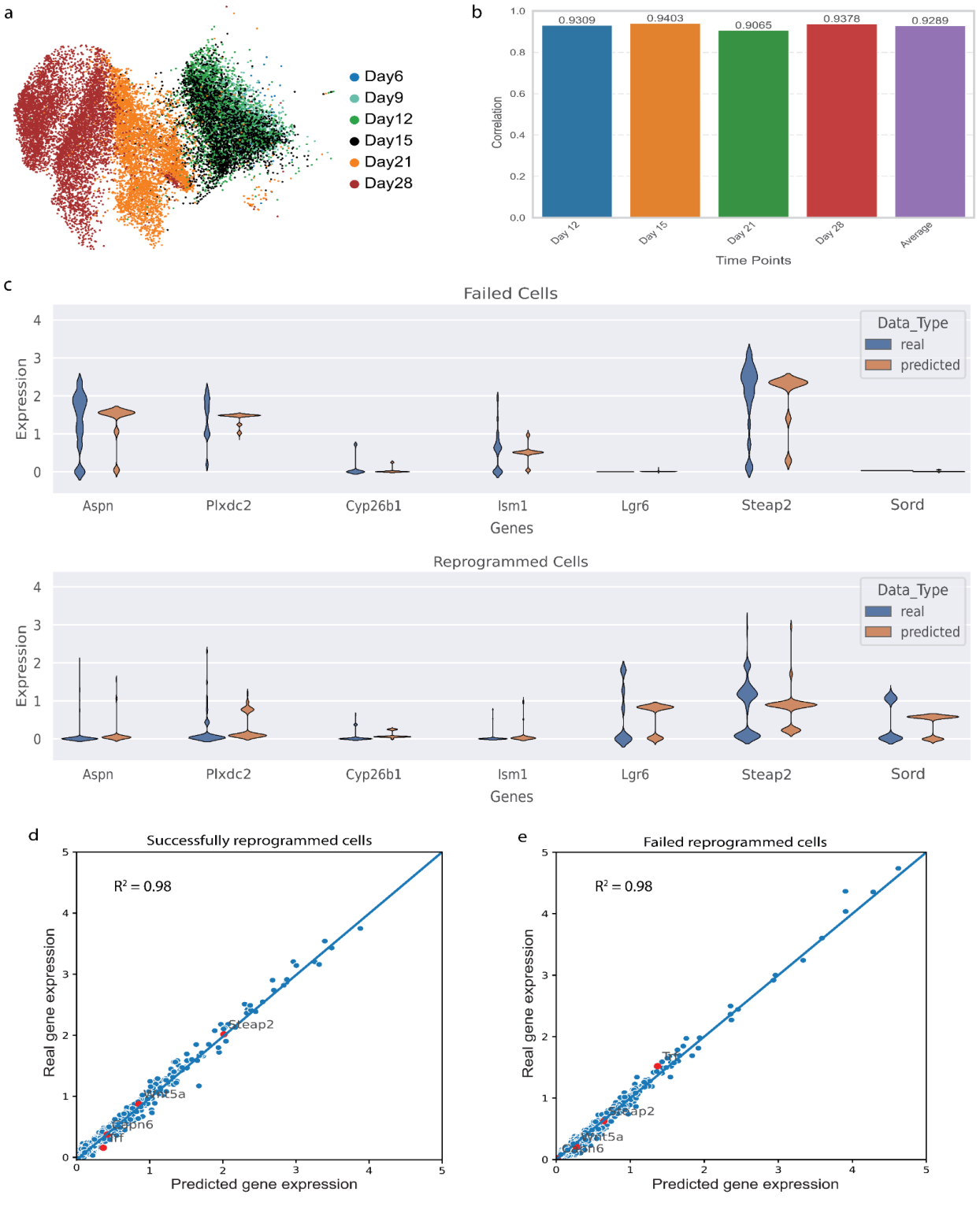

a
Day6
Day9
Day12
Day15
Day21
Day28
b
0.9309
0.9403
0.9065
0.9378
0.9289
Correlation
Day 12
Day 15
Day 21
Day 28
Average
Time Points
c
Failed Cells
Data_Type
real
predicted
Expression
Aspn
Plxdc2
Cyp26b1
Ism1
Lgr6
Steap2
Sord
Genes
Reprogrammed Cells
d
Successfully reprogrammed cells
R² = 0.98
Real gene expression
Predicted gene expression
e
Failed reprogrammed cells
R² = 0.98
Real gene expression
Predicted gene expression

Figure 7-4: Deep Lineage accurately predicts single-cell gene expression of unseen cells in reprogramming. a, UMAP visualization demonstrates fibroblast cell reprogramming into iEPs, dots represent cells and are color-coded by time point. b, Correlation values between gene expression of ground truth cells and Deep Lineage predictions for cells excluded during training across different stages of the reprogramming process. c, Violin plots show gene expression distributions between real cells and Deep Lineage predictions for randomly selected genes for both Failed and Reprogrammed cells. d,e. Comparative analysis of the top 2000 differentially expressed genes between predicted and real cells. High correlation values are observed for both successful and failed reprogramming cells, highlighting the accuracy of the model for different cell types ($\boldsymbol{R^2}$ indicates the squared Pearson correlation coefficient between the predicted and ground truth values).

We evaluate Deep Lineage's performance in predicting the outcome of the reprogramming process with ground truth labels and compare it to the current state-of-the-art method, Cellrank. (266). After training by providing the first four time points of the process (day 6, day 9, day 12, day 15) our model can predict the reprogramming outcome on day 28 with high accuracy (94%) (Fig. 5b). ROC analysis shows that Deep Lineage outperforms Cellrank irrespective of the number of days analyzed (Fig. 5c). As expected, by giving the model more early time points to use to infer later time points, the performance of both methods improved, but Deep Lineage performs better in all cases (Fig. 5c). To test our model under ablation conditions, we predict the reprogramming outcome by sampling cells from days 6, 9, 12, 15, and 21 without considering clonal information. As seen for the mouse hematopoiesis ablation study, the results show a decrease in performance, with the average accuracy dropping from 99% to 50.6%. Also similar to our findings with hematopoiesis, changing the model's architecture (LSTM vs. GRU, hyperparameter changes) has a limited impact on accuracy, which remains in the range of approximately 49% to 51%. This again highlights the importance of considering clonal structure in accurately predicting reprogramming outcomes (Supplementary Figure 5b).

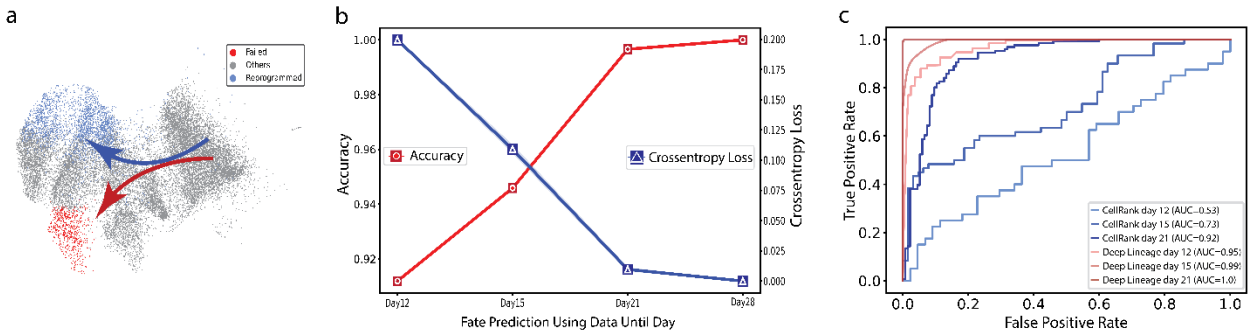


Figure 7-5: **Early cell fate prediction in reprogramming with Deep Lineage and comparison to state-of-the-art method.** a, UMAP visualization illustrates progenitor bias toward either successful or failed reprogramming outcomes. b, The model's accuracy in detecting successful or failed outcomes of progenitors is examined using gene expression data of cells up to days 12, 15, 21, and all days within a clone. c, Benchmarking comparison between Deep Lineage and CellRank in predicting fate outcome when using all time points up to a given time point (e.g. day 12) to infer time point "day 28". Using ROC curves and AUC values, the graph shows the predictive power of both models in predicting fate bias. Deep Lineage consistently outperforms CellRank.

7.3.4 Deep Lineage can identify key genes in cellular reprogramming and generate realistic in silico perturbation experiments

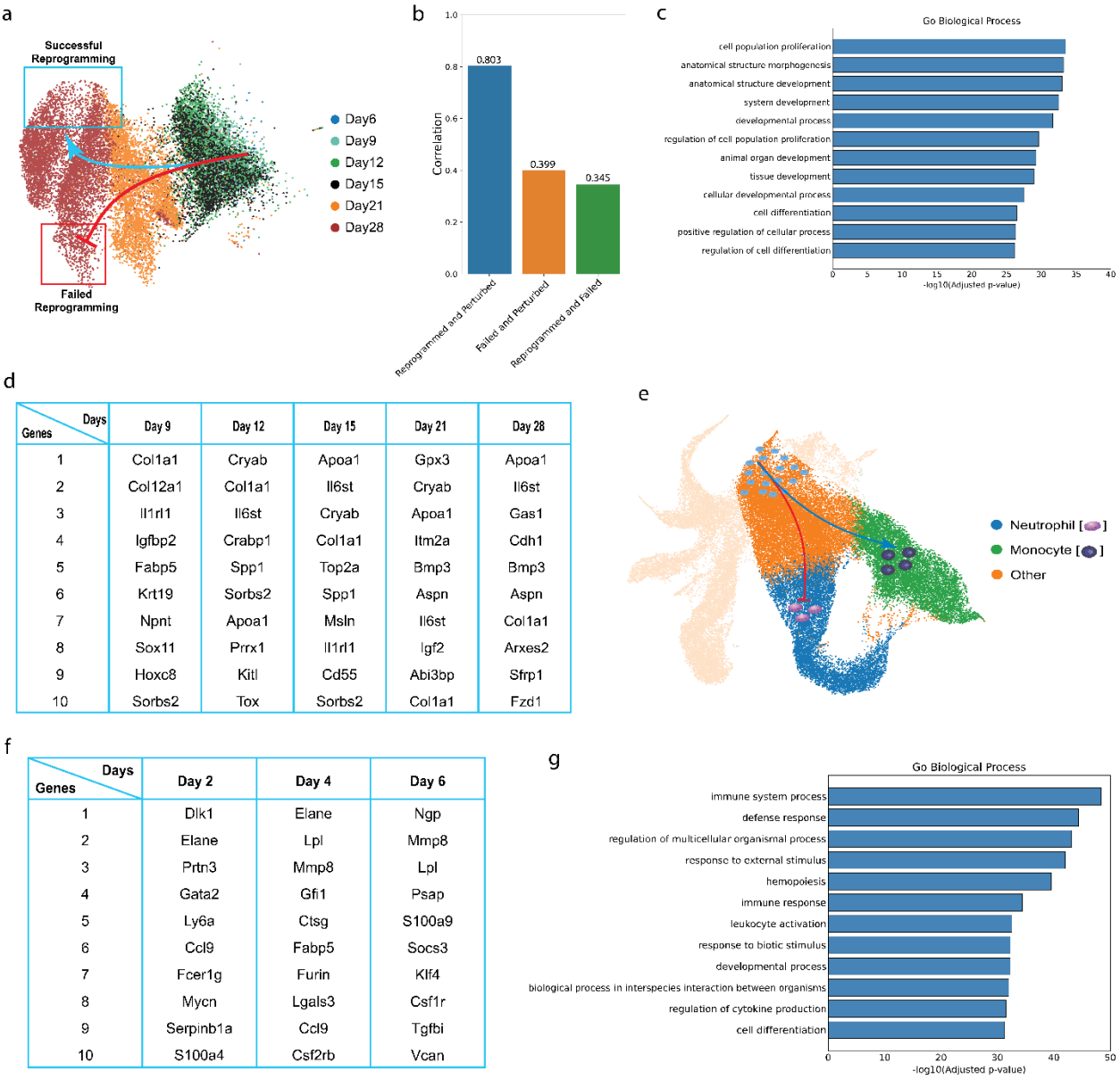
Detecting and targeting key genes involved in cell reprogramming is useful for advancing our understanding of the system and potential therapeutic applications (61). Deep Lineage uses its learned latent representation to predict cellular fate and gene expression profiles and SHAP analysis to identify key genes within the reprogramming process. We tested this ability of Deep Lineage by using both fibroblast reprogramming and mouse hematopoiesis data (81,259) and performed a series of in silico perturbations to manipulate the fate of these systems. Using SHAP, we predicted key genes for the reprogrammed and failed trajectories across various time points. Known essential marker genes of reprogramming (259,267–269) (Apoa1, Cryab, Col1a1, and Sox11) are in this list. Supplementary Table 1 presents a curated list of these genes, including their established functions in cellular reprogramming and related processes. We also performed Gene Set Enrichment Analysis (GSEA) on the top 200 gene candidates demonstrating the ability of Deep Lineage to detect reprogramming-relevant biological pathways such as system and tissue

development, proliferation signals, and cellular differentiation (Fig. 6c). Together these results suggest that Deep Lineage can be used to identify important reprogramming genes and processes.

Our goal is to alter cellular fate and reprogram cells in silico, steering them away from trajectories heading towards undesirable "dead-end" outcomes (Fig. 6a). We focused on the top 10 genes identified by the SHAP method in the failed trajectory path (Fig. 6d). After conducting a series of in silico perturbation experiments, which included over-expression and knock-out of these genes on day 15 of failed outcome clones, we applied Deep Lineage to generate gene expression profiles for days 21 and 28 (Fig. 6a). Subsequently, we input the generated perturbed time series back into Deep Lineage to evaluate the perturbations' effectiveness in redirecting the fate of the initially failed clones. After the perturbation, the classifier shows that the trajectory changed, with the fate of the failed clones changing to successfully reprogrammed, demonstrating the ability of the model to predict a way to lead to more successful cellular reprogramming. To further evaluate these predictions, we conducted a correlation analysis comparing the average gene expression of in silico reprogrammed cells on day 28 to both real reprogrammed cells and real failed cells. Our results show a significantly higher similarity between the generated in silico successful reprogrammed cells and real successful reprogramming cells (Pearson correlation = 80.3%) compared to the real failed cells (39.9%) (Fig. 6b). This additional analysis further supports Deep Lineage's ability to generate realistic gene expression for cells undergoing in silico perturbation.

We also applied Deep Lineage to hematopoiesis data to identify important genes and simulate the fate switch from neutrophils to monocytes (Fig. 6e). This analysis identified important genes, including some known genes (e.g. Dlk1, Gata2, Mmp8, Elane, and Lpl) important in the switch from neutrophils to monocytes (Fig. 6f, Supplementary Table 1) (270–274). Using the trained model and performing GSEA on the top 200 identified genes reveals significant enrichment of pathways associated with immune system development, differentiation, and hematopoiesis (Fig. 6g), indicating the model's ability to capture biological pathways relevant to hematopoiesis. By performing in silico knockouts of the top genes of progeny cells on day 2 of clones predestined to become neutrophils, we were successfully able to alter the predicted final fate of the clone, pushing

it toward monocytes by day 6. These results again demonstrate that Deep Lineage can accurately simulate and predict gene expression during in silico perturbation.



d

Genes \ Days	Day 9	Day 12	Day 15	Day 21	Day 28
1	Col1a1	Cryab	Apoa1	Gpx3	Apoa1
2	Col12a1	Col1a1	Il6st	Cryab	Il6st
3	Il1rl1	Il6st	Cryab	Apoa1	Gas1
4	Igfbp2	Crabp1	Col1a1	Itm2a	Cdh1
5	Fabp5	Spp1	Top2a	Bmp3	Bmp3
6	Krt19	Sorbs2	Spp1	Aspn	Aspn
7	Npnt	Apoa1	Msln	Il6st	Col1a1
8	Sox11	Prrx1	Il1rl1	Igf2	Arxes2
9	Hoxc8	Kitl	Cd55	Abi3bp	Sfrp1
10	Sorbs2	Tox	Sorbs2	Col1a1	Fzd1

f

Genes \ Days	Day 2	Day 4	Day 6
1	Dlk1	Elane	Ngp
2	Elane	Lpl	Mmp8
3	Prtn3	Mmp8	Lpl
4	Gata2	Gfi1	Psap
5	Ly6a	Ctsg	S100a9
6	Ccl9	Fabp5	Socs3
7	Fcer1g	Furin	Klf4
8	Mycn	Lgals3	Csf1r
9	Serpinb1a	Ccl9	Tgfbi
10	S100a4	Csf2rb	Vcan

Figure 7-6: In silico perturbations of early cells in hematopoiesis and reprogramming data successfully modify cell fate at later stages. a, UMAP representation showing the procedure for in silico perturbation of cells on day 15 within failed reprogrammed clones, aiming to switch them into successfully reprogrammed states. b, Correlation values between in silico perturbed data and actual successfully and unsuccessfully reprogrammed cells on Day 28, showing the power of Deep Lineage in performing in silico experiments. c, Gene set enrichment analysis reveals the biological processes associated with the top 200 genes identified by the SHAP method (40 genes selected from each time point) d, Identification of the top

10 gene candidates using SHAP analysis at each time point during fibroblast cell reprogramming. e, UMAP visualization depicting the in silico perturbation experiment targeting progenitor cells (Dark orange) in clones with a Neutrophil fate (blue), resulting in a transformation to Monocytes (green). f, The top 10 gene candidates for hematopoiesis detected by SHAP analysis across all time points. g, Gene set enrichment analysis of the top 200 genes identified by SHAP analysis.

7.4 Methods

7.4.1 Autoencoder for latent embedding of scRNA-seq data

To achieve the reduced dimension of the data before training the LSTM model, we employed an autoencoder, an unsupervised neural network algorithm (275). The autoencoder consists of two components: an encoder, which compresses the high-dimensional scRNA-seq data into a lower-dimensional latent space, and a decoder, which reconstructs the data back to the original input space. The objective of the autoencoder is to minimize the reconstruction error, measured using the mean squared error (MSE) loss function, expressed as:

$$L_{AE} = \frac{1}{N} * \sum_{i=1}^{N} \left|\left| x_i - \text{Decoder}\left(\text{Encoder}\left(x_{(i)}\right)\right)\right|\right|$$

Where N is the number of training samples, is the i-th input data sample which is the gene expression of a cell. *Encoder( )* and *Decoder( )* represent the encoder and the decoder parts of the autoencoder. By minimizing the above loss, the autoencoder learns a lower-dimensional representation of the data while preserving important biological features, facilitating the analysis of time series models (LSTMs, GRU)

7.4.2 RNN models for gene expression and early cell fate prediction

For the regression task of predicting gene expression for cells within a clone, we harnessed the power of RNNs, mainly LSTMs and GRU. Unlike traditional feedforward neural networks, LSTM networks have loops and gates that enable information to persist and flow through time steps, making them well-suited for tasks involving time series and sequential data analysis. The distinguishing feature of LSTM is its memory block, which maintains a hidden state and a block

state. The block state acts as a long-term memory, enabling the network to keep and update relevant information over long time ranges. The hidden state carries short-term memory and interacts with the block state through various gates. The LSTM block is composed of three main gates that control the flow of information: the input gate (i), the forget gate (f), and the output gate (o) (276). The input gate decides how much of the new input should be added to the block state. The forget gate controls what information from the previous block state should be discarded, allowing the network to selectively remember or forget information. The output gate determines what part of the block state should be exposed as the hidden state. The equations governing the LSTM block operations are as follows:

Input gate i_t:

$$i_t = \sigma(W_i * h_{t-1} + U_i * X_t + b_i)$$

Forget gate f_t:

$$f_t = \sigma\left(W_f * h_{t-1} + U_f * X_t + b_f\right)$$

Output gate o_t:

$$o_t = \sigma(W_o * h_{t-1} + U_o * X_t + b_o)$$

Block state c_t:

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_C * h_{t-1} + U_c * X_t + b_c)$$

Hidden state h_t:

$$h_t = o_t * \tanh(c_t)$$

Here, X_t represents the input at the current time step, h_{t-1} denotes the hidden state from the previous time step, and σ is the sigmoid activation function. The weights and biases, represented by W_i , W_f, W_o, W_C, U_i, U_f, U_o, and U_c, are learnable parameters that the LSTM network optimizes during training. To obtain a compact representation of the gene expression data, we used the latent layer of an autoencoder as input to the LSTM model.

GRU is another variant of the RNN models, designed to address some of the limitations of traditional RNNs, such as the vanishing gradient problem that can interfere with the learning process. Similar to LSTM, GRU is particularly effective in processing sequential data and capturing dependencies over long sequences. Each GRU block also contains a hidden state, but instead of having separate blocks and hidden states like LSTM, it uses an update gate (z) and a reset gate (r) to control the flow of information (276). These gates enable the GRU to selectively update and reset the hidden state, facilitating the retention of relevant information and reducing the number of parameters compared to LSTM. In GRUs the reset gate controls how much of the previous hidden state should be forgotten, allowing the network to decide what information to retain from the past. The update gate determines how much of the candidate's hidden state $h`_t$ should be blended with the previous hidden state h_{t-1} to generate the new hidden state h_t. This gate mechanism enables the GRU to regulate the information flow effectively. The operations of a GRU block can be summarized as follows:

Reset gate r_t:

$$r_t = \sigma(W_r * h_{t-1} + U_r * X_t + b_r)$$

Update gate z_t:

$$z_t = \sigma(W_z * h_{t-1} + U_z * X_t + b_z)$$

Candidate hidden state $h`_t$:

$$h`_t = \tanh(W_h * (r_t * h_{t-1}) + U_h * X_t + b_h)$$

Hidden state h_t:

$$h_t = (1 - z_t) * h_{t-1} + z_t * h`_t$$

Here, X_t represents the input at the current time step, h_{t-1} denotes the hidden state from the previous time step, and σ is the sigmoid activation function. The weights and biases, are

represented by $W_r, W_z, W_h, U_r, U_z, U_h, b_r, b_z,$ and b_h are learnable parameters that the GRU network optimizes during training. GRU is computationally more efficient than LSTM due to its simplified architecture, making it particularly suitable for handling extensive datasets with limited computational resources or situations where training is scarce.

The equations mentioned earlier can be used for both regression and classification tasks. In classification tasks, the model's output is then connected to a fully connected layer that handles the classification. Employing softmax activation in this layer allows the model to derive class probabilities.

$$Classification_{output} = Softmax(w * h_t + b)$$

7.4.3 **Data preprocessing**

To show Deep Lineage can handle different barcoding strategies, we applied it to data generated from two methods: Larry and CellTagging (81,259). Given the selection of a preprocessing method depends on the specific goal of the analysis, the performance of a supervised method can serve as a useful metric. Therefore, in each task (prediction of cell fate and gene expression), we compare the performance of different preprocessing steps before training the models and choose the one that leads to higher accuracy (lower reconstruction error for AE and lower loss for RNN) for that data. Our evaluation involves a multitude of combinations that incorporate different count normalization, log transformation, several scaling techniques, different numbers of highly variable genes, and different dimensionality reductions. For both reprogramming and hematopoiesis data, we realize that log transformation without count normalization achieves the highest accuracy and as expected using only raw counts or just scaling without log transformation results in poor performance which is consistent with the recently published benchmark paper (277) (Supplementary Figure 2, Supplementary Table 2). Following this, we select the highly variable genes and employ an autoencoder for dimensionality reduction (the architectures can be found in Supplementary Table 2).

7.4.4 **Modelling choices and training**

After completing the preprocessing step, the gene expression data is partitioned into three subsets: 80% for training, 10% for validation, and 10% for testing. To optimize model performance, we conduct an extensive hyperparameter search using various model types (LSTM, GRU, bidirectional LSTM), model architecture (number of layers, nodes in each layer), and activation function (Tanh, ReLU, Leaky ReLU with alpha=0.3) (the detailed results are provided in Supplementary Table 2). Subsequently, we select specific sets of parameters that demonstrate superior accuracy for both the autoencoder reconstruction and the LSTMs during regression and classification tasks, tailored to each dataset. For the autoencoder, we employ the mean square error as the loss function, to ensure effective learning of the latent representation. After training the autoencoder, the resulting latent embedding is used to train the RNN models. In the regression task, where we aim to predict gene expression levels of cells across different developmental stages, we employ the mean square error as the loss function. For predicting early cell fate bias (classification task), we adopt the categorical cross-entropy loss function. Throughout the training process of both the autoencoder and the RNN models, we implement early stopping, which enables us to halt the training once the models demonstrate optimal performance on the validation set, preventing overfitting and supporting generalization to unseen data. In each iteration, we randomly select a training sample and update the network's weights using SGD (Stochastic Gradient Descent) and Adam optimization algorithms with a default learning rate of 0.001. Also, RNN training can be slow due to the sequential nature of computations. To tackle this challenge, we use the CuDNN LSTM technique, which uses GPU-accelerated optimization to enhance the training efficiency of Long Short-Term Memory (LSTM) networks. By using the parallel processing capabilities and specialized algorithms offered by CuDNN LSTM, we achieve a significant acceleration in model training speed. In addressing instances of missing data within time series, a common approach involves the use of masking techniques. When a data point at a specific timestep is not available, the model skips that particular timestep and proceeds to the subsequent one. By using masking in both classification and regression tasks, we ensure that the time series model handles missing data consistently and effectively. Regarding the model selection, we observe

GRU, and LSTMs achieve high accuracies with LSTMs showing slightly superior performance (Supplementary Figure 2, Supplementary Tables 2 and 3). However, if computational resources are limited, we recommend using GRU over LSTMs, as GRU requires less computing power.

7.4.5 Comparison with existing methods

To establish a benchmark for evaluating the performance of Deep Lineage in early cell fate bias and predicting gene expression of cell progenies, for hematopoiesis data we selected existing relevant methods to compare to: Cospar (262), Super OT (263), WOT (264), and GAN-Based OT (265). We adopted the same task as Super OT (10), which involves evaluating the accuracy of fate prediction for day 2 cells in relation to their development into day 4 and 6 cell types. Specifically, using the clonal information from the dataset, we identify whether each day 2 cell is more likely to differentiate into a neutrophil or a monocyte, assigning this label accordingly For Cospar and WOT, we used the default parameters as reported in their documentation. For other methods where the code was not available, we attempted to reimplement the method. In cases where our reimplementation resulted in lower accuracy, we used the accuracy metrics reported in their respective studies to ensure a fair comparison of our model's performance with these established benchmarks.

In the context of early cell fate bias in the reprogramming dataset, we conduct a comparative analysis between Deep Lineage and Cellrank (266). The Cellrank framework integrates similarity-based trajectory inference with RNA velocity to uncover directed probabilistic state-change trajectories. When applying Cellrank to the reprogramming data, we followed the recommended preprocessing steps, including filtering genes based on a minimum expression threshold across ten cells and at least 20 counts in both spliced and unspliced layers. Following this, normalization by total counts per cell was performed, along with log transformation, while selecting the top 2,000 highly variable genes. Subsequently, a PCA representation was computed using the top 30 principal components (PCs) to construct a K-nearest neighbour (KNN) graph, with the parameter K set to 30.

7.4.6 Simulating and Evaluating In Silico Perturbations

Perturbation experiments in the Deep Lineage were conducted using both regression and classification models. In silico perturbations were introduced by modifying the scaled normalized expression levels of target genes. For gene knockdowns, expression was set to 0, while for overexpression, it was set to a value 5 times higher than the baseline expression level. These perturbations were applied at specific time points in the experimental timeline: Day 15 for the reprogramming dataset and Day 2 for the hematopoiesis dataset. Post-perturbation, we used the altered time series data as input for our regression models, which were then employed to predict the gene expression data for subsequent time points, specifically day 21 and day 28 for reprogramming and day 4 and day 5 for hematopoiesis. For consistency, the same neural network architecture as the baseline model (before perturbation) was used for each dataset. Following the generation of the perturbed time series data, we use the classifier that was used in our clonal fate bias experiments to classify and predict clone fates. The focus of these perturbations was on monocytes and neutrophils, driven by the availability of experimentally validated or correlated perturbation data for these cell types. The target genes for perturbation were selected based on the output of the SHAP method, which provided insights into gene importance within the model, guiding our selection process to focus on genes with a significant impact on the fate of monocytes and neutrophils.

7.4.7 Hardware used

Two computational setups are used for data processing, model training, and plotting. A distributed computing system was used extensively for data processing and model training. Nvidia A100 (40GB) and Nvidia P100 (16GB) GPUs were used on this computing server. Intel Xeon E5-2680v4 2.4 GHz (Broadwell) and AMD EPYC 7542 2.9 GHz CPUs were used for other tasks. The maximum memory used on these servers was 64GB.

7.4.8 Data availability

The data used in this study are openly accessible and can be obtained from their respective public repositories. The hematopoiesis Weinreb et al. data is available for download from the Gene Expression Omnibus (GEO) database under accession number GSE140802. The reprogramming dataset can be accessed from the GEO database with accession number GSE99915. Additionally, a preprocessed version of the data can be obtained from: https://cospar.readthedocs.io/en/latest/. We used these two datasets because they are based on two distinct methodologies: Larry and CellTagging. These methods offer unique advantages over CRISPR-based barcoding techniques, such as the ability to link cell states across time rather than only at a single endpoint. They also eliminate the need for lineage tree inference to establish sister-cell relationships, exhibit very low single-cell barcode dropout rates, and do not require the delivery of multiple components (81).

7.4.9 Code availability

Code supporting this study is available on: https://github.com/physics2001/Deep_Lineage. We used Python 3.9.13, tensorflow-gpu 2.10.0, cudatoolkit 11.3.1, cudnn 8.4.1.50, matplotlib 3.5.1, numpy 1.22.4, scanpy 1.8.2, scikit-learn 1.1.1, seaborn 0.12.2, scipy 1.10.1, keras 2.10.0, cospar 0.2.1, pip 22.1.2, pandas 1.4.3, conda 22.9.0.

Chapter 8
Conclusions and future directions

8.1 Summary of Contributions

Robert McKee's assertion that "true character is shown through the choices we make under pressure" can be extended beyond human behavior to other organisms, including cells. This perspective is also true for the cellular decision-making processes observed in biological systems, particularly under different conditions and perturbations. I personally have a strong interest in understanding aging and the dual goals of extending both lifespan and healthspan. In Chapter 2, I explored two main strategies for combating aging: cellular reprogramming and enhancing cellular resilience. Investigating the decision-making mechanisms of cells can advance both approaches. In cellular reprogramming, we can identify key regulatory genes, similar to the Yamanaka factors, that can increase regenerative capacity. To enhance cellular resistance, we can discover gene programs that help cells counteract aging-related challenges such as inflammation, mitochondrial dysfunction, and stem cell exhaustion, ultimately contributing to both extended lifespan and improved healthspan.

Single-cell technologies offer high-resolution data that enable us to investigate these biological processes with precision, and they also open the door to applying complex computational methods from fields like computer vision and natural language processing. Machine learning and deep learning, in particular, are essential tools for handling large-scale data and capturing nonlinear dynamics, which is critical since biological processes inherently exhibit nonlinearity. For example, gene regulatory networks often show nonlinear behavior where a small change in gene expression can lead to disproportionately large or cascading effects in downstream genes. This nonlinearity is crucial for processes such as bistable switches in cell fate decisions, where a cell might rapidly commit to one state or another, similar to how stem cells differentiate into specialized cells. Another example is the feedback loops found in signaling pathways like the MAPK cascade, where nonlinear interactions help amplify signals, allowing cells to respond

robustly to weak stimuli. These nonlinear processes enable biological systems to function, survive perturbations, and adapt to changing environments efficiently.

In my thesis, I worked under several key assumptions that were integrated into the developed computational models. A fundamental premise is that biological systems must make efficient decisions to survive and reproduce, which involves optimizing an energy function similar to the Hamiltonian concept in classical physics. This principle can be nicely exemplified in processes such as wound healing and tissue repair, where the body's decision to either regenerate (tissue without scarring) or repair (tissue with scar formation) is influenced by factors such as the severity of the injury and the availability of resources like oxygen, water, and inflammatory signals (278,279). The urgency of the situation also plays a critical role, dictating the type of response and the choice between fast and slow gene programs and pathways. Both repair and regeneration activate distinct and overlapping gene programs and pathways, which follow a hierarchical and modular organization. This process starts with the activation of a few key genes, which then trigger a coordinated cascade affecting other downstream genes. Interestingly, this hierarchical structure is not restricted to higher biological levels like genomics, proteomics, or metabolomics but also exists at the gene interaction level. The modular nature of these gene programs, where different modules work together to drive the response, facilitates interventions for us. Based on the timing, we can strategically choose where and what to target, and that will influence the effectiveness of the perturbation. This structure enables targeting at various scales to achieve specific outcomes, simplifying our ability to understand and manipulate the system.

Building on the aforementioned assumptions, we developed multiple computational methods primarily based on representation learning to investigate cellular decision-making processes. In Chapter 3, we introduced Fatecode, a computational method that uses a classification-supervised autoencoder to learn cell type related latent representations. By incorporating an auxiliary task related to cell type classification, Fatecode aims to identify genes whose perturbation can alter cell type distributions, thereby increasing or decreasing specific cell populations. In Chapter 4, we introduced CLERA, a computational method that leverages prior knowledge of the system to refine the space of possible latent variables. By selecting

representations that adhere to specific dynamic forms, CLERA minimizes total loss and reveals meaningful relationships between latent variables, which enables accurate cell type classification. This approach improves the interpretability of the results and facilitates the discovery of models for biological systems. Using PPR, CLERA successfully identifies active gene programs and central genes in various points of biological processes, which are consistent with previous studies. In Chapter 5, we proposed FateNet, which demonstrates the power of mechanistic models with different dynamics in training neural networks. FateNet accurately identifies the timing of cellular decision-making, the type of bifurcation, and identifies genes crucial in this process, which differ from those that are differentially expressed. Chapter 6 introduced the first latent diffusion model in the field of scRNA seq, capable of generating new cells similar to real ones for different cell types. This model predicts the effects of various perturbations, including disease and drug interventions, and leverages the hierarchical structure of latent diffusion models to illustrate how genes and pathways are activated as we interpolate between states. In Chapter 6, we presented Deep Lineage, which showed the potential of time series models and representation learning for modeling gene expression changes and early cell fate prediction using scRNA-seq and lineage tracing data. This approach highlights the power of computational systems with memory features for modeling clonal dynamics, particularly in early cell fate prediction. We applied these methods to a variety of datasets, including pancreas development, neurogenesis, hematopoiesis, induced endoderm progenitors reprogramming, and multiple perturbation data such as intestinal epithelial cells after Salmonella or H. poly infections, as well as simulated data with known gene regulatory networks and master regulators.

8.2 Future direction

From a technological perspective, several key directions could enhance our understanding of cellular dynamics. In recent years, single-cell genomics has emerged as a powerful tool for studying dynamic biological processes. However, its major limitation lies in the need to destroy cells when measuring gene expression levels, which obstructs our ability to track cellular decision-making over time. Moreover, while transcriptomics provides valuable insights, it alone may not

address some of our fundamental questions. A more robust, high-throughput single-cell proteomics method is necessary, particularly since mapping transcriptomic data to proteomic outcomes is not straightforward(280). This challenge can be partially addressed using ADT (Antibody-Derived Tags) methods, which enable simultaneous measurement of proteins and RNA at single-cell resolution, providing a more complete picture of cellular states(281).

Hopefully, improvements in lineage tracing methods could allow us to follow cells over time with greater resolution. Dynamic genetic barcoding techniques have shown significant promise in this area. Moreover, perturb-seq data should evolve to provide outputs that extend beyond gene expression, capturing details about cell fate and state changes. Large-scale perturb-seq experiments, applied across diverse cell types, could further illuminate cellular regulatory networks.

Incorporating multi-modal data, alongside perturb-seq or lineage tracing data, will deepen our understanding of causal gene interactions and offer a more holistic view of cellular development by integrating various molecular layers. A promising future direction would be adapting current predictive models to forecast perturbation responses at multiple molecular levels, which would require representation learning across all relevant modalities,

From a computational modeling perspective, future advancements will require developing more accurate computational methods that incorporate our prior knowledge. Recent attempts to apply large language models (LLMs) in biology, which aim to build one model for different tasks using millions of cell data, are unlikely to succeed with the current format. This is due to fundamental differences between biological systems and natural language. First, the volume of biological data is far smaller than what's available in NLP (scale), and biological systems are governed by physical and biochemical principles that must be incorporated into the models. Unlike language, where relationships between words can be loosely modeled based on patterns in data, biological systems are constrained by more rigid, complex, and hierarchical rules that govern molecular interactions, signaling pathways, and cellular behavior. Moreover, biological rules are harder to infer because they are intertwined with stochasticity and context dependence. Cells do not just follow predetermined scripts they adapt to environmental cues, and these responses often

follow non-linear dynamics that require more specialized models. As such, models designed for one biological task often can't be easily generalized to others without losing accuracy or interpretability.

What is needed are methods that can integrate prior knowledge of the system, grounded in physical laws and our biological knowledge. These methods must also understand planning and hierarchy. They should be able to answer questions such as: "If we want a specific phenotype, what are the ways to achieve it?" The models should propose multiple plans, determining when and what to perturb based on the current and the desired states, while incorporating uncertainty into each proposed plan (Bayesian probabilistic approaches being one potential solution but not the only one). Also, these models need to be robust and secure, especially given the sensitive nature of scRNA-seq data and its increasing relevance in health and medicine. In my PhD, I contributed to this by developing a robust cell-type classifier that withstands adversarial attacks, improving both robustness and interpretability. This work, titled *Adversarial training improves model interpretability in single-cell RNA-seq analysis*, was a collaboration with Prof. Anita Layton and Gary Bader and was published in *Bioinformatics Advances*(24). I chose not to include this project in my thesis since it focuses more on robustness and interpretability rather than cellular reprogramming.

Beyond these scientific advancements, it is important to reflect on the societal impacts of breakthroughs in the fields of biology and longevity, particularly given our success in extending the lifespan of various model organisms and reprogramming mature cells. If we achieve significant life extension, how will this affect the environment, politics, the economy, and scientific progress? For instance, if we can restore organ function and improve health, will people still prioritize healthy living? Politically, what if dictators live several decades longer? As Max Planck famously said, “A new scientific truth does not triumph by convincing its opponents, but rather because its opponents eventually die.” If we live much longer, how will this impact creativity and the generation of new ideas in scientific communities? These are complex and essential questions that need further thought and exploration as we move toward a future shaped by biological reprogramming and life extension.

In conclusion, my PhD thesis outlines several computational methods we developed specifically for scRNA-seq data to improve our understanding of cellular decision-making across various biological processes. These methods primarily focus on the prediction and analysis of latent representation rather than the observational layer. Also, we created models that integrate mechanistic modeling with deep learning to improve predictive accuracy and interpretability. We hope that these methods will significantly contribute to the analysis of single-cell data, providing valuable insights into cell reprogramming and fate decision-making.

References

1. Sadria M, Layton A, Goyal S, Bader GD. Fatecode enables cell fate regulator prediction using classification-supervised autoencoder perturbation. Cell Rep Methods. 2024 Jul 15;4(7):100819.

2. Sadria M, Swaroop V. Discovering Governing Equations of Biological Systems through Representation Learning and Sparse Model Discovery. BioRxiv. 2024 Sep 23;

3. Sadria M, Bury TM. FateNet: an integration of dynamical systems and deep learning for cell fate prediction. Bioinformatics. 2024 Sep 2;40(9).

4. Sadria M, Layton A. The Power of Two: integrating deep diffusion models and variational autoencoders for single-cell transcriptomics analysis. BioRxiv. 2023 Apr 16;

5. Sadria M, Zhang A, Bader GD. Deep Lineage: Single-Cell Lineage Tracing and Fate Inference Using Deep Learning. BioRxiv. 2024 Apr 26;

6. Keane M, Semeiks J, Webb AE, Li YI, Quesada V, Craig T, et al. Insights into the evolution of longevity from the bowhead whale genome. Cell Rep. 2015 Jan 6;10(1):112–22.

7. Nielsen J, Hedeholm RB, Heinemeier J, Bushnell PG, Christiansen JS, Olsen J, et al. Eye lens radiocarbon reveals centuries of longevity in the Greenland shark (Somniosus microcephalus). Science. 2016 Aug 12;353(6300):702–4.

8. Matsumoto Y, Piraino S, Miglietta MP. Transcriptome Characterization of Reverse Development in Turritopsis dohrnii (Hydrozoa, Cnidaria). G3 (Bethesda). 2019 Dec 3;9(12):4127–38.

9. George A. The Epic of Gilgamesh. A new translation. eprints.soas.ac.uk; 1999.

10. Lean E. The Modern Elixir: Medicine as a Consumer Item in the Early Twentieth-Century Chinese Press. UCLA Historical Journal. 1995;

11. Ghazvini FY, Fazeli M, Ashrafzadeh R. The Opposition between Good and Evil in the Story of Jamshid. Propós represent. 2021;9(SPE1).

12. Richardson A. Rapamycin, anti-aging, and avoiding the fate of Tithonus. J Clin Invest. 2013 Aug;123(8):3204–6.

13. López-Otín C, Blasco MA, Partridge L, Serrano M, Kroemer G. Hallmarks of aging: An expanding universe. Cell. 2023 Jan 19;186(2):243–78.

14. Principles of regenerative medicine. Elsevier; 2011.

15. López-Otín C, Galluzzi L, Freije JMP, Madeo F, Kroemer G. Metabolic control of longevity. Cell. 2016 Aug 11;166(4):802–21.

16. Sadria M, Layton AT. Use of Angiotensin-Converting Enzyme Inhibitors and Angiotensin II Receptor Blockers During the COVID-19 Pandemic: A Modeling Analysis. PLoS Comput Biol. 2020 Oct 8;16(10):e1008235.

17. Sadria M, Layton AT. Modeling within-Host SARS-CoV-2 Infection Dynamics and Potential Treatments. Viruses. 2021 Jun 14;13(6).

18. Layton AT, Sadria M. Understanding the dynamics of SARS-CoV-2 variants of concern in Ontario, Canada: a modeling study. Sci Rep. 2022 Feb 8;12(1):2114.

19. Sadria M, Layton AT. Interactions among mTORC, AMPK and SIRT: a computational model for cell energy balance and metabolism. Cell Commun Signal. 2021 May 20;19(1):57.

20. Sadria M, Seo D, Layton AT. The mixed blessing of AMPK signaling in Cancer treatments. BMC Cancer. 2022 Jan 25;22(1):105.

21. Sadria M, Layton AT. Aging affects circadian clock and metabolism and modulates timing of medication. iScience. 2021 Apr 23;24(4):102245.

22. Bajohr J, Scott EY, Olfat A, Sadria M, Lee K, Fahim M, et al. Direct lineage conversion of postnatal mouse cortical astrocytes to oligodendrocyte lineage cells. 2024 Jul 24;

23. Murphy KP. Probabilistic machine learning: an introduction. books.google.com; 2022.

24. Sadria M, Layton A, Bader GD. Adversarial training improves model interpretability in single-cell RNA-seq analysis. Bioinformatics Advances. 2023 Nov 23;3(1):vbad166.

25. Sadria M, Bjornstad P, Pottumarthi P, Lu-Ping L, Pyle L, Vigers T, et al. A multimodal neural network that distinguishes between type 1 and type 2 diabetes in young persons using MRI and clinical data. Physiology (Bethesda). 2023 May;38(S1).

26. Goodfellow I, Bengio Y, Courville A. Deep learning. books.google.com; 2016.

27. Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;

28. McInnes L, Healy J, Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv. 2018;

29. Jiang L, Chen H, Pinello L, Yuan G-C. GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. Genome Biol. 2016 Jul 1;17(1):144.

30. Sekar R, Rybkin O, Daniilidis K, Abbeel P, Hafner D, Pathak D. Planning to Explore via Self-Supervised World Models. arXiv. 2020;

31. Salazar J, Liang D, Nguyen TQ, Kirchhoff K. Masked language model scoring. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics; 2020. p. 2699–712.

32. Pelissetto A, Vicari E. Critical phenomena and renormalization-group theory. Physics Reports. 2002 Oct;368(6):549–727.

33. Rabbani M, Joshi R. An overview of the JPEG 2000 still image compression standard. Signal Processing: Image Communication. 2002 Jan;17(1):3–48.

34. Garrido Q, Assran M, Ballas N, Bardes A, Najman L, LeCun Y. Learning and Leveraging World Models in Visual Representation Learning. arXiv. 2024;

35. Dahl GE, Dong Yu, Li Deng, Acero A. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Trans Audio Speech Lang Process. 2012 Jan;20(1):30–42.

36. Sun C, Myers A, Vondrick C, Murphy K, Schmid C. Videobert: A joint model for video and language representation learning. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE; 2019. p. 7463–72.

37. Guan H, Liu M. Domain adaptation for medical image analysis: A survey. IEEE Trans Biomed Eng. 2022 Mar;69(3):1173–85.

38. Vincent P, Larochelle H, Bengio Y, Manzagol P-A. Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th international conference on Machine learning - ICML '08. New York, New York, USA: ACM Press; 2008. p. 1096–103.

39. Khemakhem I, Kingma DP, Monti RP, Hyvärinen A. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. arXiv. 2019;

40. Hyvärinen A, Khemakhem I, Morioka H. Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning. Patterns (N Y). 2023 Oct 13;4(10):100844.

41. Brunton SL, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci USA. 2016 Apr 12;113(15):3932–7.

42. Kutz JN, Brunton SL. Parsimony as the ultimate regularizer for physics-informed machine learning. Nonlinear Dyn. 2022 Feb;107(3):1801–17.

43. Watson DS. Conceptual challenges for interpretable machine learning. Synthese. 2022 Apr;200(2):65.

44. Brunton SL, Budišić M, Kaiser E, Kutz JN. Modern Koopman Theory for Dynamical Systems. arXiv. 2021;

45. Brunton SL, Brunton BW, Proctor JL, Kutz JN. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. PLoS ONE. 2016 Feb 26;11(2):e0150171.

46. Champion K, Lusch B, Kutz JN, Brunton SL. Data-driven discovery of coordinates and governing equations. Proc Natl Acad Sci USA. 2019 Nov 5;116(45):22445–51.

47. Kitano H. Computational systems biology. Nature. 2002 Nov 14;420(6912):206–10.

48. Kuznetsov YA. Elements of applied bifurcation theory. New York, NY: Springer New York; 2004.

49. Guckenheimer J, Holmes P. Nonlinear oscillations, dynamical systems, and bifurcations of vector fields. New York, NY: Springer New York; 1983.

50. Roesch E, Stumpf MPH. Parameter inference in dynamical systems with co-dimension 1 bifurcations. R Soc Open Sci. 2019 Oct 30;6(10):190747.

51. Longo SK, Guo MG, Ji AL, Khavari PA. Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. Nat Rev Genet. 2021 Oct;22(10):627–44.

52. Wang S, Sun S-T, Zhang X-Y, Ding H-R, Yuan Y, He J-J, et al. The Evolution of Single-Cell RNA Sequencing Technology and Application: Progress and Perspectives. Int J Mol Sci. 2023 Feb 2;24(3).

53. Wang J, Deng G, Wang S, Li S, Song P, Lin K, et al. Enhancing regenerative medicine: the crucial role of stem cell therapy. Front Neurosci. 2024 Feb 8;18:1269577.

54. Jovic D, Liang X, Zeng H, Lin L, Xu F, Luo Y. Single-cell RNA sequencing technologies and applications: A brief overview. Clin Transl Med. 2022 Mar;12(3):e694.

55. Heumos L, Schaar AC, Lance C, Litinetskaya A, Drost F, Zappia L, et al. Best practices for single-cell analysis across modalities. Nat Rev Genet. 2023 Aug;24(8):550–72.

56. Wagner DE, Klein AM. Lineage tracing meets single-cell omics: opportunities and challenges. Nat Rev Genet. 2020 Jul;21(7):410–27.

57. Dixit A, Parnas O, Li B, Chen J, Fulco CP, Jerby-Arnon L, et al. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. Cell. 2016 Dec 15;167(7):1853-1866.e17.

58. Scholkopf B, Locatello F, Bauer S, Ke NR, Kalchbrenner N, Goyal A, et al. Toward causal representation learning. Proc IEEE. 2021;1–23.

59. Shifrut E, Carnevale J, Tobin V, Roth TL, Woo JM, Bui CT, et al. Genome-wide CRISPR Screens in Primary Human T Cells Reveal Key Regulators of Immune Function. Cell. 2018 Dec 13;175(7):1958-1971.e15.

60. Ursu O, Neal JT, Shea E, Thakore PI, Jerby-Arnon L, Nguyen L, et al. Massively parallel phenotyping of coding variants in cancer with Perturb-seq. Nat Biotechnol. 2022 Jun;40(6):896–905.

61. Wang H, Yang Y, Liu J, Qian L. Direct cell reprogramming: approaches, mechanisms and progress. Nat Rev Mol Cell Biol. 2021 Jun;22(6):410–24.

62. Zimmermannova O, Caiado I, Ferreira AG, Pereira C-F. Cell fate reprogramming in the era of cancer immunotherapy. Front Immunol. 2021 Jul 21;12:714822.

63. Fleck JS, Jansen SMJ, Wollny D, Zenk F, Seimiya M, Jain A, et al. Inferring and perturbing cell fate regulomes in human brain organoids. Nature. 2023 Sep;621(7978):365–72.

64. Alyagor I, Berkun V, Keren-Shaul H, Marmor-Kollet N, David E, Mayseless O, et al. Combining Developmental and Perturbation-Seq Uncovers Transcriptional Modules Orchestrating Neuronal Remodeling. Dev Cell. 2018 Oct 8;47(1):38-52.e6.

65. Chen S, Rivaud P, Park JH, Tsou T, Charles E, Haliburton JR, et al. Dissecting heterogeneous cell populations across drug and disease conditions with PopAlign. Proc Natl Acad Sci USA. 2020 Nov 17;117(46):28784–94.

66. Lotfollahi M, Wolf FA, Theis FJ. scGen predicts single-cell perturbation responses. Nat Methods. 2019 Aug;16(8):715–21.

67. Wei X, Dong J, Wang F. scPreGAN, a deep generative model for predicting the response of single cell expression to perturbation. Bioinformatics. 2022 May 26;

68. Kamimoto K, Stringa B, Hoffmann CM, Jindal K, Solnica-Krezel L, Morris SA. Dissecting cell identity via network inference and in silico gene perturbation. Nature. 2023 Feb 8;614(7949):742–51.

69. Street K, Risso D, Fletcher RB, Das D, Ngai J, Yosef N, et al. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. BMC Genomics. 2018 Jun 19;19(1):477.

70. Ji Z, Ji H. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. Nucleic Acids Res. 2016 Jul 27;44(13):e117.

71. Tong A, Huang J, Wolf G, van Dijk D, Krishnaswamy S. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. Proc Mach Learn Res. 2020 Jul;119:9526–36.

72. Pandey K, Zafar H. Inference of cell state transitions and cell fate plasticity from single-cell with MARGARET. Nucleic Acids Res. 2022 Aug 26;50(15):e86.

73. Setty M, Kiseliovas V, Levine J, Gayoso A, Mazutis L, Pe'er D. Characterization of cell fate probabilities in single-cell data with Palantir. Nat Biotechnol. 2019 Apr;37(4):451–60.

74. Tritschler S, Büttner M, Fischer DS, Lange M, Bergen V, Lickert H, et al. Concepts and limitations for learning developmental trajectories from single cell genomics. Development. 2019 Jun 27;146(12).

75. Zhu Q, Zhang R. [1902.00220] A Classification Supervised Auto-Encoder Based on Predefined Evenly-Distributed Class Centroids. arXiv. 2019 Feb 1;

76. Abdolhosseini F, Azarkhalili B, Maazallahi A, Kamal A, Motahari SA, Sharifi-Zarchi A, et al. Cell Identity Codes: Understanding Cell Identity from Gene Expression Profiles using Deep Neural Networks. Sci Rep. 2019 Feb 20;9(1):2342.

77. Dibaeinia P, Sinha S. SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks. Cell Syst. 2020 Sep 23;11(3):252-271.e11.

78. Athanasiadis EI, Botthof JG, Andres H, Ferreira L, Lio P, Cvejic A. Single-cell RNA-sequencing uncovers transcriptional states and fate decisions in haematopoiesis. Nat Commun. 2017 Dec 11;8(1):2045.

79. Hochgerner H, Zeisel A, Lönnerberg P, Linnarsson S. Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell RNA sequencing. Nat Neurosci. 2018 Feb;21(2):290–9.

80. Paul F, Arkin Y, Giladi A, Jaitin DA, Kenigsberg E, Keren-Shaul H, et al. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. Cell. 2015 Dec 17;163(7):1663–77.

81. Weinreb C, Rodriguez-Fraticelli A, Camargo FD, Klein AM. Lineage tracing on transcriptional landscapes links state to fate during differentiation. Science. 2020 Feb 14;367(6479).

82. Press O, Galanti T, Benaim S, Wolf L. Emerging Disentanglement in Auto-Encoder Based Unsupervised Image Content Transfer. arXiv. 2020;

83. Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv. 2015 Nov 19;

84. Klys J, Snell J, Zemel R. Learning latent subspaces in variational autoencoders. Advances in neural information processing systems. 2018;31.

85. Plumerault A, Le Borgne H, Hudelot C. AVAE: adversarial variational auto encoder. 2020 25th International Conference on Pattern Recognition (ICPR). IEEE; 2021. p. 8687–94.

86. An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE. 2015;

87. Dai B, Wang Z, Wipf D. The usual suspects? Reassessing blame for VAE posterior collapse. International conference on machine learning. 2020;2313.

88. Li P, Pei Y, Li J. A comprehensive survey on design and application of autoencoder in deep learning. Appl Soft Comput. 2023 May;138:110176.

89. Bastidas-Ponce A, Tritschler S, Dony L, Scheibner K, Tarquis-Medina M, Salinno C, et al. Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. Development. 2019 Jun 17;146(12).

90. Hao Y, Hao S, Andersen-Nissen E, Mauck WM, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. Cell. 2021 Jun 24;184(13):3573-3587.

91. Rosenbauer F, Tenen DG. Transcription factors in myeloid development: balancing differentiation with transformation. Nat Rev Immunol. 2007 Feb 1;7(2):105–17.

92. Pijuan-Sala B, Griffiths JA, Guibentif C, Hiscock TW, Jawaid W, Calero-Nieto FJ, et al. A single-cell molecular map of mouse gastrulation and early organogenesis. Nature. 2019 Feb 20;566(7745):490–5.

93. Ben-David Y, Gajendran B, Sample KM, Zacksenhaus E. Current insights into the role of Fli-1 in hematopoiesis and malignant transformation. Cell Mol Life Sci. 2022 Feb 28;79(3):163.

94. Tani H, Li B, Kusu T, Okumura R, Nishimura J, Okuzaki D, et al. The ATP-hydrolyzing ectoenzyme E-NTPD8 attenuates colitis through modulation of P2X4 receptor-dependent metabolism in myeloid cells. Proc Natl Acad Sci USA. 2021 Sep 28;118(39).

95. Cai S, Paudel S, Jin L, Ghimire L, Taylor CM, Wakamatsu N, et al. NLRP6 modulates neutrophil homeostasis in bacterial pneumonia-derived sepsis. Mucosal Immunol. 2021 May;14(3):574–84.

96. Wang Z, Bunting KD. STAT5 in hematopoietic stem cell biology and transplantation. JAKSTAT. 2013 Oct 1;2(4):e27159.

97. Wang Z, Li G, Tse W, Bunting KD. Conditional deletion of STAT5 in adult mouse hematopoietic stem cells causes loss of quiescence and permits efficient nonablative stem cell replacement. Blood. 2009 May 14;113(20):4856–65.

98. Yáñez A, Ng MY, Hassanzadeh-Kiabi N, Goodridge HS. IRF8 acts in lineage-committed rather than oligopotent progenitors to control neutrophil vs monocyte production. Blood. 2015 Feb 26;125(9):1452–9.

99. Lee J, Zhou YJ, Ma W, Zhang W, Aljoufi A, Luh T, et al. Lineage specification of human dendritic cells is marked by IRF8 expression in hematopoietic stem cells and multipotent progenitors. Nat Immunol. 2017 Aug;18(8):877–88.

100. Hambleton S, Salem S, Bustamante J, Bigley V, Boisson-Dupuis S, Azevedo J, et al. IRF8 mutations and human dendritic-cell immunodeficiency. N Engl J Med. 2011 Jul 14;365(2):127–38.

101. Xie X, Liu J-X, Hu B, Xiao W. Zebrafish foxo3b negatively regulates canonical Wnt signaling to affect early embryogenesis. PLoS ONE. 2011 Sep 7;6(9):e24469.

102. Tsuchiya K, Westerterp M, Murphy AJ, Subramanian V, Ferrante AW, Tall AR, et al. Expanded granulocyte/monocyte compartment in myeloid-specific triple FoxO knockout increases oxidative stress and accelerates atherosclerosis in mice. Circ Res. 2013 Mar 29;112(7):992–1003.

103. Tothova Z, Kollipara R, Huntly BJ, Lee BH, Castrillon DH, Cullen DE, et al. FoxOs are critical mediators of hematopoietic stem cell resistance to physiologic oxidative stress. Cell. 2007 Jan 26;128(2):325–39.

104. Job F, Mai C, Villavicencio-Lorini P, Herfurth J, Neuhaus H, Hoffmann K, et al. OTUD3: A Lys6 and Lys63 specific deubiquitinase in early vertebrate development. Biochim Biophys Acta Gene Regul Mech. 2023 Mar;1866(1):194901.

105. Zhang Z, Zhao W, Li Y, Li Y, Cheng H, Zheng L, et al. YOD1 serves as a potential prognostic biomarker for pancreatic cancer. Cancer Cell Int. 2022 May 31;22(1):203.

106. Kaur S, Soto-Pantoja DR, Stein EV, Liu C, Elkahloun AG, Pendrak ML, et al. Thrombospondin-1 signaling through CD47 inhibits self-renewal by regulating c-Myc and other stem cell transcription factors. Sci Rep. 2013;3:1673.

107. Neganova I, Tilgner K, Buskin A, Paraskevopoulou I, Atkinson SP, Peberdy D, et al. CDK1 plays an important role in the maintenance of pluripotency and genomic stability in human pluripotent stem cells. Cell Death Dis. 2014 Nov 6;5(11):e1508.

108. Ben-David U, Cowell IG, Austin CA, Benvenisty N. Brief reports: Controlling the survival of human pluripotent stem cells by small molecule-based targeting of topoisomerase II alpha. Stem Cells. 2015 Mar;33(3):1013–9.

109. Zhang C, Fondufe-Mittendorf YN, Wang C, Chen J, Cheng Q, Zhou D, et al. Latexin regulation by HMGB2 is required for hematopoietic stem cell maintenance. Haematologica. 2020 Mar;105(3):573–84.

110. Fatima A, Irmak D, Noormohammadi A, Rinschen MM, Das A, Leidecker O, et al. The ubiquitin-conjugating enzyme UBE2K determines neurogenic potential through histone H3 in human embryonic stem cells. Commun Biol. 2020 May 25;3(1):262.

111. Liu P-F, Chen C-F, Shu C-W, Chang H-M, Lee C-H, Liou H-H, et al. UBE2C is a potential biomarker for tumorigenesis and prognosis in tongue squamous cell carcinoma. Diagnostics (Basel). 2020 Sep 4;10(9).

112. Mo AY-C, Kincross H, Wang X, Chang LY-T, Duns G, Kwan H, et al. Loss of FBXO11 function establishes a stem cell program in acute myeloid leukemia through dysregulation of the mitochondrial protease LONP1. BioRxiv. 2022 Sep 13;

113. Garza-Manero S, Sindi AAA, Mohan G, Rehbini O, Jeantet VHM, Bailo M, et al. Maintenance of active chromatin states by HMGN2 is required for stem cell identity in a pluripotent stem cell model. Epigenetics Chromatin. 2019 Dec 12;12(1):73.

114. Capecchi MR, Pozner A. ASPM regulates symmetric stem cell division by tuning Cyclin E ubiquitination. Nat Commun. 2015 Nov 19;6:8763.

115. Baker SJ, Ma'ayan A, Lieu YK, John P, Reddy MVR, Chen EY, et al. B-myb is an essential regulator of hematopoietic stem cell and myeloid progenitor cell development. Proc Natl Acad Sci USA. 2014 Feb 25;111(8):3122–7.

116. Li X, Sun L, Jin Y. Identification of karyopherin-alpha 2 as an Oct4 associated protein. Journal of Genetics and Genomics. 2008 Dec;35(12):723–8.

117. Yang XW, Zhong R, Heintz N. Granule cell specification in the developing mouse brain as defined by expression of the zinc finger transcription factor RU49. Development. 1996 Feb;122(2):555–66.

118. Bu S, Lv Y, Liu Y, Qiao S, Wang H. Zinc Finger Proteins in Neuro-Related Diseases Progression. Front Neurosci. 2021 Nov 18;15:760567.

119. Havrda MC, Harris BT, Mantani A, Ward NM, Paolella BR, Cuzon VC, et al. Id2 is required for specification of dopaminergic neurons during adult olfactory neurogenesis. J Neurosci. 2008 Dec 24;28(52):14074–86.

120. Pleasure SJ, Collins AE, Lowenstein DH. Unique expression patterns of cell fate molecules delineate sequential stages of dentate gyrus development. J Neurosci. 2000 Aug 15;20(16):6095–105.

121. Yang CY, Best JA, Knell J, Yang E, Sheridan AD, Jesionek AK, et al. The transcriptional regulators Id2 and Id3 control the formation of distinct memory CD8+ T cell subsets. Nat Immunol. 2011 Nov 6;12(12):1221–9.

122. Kühl NM, De Keyser J, De Vries H, Hoekstra D. Insulin-like growth factor binding proteins-1 and -2 differentially inhibit rat oligodendrocyte precursor cell survival and differentiation in vitro. J Neurosci Res. 2002 Jul 15;69(2):207–16.

123. La Manno G, Soldatov R, Zeisel A, Braun E, Hochgerner H, Petukhov V, et al. RNA velocity of single cells. Nature. 2018 Aug 8;560(7719):494–8.

124. Mukherjee C, Kling T, Russo B, Miebach K, Kess E, Schifferer M, et al. Oligodendrocytes provide antioxidant defense function for neurons by secreting ferritin heavy chain. Cell Metab. 2020 Aug 4;32(2):259-272.e10.

125. Qian L, Huang Y, Spencer CI, Foley A, Vedantham V, Liu L, et al. In vivo reprogramming of murine cardiac fibroblasts into induced cardiomyocytes. Nature. 2012 May 31;485(7400):593–8.

126. Wirsching H-G, Krishnan S, Florea A-M, Frei K, Krayenbühl N, Hasenbach K, et al. Thymosin β 4 gene silencing decreases stemness and invasiveness in glioblastoma. Brain. 2014 Feb;137(Pt 2):433–48.

127. Aibar S, González-Blas CB, Moerman T, Huynh-Thu VA, Imrichova H, Hulselmans G, et al. SCENIC: single-cell regulatory network inference and clustering. Nat Methods. 2017 Nov;14(11):1083–6.

128. Lin B, Srikanth P, Castle AC, Nigwekar S, Malhotra R, Galloway JL, et al. Modulating cell fate as a therapeutic strategy. Cell Stem Cell. 2018 Sep 6;23(3):329–41.

129. Davis TL, Rebay I. Master regulators in development: Views from the Drosophila retinal determination and mammalian pluripotency gene networks. Dev Biol. 2017 Jan 15;421(2):93–107.

130. Oestreich KJ, Weinmann AS. Master regulators or lineage-specifying? Changing views on CD4+ T cell transcription factors. Nat Rev Immunol. 2012 Nov 1;12(11):799–804.

131. Kamyshanska H, Memisevic R. The potential energy of an autoencoder. IEEE Trans Pattern Anal Mach Intell. 2015 Jun;37(6):1261–73.

132. Chi H. Regulation and function of mTOR signalling in T cell fate decisions. Nat Rev Immunol. 2012 Apr 20;12(5):325–38.

133. Tatapudy S, Aloisio F, Barber D, Nystul T. Cell fate decisions: emerging roles for metabolic signals and cell morphology. EMBO Rep. 2017 Dec;18(12):2105–18.

134. Mikolov T, Yih WT, Zweig G. Linguistic regularities in continuous space word representations. Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies. 2013;746.

135. Franz M, Lopes CT, Fong D, Kucera M, Cheung M, Siper MC, et al. Cytoscape.js 2023 update: a graph theory library for visualization and analysis. Bioinformatics. 2023 Jan 1;39(1).

136. Hirsa A, Neftci SN. An introduction to the mathematics of financial derivatives. books.google.com; 2013.

137. Camps-Valls G, Gerhardus A, Ninad U, Varando G, Martius G, Balaguer-Ballester E, et al. Discovering causal relations and equations from data. Physics Reports. 2023 Dec;1044:1–68.

138. Lejarza F, Baldea M. Data-driven discovery of the governing equations of dynamical systems via moving horizon optimization. Sci Rep. 2022 Jul 12;12(1):11836.

139. Li P-H, Kong X-Y, He Y-Z, Liu Y, Peng X, Li Z-H, et al. Recent developments in application of single-cell RNA sequencing in the tumour immune microenvironment and cancer therapy. Mil Med Res. 2022 Sep 26;9(1):52.

140. Saliba A-E, Westermann AJ, Gorski SA, Vogel J. Single-cell RNA-seq: advances and future challenges. Nucleic Acids Res. 2014 Aug;42(14):8845–60.

141. Eling N, Morgan MD, Marioni JC. Challenges in measuring and understanding biological noise. Nat Rev Genet. 2019 Sep;20(9):536–48.

142. Wen Y, Huang J, Guo S, Elyahu Y, Monsonego A, Zhang H, et al. Applying causal discovery to single-cell analyses using CausalCell. eLife. 2023 May 2;12.

143. Schölkopf B. Causality for machine learning. In: Geffner H, Dechter R, Halpern JY, editors. Probabilistic and causal inference: the works of judea pearl. New York, NY, USA: ACM; 2022. p. 765–804.

144. Hyvarinen A, Morioka H. Nonlinear ICA of temporally dependent stationary sources. Artificial Intelligence and Statistics. 2017;

145. Hyvarinen A, Sasaki H. Nonlinear ICA using auxiliary variables and generalized contrastive learning. PMLR. 2019 Apr 11;859–68.

146. Khemakhem I, Kingma D, Monti R. Variational autoencoders and nonlinear ica: A unifying framework. PMLR. 2020 Jun 3;2207–17.

147. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell RNA-seq denoising using a deep count autoencoder. Nat Commun. 2019 Jan 23;10(1):390.

148. Freedman SL, Xu B, Goyal S, Mani M. A dynamical systems treatment of transcriptomic trajectories in hematopoiesis. Development. 2023 Jun 1;150(11).

149. Lundberg S, Lee S-I. A Unified Approach to Interpreting Model Predictions. arXiv. 2017;

150. Jeh G, Widom J. Scaling personalized web search. Proceedings of the twelfth international conference on World Wide Web - WWW '03. New York, New York, USA: ACM Press; 2003. p. 271.

151. Lundberg SM, Lee S-I. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems. 2017;

152. Lee J-H, Lee J-H, Rane SG. TGF-β Signaling in Pancreatic Islet β Cell Development and Function. Endocrinology. 2021 Mar 1;162(3).

153. Szlachcic WJ, Ziojla N, Kizewska DK, Kempa M, Borowiak M. Endocrine Pancreas Development and Dysfunction Through the Lens of Single-Cell RNA-Sequencing. Front Cell Dev Biol. 2021 Apr 29;9:629212.

154. Kilic G, Wang J, Sosa-Pineda B. Osteopontin is a novel marker of pancreatic ductal tissues and of undifferentiated pancreatic precursors in mice. Dev Dyn. 2006 Jun 1;235(6):1659–67.

155. Dugnani E, Sordi V, Pellegrini S, Chimienti R, Marzinotto I, Pasquale V, et al. Gene expression analysis of embryonic pancreas development master regulators and terminal cell fate markers in resected pancreatic cancer: A correlation with clinical outcome. Pancreatology. 2018 Dec;18(8):945–53.

156. Zhu Y, Liu Q, Zhou Z, Ikeda Y. PDX1, Neurogenin-3, and MAFA: critical transcription regulators for beta cell development and regeneration. Stem Cell Res Ther. 2017 Nov 2;8(1):240.

157. Gradwohl G, Dierich A, LeMeur M, Guillemot F. neurogenin3 is required for the development of the four endocrine cell lineages of the pancreas. Proc Natl Acad Sci USA. 2000 Feb 15;97(4):1607–11.

158. Kim BM, Kim SY, Lee S, Shin YJ, Min BH, Bendayan M, et al. Clusterin induces differentiation of pancreatic duct cells into insulin-secreting cells. Diabetologia. 2006 Feb;49(2):311–20.

159. Puri S, Maachi H, Nair G, Russ HA, Chen R, Pulimeno P, et al. Sox9 regulates alternative splicing and pancreatic beta cell function. Nat Commun. 2024 Jan 18;15(1):588.

160. Rochford I, Joshi JC, Rayees S, Anwar M, Akhter MZ, Yalagala L, et al. Evidence for reprogramming of monocytes into reparative alveolar macrophages in vivo by targeting PDE4b. Am J Physiol Lung Cell Mol Physiol. 2021 Oct 1;321(4):L686–702.

161. Böiers C, Buza-Vidas N, Jensen CT, Pronk CJH, Kharazi S, Wittmann L, et al. Expression and role of FLT3 in regulation of the earliest stage of normal granulocyte-monocyte progenitor development. Blood. 2010 Jun 17;115(24):5061–8.

162. Ishiguro A, Spirin KS, Shiohara M, Tobler A, Gombart AF, Israel MA, et al. Id2 expression increases with differentiation of human myeloid cells. Blood. 1996 Jun 15;87(12):5225–31.

163. Scholz W, Platzer B, Schumich A, Höcher B, Fritsch G, Knapp W, et al. Initial human myeloid/dendritic cell progenitors identified by absence of myeloperoxidase protein expression. Exp Hematol. 2004 Mar;32(3):270–6.

164. Cremer S, Michalik KM, Fischer A, Pfisterer L, Jaé N, Winter C, et al. Hematopoietic deficiency of the long noncoding RNA MALAT1 promotes atherosclerosis and plaque inflammation. Circulation. 2019 Mar 5;139(10):1320–34.

165. Wang ZQ, Ovitt C, Grigoriadis AE, Möhle-Steinlein U, Rüther U, Wagner EF. Bone and haematopoietic defects in mice lacking c-fos. Nature. 1992 Dec 31;360(6406):741–5.

166. Konturek-Ciesla A, Olofzon R, Kharazi S, Bryder D. Implications of stress-induced gene expression for hematopoietic stem cell aging studies. Nat Aging. 2024 Feb;4(2):177–84.

167. Morisot S, Georgantas RW, Civin CI. 345. Hematopoietic Stem-Progenitor Cells Express CD52 mRNA and Membrane Protein. Mol Ther. 2006;13:S131–2.

168. Becker-Herman S, Rozenberg M, Hillel-Karniel C, Gil-Yarom N, Kramer MP, Barak A, et al. CD74 is a regulator of hematopoietic stem cell maintenance. PLoS Biol. 2021 Mar 4;19(3):e3001121.

169. Yang M, Liu F, Chen Z, Shen X, Hao J, Wang J. Causalvae: disentangled representation learning via neural structural causal models. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2021. p. 9588–97.

170. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323(6088):533–6.

171. Ma L, Zheng J. A polynomial based model for cell fate prediction in human diseases. BMC Syst Biol. 2017 Dec 21;11(Suppl 7):126.

172. Chow S-M. Practical tools and guidelines for exploring and fitting linear and nonlinear dynamical systems models. Multivariate Behav Res. 2019 Apr 5;54(5):690–718.

173. Ambika G, Kurths J. Tipping in complex systems: theory, methods and applications. Eur Phys J Spec Top. 2021 Oct;230(16–17):3177–9.

174. Dakos V, Matthews B, Hendry AP, Levine J, Loeuille N, Norberg J, et al. Ecosystem tipping points in an evolving world. Nat Ecol Evol. 2019 Mar;3(3):355–62.

175. May RM, Levin SA, Sugihara G. Complex systems: Ecology for bankers. Nature. 2008 Feb 21;451(7181):893–5.

176. Dietz S, Rising J, Stoerk T, Wagner G. Economic impacts of tipping points in the climate system. Proc Natl Acad Sci USA. 2021 Aug 24;118(34).

177. Jurczyk J, Rehberg T, Eckrot A, Morgenstern I. Measuring critical transitions in financial markets. Sci Rep. 2017 Sep 14;7(1):11564.

178. Drake JM, Brett TS, Chen S, Epureanu BI, Ferrari MJ, Marty É, et al. The statistics of epidemic transitions. PLoS Comput Biol. 2019 May 8;15(5):e1006917.

179. Murray JD, editor. Mathematical Biology II: Spatial Models and Biomedical Applications. New York: Springer-Verlag; 2003.

180. Venegas JG, Winkler T, Musch G, Vidal Melo MF, Layfield D, Tgavalekos N, et al. Self-organized patchiness in asthma as a prelude to catastrophic shifts. Nature. 2005 Apr 7;434(7034):777–82.

181. Chen L, Liu R, Liu Z-P, Li M, Aihara K. Detecting early-warning signals for sudden deterioration of complex diseases by dynamical network biomarkers. Sci Rep. 2012 Mar 29;2:342.

182. Meisel C, Kuehn C. Scaling effects and spatio-temporal multilevel dynamics in epileptic seizures. PLoS ONE. 2012 Feb 17;7(2):e30371.

183. Scheffer M, Bascompte J, Brock WA, Brovkin V, Carpenter SR, Dakos V, et al. Early-warning signals for critical transitions. Nature. 2009 Sep 3;461(7260):53–9.

184. Wissel C. A universal law of the characteristic return time near thresholds. Oecologia. 1984 Dec;65(1):101–7.

185. Kleinen T, Held H, Petschel-Held G. The potential role of spectral properties in detecting thresholds in the Earth system: application to the thermohaline circulation. Ocean Dynamics. 2003 Jun 1;53(2):53–63.

186. Brock WA, Carpenter SR. Variance as a leading indicator of regime shift in ecosystem services. Ecology and Society. 2006;11(2).

187. Pace ML, Batt RD, Buelo CD, Carpenter SR, Cole JJ, Kurtzweil JT, et al. Reversal of a cyanobacterial bloom in response to early warnings. Proc Natl Acad Sci USA. 2017 Jan 10;114(2):352–7.

188. Boers N. Early-warning signals for Dansgaard-Oeschger events in a high-resolution ice core record. Nat Commun. 2018 Jul 2;9(1):2556.

189. Boettiger C, Ross N, Hastings A. Early warning signals: the charted and uncharted territories. Theor Ecol. 2013 Aug;6(3):255–64.

190. Dakos V, Scheffer M, van Nes EH, Brovkin V, Petoukhov V, Held H. Slowing down as an early warning signal for abrupt climate change. Proc Natl Acad Sci USA. 2008 Sep 23;105(38):14308–12.

191. Ferrell JE. Bistability, bifurcations, and Waddington's epigenetic landscape. Curr Biol. 2012 Jun 5;22(11):R458-66.

192. Sadria M, Layton A, Goyal S, Bader G. Fatecode: Cell fate regulator prediction using classification autoencoder perturbation. BioRxiv. 2022 Dec 19;

193. Lee J, Hyeon DY, Hwang D. Single-cell multiomics: technologies and data analysis methods. Exp Mol Med. 2020 Sep 15;52(9):1428–42.

194. Weinreb C, Wolock S, Tusi BK, Socolovsky M, Klein AM. Fundamental limits on dynamic inference from single-cell snapshots. Proc Natl Acad Sci USA. 2018 Mar 6;115(10):E2467–76.

195. Ding J, Sharon N, Bar-Joseph Z. Temporal modelling using single-cell transcriptomics. Nat Rev Genet. 2022 Jun;23(6):355–68.

196. Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nat Biotechnol. 2019 May;37(5):547–54.

197. Gorin G, Fang M, Chari T, Pachter L. RNA velocity unraveled. PLoS Comput Biol. 2022 Sep 12;18(9):e1010492.

198. Moris N, Pina C, Arias AM. Transition states and cell fate decisions in epigenetic landscapes. Nat Rev Genet. 2016 Nov;17(11):693–703.

199. Brett TS, Drake JM, Rohani P. Anticipating the emergence of infectious diseases. J R Soc Interface. 2017 Jul;14(132).

200. Dakos V, Soler-Toscano F. Measuring complexity to infer changes in the dynamics of ecological systems under stress. Ecological Complexity. 2016 Sep;

201. Dakos V, Boulton CA, Buxton JE, Abrams JF, Armstrong McKay DI, Bathiany S, et al. Tipping Point Detection and Early-Warnings in climate, ecological, and human systems. 2023 Aug 7;

202. Zhong J, Han C, Chen P, Liu R. SGAE: single-cell gene association entropy for revealing critical states of cell transitions during embryonic development. Brief Bioinformatics. 2023 Sep 22;24(6).

203. Luo Q, Maity AK, Teschendorff AE. Distance covariance entropy reveals primed states and bifurcation dynamics in single-cell RNA-Seq data. iScience. 2022 Dec 22;25(12):105709.

204. Pathak J, Hunt B, Girvan M, Lu Z, Ott E. Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach. Phys Rev Lett. 2018 Jan 12;120(2):024102.

205. Bury TM, Sujith RI, Pavithran I, Scheffer M, Lenton TM, Anand M, et al. Deep learning for early warning signals of tipping points. Proc Natl Acad Sci USA. 2021 Sep 28;118(39).

206. Bury TM, Dylewsky D, Bauch CT, Anand M, Glass L, Shrier A, et al. Predicting discrete-time bifurcations with deep learning. Nat Commun. 2023 Oct 10;14(1):6331.

207. Deb S, Sidheekh S, Clements CF, Krishnan NC, Dutta PS. Machine learning methods trained on simple models can predict critical transitions in complex natural systems. R Soc Open Sci. 2022 Feb 16;9(2):211475.

208. Wolf FA, Hamey FK, Plass M, Solana J, Dahlin JS, Göttgens B, et al. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. Genome Biol. 2019 Mar 19;20(1):59.

209. Clements CF, McCarthy MA, Blanchard JL. Early warning signals of recovery in complex systems. Nat Commun. 2019 Apr 11;10(1):1681.

210. Glass L, Mackey MC. From clocks to chaos: the rhythms of life. Princeton University Press; 2020.

211. Zhou JX, Huang S. Understanding gene circuits at cell-fate branch points for rational cell reprogramming. Trends Genet. 2011 Feb;27(2):55–62.

212. Goldbeter A. Dissipative structures in biological systems: bistability, oscillations, spatial patterns and waves. Philos Transact A Math Phys Eng Sci. 2018 Jul 28;376(2124).

213. Hashimoto H, Olson EN, Bassel-Duby R. Therapeutic approaches for cardiac regeneration and repair. Nat Rev Cardiol. 2018 Oct;15(10):585–600.

214. Matsuda M, Koga M, Woltjen K, Nishida E, Ebisuya M. Synthetic lateral inhibition governs cell-type bifurcation with robust ratios. Nat Commun. 2015 Feb 5;6:6195.

215. Huang S, Guo Y-P, May G, Enver T. Bifurcation dynamics in lineage-commitment in bipotent progenitor cells. Dev Biol. 2007 May 15;305(2):695–713.

216. Bargaje R, Trachana K, Shelton MN, McGinnis CS, Zhou JX, Chadick C, et al. Cell population structure prior to bifurcation predicts efficiency of directed differentiation in human induced pluripotent cells. Proc Natl Acad Sci USA. 2017 Feb 28;114(9):2271–6.

217. Hat B, Kochańczyk M, Bogdał MN, Lipniacki T. Feedbacks, Bifurcations, and Cell Fate Decision-Making in the p53 System. PLoS Comput Biol. 2016 Feb 29;12(2):e1004787.

218. Raju A, Siggia ED. A geometrical model of cell fate specification in the mouse blastocyst. Development. 2024 Apr 15;151(8).

219. Zhong J, Han C, Wang Y, Chen P, Liu R. Identifying the critical state of complex biological systems by the directed-network rank score method. Bioinformatics. 2022 Dec 13;38(24):5398–405.

220. Zhong J, Han C, Zhang X, Chen P, Liu R. scGET: Predicting Cell Fate Transition During Early Embryonic Development by Single-cell Graph Entropy. Genomics Proteomics Bioinformatics. 2021 Jun;19(3):461–74.

221. Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. PLoS ONE. 2015 Jul 10;10(7):e0130140.

222. Korolev KS, Xavier JB, Gore J. Turning ecology and evolution against cancer. Nat Rev Cancer. 2014 May;14(5):371–80.

223. Kembro JM, Cortassa S, Lloyd D, Sollott SJ, Aon MA. Mitochondrial chaotic dynamics: Redox-energetic behavior at the edge of stability. Sci Rep. 2018 Oct 18;8(1):15422.

224. Glass L. Dynamical disease: Challenges for nonlinear dynamics and medicine. Chaos. 2015 Sep;25(9):097603.

225. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997;9(8):1735–80.

226. Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Comput. 2000 Oct;12(10):2451–71.

227. Bury TM. ewstools: A Python package for early warning signals of bifurcations in time series data. JOSS. 2023 Feb 11;8(82):5038.

228. Dakos V, Carpenter SR, Brock WA, Ellison AM, Guttal V, Ives AR, et al. Methods for detecting early warnings of critical transitions in time series illustrated using simulated ecological data. PLoS ONE. 2012 Jul 17;7(7):e41010.

229. Flood MW, Grimm B. EntropyHub: An open-source toolkit for entropic time series analysis. PLoS ONE. 2021 Nov 4;16(11):e0259448.

230. Merico D, Isserlin R, Stueker O, Emili A, Bader GD. Enrichment map: a network-based method for gene-set enrichment visualization and interpretation. PLoS ONE. 2010 Nov 15;5(11):e13984.

231. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv. 2021;

232. Pandey K, Mukherjee A, Rai P, Kumar A. DiffuseVAE: Efficient, Controllable and High-Fidelity Generation from Low-Dimensional Latents. arXiv. 2022;

233. Preechakul K, Chatthee N, Wizadwongsa S, Suwajanakorn S. Diffusion Autoencoders: Toward a Meaningful and Decodable Representation. arXiv. 2021;

234. Shi C, Pamer EG. Monocyte recruitment during infection and inflammation. Nat Rev Immunol. 2011 Oct 10;11(11):762–74.

235. Isogai S, Yamamoto N, Hiramatsu N, Goto Y, Hayashi M, Kondo M, et al. Preparation of induced pluripotent stem cells using human peripheral blood monocytes. Cell Reprogram. 2018 Dec;20(6):347–55.

236. Jung H, Kim MJ, Kim DO, Kim WS, Yoon S-J, Park Y-J, et al. TXNIP maintains the hematopoietic cell pool by switching the function of p53 under oxidative stress. Cell Metab. 2013 Jul 2;18(1):75–85.

237. Gribble SL, Kim H-S, Bonner J, Wang X, Dorsky RI. Tcf3 inhibits spinal cord neurogenesis by regulating sox4a expression. Development. 2009 Mar;136(5):781–9.

238. Dumon S, Walton DS, Volpe G, Wilson N, Dassé E, Del Pozzo W, et al. Itga2b regulation at the onset of definitive hematopoiesis and commitment to differentiation. PLoS ONE. 2012 Aug 28;7(8):e43300.

239. Cox JL, Mallanna SK, Ormsbee BD, Desler M, Wiebe MS, Rizzino A. Banf1 is required to maintain the self-renewal of both mouse and human embryonic stem cells. J Cell Sci. 2011 Aug 1;124(Pt 15):2654–65.

240. Schlachetzki JCM, Prots I, Tao J, Chun HB, Saijo K, Gosselin D, et al. A monocyte gene expression signature in the early clinical course of Parkinson's disease. Sci Rep. 2018 Jul 17;8(1):10757.

241. Carmona SJ, Teichmann SA, Ferreira L, Macaulay IC, Stubbington MJT, Cvejic A, et al. Single-cell transcriptome analysis of fish immune cells provides insight into the evolution of vertebrate immune cell types. Genome Res. 2017 Mar;27(3):451–61.

242. Huber R, Pietsch D, Panterodt T, Brand K. Regulation of C/EBPβ and resulting functions in cells of the monocytic lineage. Cell Signal. 2012 Jun;24(6):1287–96.

243. Hu Y, Hu Y, Xiao Y, Wen F, Zhang S, Liang D, et al. Genetic landscape and autoimmunity of monocytes in developing Vogt-Koyanagi-Harada disease. Proc Natl Acad Sci USA. 2020 Oct 13;117(41):25712–21.

244. Ho J, Jain A, Abbeel P. Denoising Diffusion Probabilistic Models. arXiv. 2020;

245. Kang HM, Subramaniam M, Targ S, Nguyen M, Maliskova L, McCarthy E, et al. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. Nat Biotechnol. 2018;36(1):89–94.

246. Zheng M, Hu Z, Mei X, Ouyang L, Song Y, Zhou W, et al. Single-cell sequencing shows cellular heterogeneity of cutaneous lesions in lupus erythematosus. Nat Commun. 2022 Dec 5;13(1):7489.

247. Zhou D, Luo N, Wu Q, You Y, Zhai Z, Mou Z, et al. Transcellular distribution heterogeneity of Annexin A5 represents a protective response to lupus-related thrombophilia: a pilot Proteomics-based study. Biochem Biophys Res Commun. 2012 Apr 6;420(2):357–63.

248. Yanaba K, Asano Y, Noda S, Akamata K, Aozasa N, Taniguchi T, et al. Increased circulating fibrinogen-like protein 2 in patients with systemic sclerosis. Clin Rheumatol. 2013 Jan;32(1):43–7.

249. Santer DM, Wiedeman AE, Teal TH, Ghosh P, Elkon KB. Plasmacytoid dendritic cells and C1q differentially regulate inflammatory gene induction by lupus immune complexes. J Immunol. 2012 Jan 15;188(2):902–15.

250. Zhao Y, Zhou L, Li H, Sun T, Wen X, Li X, et al. Nuclear-Encoded lncRNA MALAT1 Epigenetically Controls Metabolic Reprogramming in HCC Cells through the Mitophagy Pathway. Mol Ther Nucleic Acids. 2021 Mar 5;23:264–76.

251. Chang C-C, Hsu W-H, Wang C-C, Chou C-H, Kuo MY-P, Lin B-R, et al. Connective tissue growth factor activates pluripotency genes and mesenchymal-epithelial transition in head and neck cancer cells. Cancer Res. 2013 Jul 1;73(13):4147–57.

252. Katsuda T, Sussman JH, Ito K, Katznelson A, Yuan S, Takenaka N, et al. Cellular reprogramming in vivo initiated by SOX4 pioneer factor activity. Nat Commun. 2024 Feb 26;15(1):1761.

253. Lazure F, Farouni R, Sahinyan K, Blackburn DM, Hernández-Corchado A, Perron G, et al. Transcriptional reprogramming of skeletal muscle stem cells by the niche environment. Nat Commun. 2023 Feb 1;14(1):535.

254. Lim WK, Lyashenko E, Califano A. Master regulators used as breast cancer metastasis classifier. Pac Symp Biocomput. 2009;504–15.

255. Nachmani E, Roman RS, Wolf L. Denoising Diffusion Gamma Models. arXiv. 2021;

256. Kreis K, Dockhorn T, Li Z, Zhong E. Latent Space Diffusion Models of Cryo-EM Structures. arXiv. 2022;

257. Song J, Meng C, Ermon S. Denoising Diffusion Implicit Models. arXiv. 2020;

258. Haber AL, Biton M, Rogel N, Herbst RH, Shekhar K, Smillie C, et al. A single-cell survey of the small intestinal epithelium. Nature. 2017 Nov 16;551(7680):333–9.

259. Biddy BA, Kong W, Kamimoto K, Guo C, Waye SE, Sun T, et al. Single-cell mapping of lineage and identity in direct reprogramming. Nature. 2018 Dec 5;564(7735):219–24.

260. Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 2019 Jul;31(7):1235–70.

261. Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv. 2014;

262. Wang S-W, Herriges MJ, Hurley K, Kotton DN, Klein AM. CoSpar identifies early cell fate biases from single-cell transcriptomic and lineage information. Nat Biotechnol. 2022 Jul;40(7):1066–74.

263. Prasad N, Yang K, Uhler C. Optimal Transport using GANs for Lineage Tracing. arXiv. 2020;

264. Schiebinger G, Shu J, Tabaka M, Cleary B, Subramanian V, Solomon A, et al. Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming. Cell. 2019 Feb 7;176(4):928-943.e22.

265. Yang KD, Uhler C. Scalable unbalanced optimal transport using generative adversarial networks. arXiv preprint arXiv:181011447. 2018;

266. Lange M, Bergen V, Klein M, Setty M, Reuter B, Bakhti M, et al. CellRank for directed single-cell fate mapping. Nat Methods. 2022 Feb;19(2):159–70.

267. Luo C, Lee QY, Wapinski O, Castanon R, Nery JR, Mall M, et al. Global DNA methylation remodeling during direct reprogramming of fibroblasts to neurons. eLife. 2019 Jan 15;8.

268. Chen Y, Yang S, Lovisa S, Ambrose CG, McAndrews KM, Sugimoto H, et al. Type-I collagen produced by distinct fibroblast lineages reveals specific function during embryogenesis and Osteogenesis Imperfecta. Nat Commun. 2021 Dec 10;12(1):7199.

269. Miao Q, Hill MC, Chen F, Mo Q, Ku AT, Ramos C, et al. SOX11 and SOX4 drive the reactivation of an embryonic gene program during murine wound repair. Nat Commun. 2019 Sep 6;10(1):4042.

270. Li L, Forman SJ, Bhatia R. Expression of DLK1 in hematopoietic cells results in inhibition of differentiation and proliferation. Oncogene. 2005 Jun 23;24(27):4472–6.

271. Castaño J, Aranda S, Bueno C, Calero-Nieto FJ, Mejia-Ramirez E, Mosquera JL, et al. GATA2 promotes hematopoietic development and represses cardiac differentiation of human mesoderm. Stem Cell Reports. 2019 Sep 10;13(3):515–29.

272. Saw S, Weiss A, Khokha R, Waterhouse PD. Metalloproteases: on the watch in the hematopoietic niche. Trends Immunol. 2019 Nov;40(11):1053–70.

273. Liu C, Han T, Stachura DL, Wang H, Vaisman BL, Kim J, et al. Lipoprotein lipase regulates hematopoietic stem progenitor cell maintenance through DHA supply. Nat Commun. 2018 Apr 3;9(1):1310.

274. Nasri M, Ritter M, Mir P, Dannenmann B, Aghaallaei N, Amend D, et al. CRISPR/Cas9-mediated ELANE knockout enables neutrophilic maturation of primary hematopoietic stem and progenitor cells and induced pluripotent stem cells of severe congenital neutropenia patients. Haematologica. 2020 Mar;105(3):598–609.

275. Zhai J, Zhang S, Chen J, He Q. Autoencoder and its various variants. 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE; 2018. p. 415–9.

276. Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing [review article]. IEEE Comput Intell Mag. 2018 Aug;13(3):55–75.

277. Ahlmann-Eltze C, Huber W. Comparison of transformations for single-cell RNA-seq data. Nat Methods. 2023 May;20(5):665–72.

278. Sen CK. Wound healing essentials: let there be oxygen. Wound Repair Regen. 2009;17(1):1–18.

279. Atala A, Irvine DJ, Moses M, Shaunak S. Wound healing versus regeneration: role of the tissue environment in regenerative medicine. MRS Bull. 2010 Aug 1;35(8).

280. Liu Y, Beyer A, Aebersold R. On the Dependency of Cellular Protein Levels on mRNA Abundance. Cell. 2016 Apr 21;165(3):535–50.

281. Stoeckius M, Hafemeister C, Stephenson W, Houck-Loomis B, Chattopadhyay PK, Swerdlow H, et al. Simultaneous epitope and transcriptome measurement in single cells. Nat Methods. 2017 Sep;14(9):865–8.

282. Kolodziejczyk AA, Kim JK, Tsang JCH, Ilicic T, Henriksson J, Natarajan KN, et al. Single Cell RNA-Sequencing of Pluripotent States Unlocks Modular Transcriptional Variation. Cell Stem Cell. 2015 Oct 1;17(4):471–85.

283. Wang C, Rabadan Ros R, Martinez-Redondo P, Ma Z, Shi L, Xue Y, et al. In vivo partial reprogramming of myofibers promotes muscle regeneration by remodeling the stem cell niche. Nat Commun. 2021 May 25;12(1):3094.

284. Kamimoto K, Adil MT, Jindal K, Hoffmann CM, Kong W, Yang X, et al. Gene regulatory network reconfiguration in direct lineage reprogramming. Stem Cell Reports. 2023 Jan 10;18(1):97–112.

285. Barroca V, Lewandowski D, Jaracz-Ros A, Hardouin S-N. Paternal Insulin-like Growth Factor 2 (Igf2) Regulates Stem Cell Activity During Adulthood. EBioMedicine. 2017 Feb;15:150–62.

286. Songstad AE, Worthington KS, Chirco KR, Giacalone JC, Whitmore SS, Anfinson KR, et al. Connective Tissue Growth Factor Promotes Efficient Generation of Human Induced Pluripotent Stem Cell-Derived Choroidal Endothelium. Stem Cells Transl Med. 2017 Jun;6(6):1533–46.

287. Orlando L, Tanasijevic B, Nakanishi M, Reid JC, García-Rodríguez JL, Chauhan KD, et al. Phosphorylation state of the histone variant H2A.X controls human stem and progenitor cell fate decisions. Cell Rep. 2021 Mar 9;34(10):108818.

288. Balsalobre A, Drouin J. Pioneer factors as master regulators of the epigenome and cell fate. Nat Rev Mol Cell Biol. 2022 Jul;23(7):449–64.

289. Spanò DP, Scilabra SD. Tissue Inhibitor of Metalloproteases 3 (TIMP-3): In Vivo Analysis Underpins Its Role as a Master Regulator of Ectodomain Shedding. Membranes (Basel). 2022 Feb 11;12(2).

290. Mnatsakanyan H, Sabater I Serra R, Salmeron-Sanchez M, Rico P. Zinc maintains embryonic stem cell pluripotency and multilineage differentiation potential via AKT activation. Front Cell Dev Biol. 2019 Aug 30;7:180.

291. Ragina NP, Schlosser K, Knott JG, Senagore PK, Swiatek PJ, Chang EA, et al. Downregulation of H19 improves the differentiation potential of mouse parthenogenetic embryonic stem cells. Stem Cells Dev. 2012 May 1;21(7):1134–44.

292. Desai RV, Chen X, Martin B, Chaturvedi S, Hwang DW, Li W, et al. A DNA repair pathway can regulate transcriptional noise to promote cell fate transitions. Science. 2021 Aug 20;373(6557).

293. Chen P-C, Kuo Y-C, Chuong C-M, Huang Y-H. Niche Modulation of IGF-1R Signaling: Its Role in Stem Cell Pluripotency, Cancer Reprogramming, and Therapeutic Applications. Front Cell Dev Biol. 2020;8:625943.

294. Chen X, He L, Zhao Y, Li Y, Zhang S, Sun K, et al. Malat1 regulates myogenic differentiation and muscle regeneration through modulating MyoD transcriptional activity. Cell Discov. 2017 Mar 14;3:17002.

295. Sharanek A, Burban A, Hernandez-Corchado A, Madrigal A, Fatakdawala I, Najafabadi HS, et al. Transcriptional control of brain tumor stem cells by a carbohydrate binding protein. Cell Rep. 2021 Aug 31;36(9):109647.

296. Zhu B, Xue F, Li G, Zhang C. CRYAB promotes osteogenic differentiation of human bone marrow stem cells via stabilizing β-catenin and promoting the Wnt signalling. Cell Prolif. 2020 Jan;53(1):e12709.

297. Li G, Qin Y, Cheng Z, Cheng X, Wang R, Luo X, et al. Gpx3 and Egr1 Are Involved in Regulating the Differentiation Fate of Cardiac Fibroblasts under Pressure Overload. Oxid Med Cell Longev. 2022 Jun 28;2022:3235250.

298. Izu Y, Birk DE. Collagen XII mediated cellular and extracellular mechanisms in development, regeneration, and disease. Front Cell Dev Biol. 2023 Mar 2;11:1129000.

299. Brady JJ, Li M, Suthram S, Jiang H, Wong WH, Blau HM. Early role for IL-6 signalling during generation of induced pluripotent stem cells revealed by heterokaryon RNA-Seq. Nat Cell Biol. 2013 Oct;15(10):1244–52.

300. Marczenke M, Sunaga-Franze DY, Popp O, Althaus IW, Sauer S, Mertins P, et al. GAS1 is required for NOTCH-dependent facilitation of SHH signaling in the ventral forebrain neuroepithelium. Development. 2021 Nov 1;148(21).

301. Lee TC, Li L, Philipson L, Ziff EB. Myc represses transcription of the growth arrest gene gas1. Proc Natl Acad Sci USA. 1997 Nov 25;94(24):12886–91.

302. Grassi ES, Pietras A. Emerging roles of DLK1 in the stem cell niche and cancer stemness. J Histochem Cytochem. 2022 Jan;70(1):17–28.

303. Peters IJA, de Pater E, Zhang W. The role of GATA2 in adult hematopoiesis and cell fate determination. Front Cell Dev Biol. 2023 Nov 14;11:1250827.

304. Liu H, Sun L, Zhao H, Zhao Z, Zhang S, Jiang S, et al. Proteinase 3 depletion attenuates leukemia by promoting myeloid differentiation. Cell Death Differ. 2024 Jun;31(6):697–710.

APPENDICES

A. Appendix to chapter 3

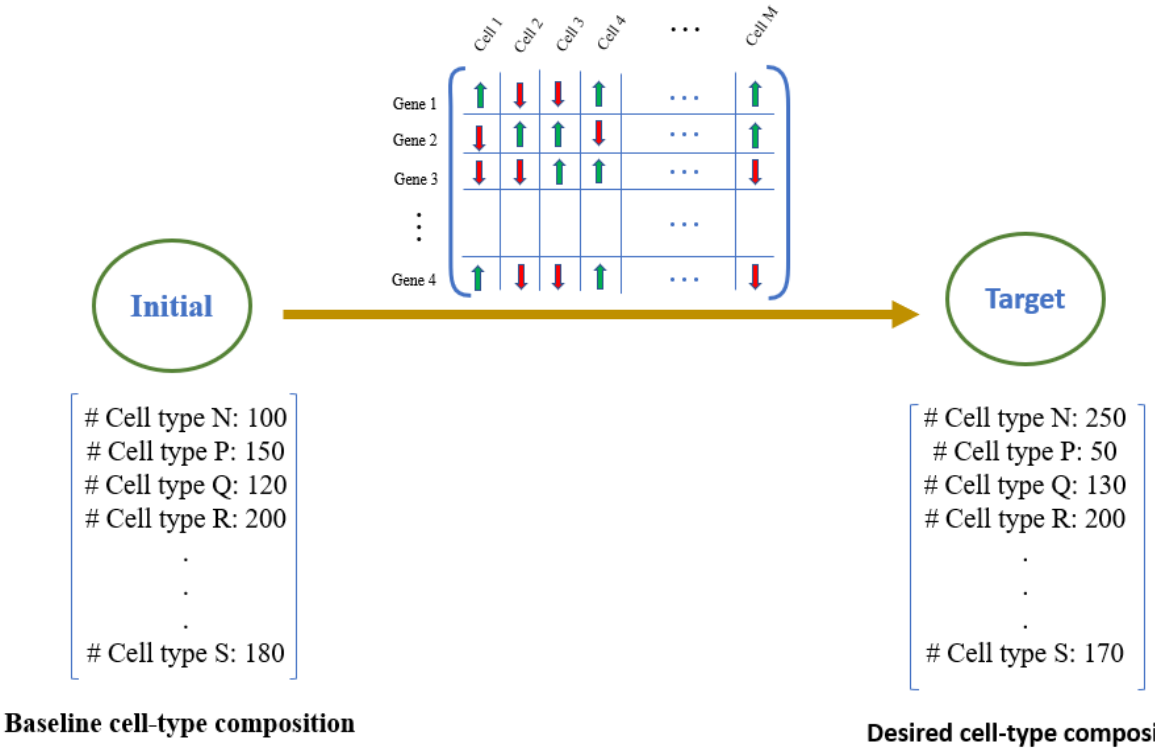


Figure 9-1 Supplementary Fig. 1: Gene expression perturbations to change the cell-type distribution. This figure demonstrates the transition of an initial cell type distribution to a desired target distribution through gene-level adjustments. The circles represent the system's state, with the frequencies of cell types indicated below.

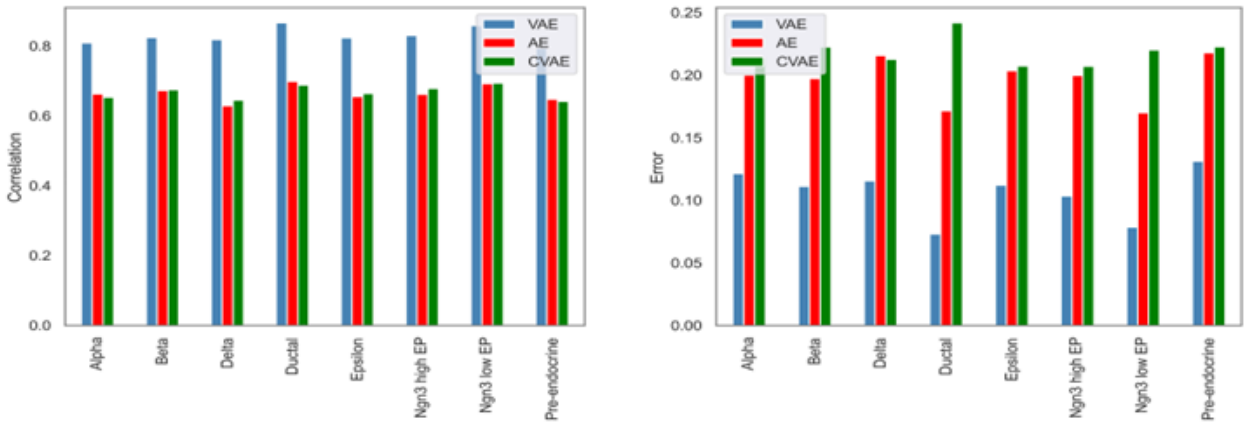


Figure 9-2 Supplementary Fig. 2: Comparison of autoencoder architectures for analyzing data for endocrine development in the mouse pancreas. a, comparison of the input-output correlation for the AE, the variational autoencoder (VAE), and the conditional variational autoencoder (CVAE). b, the mean square error of the three autoencoder architectures' input and output. VAE performs better than other architectures for this data.

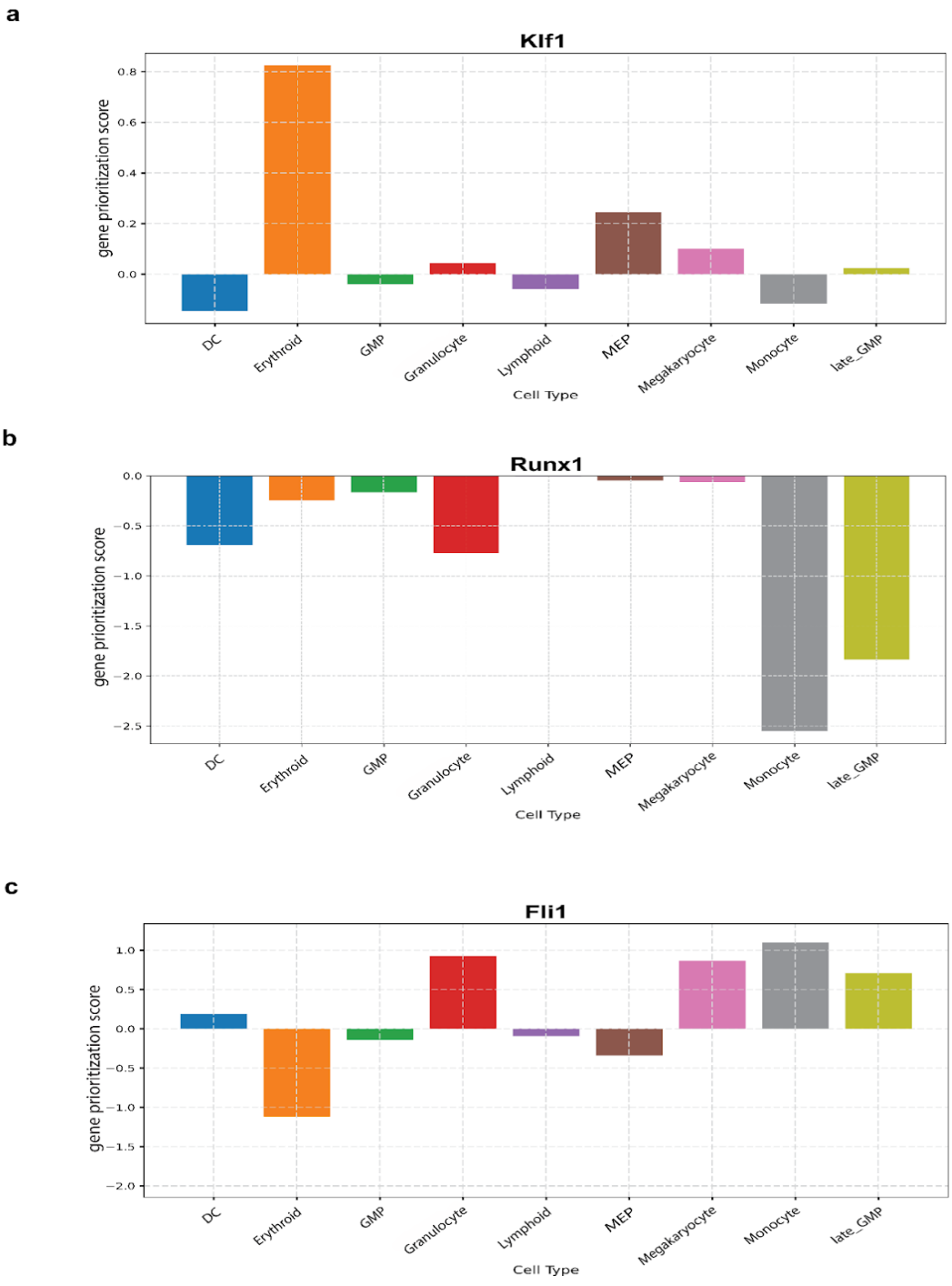


Figure 9-3 Supplementary Fig. 3: Gene prioritization score for the mouse hematopoiesis data. a, b, c, Fatecode accurately determines gene prioritization scores for various genes, including *Klf1*, *Runx1*, and *Fli1*, across different cell types.

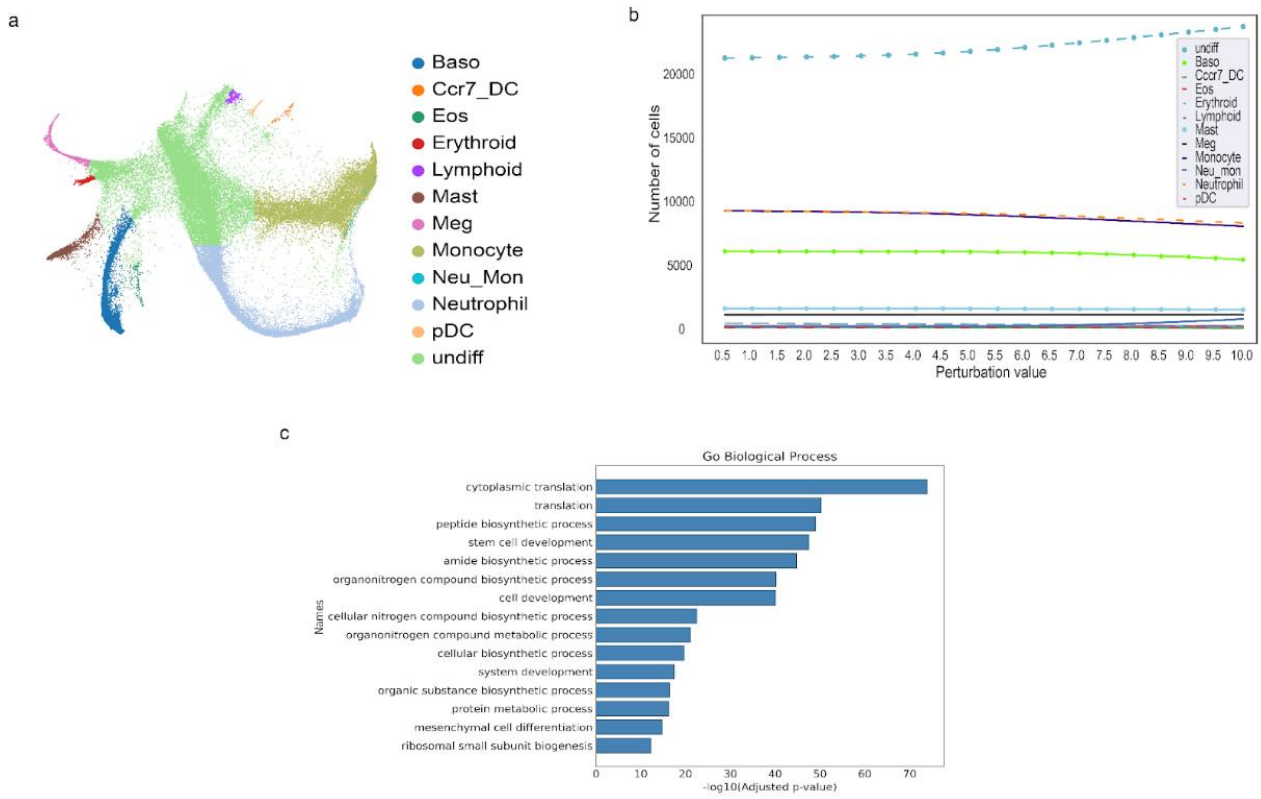


Figure 9-4 Supplementary Fig. 4: Fatecode analysis of hematopoiesis data identifies master regulators governing cell switching dynamics. a, visualization of the hematopoiesis dataset from Weinreb et al. hematopoietic progenitors differentiate into different cell types such as mast cell (Ma), basophil (Ba), eosinophil (Eos), megakaryocyte (Mk), lymphoid precursor (Ly), migratory dendritic cell (mDC) and plasmacytoid dendritic cell (pDC), erythrocyte (Er), neutrophil (Neu), monocyte (Mo). b, The effect of different perturbation sizes of a node in the latent layer on the cell distribution. c, gene set enrichment analysis results. Gene ontology (GO) biological processes enrichment analysis shows significant process terms related to mouse hematopoiesis, stem cell development, and mesenchymal cell differentiation.

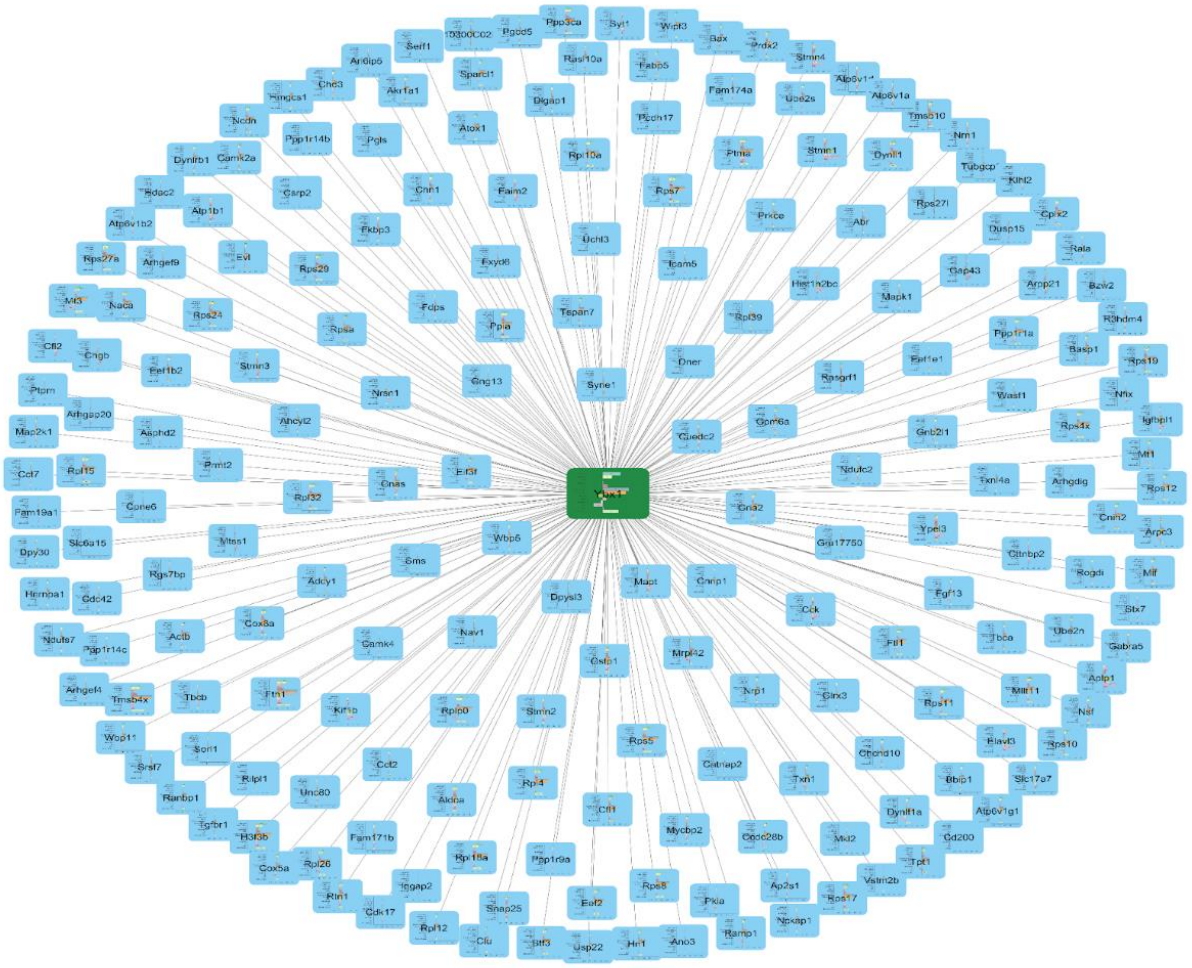


Figure 9-5 Supplementary Fig. 5: Gene Regulatory Network of Ybx1 and its downstream target genes along with gene prioritization scores. The GRN was constructed using SCENIC, by filtering the top 2000 interactions with the highest SCENIC Importance Measure (IM) scores. Additionally, the top 400 predicted master regulators from Fatecode were mapped onto the GRN, and the resulting network is presented here using a network bar chart in which each value of a bar plot shows the Fatecode gene prioritization score of the gene for that cell type.

Supplementary Note 1- Hyperparameter search

Hyperparameter tuning was conducted using a grid search approach. We explored various combinations of hyperparameters, including learning rate, batch size, number of layers, number of neurons per layer, and activation functions. The hyperparameter space for each parameter was defined as follows:

- Autoencoder type: [AE, VAE, CVAE]
- Activation function: [LeakyReLU, Relu, linear]
- Learning rate: [0.001, 0.01, 0.1]
- Batch size: [400, 500, 600]
- Number of hidden layers(encoder): [1, 2]
- Number of neurons in latent_layer: [input_size/40, input_size/60, input_size/80, input_size/ 100, input_size/125, input_size/150]

B. Appendix to chapter 4

Supplementary Figures

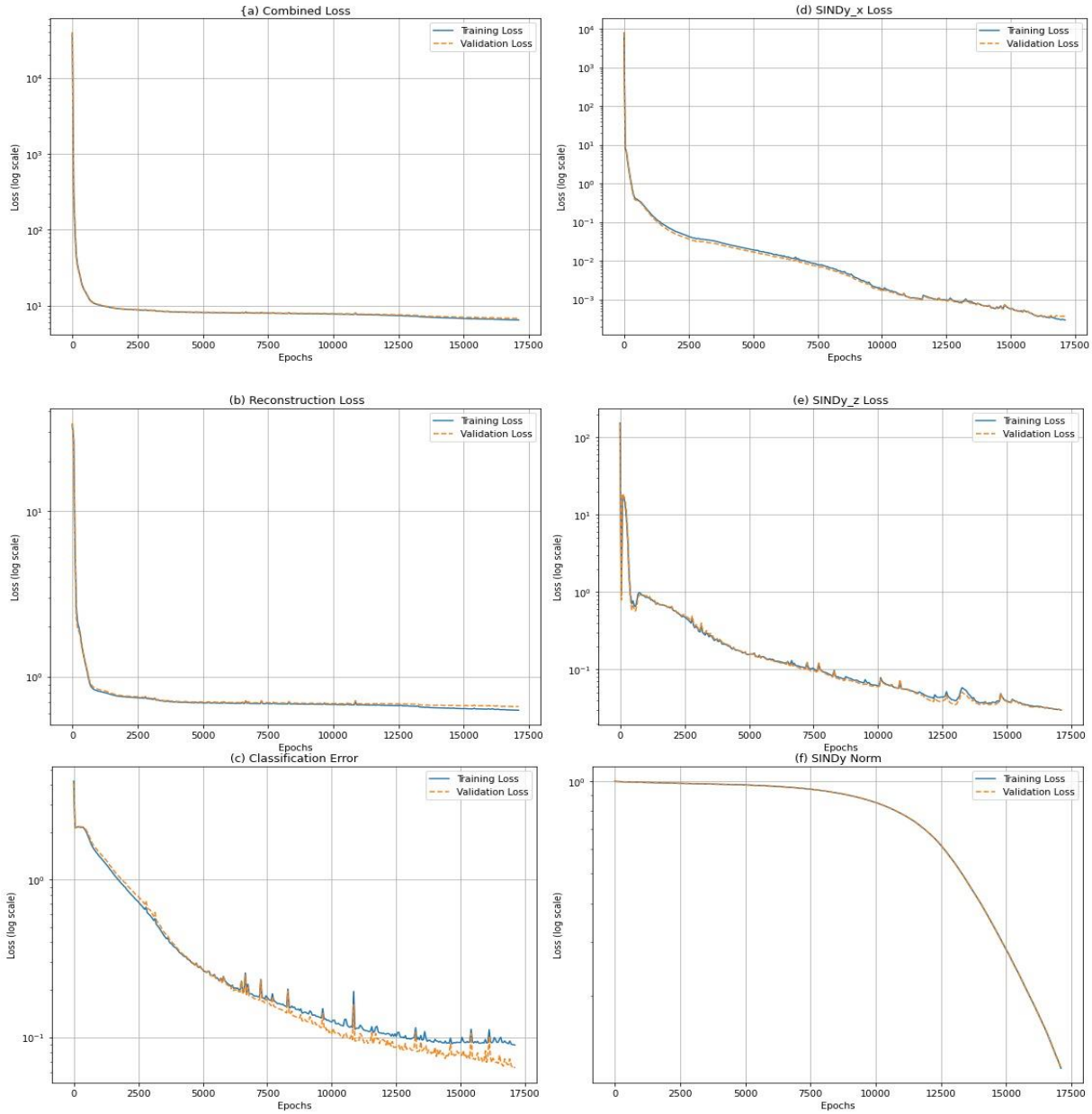


Figure 9-6 Supplementary Figure 1: Training and validation loss curves for training CLERA on the data generated using SERGIO. The low losses in (a)-(e) indicate a high accuracy and strong generalization. Refer to the Methods section for the description of each loss term.

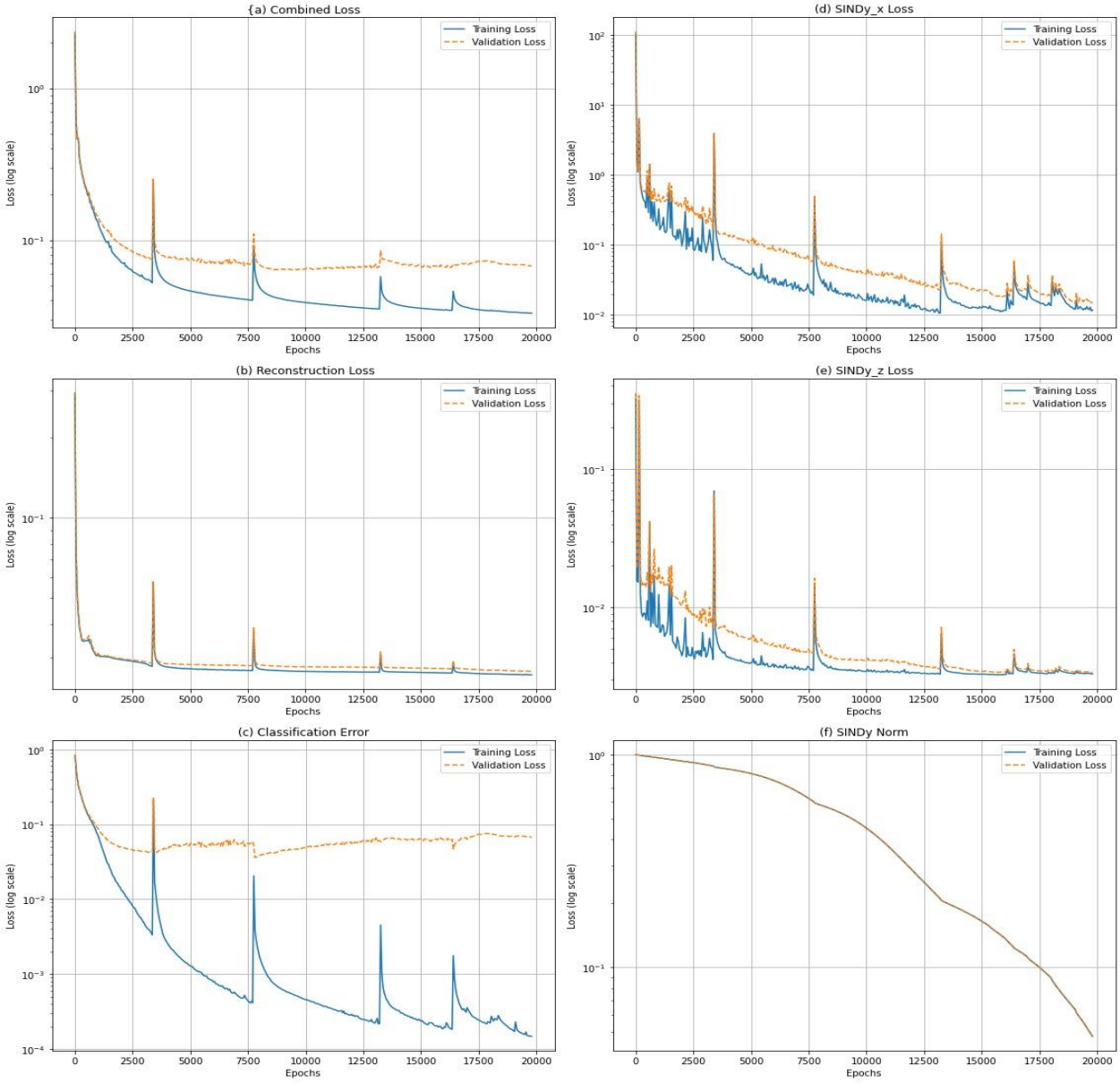


Figure 9-7 Supplementary Figure 2: Training and validation loss curves for training CLERA on the pancreas data. The low losses in (a)-(e) indicate that CLERA successfully builds the latent layer with the desired characteristics. Refer to the Methods section for the description of each loss term.

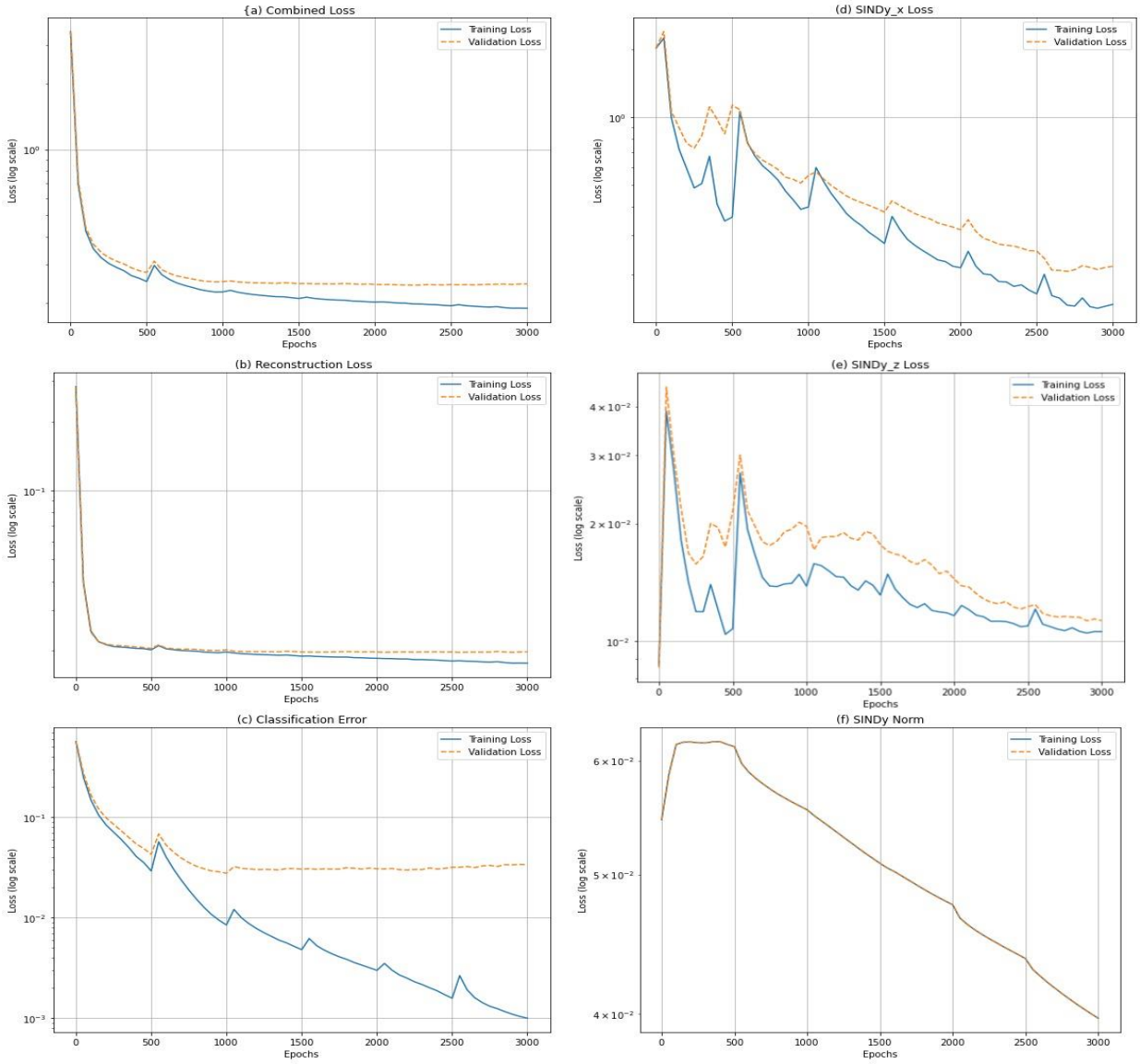


Figure 9-8 Supplementary Figure 3: Training and validation loss curves for the bone marrow data in the transfer learning setting. Due to transfer learning, the model converges in fewer epochs as compared to the training of the Pancreas data.

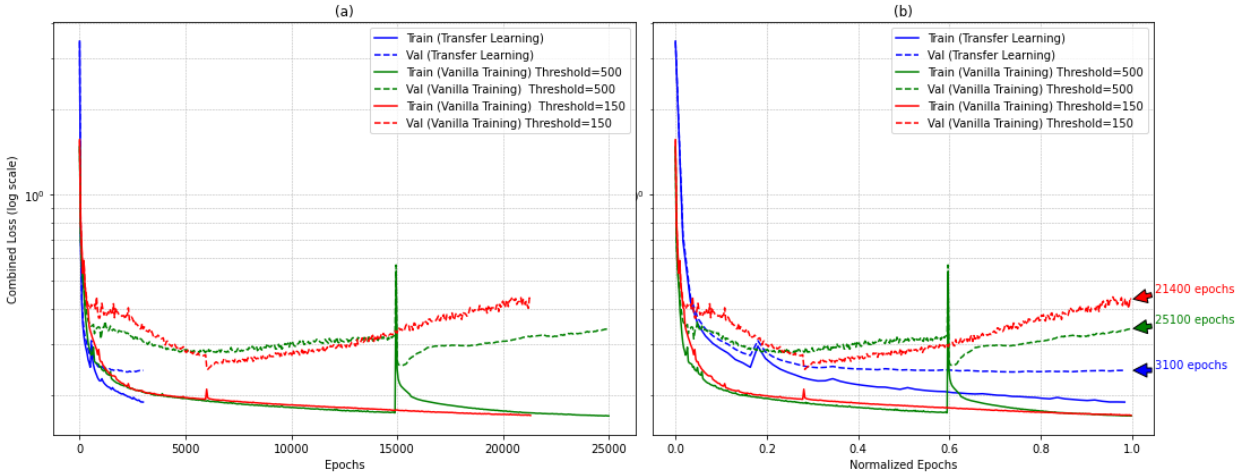


Figure 9-9 Supplementary Figure 4: Training comparison on the bone marrow dataset using transfer learning (SINDy threshold frequency- 500) and vanilla training with two different thresholds (SINDy threshold frequency- 150 and 500). (a) shows the combined loss plotted against the number of epochs, illustrating the slower convergence and signs of overfitting in the vanilla models, particularly with the lower threshold (red). (b) presents the data with normalized epochs, highlighting the efficiency of transfer learning, which reaches stability in a fraction of the time compared to vanilla training. While the vanilla models require most of the training period to converge (especially with a lower threshold), the transfer learning model generalizes well much earlier in the training process, as shown by the stabilized validation loss.

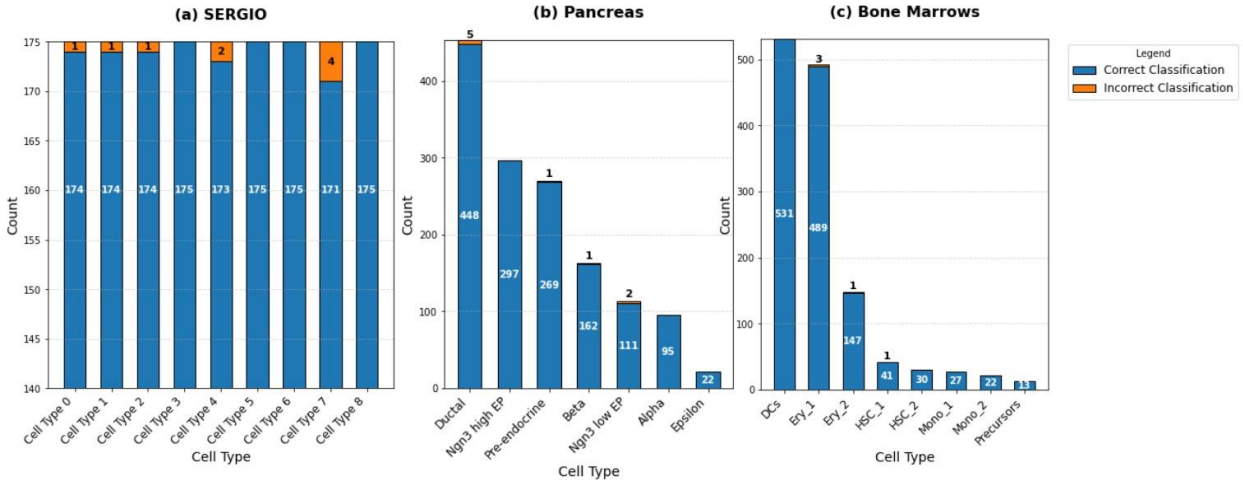


Figure 9-10 Supplementary Figure 5: Classification results of (a) SERGIO generated data (b) Pancreas data and (c) Bone marrows data. After training CLERA on different datasets. the prediction of the cell type for all the preprocessed input (validation and training) by the classification network is shown, The classifier achieves high accuracy with very few misclassifications. Even for cell types with fewer instances, the model can accurately predict the cell type from the latent space. This strong classification performance suggests that the latent space captures cell-type-specific features. Accurate classification ensures that the predicted ODEs reliably infer relationships and can predict cellular behavior across different conditions.

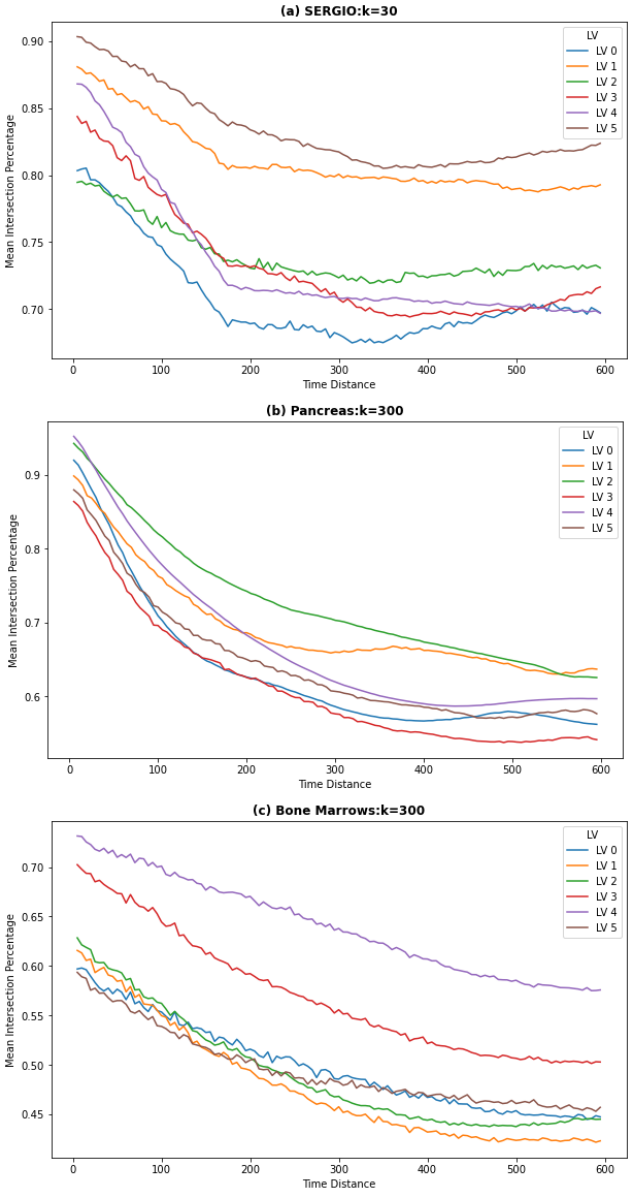


Figure 9-11 Supplementary Figure 6: Mean intersection percentage between top-k genes of different latent variables present at various time distances for (a) SERGIO generated dataset (b) Pancreas dataset (c) Bone marrows dataset. The mean intersection percentage between the top-k genes for various time distances in the SERGIO-generated dataset with k=30 (left), the Pancreas dataset with k=300 (middle), and Bone marrow data with k=300 (right) datasets. The intersection percentage, plotted for each latent variable, is higher for closer time steps and gradually decreases as the time distance grows. This shows the model's ability to capture temporal coherence in gene expression patterns, where similar gene sets are shared between adjacent time points, reflecting the gradual changes in biological processes. The consistency decreases with larger time distances, indicating the differences in active gene programs.

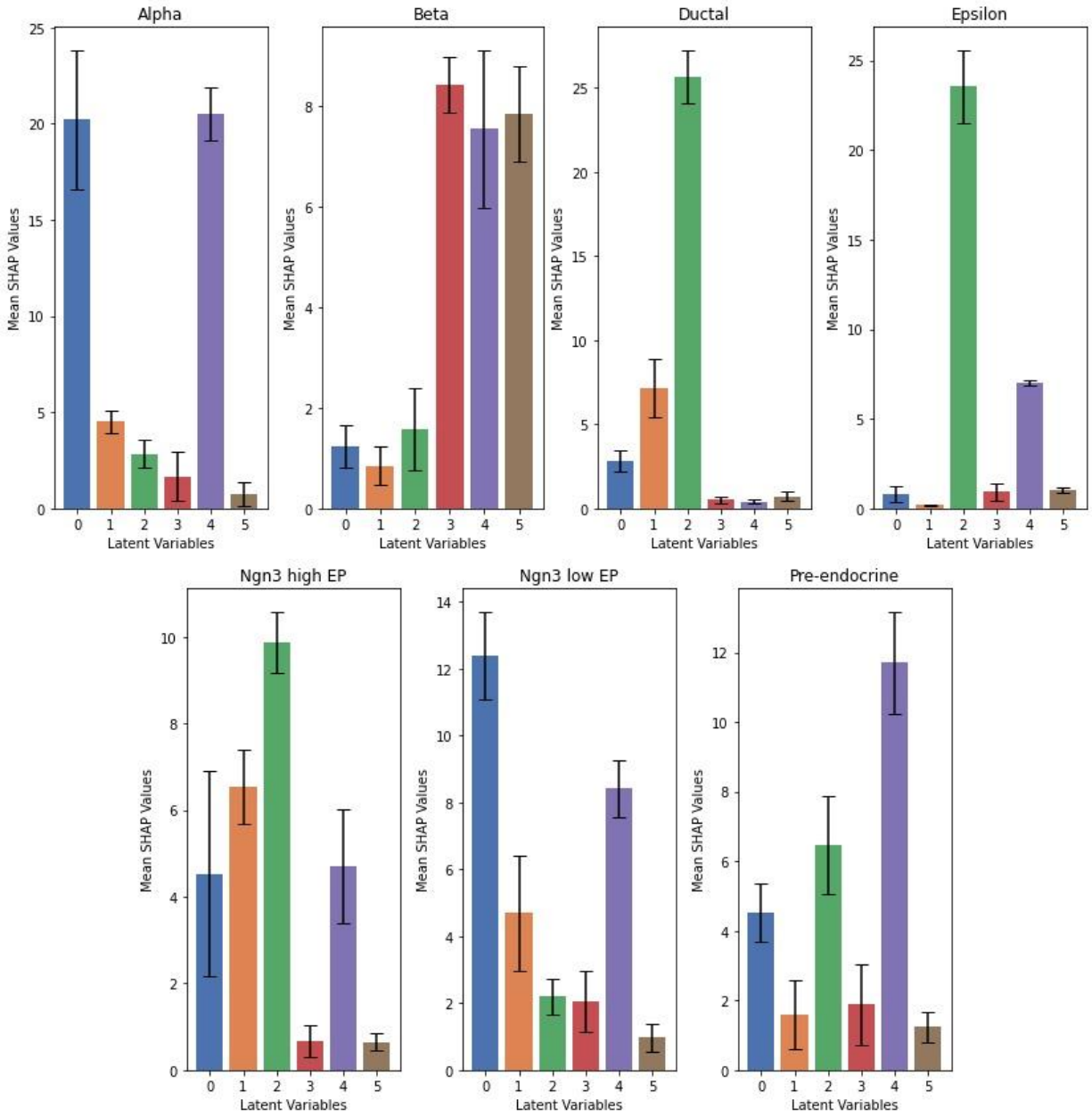


Figure 9-12 Supplementary Figure 7: The mean and standard deviation of SHAP values between each cell type and latent variable for the pancreas data. The SHAP values calculated using the classifier network are plotted which show that each cell type is primarily driven by one or more latent variables for classification. This demonstrates that certain latent variables are key drivers in predicting particular cell types, suggesting a cell type-level abstraction within the latent space.

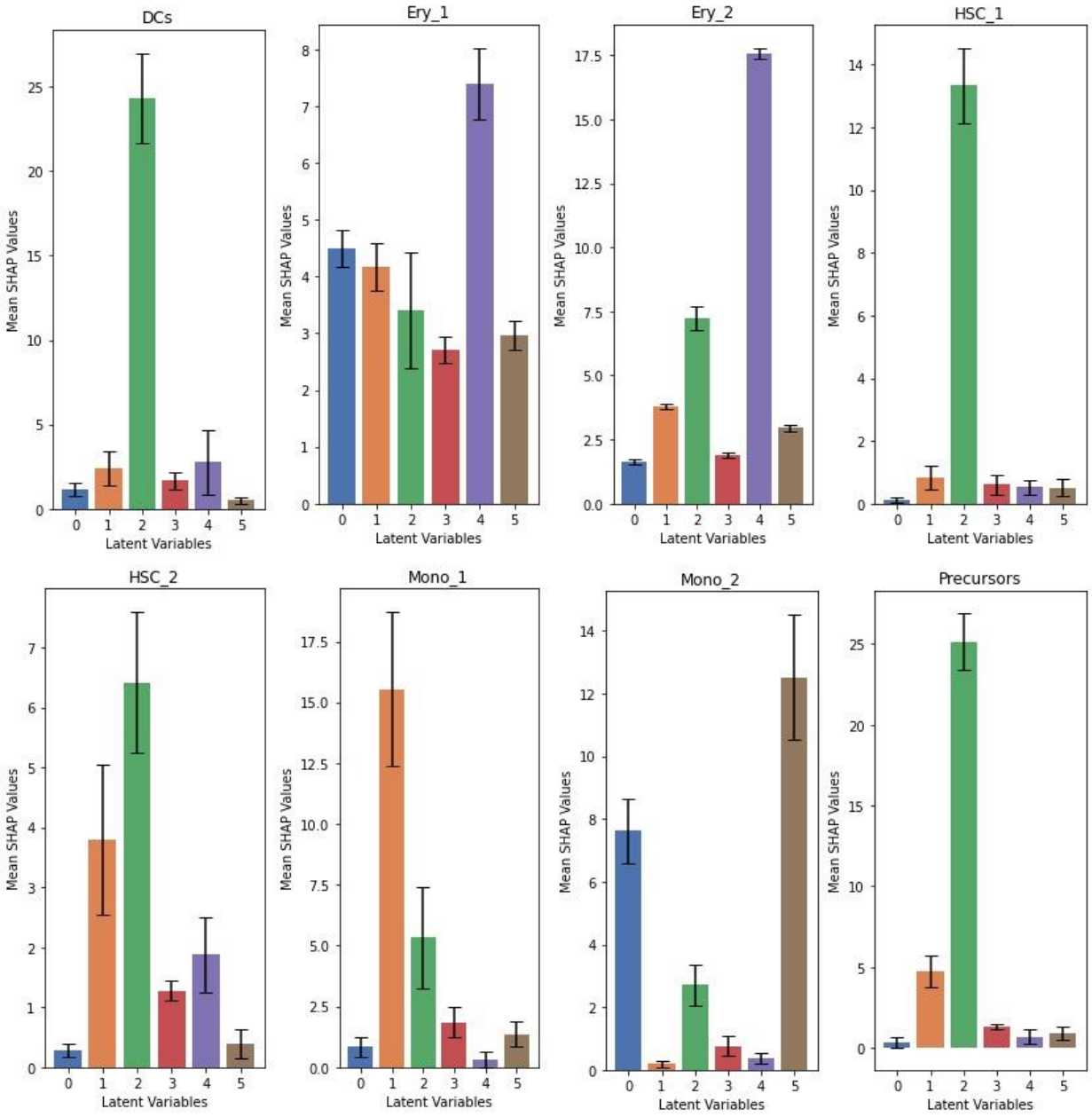


Figure 9-13 Supplementary Figure 8: SHAP Value Analysis of Latent Variables in hematopoietic differentiation. This figure illustrates the mean and standard deviation of SHAP values for each cell type and latent variable. The values, derived from the classifier network, demonstrate that specific latent variables significantly influence cell type classification, indicating a cell type abstraction in the latent space.

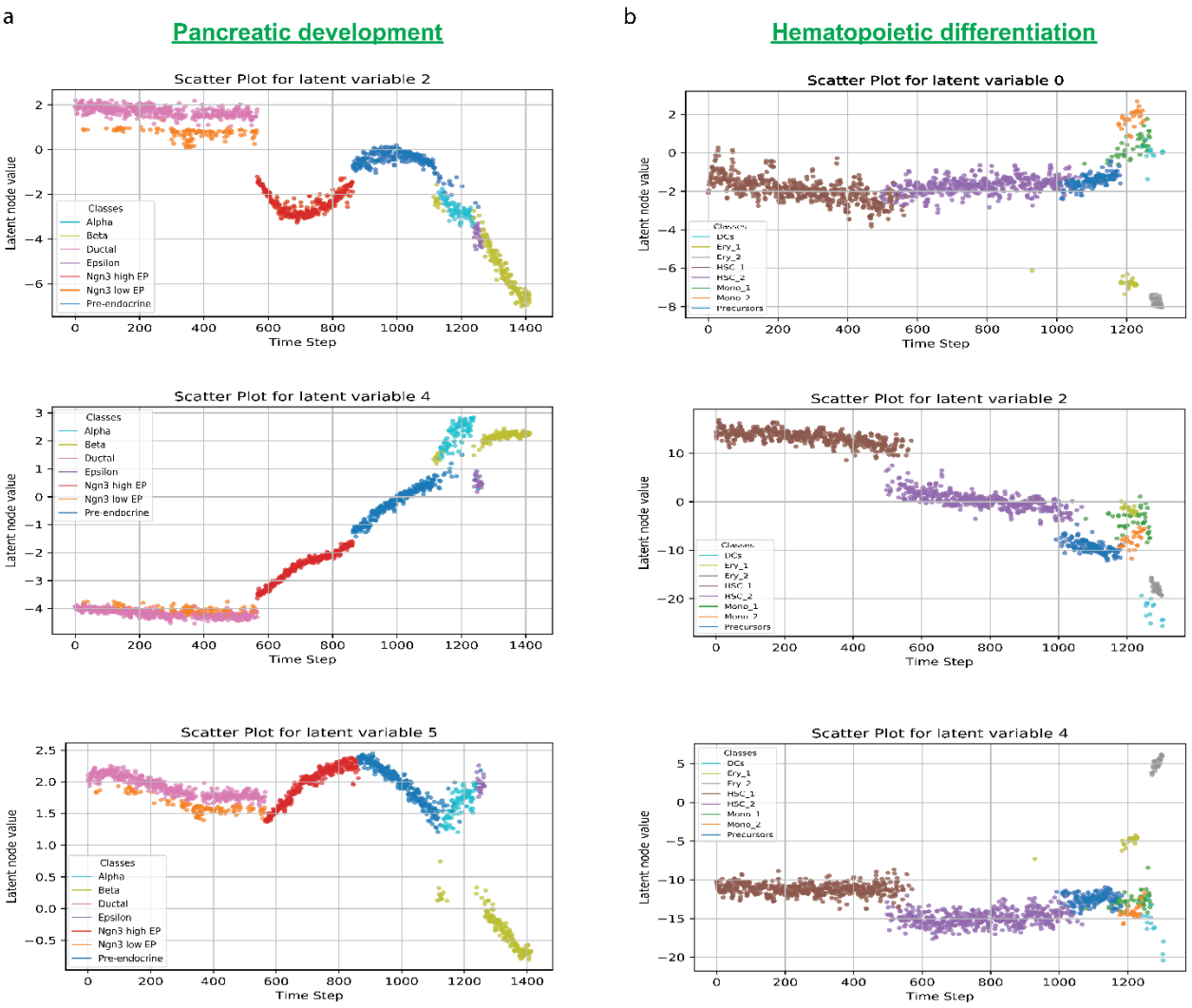


Figure 9-14 Supplementary Figure 9: Latent Variable Time Series discovered using CLERA for (a) pancreas dataset and (b) Bone marrows dataset. Colors have been assigned as the cell type labels. The latent space demonstrates clear cell type differentiation, with distinct cell types following unique trends and values over time. This pattern indicates that the learned latent space successfully captures cell type-specific dynamics, ensuring that the model reflects latent qualities at a cell type level granularity.

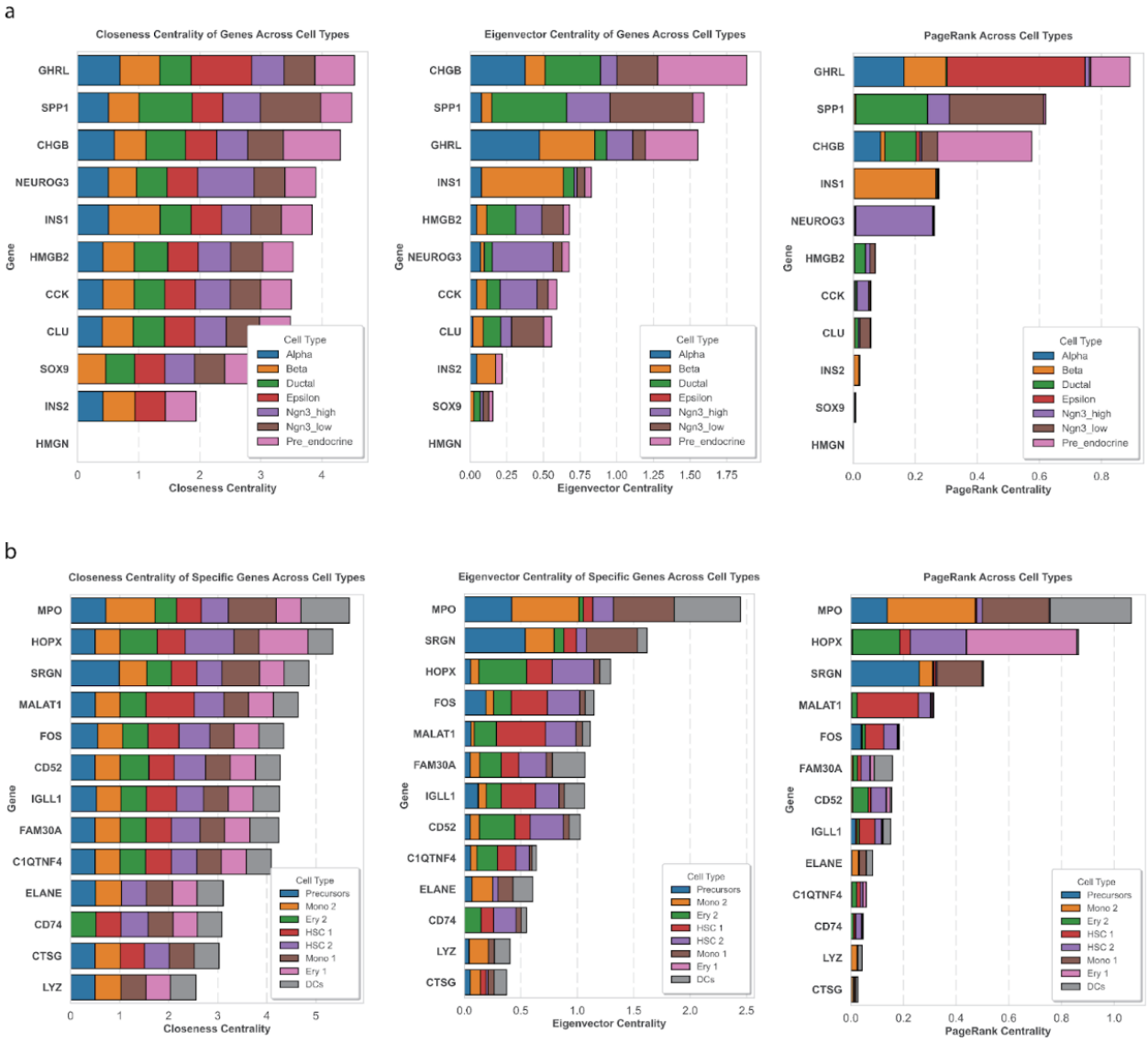


Figure 9-15 Supplementary Figure 10: Results of centrality measures on (a) Pancreas dataset and (b) Bone marrows dataset. The figure shows the distribution of three centrality measures—Closeness, Eigenvector, and PageRank—for the most central genes across different cell types in (a) the pancreas dataset and (b) the bone marrow dataset. In both datasets, some genes exhibit consistently high centrality values across multiple cell types, indicating their importance in the gene regulatory networks.

Supplementary Notes

Supplementary Note 1: Training Hyperparameters

Hyperparameter tuning is conducted to train different datasets. We explore various combinations of hyperparameters, including learning rate, batch size, number of layers, number of neurons per layer, and activation functions. Here is a list of different training hyperparameters that we try:

- Sliding Window Length- 2, 4, 7, 10
- Sliding Window Stride- 2,3,4
- Number of Autoencoder hidden layers (the encoder part)- 2, 3, 4, 5
- Size of latent dimensions- 4, 5, 6, 7, 10, 15, 20
- Learning Rate-1e-3, 1e-4, 1e-5
- Activation Function- ELU, sigmoid, ReLu
- SINDy Coefficient Initialisation- Xavier Initialisation, Normal Initialisation, Uniform Initialisation of 1, Random Sampling from {-1,1}, Random Sampling from {-5,5}
- SINDy Threshold Interval- 150, 500
- Classifier width- 7, 8, 12
- SINDy Threshold Coefficient- 1e-3, 1e-2, 1e-1
- Number of refinement epochs- 0, 150, 500, 1000

Along with these, we also experiment with various CLERA's loss weights

- Reconstruction Loss Weight- 1e-2, 1e-3, 1e-4
- SINDy_z Weight- 1e-2, 1e-3, 1e-4
- SINDy_x Weight- 1e-1, 1e-2, 1e-3
- SINDy regularization Weight- 1e-4, 5e-3, 8e-3, 1e-2,
- Classifier Weight- 5e-1, 1
- Autoencoder regularization- 1e-6, 1e-5, 5e-5

Supplementary Note 2: Inference Hyperparameters

CLERA generates SHAP values between genes and latent variables across all time steps. To build an interaction graph at the cell-type level, we heuristically average the SHAP values across time steps for each cell type, to enable analysis at a broader cell-type granularity. For a more granular analysis, using SHAP values from individual time steps can provide more precise insights. Notably, it is observed that taking the absolute SHAP values instead of signed SHAP yields better results.

We formulate the interaction between genes and the latent space as a bipartite graph by selecting the top K genes per latent variable based on their SHAP values. For the bone marrow and pancreas development data, we set K=300, while for the SERGIO-generated dataset, we use K=30. A smaller K might overlook important genes, while a larger K can introduce computational inefficiencies and also capture less relevant genes. We experiment with various values of K, such as 100, 200, and 300 for both datasets, and 20, 30, and 60 for the SERGIO-generated data, to determine the optimal balance.

Singular analysis of the top K genes captures gene-latent space interactions but neglects direct relationships between latent variables. To enhance inference, we introduce latent variable interdependencies identified by the predicted ordinary differential equations. For each latent variable, its corresponding right-hand side in the ODE defines its direct dependencies on other latent variables. These dependencies are incorporated as edges in the graph, to represent both gene-latent space interactions and latent variable relationships. We apply Louvain clustering to the resulting graph to identify potentially co-regulated genes. This enables the discovery of active gene programs.

Selecting the top E edges during graph rewiring is essential for identifying central genes and key interactions. We experiment with different E values: 500, 600, and 700 for the pancreas development and bone marrow datasets, and 50, 60, and 70 for the SERGIO-generated dataset to explore the trade-off between network sparsity and capturing meaningful indirect relationships.

This allows us to fine-tune the balance between retaining crucial regulatory interactions and minimizing unnecessary complexity in the network.

Supplementary Tables

Hyperparameters	**Value**
Sliding Window Length	4
Number of Autoencoder hidden layers (encoder part)	2
Size of latent layer	6
Autoencoder Network Shape	[2000, 512, 64, 6, 64, 512, 2000] *
Activation	ReLU
Learning Rate	1e-3
Classifier Width	8 with architecture as [6, 8, num_cell_types]*
SINDy Coefficient Initialisation	Random Sampling from {-1,1}

SINDy Threshold Interval	150		
SINDy threshold Coefficient	0.1		
SINDy library terms	Constant, Polynomial (upto 3rd order), sin(z), e^z, $\frac{1}{1+z^2}$		
Maximum Active terms	37		
Loss Weights	Loss Type	Pancreas	Bone Marrows
	Reconstruction	1	10
	SINDy_z	1e-2	1e-2
	SINDy_x	1e-1	1e-1
	SINDy regularization	8e-3	1e-2
	Classifier	5e-2	1e-5
	Autoencoder regularization	1e-5	1e-6

Table 9-1 Supplementary Table 1: The hyperparameters used to train CLERA on bone marrow and pancreas development data. Several hyperparameters are tested for training the models. To evaluate the performance of these choices, we used metrics such as the training and validation curves, the number of active terms in the discovered ODE, the structure of the identified ODE, and the Gene Set Enrichment Analysis (GSEA). *Each consecutive pair in this list corresponds to a linear layer followed by an activation function. Activation is not applied after the last layer.

C.Appendix to chapter 5

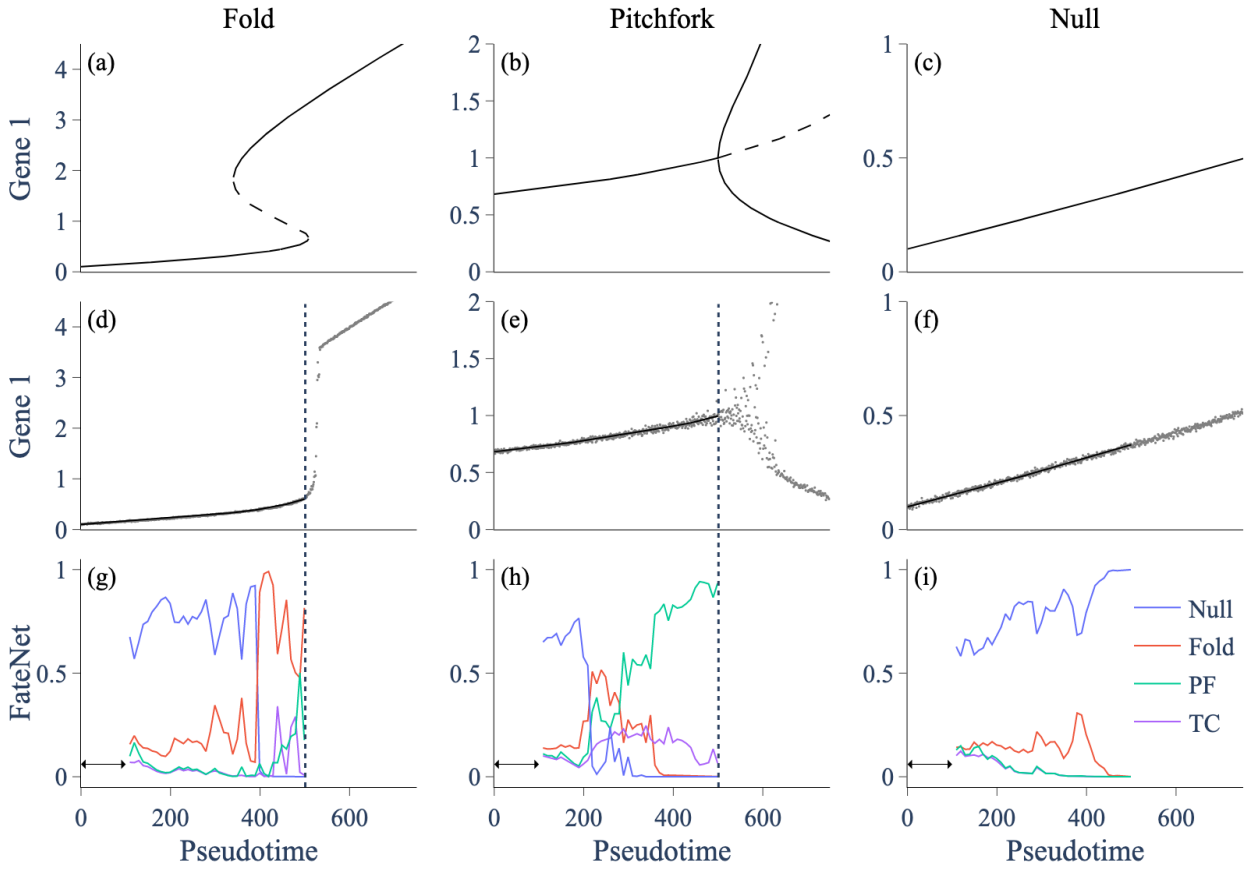


Figure 9-16 Supplementary Figure 1*:* Simulations and predictions in the simple gene regulatory network model using a smaller noise amplitude of 0.01. Refer to the caption of Figure 1 for further details.

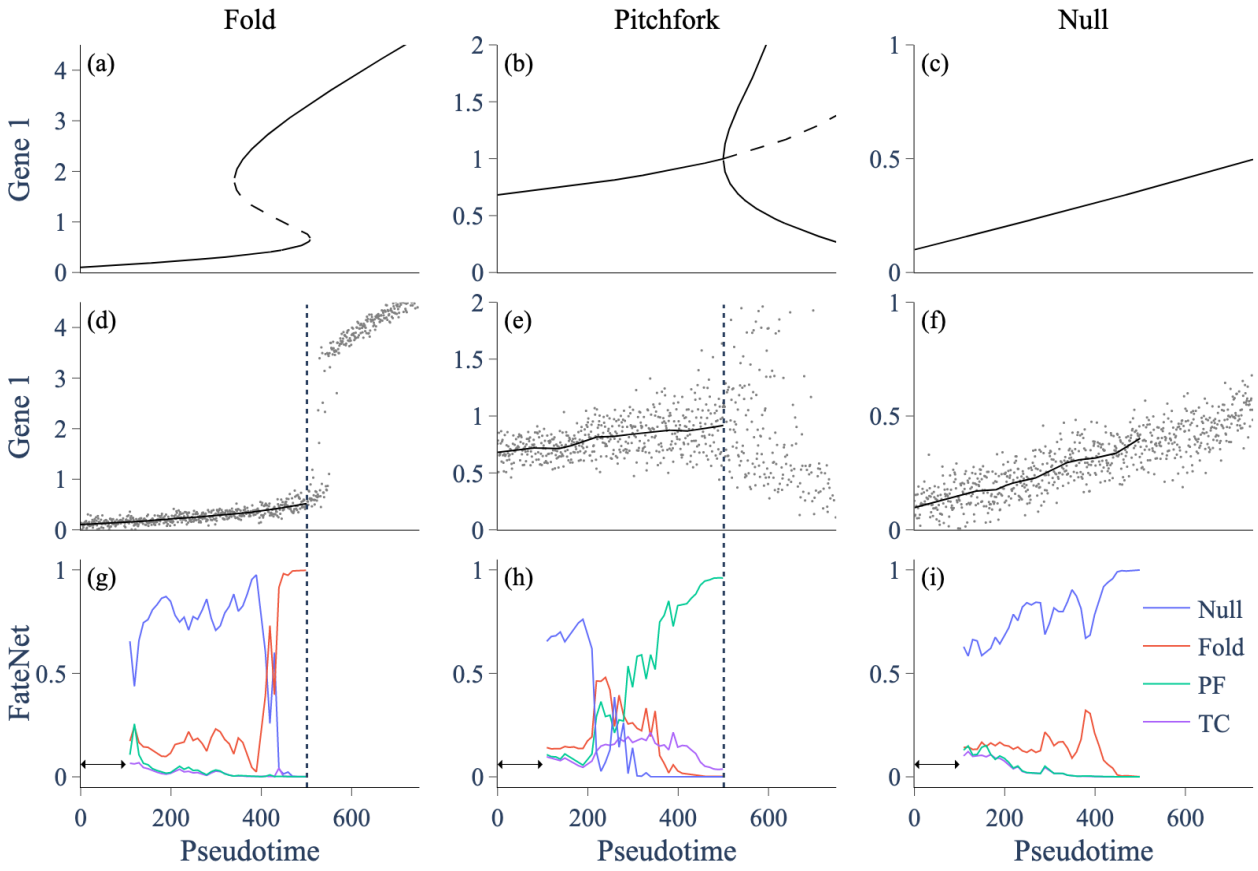


Figure 9-17 Supplementary Figure 2*:* Simulations and predictions in the simple gene regulatory network model using a larger noise amplitude of 0.1. Refer to the caption of Figure 1 for further details

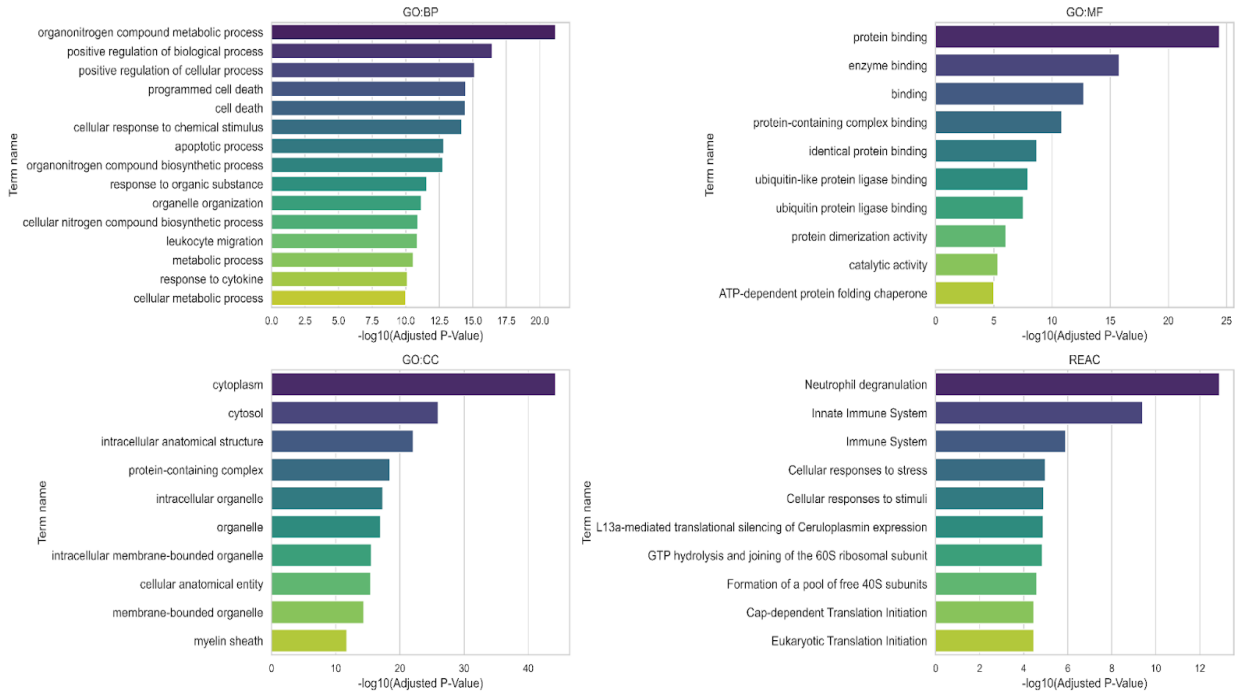


Figure 9-18 Supplementary Figure 3: Gene Ontology (GO) term and Reactome pathway enrichment analysis for the top 250 significant genes. a) GO: Biological Process (BP) showing the enriched biological processes such as different metabolic processes, cellular response to different stimulus, and cell apoptosis indicating relevant biological activities associated with these genes. b) GO: Molecular Function (MF) highlights the enriched molecular functions including protein binding and enzyme binding. c) GO: Cellular Component (CC) displays the enriched cellular components such as cytoplasm and intracellular organelles, suggesting the subcellular localizations where these genes are active. d) The REAC panel represents the enriched pathways from the Reactome database, including immune system processes and cellular responses to stress, which are related to the process of neutrophil development.

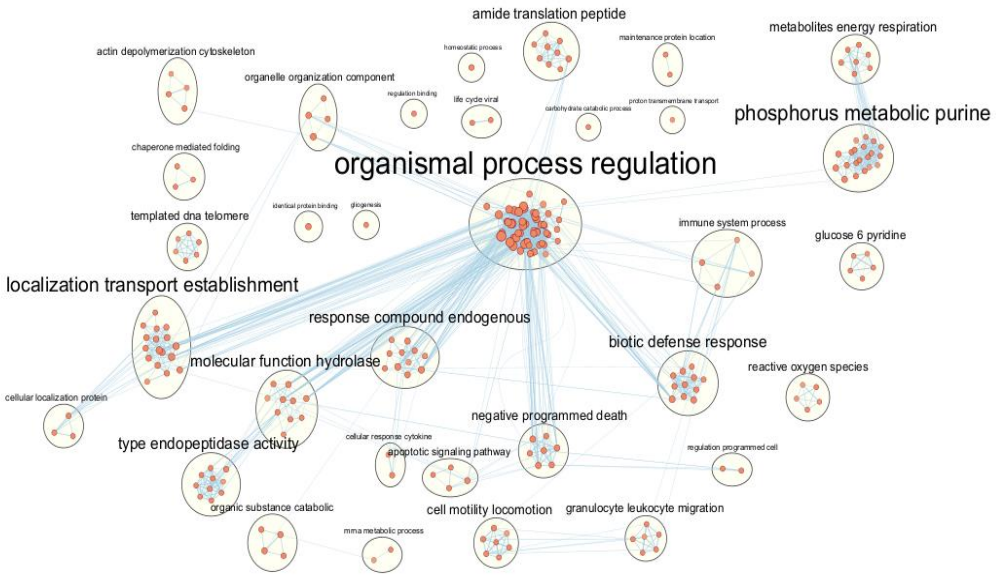


Figure 9-19 Supplementary Figure 4: Enrichment map analysis of biological pathways. The figure shows the complex connections and interactions between different active biological processes. Blue edges of varying thickness connect the nodes, indicating the strength of the connection between pathways by the number of mutual genes. To simplify the network, only edges and nodes meeting the cutoff (Node Cutoff P-value > 0.01, Edge Cutoff > 0.35) are shown.

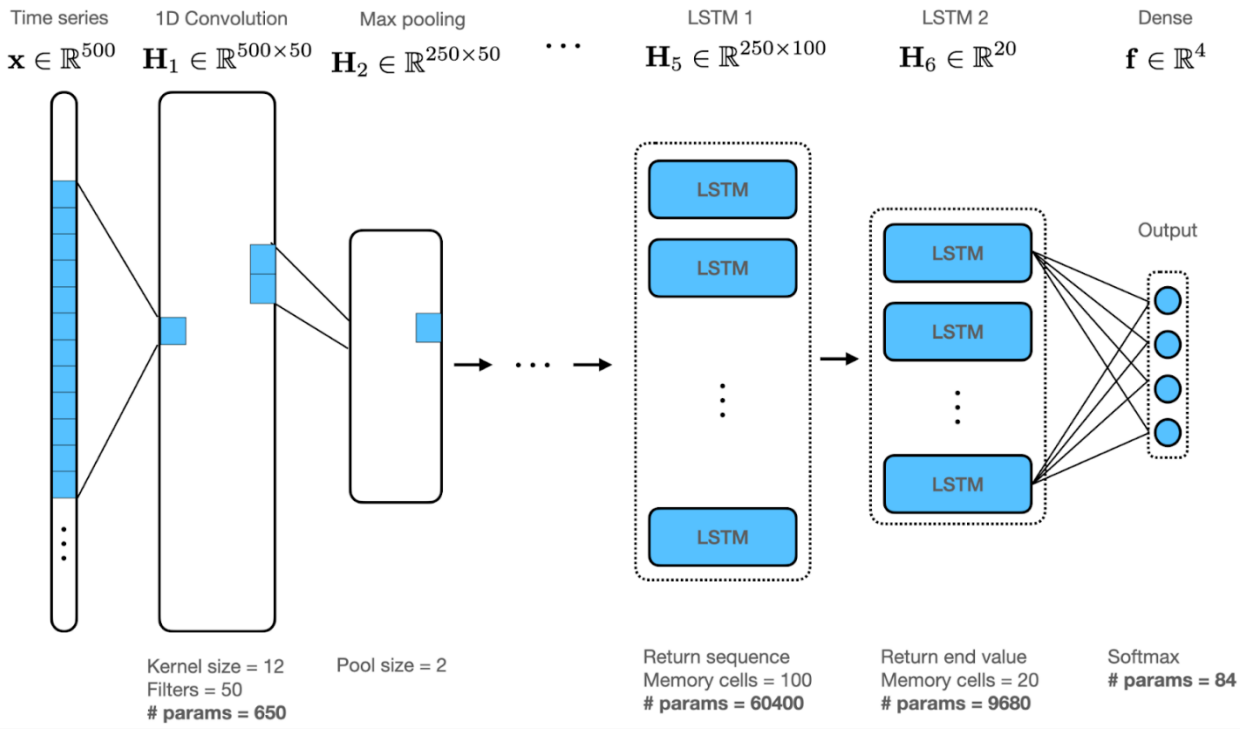


Figure 9-20 Supplementary Figure 5: Architecture for the neural networks in FateNet. Each network consists of 2 convolutional and 2 long-short-term memory (LSTM) layers. Hyperparameters are shown for Network 1. The input to the network is a time series of length 500. This is passed through a 1D convolution, with a kernel size of 12, 50 filters, and the ReLU activation function. Padding is applied to the ends of the time series to maintain the input dimension. This is then passed through a max pooling operation with pool size 2, reducing the dimension by a factor of 2. This process repeats through another convolution and max pooling layer. This is then passed to the first LSTM layer with 100 memory cells. For this layer, the cells return the entire sequence. This then enters the second LSTM layer with 20 memory cells, which returns only the end value of the sequence. Each memory cell is then connected with the 4 output nodes via a dense layer. The output is passed through a softmax filter to obtain a probability distribution over the 4 possible outcomes.

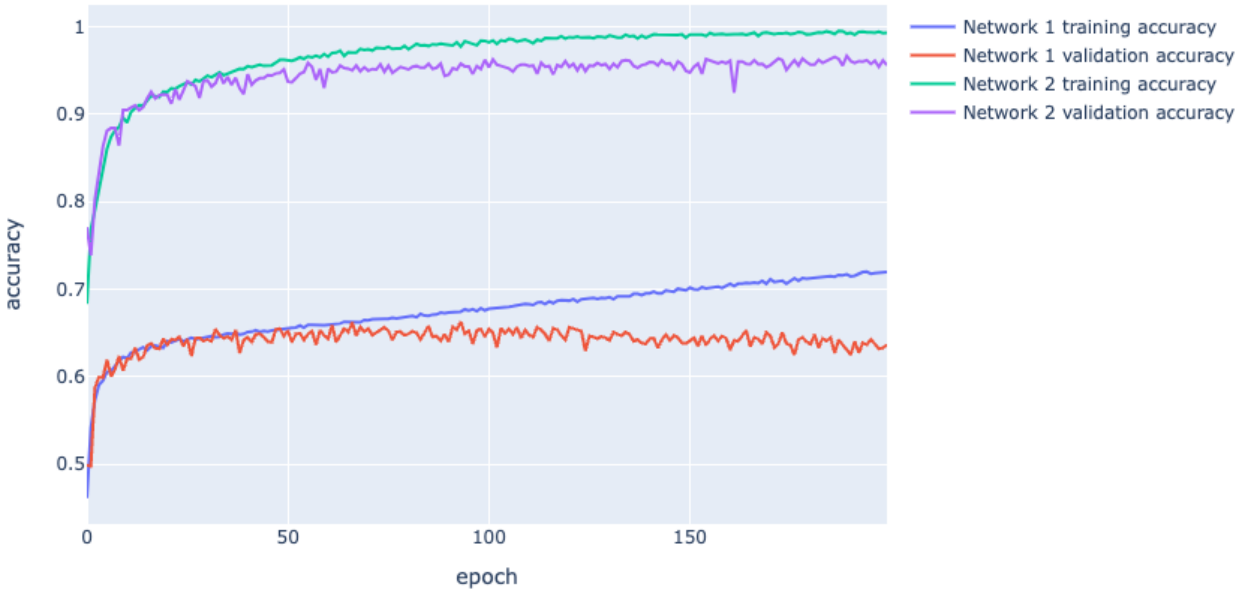


Figure 9-21 Supplementary Figure 6: training and validation accuracy vs. epoch for Network 1 and Network 2 during the training phase. The accuracy on the validation set peaks at epoch 71 and 154, respectively, at which point the models are selected.

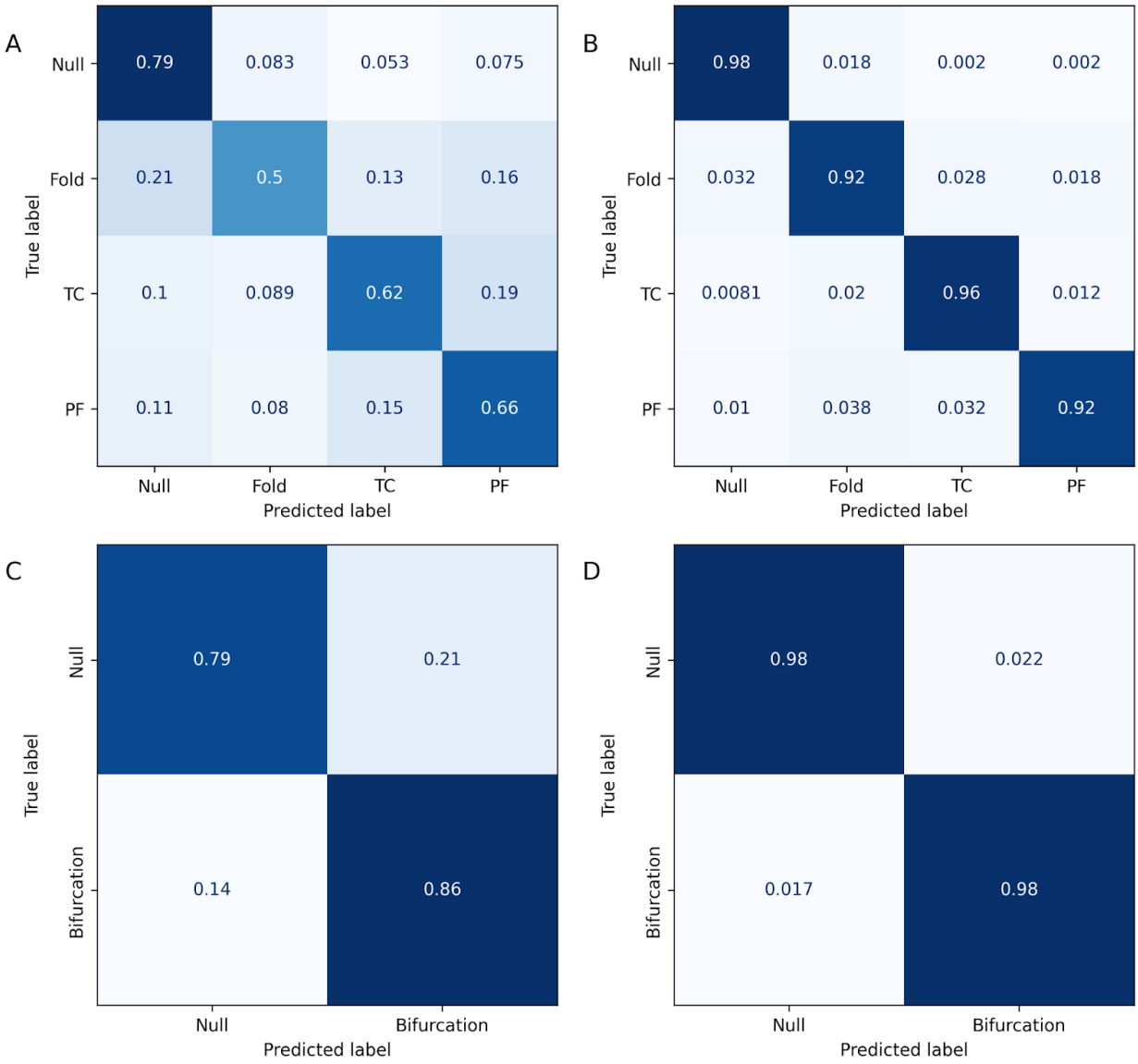


Figure 9-22 Supplementary Figure 7: Confusion matrices for Network 1 and Network 2 when evaluated on their test sets from the library of generated dynamical systems. We show performance on the multi-class classification problem, where the model must predict the specific event, and the binary classification problem, where the model only needs to predict whether or not a bifurcation is approaching. Cell values show (row-)normalized classification rates for each class. (A) Network 1 on the multi-class classification problem obtains an F1 score of 0.64. (B) Network 2 on the multi-class classification problem obtains an F1 score of 0.95. (C) Network 1 on the binary classification problem obtains an F1 score of 0.80. (D) Network 2 on the binary classification problem obtains an F1 score of 0.98. TC: transcritical. PF: pitchfork.

Gene name	Reference	Role
Ybx1	PMID: 21369783	Ybx1 is a transcription factor that plays a key role in directing the commitment and differentiation of hematopoietic stem cells
Tmsb4x	PMID: 33310787	Tmsb4x depletion hinders cell migration and differentiation and interferes with coronary vessel development
Set	PMID: 33296674	Embryonic Stem Cell Differentiation is Regulated by SET through Interactions with p53 and β-Catenin
Myc	PMID: 31951039	Myc is a family of regulator genes that code for Tfs. They are indispensable in the generation of granulocyte/monocyte progenitors (GMPs) from HPSCs
H2afy	PMID:21969577 PMID: 16428466	It is involved in chromatin remodelling and gene expression regulation. Also, modulate transcription factor binding, X-chromosome inactivation, and transcription repression.
S100a9	PMID: 28137827	Encodes a calcium-binding protein, important in immune responses, particularly in myeloid cells. S100A8/A9 induces neutrophil activation, including adhesion, CD11b upregulation, and shedding of CD62L.
Prdx5	PMID: 29497036	Prx5 transcript is highly expressed early along the transition of hematopoietic stem cells to mature neutrophils

S100a8	PMID: 28137827	Works alongside S100a9, involved in inflammatory responses in myeloid cells. S100A8/A9 induces neutrophil activation, including adhesion, CD11b upregulation, and shedding of CD62L.
C1qbp	PMID: 34860557	Associated with pre-mRNA splicing factor and can promote differentiation of immune cells via metabolic-epigenetic reprogramming
Cxcr4	PMID: 29734477 PMID: 29466759	Cxcr4 is a master regulator of neutrophil development, proliferation and retention.

Table 9-2 Supplementary Table 1. Significant genes identified during the critical period of cell decision-making (highlighted in the yellow box in Figure 3). The table includes columns for gene names, corresponding references to previous studies, and a description of their roles.

D.Appendix to chapter 6

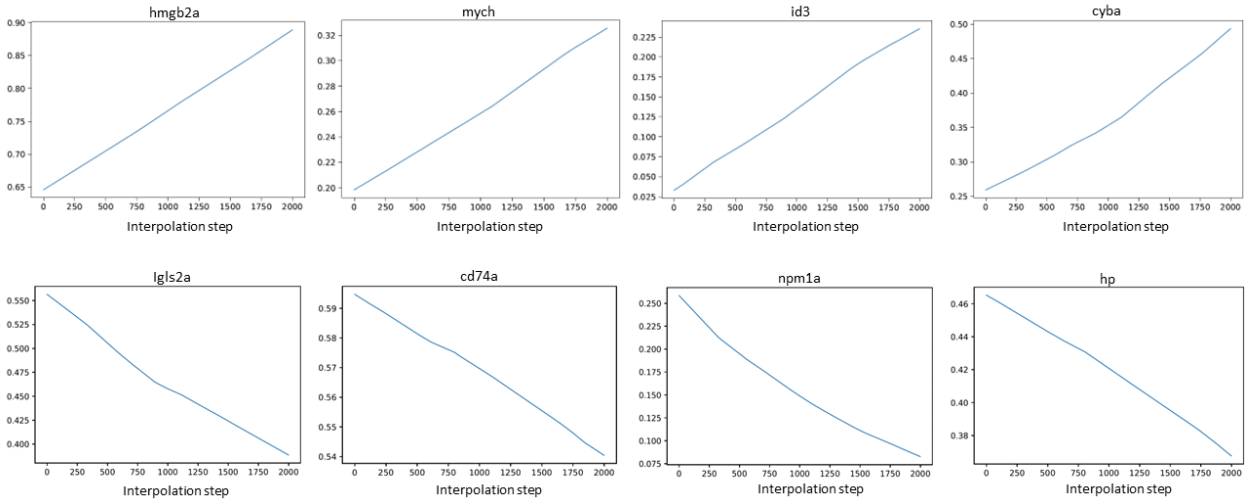


Figure 9-23 Supplementary Figure 1: Gene expression changes as a result of interpolation. The expression of selected maker genes is upregulated or downregulated as we interpolate between monocytes to HSPCs with 2000 equidistant steps.

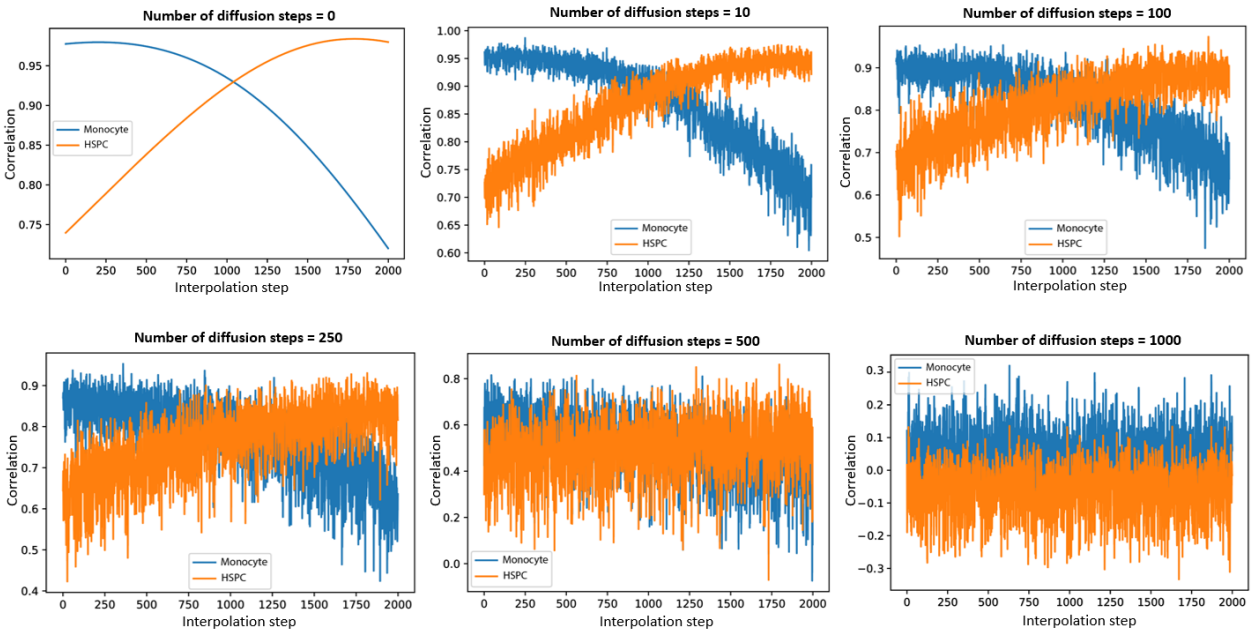


Figure 9-24 Supplementary Figure 2: The number of diffusion steps changes the level of detail in generated data Samples with different levels of granularity are generated by adjusting the number of diffusion steps before interpolation. If a large number of diffusion steps are used, more structure in the initial data will be destroyed, causing a gradual loss of information, resulting in coarser interpolations, and potentially causing abrupt changes in the correlation values between successive steps.

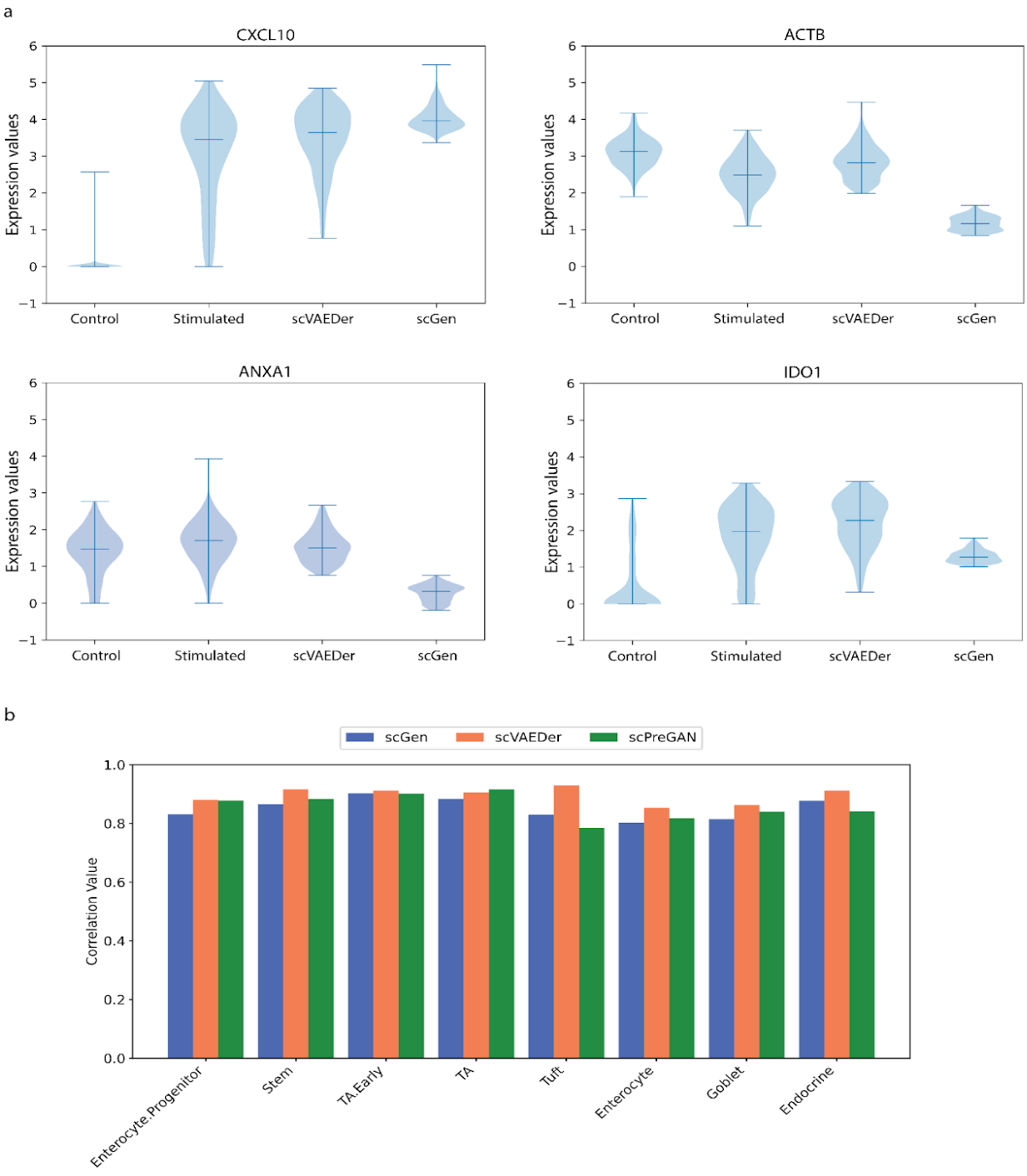


Figure 9-25 Supplementary Figure 3: scVAEDer can correctly predict the distribution of genes in the stimulated condition. a, Violin plots showing the distribution of specific key response genes across control, real stimulated condition, and prediction of stimulation using scVAEDer and scGen in Dendritic cells (DC). scVAEDer more accurately predicts the effect of the perturbation on genes. b, Comparison of the correlation

values of various cell types obtained using scGen, scPreGAN and scVAEDer for intestinal epithelial cells after Salmonella infections data.

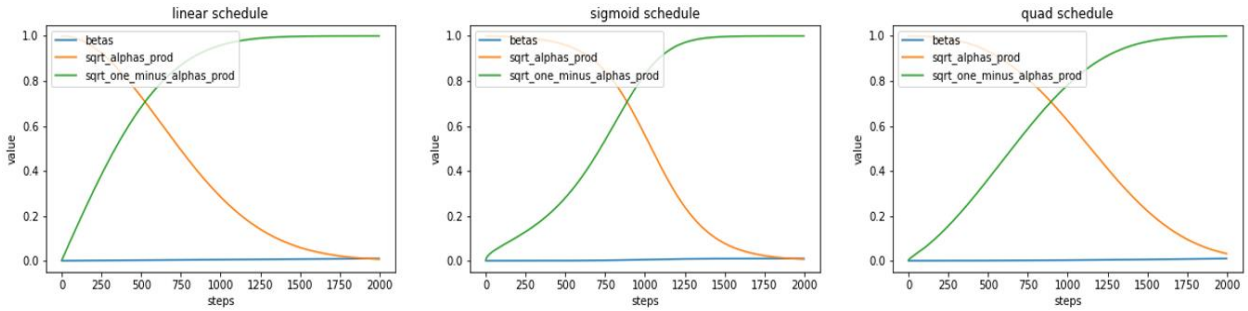


Figure 9-26 Supplementary Figure 4: DDM training and variance scheduling. A comparison between the effect of different variance scheduling methods on and values. The user can choose between one of these 3 methods prior to training the model.

Rank	Gene candidate	Gene Function	References
1	Ptma	*Ptma* is involved in regulating the exit from pluripotency and its repression resulted in the strongest deviation from pluripotency.	(282)
2	Wnt4	Loss of *Wnt4* induces the activation of stem cells and accelerates muscle regeneration.	(283)
3	Kng1	Reprogrammed iEP marker gene.	(284)
4	Igf2	*Igf2* deficiency increases HSC self-renewal and differentiation.	(285)
5	Ctgf	*Ctgf* is a reliable factor to differentiate iPSCs efficiently.	(286)
6	H2afx	H2afx is capable of controlling human stem cell self-renewal and lineage-specific differentiation.	(287)
7	FoxA1	It is known as the master regulator of the epigenome and fate.	(288)
8	Apoa1	Reprogramming initiation markers.	(284)
9	Timp3	*In vivo* analysis underpins its role as a master regulator of ectodomain shedding.	(289)
10	Mt1 and 2	Zinc maintains embryonic stem cell pluripotency	(290)

11	H19	Downregulation of *H19* improves the differentiation potential of mouse embryonic stem cells.	(291)
12	Top2a	It is uniquely expressed in HSPCs and is important in the survival of human pluripotent stem cells.	(292)
13	Igfbp	The Igfbp family is crucial for fibroblasts to progenitor cell dedifferentiation.	(293)
14	Malat1	Regulates myogenic differentiation and muscle regeneration.	(294)
15	Lgals1	Lgals1 is a transcriptional target of STAT3 and promotes mesenchymal glioblastoma.	(295)

Table 9-3 Supplementary Table 1: Top 15 genes ranked based on their velocity values, along with references that demonstrate their association with stem cell function and cellular reprogramming.

Name	**Type**	**Size_zebrafish**	**Activation_ zebrafish**	**Size_ reprogramming & IFN-β**	**Activation_ reprogramming & IFN-β**
Input	…	Number of highly variable genes	…	Number of highly variable genes	…
Layer 2	Fully connected	150	ReLU	250	ReLU and Dropout=0.15
Mean	Fully connected	30	Linear	45	Linear
Variance	Fully connected	30	Linear	45	Linear
Sampling	Fully connected	30	Linear	45	Linear
Layer 4	Fully connected	150	ReLU and Dropout=0.2	250	ReLU
Output	Fully connected	Number of highly variable genes	ReLU	Number of highly variable genes	ReLU

- Optimization: Optimizer: Adam, Learning rate: 0.001, batch_size: 400, epochs: 250
- Number of highly variable genes: 2000 for all datasets (except for zebrafish hematopoiesis, for which 1845 genes are selected)

Hyperparameter search:

Layer size: for Zebrafish [200, 150, 100], for other datasets [300, 250, 200] / Latent_layer: [15, 30, 45, 60] / Dropout: [0.15, 0.4]

Table 9-4 Supplementary Table 2: Detailed architecture of VAE part of scVAEDer.

Name	**Type**	**Size_ zebrafish**	**Activation_ zebrafish**	**Size_ reprogramming & IFN-β**	**Activation_ reprogramming & IFN-β**
Input	…	Size of latent layer	…	Size of latent layer	…
Layer1	Fully connected	512	ReLU	1200	ReLU
Layer 2	Fully connected	512	ReLU	1200	ReLU
Layer 3	Fully connected	512	ReLU	1200	ReLU
Output	Fully connected	Size of latent layer	…	Size of latent layer	…
•Optimization_Zebrafish: Optimizer: Adam, Learning rate: 0.001, batch_size: 1500, epochs: 4000					

- Optimization_Reprogramming: Optimizer: Adam, Learning rate: 0.001, batch_size: 1500, epochs: 4000
- Optimization_IFN-β: Optimizer: Adam, Learning rate: 0.001, batch_size: 10000, epochs: 4000

Hyperparameter search:

Layer size: for Zebrafish [512, 256, 128, 64], for other datasets [1200,1000,800,600,400,100] / Number of steps: [500, 750, 1000, 1250, 1500, 2000] / Noise schedule functions: [Linear, Quadratic, Sigmoid] / beta_start: [1e-5, 1e-4], beta_end: [1e-2, 5e-2]

Table 9-5 Supplementary Table 3: Detailed architecture of DDM part of scVAEDer.

E. Appendix to chapter 7

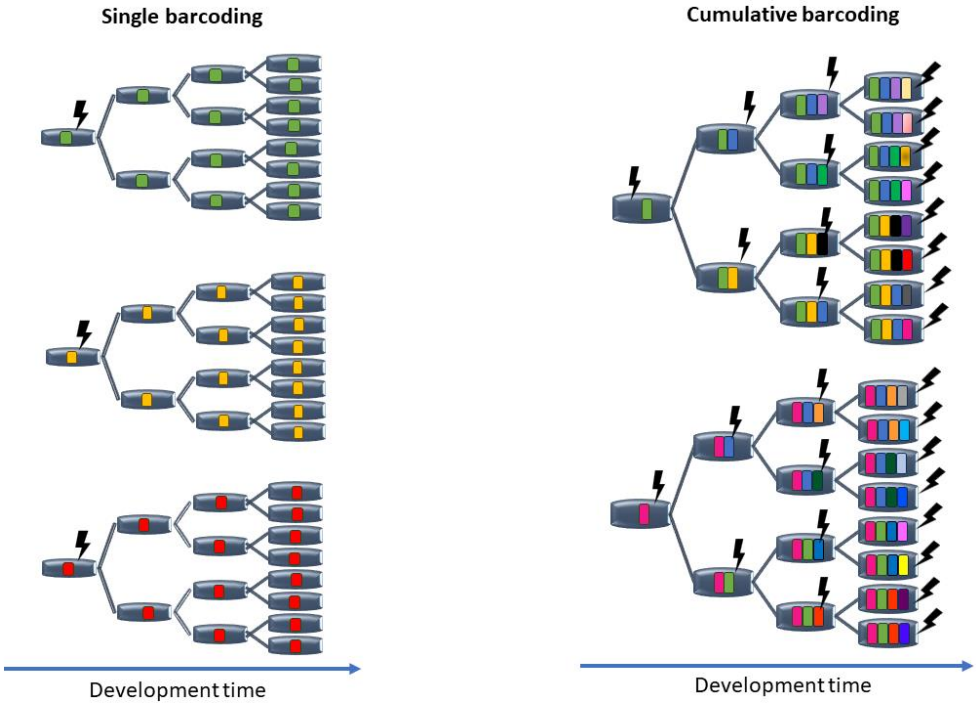


Figure 9-27 Supplementary Figure 1: Deep Lineage is compatible with both single and cumulative barcoding data. (a) Illustrates single barcoding, while (b) presents a cumulative barcoding strategy.

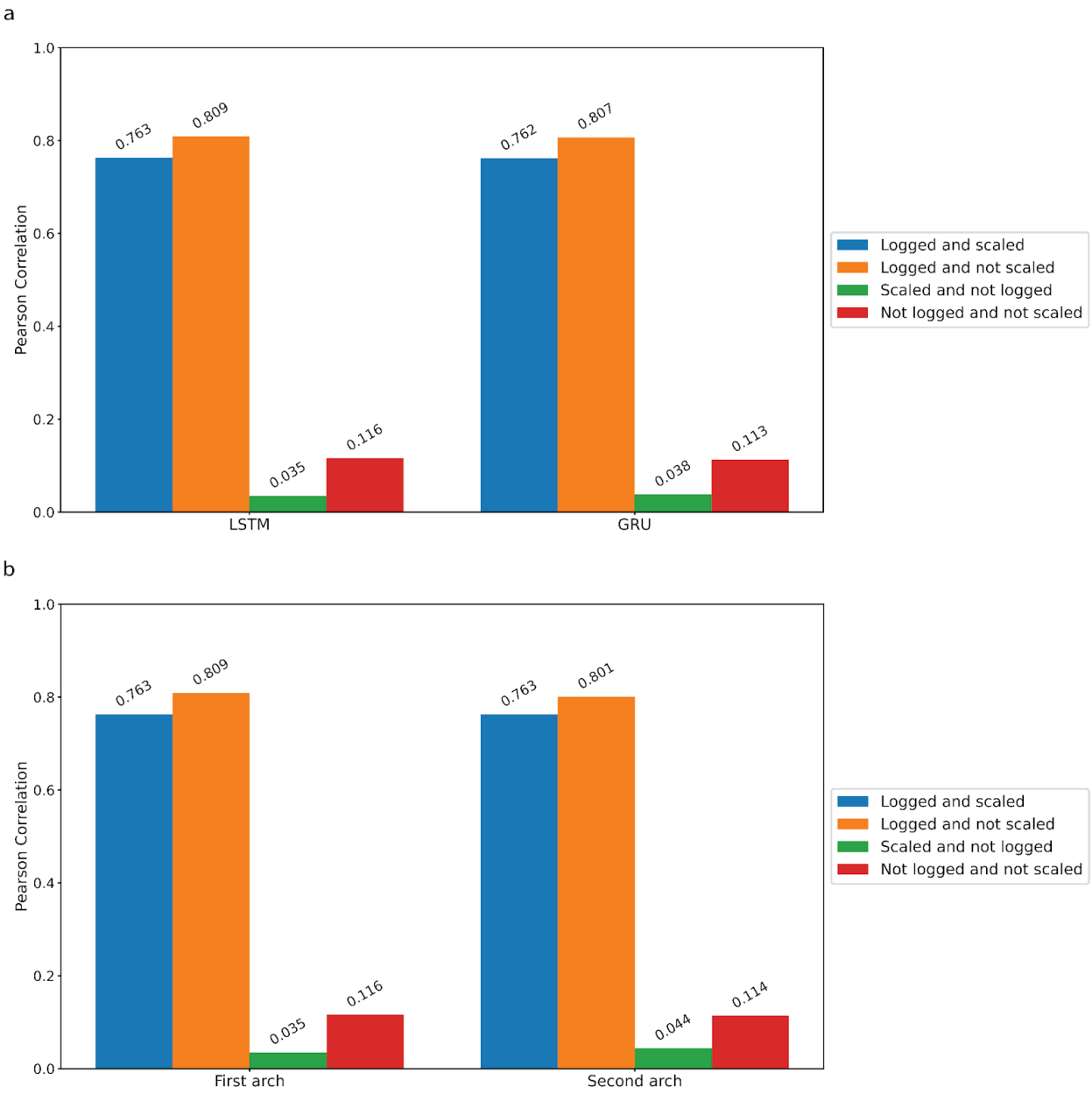


Figure 9-28 Supplementary Figure 2: The impact of different preprocessing steps and model types on the performance of Deep Lineage for hematopoiesis data. a, The effect of different preprocessing on the performance of LSTM and GRU models. b, The effect of preprocessing steps and model architecture on the LSTM performance. The first architecture includes: AE_first_hidden_layer:500, AE_second hidden_layer:100, AE_latent:75, Alpha:0.3, Num_Layers_BRNN:4, Reg_Dropout:0.3. The second architecture includes: AE_first_hidden_layer:500, AE_second_hidden_layer:200, AE_latent:50, Alpha:0.1, Num_Layers_BRNN: 2, Reg_Dropout: 0.3. As expected, using only raw counts or just scaling without log transformation results in poor performance.

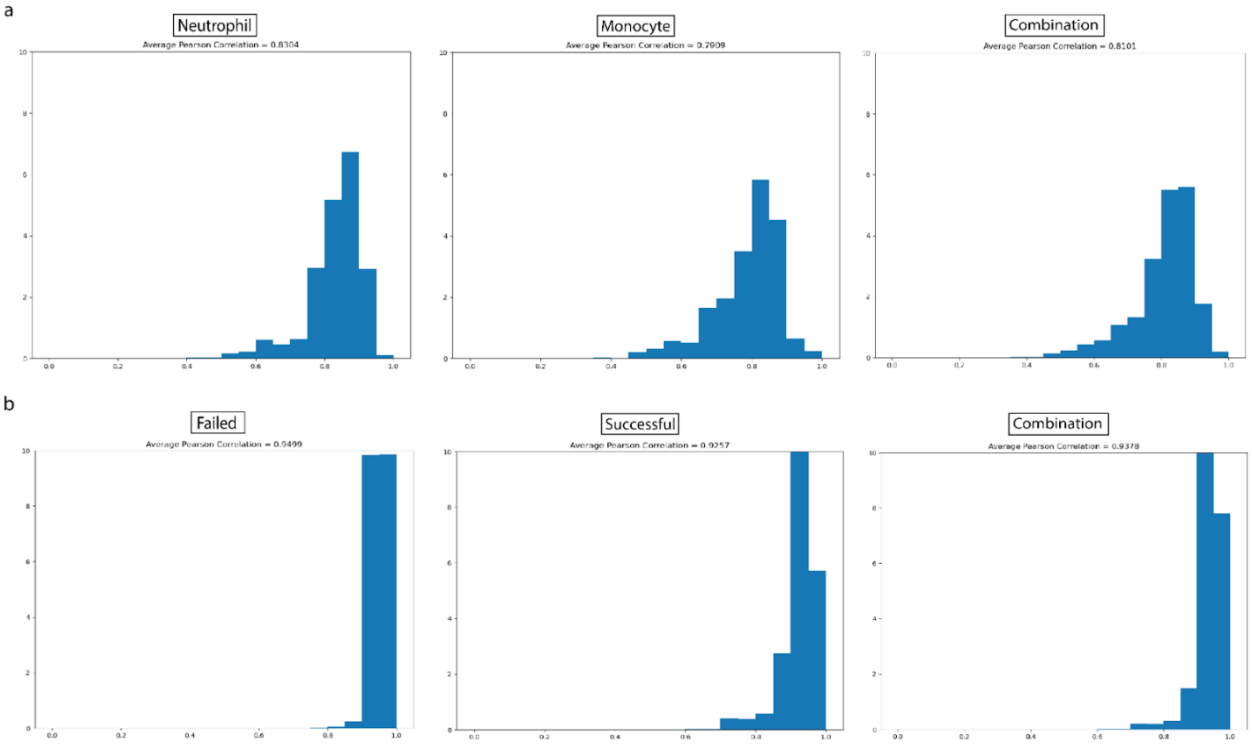


Figure 9-29 Supplementary Figure 3: Correlation values show the predictive power of Deep Lineage in estimating gene expression profiles of unseen cells. a, The distribution of correlation values for monocytes, neutrophils, and combination of cells on day 6 in hematopoiesis. b, The distribution of correlation values for failed, successfully reprogrammed cells, and the combination of cells on day 28 of the reprogramming process.

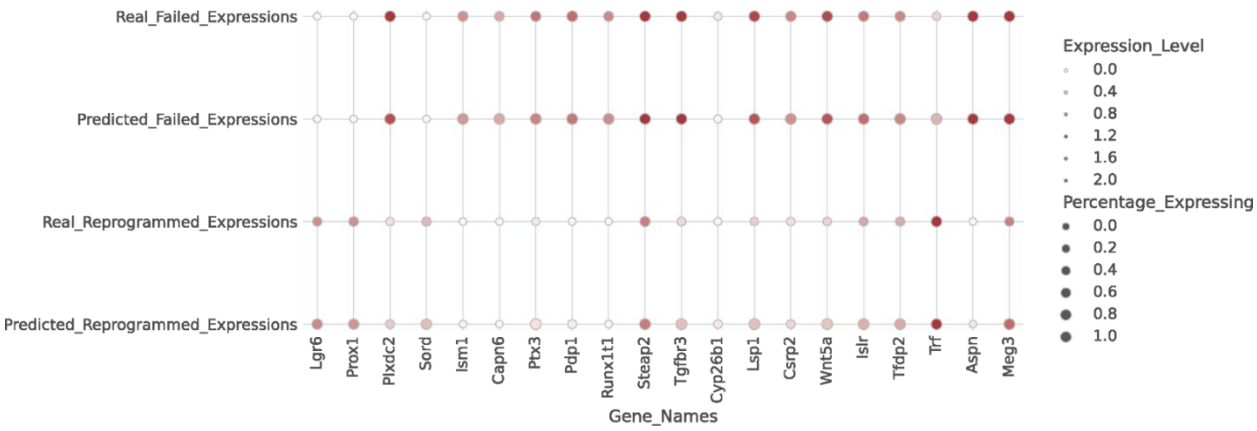


Figure 9-30 Supplementary Figure 4: Expression Prediction Accuracy in fibroblast reprogramming. The dot plot illustrates the average expression of various genes between real cells and Deep Lineage predictions, for successful and failed reprogrammed cells.

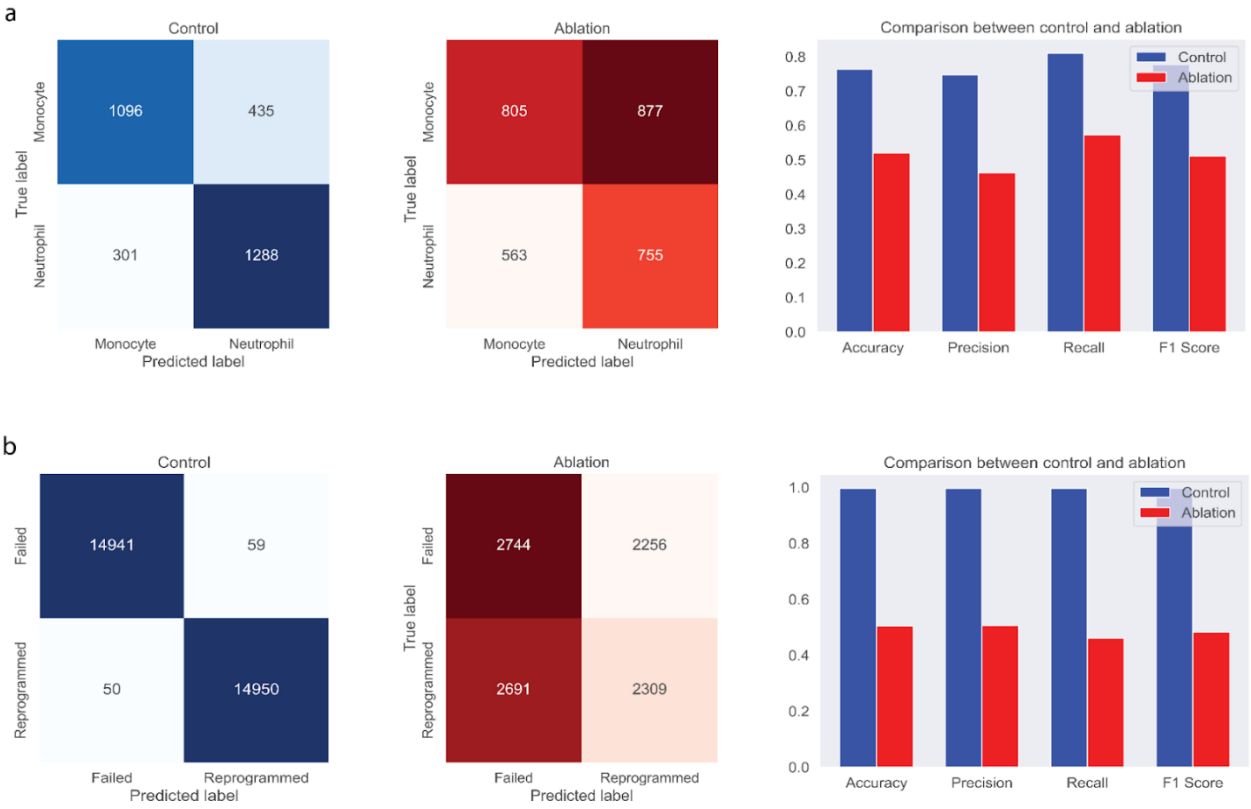


Figure 9-31 Supplementary Figure 5: Ablation studies were performed on the hematopoiesis and reprogramming data. Heatmaps display the accuracy of Deep Lineage in predicting early cell fate in control and ablation conditions for both hematopoiesis (a) and reprogramming (b). The bar plot compares these conditions using accuracy, precision, recall, and F1 score, emphasizing the impact of clonal structure ablation on model prediction.

Supplementary Tables

Gene	Roles	References
Cryab	Promotes osteogenic differentiation of human bone marrow stem cells via stabilizing β-catenin and promoting the Wnt signalling.	(296)
Gpx3	Involved in Regulating the Differentiation Fate of Cardiac Fibroblasts.	(297)
Col12a1	Mediates cellular and extracellular mechanisms in development, regeneration, and disease.	(298)
Il6st	Plays a key role in cellular reprogramming, particularly in the generation of induced pluripotent stem cells.	(299)
Gas1	Modulates critical signaling pathways (Notch, SHH, Myc) making it a key player in cellular reprogramming and differentiation.	(300,301)
Dlk1	It regulates stem cell pools and tissue differentiation by playing a pivotal role in controlling stem cell fate decisions and maintaining tissue homeostasis.	(302)
Gata2	GATA2 is crucial for maintaining (HSCs) and guiding their differentiation into various blood lineages. Mutations in GATA2 can result in cytopenias, bone marrow failure, and an increased risk of myeloid malignancies, known as GATA2 deficiency syndrome.	(303)

Prtn3	Prtn3 regulates myeloid differentiation by downregulating STAT3 expression, indicating a promising new therapeutic target for the treatment of acute myeloid leukemia.	(304)
Mmp8	MMP8 plays a crucial role in fate and function in the hematopoietic niche. It facilitates HSC mobilization through the cleavage of extracellular matrix components and cell adhesion molecules in the bone marrow niche.	(272)
Lpl	Lpl regulates hematopoietic stem progenitor cell maintenance through DHA supply.	(273)

Table 9-6 Supplementary Table 1: Key genes identified in reprogramming and hematopoiesis with their associated functions. This table presents a curated list of top genes identified by Deep Lineage as significant in both cellular reprogramming and hematopoiesis. Each row includes the gene symbol and a brief description of its known function or role, supported by relevant literature. The genes listed play diverse roles in stem cell maintenance, differentiation, signaling pathways, and tissue homeostasis.

Num_genes	Logged	Scaled	AE_dim1	AE_dim2	AE_latent	Alpha	Reg_Model_Type	Num_Layers	Reg_Dropout	Reg_P_Corr	Reg_Sp_Corr
1000	TRUE	FALSE	500	100	75	0.3	biLSTM	4	0.3	0.944548853	0.942229164
1000	TRUE	FALSE	500	100	75	0.3	LSTM	4	0.3	0.942448853	0.941129164
1000	TRUE	FALSE	500	100	75	0.3	GRU	4	0.3	0.941784166	0.940391497
1000	TRUE	FALSE	500	100	75	0.3	biLSTM	2	0.3	0.941266451	0.93999151
1000	TRUE	FALSE	500	100	75	0.3	GRU	2	0.3	0.941147014	0.939845989
1000	TRUE	FALSE	500	200	50	0.3	biLSTM	2	0.3	0.850016573	0.817333831
1000	FALSE	TRUE	500	100	75	0.3	LSTM	4	0.3	0.849379686	0.842069225
1000	FALSE	TRUE	500	100	75	0.3	GRU	4	0.3	0.849195238	0.841825634
1000	TRUE	FALSE	500	200	50	0.1	biLSTM	4	0.3	0.848818795	0.816337899
1000	TRUE	FALSE	500	200	50	0.1	GRU	2	0.3	0.847761057	0.815014231
1000	TRUE	FALSE	500	200	50	0.1	LSTM	2	0.3	0.847498949	0.814677398
1000	FALSE	TRUE	500	100	75	0.3	GRU	2	0.3	0.84685067	0.839635916
1000	FALSE	TRUE	500	100	75	0.3	LSTM	2	0.3	0.846769571	0.839413632
1000	TRUE	TRUE	500	200	50	0.1	GRU	4	0.3	0.810635694	0.80122234
1000	TRUE	TRUE	500	200	50	0.1	LSTM	4	0.3	0.810017833	0.800804507
1000	TRUE	TRUE	500	100	75	0.3	LSTM	4	0.3	0.80894793	0.809193097
1000	TRUE	TRUE	500	200	50	0.1	LSTM	2	0.3	0.808315824	0.799211364
1000	FALSE	TRUE	500	200	50	0.1	GRU	4	0.3	0.808314441	0.783773673
1000	TRUE	TRUE	500	100	75	0.3	GRU	4	0.3	0.807893351	0.808067781
1000	FALSE	TRUE	500	200	50	0.1	LSTM	4	0.3	0.807351411	0.782850232
1000	FALSE	FALSE	500	100	75	0.3	LSTM	4	0.3	0.806950121	0.78797323
1000	TRUE	TRUE	500	200	50	0.1	GRU	2	0.3	0.80680291	0.797987988
1000	TRUE	TRUE	500	100	75	0.3	GRU	2	0.3	0.806125402	0.806454036
1000	TRUE	TRUE	500	100	75	0.3	LSTM	2	0.3	0.80610008	0.806304019
1000	FALSE	TRUE	500	200	50	0.1	LSTM	2	0.3	0.805871804	0.781350875
1000	FALSE	TRUE	500	200	50	0.1	GRU	2	0.3	0.805663899	0.78077926
1000	FALSE	FALSE	500	100	75	0.3	GRU	4	0.3	0.805486588	0.786351988
1000	FALSE	FALSE	500	100	75	0.3	LSTM	2	0.3	0.803153346	0.784039623
1000	FALSE	FALSE	500	100	75	0.3	GRU	2	0.3	0.802871588	0.783584771
1000	FALSE	FALSE	500	200	50	0.1	LSTM	4	0.3	0.798550871	0.783676193
1000	FALSE	FALSE	500	200	50	0.1	GRU	4	0.3	0.798207376	0.783414733
1000	FALSE	FALSE	500	200	50	0.1	LSTM	2	0.3	0.795993826	0.781043827
1000	FALSE	FALSE	500	200	50	0.1	GRU	2	0.3	0.795735324	0.780444273

Table 9-7 Supplementary table 2: Preprocessing Steps. This table shows the different preprocessing steps and hyperparameters applied to both datasets used in the study.

Num_gene	AE_dim1	AE_dim2	AE_latent	Alpha	Reg_Mode	Num_Laye	Reg_Dropc	Reg_P_Cor	Reg_Sp_Corr
1000	500	100	75	0.3	GRU	4	0.5	0.805312	0.78400909
1000	500	100	75	0.3	biLSTM	2	0.25	0.808028	0.78826535
1000	500	100	75	0.3	biLSTM	4	0.25	0.810279	0.78992654
1000	500	100	75	0.3	GRU	2	0.5	0.804575	0.78324911
1000	500	100	75	0.3	LSTM	2	0.25	0.807077	0.78634727
1000	500	100	75	0.3	GRU	4	0.25	0.809417	0.78902639
1000	500	100	75	0.3	biLSTM	2	0.5	0.803954	0.78245209
1000	500	100	75	0.3	GRU	1	0.25	0.804275	0.78339226
1000	500	100	75	0.3	GRU	2	0.25	0.806904	0.78635154
1000	500	100	75	0.3	biLSTM	1	0.5	0.79999	0.77845038
1000	500	100	75	0.3	biLSTM	1	0.25	0.805203	0.78462392
1000	500	100	75	0.3	LSTM	4	0.5	0.805791	0.78450958
1000	500	250	75	0.3	GRU	1	0.25	0.697021	0.66057573
1000	500	250	75	0.3	GRU	4	0.5	0.699193	0.66219704
1000	500	250	75	0.3	GRU	4	0.25	0.70304	0.66712391
1000	500	250	75	0.3	LSTM	2	0.5	0.697026	0.65977952
1000	500	250	75	0.3	LSTM	4	0.5	0.699848	0.66282786
1000	500	100	75	0.3	GRU	1	0.5	0.799607	0.77773103
1000	500	250	75	0.3	LSTM	1	0.25	0.694762	0.65835216
1000	500	250	75	0.3	GRU	2	0.5	0.697251	0.66033044
1000	500	250	75	0.3	LSTM	4	0.25	0.702469	0.66635414
1000	500	250	75	0.3	LSTM	1	0.5	0.69084	0.65393898
1000	500	250	75	0.3	LSTM	2	0.25	0.698557	0.66276796
1000	500	250	75	0.3	GRU	2	0.25	0.699621	0.66348044
5000	2500	1200	400	0.3	GRU	1	0.25	0.46879	0.44694201
5000	2500	1200	400	0.3	LSTM	1	0.5	0.472332	0.45050686
5000	2500	1200	400	0.3	LSTM	4	0.5	0.479761	0.45810421
5000	2500	1200	400	0.3	GRU	2	0.25	0.471762	0.44996175
5000	2500	1200	400	0.3	LSTM	2	0.25	0.472282	0.4505922
5000	2500	1200	400	0.3	LSTM	2	0.5	0.476202	0.45444539
5000	2500	1200	400	0.3	GRU	1	0.5	0.473874	0.45197846

Table 9-8 Supplementary table 3: Regression Architecture for Hematopoiesis Data. This table presents various architectures and parameters evaluated for the regression part of the hematopoiesis data analysis.

Num_genes	Logged	Scaled	AE_dim1	AE_dim2	AE_latent	Alpha	Reg_Model_Type	Num_Layers_BRNN	Reg_Dropout	Reg_P_Corr	Day
1000	TRUE	FALSE	500	100	75	0.3	biLSTM	2	0.3	0.94030711	15
1000	TRUE	FALSE	500	100	75	0.3	biLSTM	4	0.3	0.94027223	15
1000	TRUE	FALSE	500	100	75	0.3	LSTM	2	0.3	0.94027071	15
1000	TRUE	FALSE	500	100	75	0.3	GRU	4	0.3	0.94026846	15
1000	TRUE	FALSE	500	100	75	0.3	GRU	2	0.3	0.94026673	15
1000	TRUE	FALSE	500	100	75	0.3	LSTM	4	0.3	0.9378291	28
1000	TRUE	FALSE	500	100	75	0.3	LSTM	2	0.3	0.93348754	28
1000	TRUE	FALSE	500	100	75	0.3	GRU	2	0.3	0.93085606	12
1000	TRUE	FALSE	500	100	75	0.3	LSTM	4	0.3	0.93083982	12
1000	TRUE	FALSE	500	100	75	0.3	LSTM	2	0.3	0.93083606	12
1000	TRUE	FALSE	500	100	75	0.3	GRU	4	0.3	0.9307899	12
1000	TRUE	FALSE	500	100	75	0.3	GRU	4	0.3	0.92778345	28
1000	TRUE	FALSE	500	100	75	0.3	GRU	2	0.3	0.92155994	28
1000	TRUE	FALSE	500	100	75	0.3	LSTM	2	0.3	0.90649628	21
1000	TRUE	FALSE	500	100	75	0.3	GRU	2	0.3	0.90455302	21
1000	TRUE	FALSE	500	100	75	0.3	LSTM	4	0.3	0.89629499	21
1000	TRUE	FALSE	500	100	75	0.3	GRU	4	0.3	0.89627319	21

Table 9-9 Supplementary table 4: Regression Architecture for Reprogramming Data. This table presents various architectures and parameters evaluated for the regression part of the reprogramming data analysis.

Num_Genes	Num_Features	Classification_Model_Type	Num_Layers	Classification_Dropout	Accuracy	Categorical_Crossentropy
1000	75	biLSTM	2	0.25	0.996833	0.008782554
1000	75	biLSTM	4	0.25	0.996367	0.009584519
1000	75	GRU	2	0.25	0.996767	0.009473658
1000	75	biLSTM	4	0.25	0.9964	0.010703371
1000	75	biLSTM	2	0.5	0.986933	0.045165885
1000	75	biLSTM	4	0.5	0.9802	0.06844167
1000	75	GRU	2	0.5	0.9826	0.06100791
1000	75	GRU	4	0.5	0.975467	0.081366405

Table 9-10 Supplementary table 5: Cell Fate Prediction Architecture and Parameter Search for Reprogramming Dataset. This table describes different architectures and parameters tested for the fate prediction component in the reprogramming dataset, specifically for the task where all days except day 28 of clones are given to predict day 28.

Num_Genes	Num_Features	Classification_Model_Type	Num_Layers	Classification_Dropout	Accuracy	Categorical_Crossentropy
1000	75	biLSTM	2	0.25	0.9461333	0.10827406
1000	75	biLSTM	4	0.25	0.9462333	0.10588061
1000	75	GRU	2	0.25	0.9455333	0.112969115
1000	75	GRU	4	0.25	0.9456333	0.10925144
1000	75	biLSTM	2	0.5	0.9404667	0.13359515
1000	75	biLSTM	4	0.5	0.9348333	0.14498238
1000	75	GRU	2	0.5	0.9399333	0.13637076
1000	75	GRU	4	0.5	0.9328667	0.14783286

Table 9-11 Supplementary table 6: Cell Fate Prediction Architecture and Parameter Search for Reprogramming Dataset. This table describes different architectures and parameters tested for the fate prediction component in the reprogramming dataset. The task involves using data from all days except day 21 and day 28 of clones to predict day 28.

Num_Genes	Num_Features	Classification_Model_Type	Num_Layers	Classification_Dropout	Accuracy	Categorical_Crossentropy
1000	75	biLSTM	2	0.25	0.9119	0.19947106
1000	75	biLSTM	4	0.25	0.9119	0.19950864
1000	75	GRU	2	0.25	0.9119	0.19949198
1000	75	GRU	4	0.25	0.9119	0.19950809
1000	75	biLSTM	2	0.5	0.9119	0.19966803
1000	75	biLSTM	4	0.5	0.9119	0.20008819
1000	75	GRU	2	0.5	0.9119	0.19969603
1000	75	GRU	4	0.5	0.9119	0.20006862

Table 9-12 Supplementary table 7: Comparison of Deep Learning Architectures for Cell Fate Classification in Reprogramming. This table describes different architectures and parameters tested for the fate prediction component in the reprogramming dataset. The task involves using data from all days except day 15, 21, and day 28 of clones to predict day 28.

Num_Genes	Num_Features	Classification_Model_Type	Num_Layers	Classification_Dropout	Accuracy	Categorical_Crossentropy
1000	75	biLSTM	1	0.25	0.52	1.0655743
1000	75	biLSTM	2	0.25	0.508	1.2692021
1000	75	biLSTM	4	0.25	0.517333	1.4407775
1000	75	GRU	1	0.25	0.503333	1.054049
1000	75	GRU	2	0.25	0.512667	1.4320257
1000	75	GRU	4	0.25	0.513	1.4005127
1000	75	biLSTM	1	0.5	0.507	0.9063669
1000	75	biLSTM	2	0.5	0.509	1.010492
1000	75	biLSTM	4	0.5	0.508667	1.0225995
1000	75	GRU	1	0.5	0.507	0.91736084
1000	75	GRU	2	0.5	0.522333	1.0038046

Table 9-13 Supplementary table 8: Ablation Study for Hematopoiesis Dataset. This table provides results from ablation studies conducted on the hematopoiesis dataset, showing the importance of clonal data and also how different parameters contribute to overall performance.

Num_Genes	Num_Features	Classification_Model_Type	Num_Layers	Classification_Dropout	Accuracy	Categorical_Crossentropy
1000	75	biLSTM	2	0.25	0.5012	2.5337613
1000	75	biLSTM	4	0.25	0.5053	2.6043837
1000	75	GRU	2	0.25	0.5057	2.2953575
1000	75	GRU	4	0.25	0.4951	3.0412765
1000	75	biLSTM	2	0.5	0.4916	2.2944665
1000	75	biLSTM	4	0.5	0.4955	2.772279
1000	75	GRU	2	0.5	0.4996	2.0155866
1000	75	GRU	4	0.5	0.5022	3.0623906

Table 9-14 Supplementary table 9: Ablation Study for Reprogramming Dataset. This table provides results from ablation studies conducted on the hematopoiesis dataset, showing the importance of clonal data and also how different parameters contribute to overall performance.