北京大学

本科生毕业论文

题目：**基于单细胞发育轨迹的基因表达调控网络推断**

姓　　名：	吴仁杰
学　　号：	2000012186
院　　系：	生命科学学院
专　　业：	生物信息学
导师姓名：	王劲卓助理研究员

二〇二四 年 五 月

版权声明

任何收存和保管本论文各种版本的单位和个人，未经本论文作者同意，不得将本论文转借他人，亦不得随意复制、抄录、拍照或以任何方式传播。否则，引起有碍作者著作权之问题，将可能承担法律责任。

摘要

在生物医学领域，基因表达调控网络（GRN）的研究对于理解细胞状态转变至关重要。本研究聚焦于从单细胞 RNA 测序（scRNA-seq）数据中推断 GRN。传统 GRN 模型因缺乏对细胞状态动态变化的考虑而受限。我们提出一种基于 RNA 速率向量场的 GRN 模型，使用深度学习技术如图卷积网络和注意力机制对 RNA 速率向量场进行学习，以揭示基因间的调控关系。通过对小鼠胚胎干细胞的 scRNA-seq 数据进行模型训练，我们成功构建了基因表达状态到 RNA 速率的映射，并从伪时间、高变基因、基因模块等多方面对其进行了分析。结果显示，所推断的 GRN 与已知的生物学过程高度吻合，证明了方法的有效性。本研究还讨论了编码过程中该模型训练中的必要性以及编码维度的重要性。总而言之，本研究的方法为单细胞基因组学数据提供了新的分析框架，对于进一步的生物学发现和生物医学机制研究具有重要意义。

关键词：单细胞 RNA 测序（scRNA-seq）；基因表达调控网络（GRN）；RNA 速率向量场；深度学习；发育轨迹

Inference of gene regulatory network based on single-cell developmental trajectories

Renjie Wu (Bioinformatics)

Directed by Jinzhuo Wang

ABSTRACT

In biomedical research, understanding gene regulatory networks (GRNs) is crucial for elucidating cellular state transitions. This study focuses on inferring GRNs from single-cell RNA sequencing (scRNA-seq) data. Traditional GRN models are limited due to their lack of consideration for dynamic changes in cellular states. We here propose a novel GRN model based on RNA velocity vector fields, utilizing deep learning techniques such as graph convolutional networks and attention mechanisms to learn the vector fields and reveal gene regulatory relationships. By training models on scRNA-seq data from mouse embryonic stem cells, we successfully constructed a mapping RNA velocity from gene expression state and analyzed it from various aspects including pseudo-time, highly variable genes, and gene modules. The results demonstrate that the inferred GRN aligns well with known biological processes, validating our method's effectiveness. The necessity of encoding and the significance of embedding dimension in the model training are also discussed. To sum up, this study provides a new analytical framework for single-cell genomics data, which is significant for further biological discoveries and biomedical mechanism research.

KEY WORDS: single-cell RNA sequencing (scRNA-seq); gene regulatory network (GRN); RNA velocity vector field; deep learning; developmental trajectories

目录

主要符号和缩写对照表

缩略词	英文全称	中文全称
GRN	gene regulatory network	基因调控网络
ChIP-seq	chromatin immunoprecipitation sequencing	染色质免疫共沉淀测序
dGRN	cell state-dependent GRN	细胞状态依赖的基因调控网络
scRNA-seq	single-cell RNA sequencing	单细胞 RNA 测序
mESC	mouse embryonic stem cell	小鼠胚胎干细胞
GEO	Gene Expression Omnibus	基因表达总览（数据库）
TPM	transcripts per kilobase per million mapped reads	每千碱基每百万读数的转录本计数
MGI	Mouse Genome Informatics	小鼠基因组信息（数据库）
H	hour	小时
MLP	multilayer perceptron	多层感知机
GCN	graph convolutional network	图卷积网络
PCA	principal component analysis	主成分分析
GO	gene ontology	基因本体
HVG	highly variable gene	高度可变基因
BP	biological process	生物学过程
TPR	true positive rate	真阳性率
FPR	false positive rate	假阳性率
AUROC	area under the receiver operating characteristic curve	接受者操作特征曲线下面积

第一章 引言

1.1 研究问题、意义及现状

1.1.1 基因表达调控与细胞状态转变

基因表达调控是现代分子生物学中的一个核心概念，指基因何时何地何种条件下被激活或抑制。基因表达调控在多种细胞状态转变过程中发挥着至关重要的作用[1]。

细胞状态转变是指细胞从一种状态转换到另一种状态的过程，受外界环境信号刺激和内部反馈网络共同调控，涉及各种细胞形态功能的改变，如干细胞的分化、免疫细胞的激活或者肿瘤细胞的恶化。在这些转变过程中，基因表达调控就是多种细胞内反馈网络之一，一方面感受、响应外界和内部其他的信号，另一方面将信号传播、处理，作用于自身以及其他细胞内反馈网络。基因表达调控中的特定模块被激活或抑制，导致特定基因表达模式的改变，从而引导细胞走向特定的命运[2]。

例如，在小鼠胚胎干细胞向原始内胚层细胞的分化过程中，转录因子 Gata6 发挥了核心作用[3]。FGF 等胞外信号以及这些信号通路的相互作用促进 Gata6 表达，进一步激活下游与加强细胞向内胚层分化相关的基因，有序地保障内胚层的细胞命运[4]。总之，特定的生物学过程需要启动相应的基因调控以确保细胞正确地理解了外界的信息和内在的程序，从而进入正确的轨迹与实现相应的功能。

因此，基因表达调控和细胞状态转变是两个密不可分的概念。一方面，基因表达调控是细胞状态改变的细胞分子基础与动力源；另一方面，对于细胞状态转变过程中的受激活或抑制的基因模块的确定可以帮助揭示控制该过程的关键基因或信号分子，深化我们对细胞状态转变的理解[5]。这表明基因表达调控的推断的重要生物医学意义，以及借助细胞基因表达状态改变的轨迹信息来推断基因表达调控的可能性。

1.1.2 传统的基因表达调控网络模型的定义与推断方法

尽管基因表达调控的范围很广，但是有些时候我们仅关注基因与基因之间的关系，并用基因表达调控网络（GRN）模型对其抽象。通常来说，该模型用有向有权图表示，其中节点表示基因，边的方向和权重表示基因间调控关系和强弱。[6]尽管GRN的概念在理论上为我们提供了一个强大的框架来理解基因表达的调控机制，但在实际定义上存在一定的不准确性和模糊性。

比如，传统的GRN模型常常聚焦于转录因子和靶基因之间的直接相互作用上，因此GRN的边的权重往往由转录因子对靶基因的结合强弱来定义，评价标准也以相应的如染色质免疫共沉淀测序（ChIP-seq）的实验数据或数据库为主[6]。然而，这样的定义或者评价标准忽略了基因表达调控的复杂性和动态性：首先，转录因子对基因表达的调控作用是动态的、时空特异的。对于某个培养条件下某个发育阶段的某种类型细胞的基因调控状态不可以直接应用于其他细胞上，而湿实验手段无法枚举出所有条件的组合。其次，基因调控的水平不仅限于转录水平，基因之间的调控作用也有非直接的，而这些调控层次都是ChIP-seq等方法无法捕捉的[6]。

单细胞测序技术和基因组学的出现允许在单个细胞尺度上测量和描述基因表达状态，使得更精细地、按照不同细胞状态地构建GRN成为可能[6,7]。这些细胞状态依赖的GRN（cell state-dependent GRN, dGRN）相比于对一群异质细胞群体计算平均的GRN来说更加符合实际。然而这些方法同样没有对GRN进行明确定义，定义的模糊性导致模型评估方法相对有限或主观。这些评估方法中，一类是上述的用检验转录因子调控作用的实验（如染色质免疫共沉淀测序）数据来评价模型，虽然存在明确的“真值”用于比较和评价，但可能受批次效应、数据有限影响，且在应用于下游分析时需要注意这种方法对基因调控的定义局限性。另一类方法评价模型在下游数据分析的效果，例如细胞聚类效果[8]。但是不存在用于比较的“真值”会导致无法客观评价模型。

在以上讨论中，要求GRN定义的准确性和明确性是为了服务下游分析。例如，如果下游分析希望通过GRN预测将来的基因表达状态，那么使用蛋白-DNA结合强弱定义的GRN就不够恰当。因此，为了保证对于GRN推断结果解读与应用于下游分析的合理性，GRN模型需要更准确且明确的定义以及相应的更加客观的评价方法。

1.1.3 由 RNA 速率向量场定义的基因调控网络模型

RNA速率是指RNA分子在细胞内含量随时间变化率[9]。由于单细胞RNA测序（scRNA-seq）中，内含子和外显子在序列比对过程中可以一定程度区分开来，且二者与未剪接mRNA和剪接后mRNA含量有一定对应关系，因此可以从测序数据中推断每个细胞中的这两种mRNA的比例，进而通过数学建模推断细胞当前状态下mRNA合成或降解的速率[9,10]，即RNA速率。这样的方法可以从单个时刻的单细胞基因表达数据推断该时刻的将来时刻的基因表达水平，进而推断不同细胞类型的出现先后顺序以及路径。

如果将RNA表达状态抽象成一个向量空间，那么每一个单细胞的基因表达状态$\boldsymbol{x}(t)$可以被放在这样的空间里，并且可以假设每个单细胞的RNA速率$\dot{\boldsymbol{x}}(t)$可以作为空间上的每个点的映射即$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(\boldsymbol{x}(t)\big)$，如果忽略随机因素[11]。这样的抽象和假设是生物学上合理的，因为一个细胞的RNA速率，或者说基因表达状态变化，一定程度上就是由该细胞当前的基因调控网络控制的（见1.1.1），尽管有其他因素（比如没有在模型中考虑到的其他信号分

子的调控作用）或者随机因素等也会影响RNA速率。这样的合理性表明从RNA表达数据$\{\boldsymbol{x}_i(t)\}$和RNA速率数据$\{\dot{\boldsymbol{x}}_i(t)\}$通过机器学习方法拟合向量场的合理性[11]。

上述的抽象有若干好处：首先，向量场$\boldsymbol{f}$隐式地体现了GRN，比如向量场的雅各布矩阵就体现了一个基因对另一个基因的变化速率的影响。此外，相比于传统GRN，这样的GRN是明确定义的，细胞状态依赖的，并且不需要像其他的dGRN推断方法一样划分出离散的细胞状态。其次，由于$\{\boldsymbol{x}_i(t)\}$和$\{\dot{\boldsymbol{x}}_i(t)\}$均可以测得或者推断得到，向量场拟合的好坏有明确和客观的评价方法。第三，这样定义的GRN是有下游应用场景的。建立RNA速率向量场可以方便的提供细胞状态轨迹，而不需要单独地对每一个数据集进行建模和轨迹推断。向量场还可以用于微分几何学上的分析，并将其与生物学模式关联起来[11]。

尽管当前已有一些使用RNA速率或者其他单细胞发育轨迹定义GRN的工作，但是目前这样的研究并不多，使用的机器学习模型尚为有限。因此，本研究将延续这一GRN的定义方法以体现基因调控的动态性和复杂性，探索更加丰富的前沿深度学习方法。

1.2 研究内容及方法简介

基于上述的dGRN推断的重要意义以及潜在可行的推断方法，本研究将专注于从scRNA-seq数据以及由其推断的RNA速率数据中，使用注意力机制、图卷积网络等深度学习方法对每个细胞每个基因的表达状态进行编码，以便学习RNA速率向量场。

注意力机制[12]在处理时序信息时，通过模型动态分配不同时间点的信息之间的关系权重，使其能够更加集中于重要的信息输入。Crossformer[13]是最近提出的一种对多变量时序进行预测的模型，它通过两阶段的注意力处理过程——例如在基因表达矩阵中，首先关注不同时间点的细胞的关系，随后整合基因之间的信息——以提高模型的预测精度。

图卷积网络[14]通过在图结构上应用卷积操作，能够直接捕捉节点间的复杂关系，例如基因间的调控作用。因此，在提供先验基因表达调控网络时，使用图神经网络可以有效地利用这种结构信息来增强模型对基因调控动态的理解和预测能力。

训练得到模型后，本研究除了用机器学习上的表现指标来客观评价模型外，还将从多样的下游分析任务表现上评价模型。最后，本研究将对模型中的组件以及超参数的选择进行实验。本研究强调了一种新兴的GRN定义和研究思路，说明了使用深度学习捕捉RNA速率向量场的可行性，并为构建更普遍的RNA速率向量场提供模型和训练的框架，这将为单细胞基因组学领域的众多下游分析提供宝贵的资源，为生物医学领域的研究提供新的视角和方法。

第二章 实验材料和方法

2.1 实验设备与软件

本研究使用的服务器信息如下：

表 1 服务器信息

操作系统	GNU/Linux	**硬盘容量**	18TB
内核版本	5.4.0-137-generic x86_64	**GPU 型号**	NVIDIA A100-SXM4
CPU 型号	Intel(R) Xeon(R) Platinum 8358 CPU @ 2.60GHz	**GPU 数量**	8
CPU 核数	32	**每块 GPU 显存容量**	80GB
CPU 线程数	128	**CUDA 版本**	11.8
内存容量	1TB	**NVIDIA 驱动版本**	520.61.05

本研究所用到的软件、版本号及来源如下（表 2 和表 3 分为用于 Python 的软件和用于 R 的软件）：

表 2 Python 软件信息

软件名	版本号	来源文献	来源代码
einops	0.7.0	Rogozhnikov, A., 2022	https://github.com/arogozhnikov/einops
matplotlib	3.6.3	Hunter, J. D., 2007	https://github.com/matplotlib/matplotlib
numpy	1.24.4	Harris, C. R. et al., 2020	https://github.com/numpy/numpy
pandas	1.5.3	/	https://github.com/pandas-dev/pandas
pyscenic[15]	0.12.1	Aibar, S. et al., 2017	https://github.com/aertslab/pySCENIC
pytorch	2.0.1	Ansel, J. et al., 2024	https://github.com/pytorch/pytorch
pytorch_warmup	0.1.1	/	https://github.com/Tony-Y/pytorch_warmup
python	3.8.17	/	/
rpy2	3.5.15	/	https://github.com/rpy2/rpy2
scipy	1.10.1	Virtanen, P. et al., 2020	https://github.com/scipy/scipy
scikit-learn	1.3.2	Pedregosa, F. et al., 2011	https://github.com/scikit-learn/scikit-learn
scvelo[10]	0.2.5	Bergen, V. et al., 2020	https://github.com/theislab/scvelo
seaborn	0.12.2	Waskom, M. L., 2021	https://github.com/mwaskom/seaborn
torch_geometric	2.4.0	Fey, M. et al., 2019	https://github.com/pyg-team/pytorch_geometric

表 3 R 软件信息

软件名	版本号	来源文献	来源代码
biomaRt	2.58.2	Durinck, S., et al., 2005	https://github.com/grimbough/biomaRt
gam	1.22-3	/	https://github.com/cran/gam
ggplot2	3.4.4	Wickham, H., 2016	https://github.com/tidyverse/ggplot2
monocle	2.30.1	Trapnell, C. et al., 2014	https://github.com/cole-trapnell-lab/monocle-release
PRROC	1.3.1	/	https://github.com/cran/PRROC
R	4.3.1	/	/
RColorBrewer	1.1-3	/	https://github.com/cran/RColorBrewer
slingshot[16]	2.10.0	Street, K. et al., 2018	https://github.com/kstreet13/slingshot

2.2 实验数据

本研究主要使用的数据集为小鼠胚胎干细胞（mESC）的 scRNA-seq 数据集，包含了 421 个由 mESC 诱导分化成原始内胚层过程中的细胞（加入地塞米松以诱导外源 GATA6 入核，从而诱导分化），采样于加入诱导剂后的 0h，12h，24h，48h 和 72h，详细介绍参考原始文献 [17]。该数据集在多个 GRN 推断软件（GRISLI，SINGE，SCODE）以及 GRN 推断评估方法（BEELINE）相关的文章中用于评价模型表现 [18]。原始数据在 GEO 数据库的序列号为 GSE98664，是该 421 个细胞的每千碱基每百万读数的转录本计数（TPM）矩阵。

本研究使用的小鼠先验网络来源于 2023 年 Peizhuo Wang 等人的工作 [19]，包含 18137 个基因和 5018393 条边。该先验网络由 NicheNet 和相应用到的数据库预测得到，并经过一些预处理。具体处理方法参考原文献。

本研究使用的小鼠转录因子列表由 BEELINE 工作整理产生 [18]，这些转录因子来源于包括 ESCAPE、RegNetwork、TRRUST 三个数据库。具体处理方法参考原文献。

本研究用于参考的基准值 GRN 来源于 BEELINE 工作中的 mESC 专门的 ChIP-seq 数据集得到的 GRN[18]，其中 ChIP-seq 数据是从 ChIP-Atlas、ENCODE、ESCAPE 三个数据集中与 mESC 相似或相同的细胞类型的数据收集和处理而来。具体处理方法参考原文献。

2.3 实验方法

本研究的所有代码均整理于 https://github.com/vo-olb/veloGRN。产生本研究结果的每一步方法见下。

2.3.1 mESC 数据集的预处理

该数据集的预处理方式基本与 BEELINE 中的相同[18]。通过 biomaRt 软件，转录本 TPM 矩阵按照相同的 MGI 数据库中的符号聚集求和，得到基因的 TPM 矩阵。421 个细胞中，1 个细胞的总读数偏离样本均值的 5 倍标准差以上因而被剔除。基因表达矩阵进一步被对数处理，质量控制（表达比例小于 10%的基因被剔除）。

接着，对数化的基因表达矩阵经过 PCA 降维，前三个维度用于伪时间分析（具体见下文 2.3.2）。以推算的伪时间为自变量，基因表达为因变量，用广义相加模型分别对每个基因进行方差分析，计算 F 统计量及其 P 值，并用 Bonferroni 方法来校正多重假设检验。在最终用于训练的基因表达矩阵中，仅保留了所有 P 值至少小于 0.01 的转录因子，以及 1000 个额外的 P 值至少小于 0.01 的方差最大的基因。最终的基因表达矩阵中有 420 个细胞，1620 个基因。

2.3.2 伪时间、RNA 速率的推断与可视化

mESC 数据集预处理中，我们使用 slingshot 软件[16]根据基因表达矩阵的 PCA 的前三个维度，以不同采样时间点为不同细胞群体，0h 为起始细胞群体，72h 为终点，推断每个细胞的伪时间。

由于该数据集仅提供了全体转录本表达量的数据而未区分剪接和未剪接表达量，难以进行一般意义上的 RNA 速率分析；然而，为了将细胞在状态空间中随时间的变化可视化，本研究以某一时间点的细胞之后的若干个时间点的细胞状态减去该细胞状态并加权求和来近似 RNA 速率。具体而言，对于 mESC 数据集，采用的是 2、8、32 个细胞之后的状态，权重均为 1/15：

$$\hat{\boldsymbol{v}}_t = \big(0.2(\boldsymbol{x}_{t+2} - x_t) + 0.2(\boldsymbol{x}_{t+8} - x_t) + 0.2(\boldsymbol{x}_{t+32} - x_t)\big)/3$$

对于模型预测的 RNA 速率计算，方法相同，但是将来时刻基因表达$\boldsymbol{x}_{t+2}, \boldsymbol{x}_{t+8}, \boldsymbol{x}_{t+32}$均由预测数据$\tilde{\boldsymbol{x}}_{t+2}, \tilde{\boldsymbol{x}}_{t+8}, \tilde{\boldsymbol{x}}_{t+32}$替换。我们将在附录图 A.1 中比较用这种方式得到的 RNA 速率可以推算出与 slingslot 推算的伪时间相似的伪时间，以说明这样的 RNA 速率计算的合理性。

得到用真实数据和预测数据计算的 RNA 速率后，将二者投影到同一 PCA 空间中。具体来说，先计算下一时刻的基因表达$\tilde{\boldsymbol{x}}_{t+1} = \boldsymbol{x}_t + \delta \cdot \hat{\boldsymbol{v}}_t$，然后将$\boldsymbol{x}_t$和$\tilde{\boldsymbol{x}}_{t+1}$进行同之前一样的 PCA 变化，得到起点和终点在二维空间上的位置，进而绘制矢量。

2.3.3 深度学习模型结构与训练方法

模型的输入数据为形状为(c, g)的基因表达矩阵，其中c为细胞数，g为基因数，表达矩阵按 2.3.2 中计算的伪时间从小大大排序。每个用于训练的样本点由前T个伪时间点和后τ个伪时间点的基因表达组成，其中T和τ由 in_len 和 out_len 参数指定：

$$\boldsymbol{X}_i = (\boldsymbol{x}_{i-T+1}, \dots, \boldsymbol{x}_i)^T, \boldsymbol{Y}_i = (\boldsymbol{x}_{i+1}, \dots, \boldsymbol{x}_{i+\tau})^T$$

模型由编码器、预测器和解码器组成。模型首先将样本矩阵$\boldsymbol{X}_i$按细胞分成不同的节段（节段长度sl由 seg_len 参数指定），然后通过线性层将sl个维度变换到d个维度（d由 d_model 参数指定），得到形状为$(g, s \coloneqq T/sl, d)$的第i个样本点的编码$\boldsymbol{E}_i^{(1)}$：

$$\boldsymbol{X}_{i,j} = \left(\boldsymbol{x}_{i-T+1+j*sl}, \dots, \boldsymbol{x}_{i-T+(j+1)*sl}\right)$$

$$\boldsymbol{E}_i^{(1)} = rearrange\left(f_{lin(sl,d)}\left(\boldsymbol{X}_{i,1}, \dots, \boldsymbol{X}_{i,s}\right), "s\ g\ d\ \rightarrow\ g\ s\ d"\right)$$

$$f_{lin(sl,d)}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{W} + \boldsymbol{B}, \qquad \boldsymbol{W}.shape = (sl, d)$$

然后，为了记录$\boldsymbol{E}_i^{(1)}$中每个节段的顺序，模型对其位置进行编码（其中$\boldsymbol{E}_{Position}$是形状$(s, d)$的对于每一个基因相同的可学习参数），然后对特征标准化：

$$\boldsymbol{E}_i^{(2)} = f_{LayerNorm}\left(\boldsymbol{E}_i^{(1)} + \boldsymbol{E}_{Position}\right)$$

接着，我们模仿 Crossformer 模型中的两步注意力机制 [13]，实现能体现细胞间以及基因间关系的编码。模型第一步计算细胞之间的注意力，也即，将d视作语言模型中单词的编码维度，将s视作语句中的单词数：

$$\boldsymbol{E}_i^{(3)} = f_{Add\&Norm}\left(f_{Add\&Norm}\left(\boldsymbol{E}_i^{(2)}, f_{selfAttention}\right), f_{MLP(d,d'',d)}\right)$$

$$f_{Add\&Norm}(\boldsymbol{X}, f) = f_{LayerNorm}\left(\boldsymbol{X} + f_{Dropout}\left(f(\boldsymbol{X})\right)\right)$$

$$f_{MLP(d,d'',d)}(\boldsymbol{X}) = f_{lin(d'',d)}\left(f_{Dropout}\left(f_{GELU}\left(f_{lin(d,d'')}(\boldsymbol{X})\right)\right)\right)$$

$$f_{selfAttention}(\boldsymbol{X}) = f_{Attention}(\boldsymbol{X}, \boldsymbol{X}, \boldsymbol{X})$$

$$f_{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = f_{reproj(d,d',h)}\left(f_{Dropout}\left(f_{softmax}\left(f_{proj(d,d',h)}(\boldsymbol{Q}) @ rearrange\left(f_{proj(d,d',h)}(\boldsymbol{K}), "g\ h\ s\ d' \rightarrow\ g\ h\ d'\ s"\right)\right)\right) @ f_{proj(d,d',h)}(\boldsymbol{V})\right)$$

$$f_{proj(d,d',h)}(\boldsymbol{X}) = rearrange\left(f_{lin(d,d'\cdot h)}(\boldsymbol{X}), "g\ s\ (d'\ h)\ \rightarrow\ g\ h\ s\ d'"\right)$$

$$f_{reproj(d,d',h)}(\boldsymbol{X}) = f_{lin(d'\cdot h,d)}\left(rearrange(\boldsymbol{X}, "g\ h\ s\ d'\ \rightarrow\ g\ s\ (d'\ h)")\right)$$

其中h, d', d''分别由 n_heads，d_keys 和 d_ff 参数指定，@表示矩阵乘法，$f_{LayerNorm}$，$f_{Dropout}$和$f_{softmax}$参考 pytorch 框架下的一般算法，在此不赘述，$f_{Dropout}$的丢失率由 dropout 参数指定。

第二步，模型计算基因之间的注意力，也即，仍将d视作单词的编码维度，但将g视作语句中的单词数：

$$\boldsymbol{E}_i^{(4)} = rearrange\left(\boldsymbol{E}_i^{(3)}, "g\ s\ d\ \rightarrow\ s\ g\ d"\right)$$

$$\boldsymbol{E}_i^{(5)} = f_{Add\&Norm}\left(f_{Add\&Norm}\left(\boldsymbol{E}_i^{(4)}, f_{routerAttention}\right), f_{MLP}\right)$$

$$f_{routerAttention}(\boldsymbol{X}) = f_{Attention}(\boldsymbol{X}, \boldsymbol{Y} \coloneqq f_{Attention}(\boldsymbol{W}_{router}, \boldsymbol{X}, \boldsymbol{X}), \boldsymbol{Y})$$

我们没有直接计算$f_{selfAttention}$，而是引入了一个形状为(s, r, d)的可学习参数$\boldsymbol{W}_{router}$作为中转器来计算$f_{routerAttention}$，从而减少参数量和训练时间，其中r由 router 参数指定。

Crossformer 编码之后，我们希望将先验网络融入基因表达状态的编码，因此我们采用了两层 GCN[14]。从此之后我们不再区分不同的节段，因此首先将$\boldsymbol{E}_i^{(4)}$转换为形状为$(g, f \coloneqq s \cdot d)$的矩阵：

$$\boldsymbol{E}_i^{(6)} = rearrange\left(\boldsymbol{E}_i^{(5)}, "s\ g\ d\ \rightarrow\ g\ (s\ d)"\right)$$
$$\boldsymbol{E}_i^{(7)} = f_{MLP(2\cdot f,f,f)}\left(f_{GCN}\left(\boldsymbol{E}_i^{(6)}, \boldsymbol{G}\right), f_{GCN}\left(\boldsymbol{E}_i^{(6)}, \boldsymbol{G}^T\right)\right)$$
$$\boldsymbol{E}_{i_1}^{(8)} = f_{MLP(2\cdot f,f,f)}\left(f_{GCN}\left(\boldsymbol{E}_i^{(7)}, \boldsymbol{G}\right), f_{GCN}\left(\boldsymbol{E}_i^{(7)}, \boldsymbol{G}^T\right)\right)$$
$$f_{GCN}(\boldsymbol{X}, \boldsymbol{G}) = \widehat{\boldsymbol{D}}_{in}^{-\frac{1}{2}}\widehat{\boldsymbol{G}}\widehat{\boldsymbol{D}}_{out}^{-\frac{1}{2}} f_{lin(f,f)}(\boldsymbol{X}), \qquad \widehat{\boldsymbol{G}} = \boldsymbol{G} + \boldsymbol{I}, \widehat{\boldsymbol{D}}_{in}, \widehat{\boldsymbol{D}}_{out}\text{为}\widehat{\boldsymbol{G}}\text{的入度和出度}$$

至此，我们完成了模型的编码：$\boldsymbol{Z}_t = \boldsymbol{E}_t^{(8)} = f_{Encoder}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \dots, \boldsymbol{x}_{t-T+1})$，其中$\boldsymbol{Z}_t$的形状为$(g, f)$。模型的预测器为对基因维度的线性变换：

$$\widetilde{\boldsymbol{Z}_{t+1}} = \boldsymbol{W}_{predictor}\boldsymbol{Z}_t + \boldsymbol{B}_{predictor}$$

模型最后的解码器类似 MLP，但是同时对基因维度和特征维度进行变换：

$$\boldsymbol{E}_i^{(9)} = f_{Dropout}\left(f_{GELU}\left(\boldsymbol{W}_{gene}^{(1)}\left(\widetilde{\boldsymbol{Z}_{i+1}}\boldsymbol{W}_{feature}^{(1)} + \boldsymbol{B}_{feature}^{(1)}\right) + \boldsymbol{B}_{gene}^{(1)}\right)\right)$$
$$\boldsymbol{E}_i^{(10)} = f_{Dropout}\left(f_{GELU}\left(\boldsymbol{W}_{gene}^{(2)}\left(\boldsymbol{E}_i^{(9)}\boldsymbol{W}_{feature}^{(2)} + \boldsymbol{B}_{feature}^{(2)}\right) + \boldsymbol{B}_{gene}^{(2)}\right)\right)$$
$$\widetilde{\boldsymbol{Y}_i}^T = (\widetilde{\boldsymbol{x}}_{i+1}, \dots, \widetilde{\boldsymbol{x}}_{i+\tau}) = f_{lin(f,\tau)}\left(\boldsymbol{E}_i^{(10)}\right)$$

模型的最终输出为形状为(g, τ)的$\widetilde{\boldsymbol{Y}_i}, i = T-1, \dots, c-\tau-1$。我们将$\widetilde{\boldsymbol{Y}_i}$和$\boldsymbol{Y}_i$比较，同时以计算$\boldsymbol{W}_{predictor}$的稀疏度作为正则项，计算损失函数：

$$loss = (1-\gamma) f_{RMSE}\left(\boldsymbol{Y}, \widetilde{\boldsymbol{Y}}\right) + 100\gamma\left\|\boldsymbol{W}_{predictor}\right\|_1$$

其中γ用于控制两项的权重，由 loss_gamma 参数指定，f_{RMSE}即根均方误差。我们使用 Adam 优化器以学习模型参数，学习率由 learning_rate 参数指定。我们同时使用线性预热器和余弦退火规划器以控制优化器的学习率在每一个批次训练中先线性增加再余弦下降，预热步数（批次数）和总迭代数分别由 warm_up_steps 和 train_epochs 指定。我们还使用提前终止器（由 patience 参数控制）提前终止训练，以节省时间和避免过拟合。

结果图 2 中的模型训练参数均使用默认参数（见 github 代码）。结果图 3、4、5 所依赖的模型与结果图 2 属同一模型。

2.3.4 高变基因的鉴别与分析

我们使用 scvelo 软件[10]从 RNA 速率推算伪时间。2.3.1 中已述 mESC 数据集的表达矩阵（归一化过的）如何依次对数化、PCA 降维、选取 1620 个基因。PCA 的前 30 个维度用于计算细胞之间的近邻图（`scvelo.pp.neighbors`），近邻数为 30。剪接 RNA 矩阵（`adata.layers['spliced']`）使用未对数化的总表达矩阵（因为原始数据中只包含总矩阵，且一般来说剪接分子数占总分子数的 75%到 90%[10]，因此用总矩阵近似剪接矩阵），用于计算一阶矩（`scvelo.pp.moments`）。我们将自行计算的 RNA 速率放置在单细胞数据对象相应的位置（`adata.layers['velocity']`）中，与一阶矩和近邻图一同用于构建速率依赖的近邻图（`scvelo.tl.velocity_graph`），后者进一步用于计算速率依赖的伪时间（`scvelo.tl.velocity_pseudotime`）。以上函数中未提及的参数均使用默认值。为了更好地可视化，我们仅保留该伪时间的顺序，依此生成均匀分布作为伪时间。由实际和预测 RNA 速率计算的“均匀伪时间”被用于线性拟合以及后续分析。

对于高变基因的鉴别，类似 2.3.1 地，用广义相加模型计算每个基因的 F 统计量的 P 值。另外，以在不同细胞中的表达量的标准化值（减去该基因的均值，除去该基因的标准差）为特征对基因做 kmeans 聚类（k 取 5），得到每个基因的类别以及每个基因到所属类别的中心的 L2 距离（“中心距离”）。我们同时用中心距离和 P 值来筛选基因。具体地，我们保留了 P 值小于 1e-20 且中心距离在所属类别前 30 的共 150 个基因，用于基因富集分析。热图展示选取了这 150 个基因中每类的 P 值前三小的基因。

对于 GO 富集分析中，我们使用 ToppGene Suite[20] 中的 ToppFun 功能，使用该 150 个基因/第 1、2 类的 60 个基因/第 4、5 类的 60 个基因作为基因集，使用默认背景基因和参数查询。展示结果选取类别为“GO: Biological Process”且 Bonferroni 校正 P 值小于 0.05 中 P 值在前 10 的术语。

2.3.5 形式 GRN 的提取与评估

提取的 GRN 的每一项为对应行列的基因被模型编码的向量的内积。对于整个数据集的 GRN，计算每个细胞 GRN 的平均值，然后保留转录因子所在列的项，去除自回边。与参考值 GRN 比较时，将参考值 GRN 的源节点和总结点分别作为背景 GRN 的所有可能源节点和目标节点，然后将预测值 GRN 限制在背景 GRN 内。我们使用 PRROC 软件计算不同阈值下的预测 GRN 的 TPR 和 FPR，绘制 TPR-FPR 曲线并计算 AUROC。其他软件的 AUROC 数据来源于 GROD 工作 [21]，整理于附录 4。

对于基因模块推断，我们先对 GRN 标准化以避免大多数转录因子最强的靶基因大量重叠。GRN 矩阵的每一项除去了所在行的均值和所在列的均值。每个转录因子的作用最强的前 50%的靶基因以及每个靶基因的作用最强的前 20 个转录因子被保留。最后，靶基因不足 20 个的转录因子被去除。我们使用 pyscenic[15] 中的 aucell 函数计算这些基因模块在每个细胞中的活性并绘制热图。

2.3.6 模型结构及超参数的选择与结果比较

模型不同的超参数设置可以通过 2.3.3 中所述参数指定，具体可通过`model/main.py -h`查看。不同模型结构可以通过 test 参数设置为 normal / no_encoder / no_Attention / no_GCN 之一，分别对应原始模型、去除编码器、去除编码器中的两步注意力、去除编码器中的图卷积网络。对于去除编码器的模型，用于计算下一时刻的基因特征直接取过去和当前时刻的表达向量，即$\boldsymbol{Z}_i = \boldsymbol{X}_i^T$。对于去除注意力编码的模型，直接将$\boldsymbol{E}_i^{(2)}$的后两个维度合并作为特征，传入 GCN，即$\boldsymbol{E}_i^{(6)} = rearrange\left(\boldsymbol{E}_i^{(2)}, "g\ s\ d \rightarrow\ g\ (s\ d)"\right)$。对于去除图卷积网络的模型，注意力编码直接作为基因特征计算下一时刻，即$\boldsymbol{Z}_i = \boldsymbol{E}_i^{(6)}$。

我们测试不同模型在相同的默认参数下，或者原始模型在不同参数组合（其他参数为默认值）下在验证集上的损失函数和耗时。对于不同模型的测试，我们重复五次实验并计算均值和标准误差；对于超参数测试，考虑到组合数量及耗时，我们仅进行一次实验。

第三章 实验结果

3.1 模型假设与结构

本研究中，不受 RNA 水平直接或间接调控的且可以影响 RNA 水平的胞内外信号和随机因素将被统称为“随机因素”；受 RNA 水平直接或间接调控的且可以影响 RNA 水平的非 RNA 信号如蛋白质水平被统称为隐变量。（对于部分受 RNA 水平调控的信号，假设可以拆分成随机因素分量和隐变量分量。）RNA 水平之间通过隐变量相互调控形成的反馈网络即本研究讨论的 GRN。

我们假设，随机因素在一定程度上可以忽略，从而基因表达状态的变化$\dot{\boldsymbol{x}}(t)$对于一个给定的$\boldsymbol{x}(t)$是确定的、由当下的 GRN 所决定的。因此存在一个向量场$\boldsymbol{f}$使得$\dot{\boldsymbol{x}}(t)=\boldsymbol{f}(\boldsymbol{x}(t))$，并且向量场和 GRN 可以看作等价的两种表示。我们以简单的二元调控模型（这样的模型广泛存在于细胞分化过程，例如造血作用中的 PU.1 和 GATA1，因此用来举例是合理的）来说明向量场和 GRN 之间的关系（图 1A）。基因之间的激活或抑制作用可以被概括成常微分方程组，后者定义了任意的基因表达状态的速率，也就是向量场。

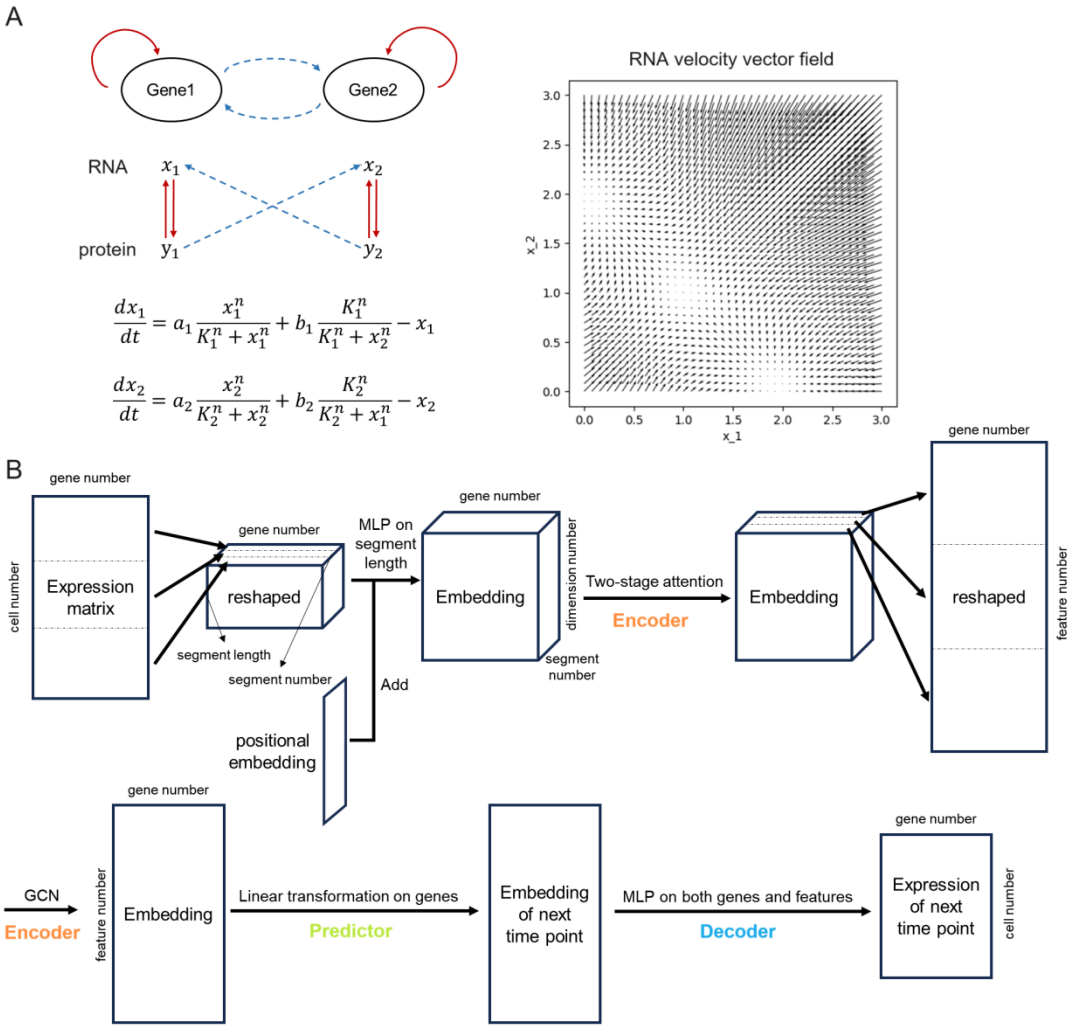

图 1 模型假设与结构。 A. 二元调控模型及对应的 RNA 速率向量场。左上，基因间调控关系，红实线表示激活，蓝虚线表示抑制。左下，对应的微分方程。右，方程对应的向量场，方程中的$a_1, a_2, b_1, b_2, K_1, K_2$均取 1，$n$取 4。改图仿自 2022 年 Xiaojie Qiu 等人的工作 [11]。B. 深度学习模型大致结构。MLP，多层感知机。GCN，图卷积网络。最开始的表达矩阵对应$(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \dots, \boldsymbol{x}_{t-T+1})^T$，GCN 之后的状态依次对应$\boldsymbol{Z}_t$，$\widetilde{\boldsymbol{Z}_{t+1}}$和$\tilde{\boldsymbol{x}}_{t+1}$。

基于上述假设，我们将建立一个深度学习模型以捕捉向量场，体现基因调控规律。模型从基因表达数据$\boldsymbol{x}(t)$预测将来时刻基因表达数据$\boldsymbol{x}_{t+1}(t) = \boldsymbol{x}(t) + \delta \cdot \dot{\boldsymbol{x}}(t)$，其中$\delta$是一个微小值。这与预测$\dot{\boldsymbol{x}}(t)$本质上是相同的，但是定义域和值域相同可以为模型训练带来一些便利。

具体而言，模型先对基因表达状态进行编码：

$$\boldsymbol{Z}_t = f_{Encoder}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \dots, \boldsymbol{x}_{t-T+1})$$

我们希望通过训练编码器的参数来捕捉不同时间点的细胞之间以及不同基因之间的关联，并认为具有关联的细胞或者基因之间的信息传递可以使得编码后的状态更能体现调控网络关系。因此，我们首先采用了两步注意力机制：先对不同时间点的细胞之间计算自注意力查询结果，然后对不同的基因之间计算自注意力查询结果。我们接着以一先验网络（见实验方法）作为引导，使用两层图卷积网络进一步将基因之间的关系整合进编码状态中。我们猜测，这样的整合了网络关系的基因表达状态可以更容易地用于将来状态的预测。

然后，模型预测下一时刻的编码后的基因表达状态。由于基因之间的调控作用，我们假设一个基因的某一个特征在下一个时刻的值依赖于自己和其他基因的这一特征在上一个时刻的值：

$$\widetilde{\boldsymbol{Z}_{t+1}} = \boldsymbol{W}\boldsymbol{Z}_t + \boldsymbol{B}$$

其中$\boldsymbol{Z}_t$的维度为基因数×特征数，$\boldsymbol{W}$和$\boldsymbol{B}$为线性预测的参数，维度分别为基因数×基因数，基因数×特征数（见图 1A）。

最后，模型将预测的基因表达状态进行解码：

$$\tilde{\boldsymbol{x}}_{t+1} = f_{Decoder}(\widetilde{\boldsymbol{Z}_t})$$

我们使用类似多层感知机的结构对其进行解码，但是区别在于，线性变换层既会对基因维度也会对特征维度做线性变换。这是因为编码器阶段有整合基因间的数据，所以在解码阶段同样需要对基因维度进行变换。如果在解码阶段没有同时对两个维度做变换，模型预测性能会明显受影响（结果未展示）。解码后的状态$\tilde{\boldsymbol{x}}_{t+1}$与$\boldsymbol{x}_{t+1}$进行比较，可以优化编码器对细胞间和基因间关系的学习、预测将来时刻的线性层参数学习以及解码器参数的学习（模型的具体结构、原理以及训练方法见实验方法部分）。

本研究中，我们对一个 mESC 的于不同时间点采样的 scRNA-seq 数据（见实验数据和实验方法部分）进行分析，推断 RNA 速率和伪时间，并用该结果来训练上述模型，并记录损失函数曲线（图 2A）。训练集与验证集的损失函数显著且同步下降。

为了进一步可视化训练效果，我们将基因表达状态空间进行 PCA 变换并选取最重要的两个维度，从而投影到二维平面。我们将用于训练的 RNA 速率以及训练得到的向量场计算的 RNA 速率进行相同的变换，投影到该二维平面上进行比较（图 2B）。可以看出实际的和预测的 RNA 速率非常接近，大体方向均从红色的细胞群体经过蓝色、绿色的到达紫色和橙色的细胞群体。这和前述的数值上的优秀预测表现共同说明了模型对 RNA 速率向量场预测的有效性。

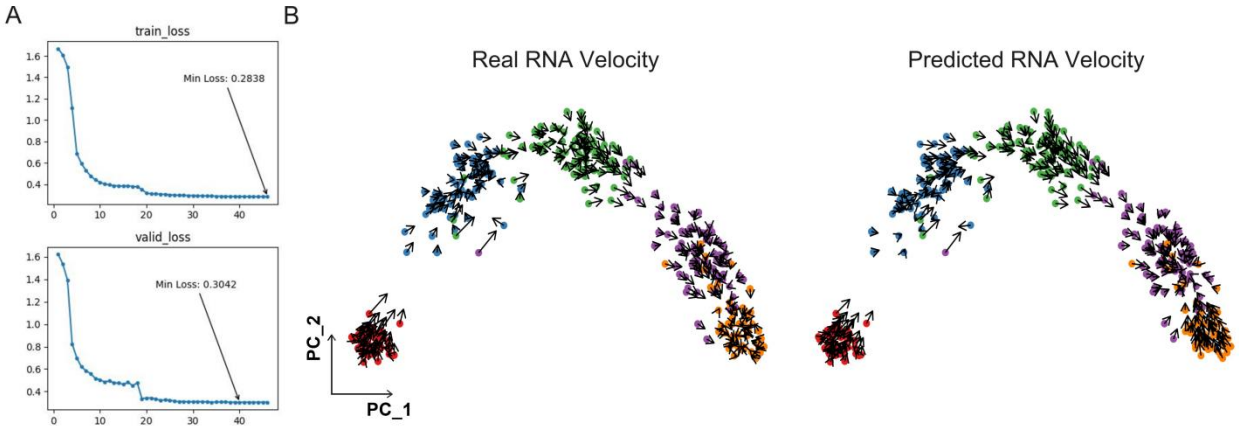

图 2 mESC 数据集的 RNA 速率训练结果。A. 训练集与验证集损失函数曲线图，横轴为训练轮数。B. 真实 RNA 速率与预测 RNA 速率于同一 PCA 变换空间的投影。不同颜色代表不同采样时间的细胞。红，蓝，绿，紫，橙五种颜色分别对应采样于 0h，12h，24h，48h，72h。

3.2 模型评估

我们对模型学习的向量场进行下游分析，以体现其在生物学上的意义，同时从更丰富的角度评估模型。首先，我们从学习的单细胞发育轨迹中鉴别高变基因，并检查这些基因的功能是否与已知的生物学过程相关。具体来讲，我们使用 mESC 数据集以及在 3.1 中计算的实际的与预测的 RNA 速率，推断每个细胞的伪时间，可以直接看出两种速率推断的伪时间的相似性（图 3A），同时线性拟合的决定系数（0.9528）也说明二者的强线性相关性（图 3B）。

根据伪时间，我们进而鉴别表达量随时间显著变化的基因（“高变基因”，根据广义相加模型拟合的 P 值大小决定是否显著变化）。我们将基因无监督聚类，可以分辨出明显的五类基因：第 1 类主要在 0h 细胞中高表达，到 24h 细胞几乎停止表达；第 2 类同样在 0h 细胞中高表达但不如第 1 类基因，并且在 24h 细胞中仍有低表达；第 3 类从 0h 到 72h 细胞都有一定程度表达，表达峰值主要在 12h 和 24h 细胞；第 4、5 类均从 12h 开始有低表达，表达量持续升高，在 48h 和 72h 细胞中高表达，但是第 5 类起始表达稍晚些（如果展示更多的基因，4 和 5 类的差别会更明显。见附录图 A.2）。对于两种方法得到的伪时间，我们均鉴别出来了这 5 类高变基因，并且每类中 P 值最小的前 3 个基因有很大重叠（图 3C）。

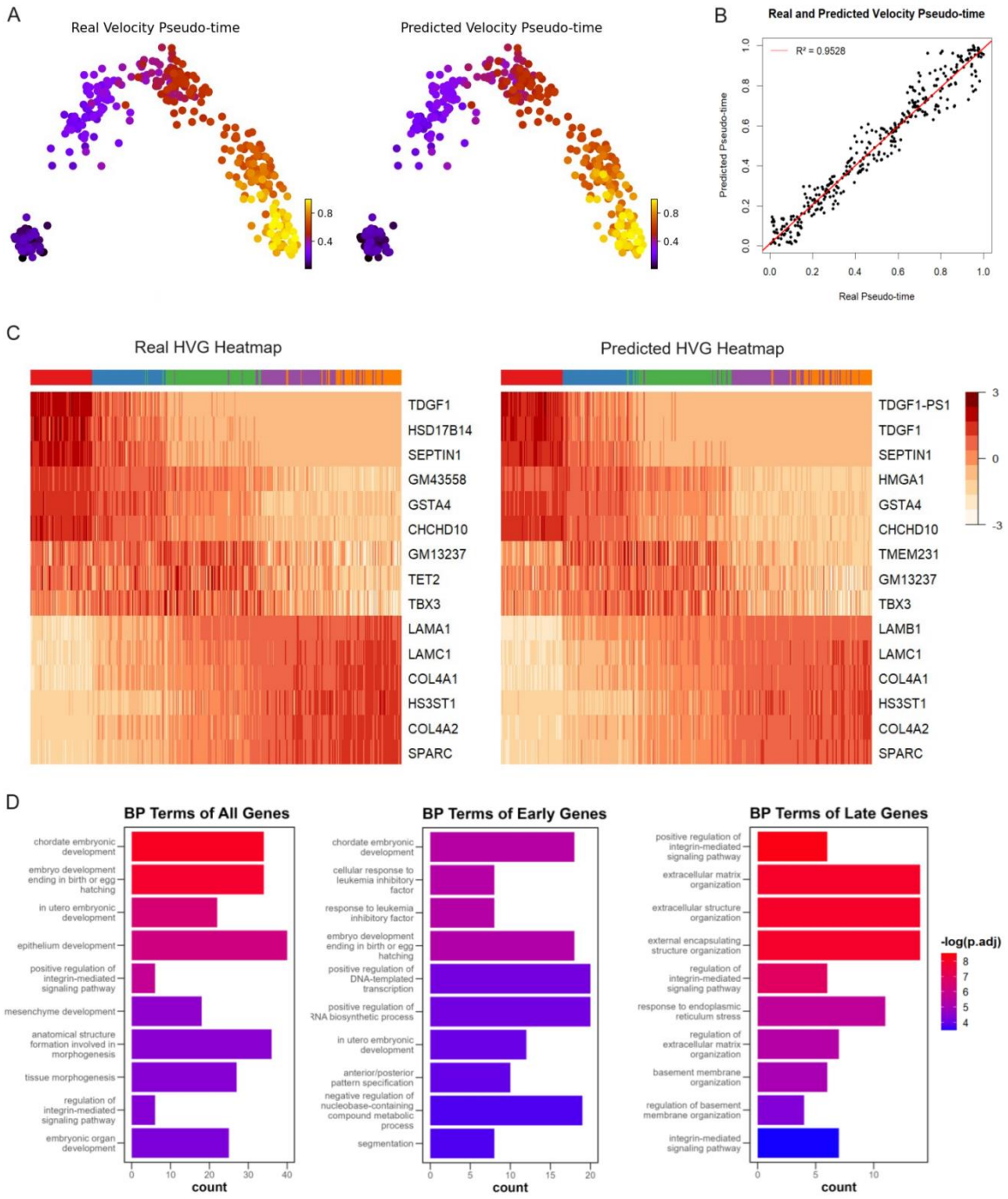

图 3 mESC 数据集的高变基因鉴别与分析。 A. 实际和预测 RNA 速率分别推断的伪时间。不同颜色代表不同的伪时间，降维方式同前。B. 两种伪时间的线性拟合。C. 关于两种伪时间的高变基因表达热图。热图中颜色深浅代表基因表达量的标准化值（Z-score）大小，两张图共同颜色标度在右上。每列上方的色块对应细胞采样时间，颜色映射同前，细胞顺序分别为两种伪时间顺序。HVG，高变基因。D. 预测 RNA 速率伪时间对应的高变基因富集的生物过程。从左到右依次为所有高变基因、早期激活基因、晚期激活基因富集结果。柱状图颜色对应 Bonferroni 校正的 P 值的负对数，三张图共同颜色标度在右侧。柱状图长度对应匹配到术语注释基因的个数。BP，生物学过程。

我们进一步对鉴定的不同阶段被激活的高变基因进行基因功能分析。由于两种数据得到的高变基因高度重叠，以及本研究主要目的是说明 RNA 速率预测的可靠性，下面我们仅分析预测数据鉴别的高变基因。我们从每一类基因各选取前 30 个基因，分别或者共同用于 GO 富集分析（图 3D）。其中，5 类高变基因（见附录表 A.3）一同富集到了多个与胚胎发育、形态发生相关术语以及“上皮细胞发育”和“间充质细胞发育”等多种细胞类型形成相关术语，与胚胎干细胞的多能性相符。此外，第 1、2 类早期激活基因（“早期基因”）和第 4、5 类晚期激活基因（“晚期基因”）富集的术语不尽相同。早期基因特异地富集了与体轴和分节模式建立相关术语，与胚胎早期发育事件相吻合。晚期基因则特异地富集了与细胞外基质重塑和调控的术语，而这是细胞分化后期和组织成熟必需的过程。这些结果共同说明了预测的 RNA 速率的准确性及其在寻找潜在的标记基因或生物过程关键基因上的可靠性。

此外，我们从之前构建的模型中提取一个形式上的 GRN 以供可能的下游分析。尽管一种可行的提取 GRN 的方法是对每个细胞计算向量场在当下基因表达状态的雅各布矩阵，但是这样计算量过大，尤其当应用于更多基因或者特征的数据时。考虑到计算时间以及本研究的重心，我们使用基因编码的相关性作为更简单快速的 GRN 表示方式。

我们首先比较预测的 GRN 与参考基准值 GRN（由 ChIP-seq 实验数据推断的 GRN，见实验数据部分）以说明该形式上的 GRN 的有效性。需要注意的是，虽然本研究并不提倡这种传统的 GRN，且我们的 GRN 和实验的 GRN 的含义也不完全相同，但是实验 GRN 作为传统且被广泛使用的参考值仍有其参考意义，我们在这里也将其用于我们的 GRN 评估以供同行参考。

通过比较参考值 GRN 和预测值 GRN，我们可以绘制 TPR-FPR 曲线并计算 AUROC，并以此为指标衡量 GRN。许多现有方法在这一数据集上的 AUROC 集中在 0.50-0.53（图 4B），整体较差的表现可能源于参考值 GRN 的稀疏性，即正负样本不平衡，以负例为主。然而，本研究计算的 GRN 的 AUROC 高达 0.778（图 4A），即使与参考 GRN 的接近程度并不是我们在训练模型时的优化目标。与参考值 GRN 比较的优异表现再次证实了我们的模型的有效性和 GRN 提取方法的可行性。

我们进一步使用提取的 GRN 推断基因模块，即转录因子及与其共同参与生物学过程调控的基因集。我们从预测值 GRN 中按一定筛选标准保留部分调控边和部分转录因子，得到共 64 个转录因子的基因模块，并计算其活性。从热图可以看出，仅使用基因模块活性这 64 个特征就可以较好地分辨出五个阶段的细胞（图 5）。这一方面相比于全部的 1620 个基因明显地降低了特征维度，有利于后续分析；另一方面这些特征对于不同细胞的分辨能力说明该特征提取方式在一定程度上的有效性以及用于计算该特征的 GRN 的有效性。但仍需要承认的是，48h 和 72h 细胞的区分以及 12h 和 24h 细胞的区分并不是很明显，这可能说明形式 GRN 的提取方式有待依据需求地改进，以更好地满足多样的下游分析。

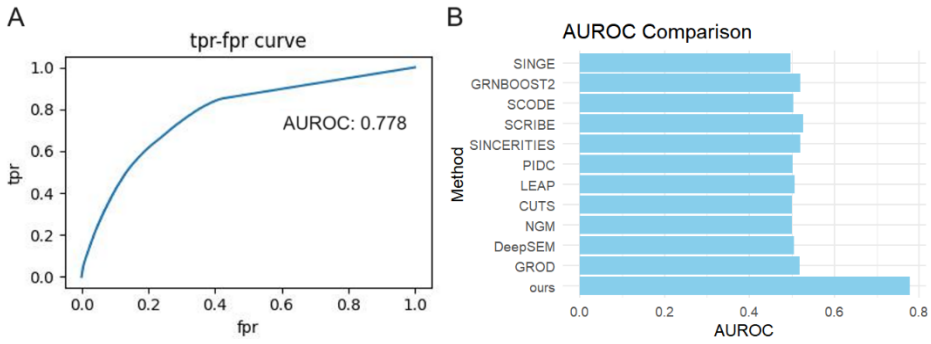

图 4 mESC 数据集的 GRN 预测性能。A. 预测值 GRN 与参考值 GRN 比较的 TPR-FPR 曲线。B. 不同方法该数据集以及该参考值 GRN 的 AUROC。最底下的 ours 行即本研究的模型。

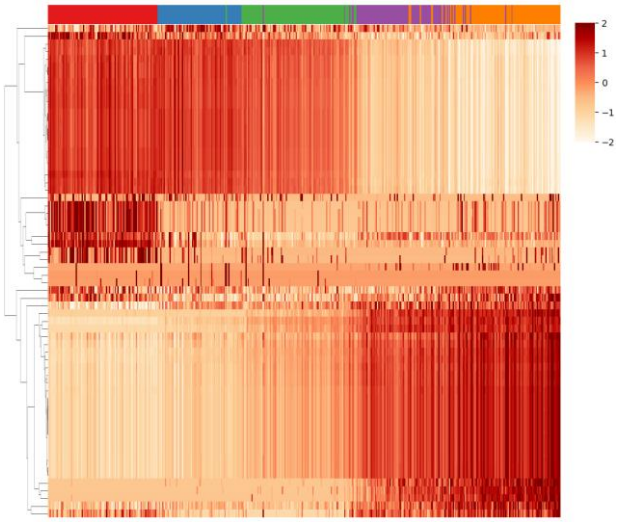

图 5 mESC 数据集的基因模块活性。 热图每行代表不同基因模块（聚类后），每列上方的色块对应细胞采样时间，颜色映射同前，细胞顺序为 slingshot 伪时间顺序。颜色深浅代表基因模块活性的标准化值（Z-score）大小，颜色标度在右上。

3.3 模型结构及超参数选择

我们对原始模型进行消融实验，测试不使用编码器、仅使用 Crossformer 编码、仅使用 GCN 编码，以及同时使用二者编码（即原始模型）的损失函数（图 6）。可以看出，预测性能最差的是无编码器模型，最好的是原始模型，另两个模型略差于原始模型。尽管四种模型两两之间差异显著，但是原始模型相较于仅选择一个编码器的模型性能提升并不太大，因此在计算资源和时间有限时可以考虑仅用 GCN 或 Crossformer 编码。

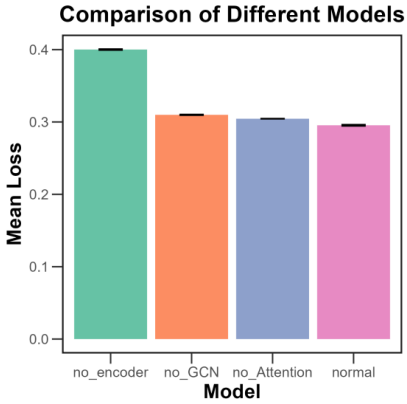

图 6 模型消融实验。柱状图从左到右模型依次为不使用编码器、仅使用 Crossformer 编码、仅使用 GCN 编码和原始模型，高度对应五次实验损失函数平均值，误差线为标准误差。

我们还对原始模型的超参数进行测试。首先，对于训练过程的学习率 learning_rate 和预热步数 warm_up_steps 参数，本研究尝试了共 30 种组合，学习率的范围从 3e-5 到 3e-2，预热步数范围从 1 到 7500（图 7）。可以看出，当学习率较大而预热步数不足时，损失函数可能会较大；而当预热步数达 2500 时，测试范围内的学习率都会有较好的表现，提示预热的重要性。

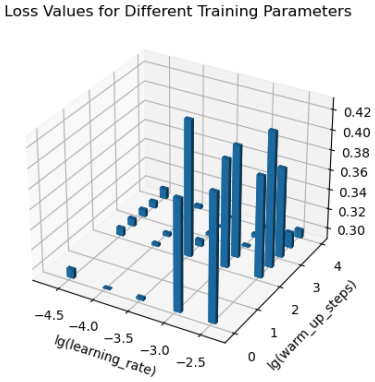

图 7 原始模型的训练参数测试。x 和 y 轴分别是学习率和预热步数的常用对数，z 轴是损失函数。

接着，我们测试几个比较重要的模型结构超参数，包括 Crossformer 前的节段长度 seg_len、Crossformer 中的编码维数 d_model、注意力头数 n_heads，以及中转参数长度 router（图 8）。本研究意图对每种参数尝试三种不同取值共 81 种组合，但是在 d_model 尝试中发现过大的 d_model 的损失函数较大且耗时明显过长，因此对于 d_model=256 仅尝试了四种组合并认为不必尝试更多组合。比较发现，模型表现受除 d_model 外的三个参数影响较小，一定程度上说明了模型的稳健性。d_model 较小取值（如 16）表现可能略不如适中取值（如 64），但是过大取值（比如 256）训练效果反而不好，这可能是因为过于复杂的模型在比较有限的训练迭代数（本研究中，train_epochs=200）无法达到最优解或者比较好的次优解。另外，耗时同样显著地受 d_model 影响。耗时也同样受 seg_len 参数影响，尽管

在测试的区间内 seg_len 的影响不如 d_model 的。这可能是因为 seg_len 越小（对应 seg_num 越大）或者 d_model 越大均直接放大基因表达特征数，因此直接注意力分数的计算量（对 seg_num 和 d_model 均呈二次方增长）。总之，d_model 是在模型训练中需要重点调试的超参数。

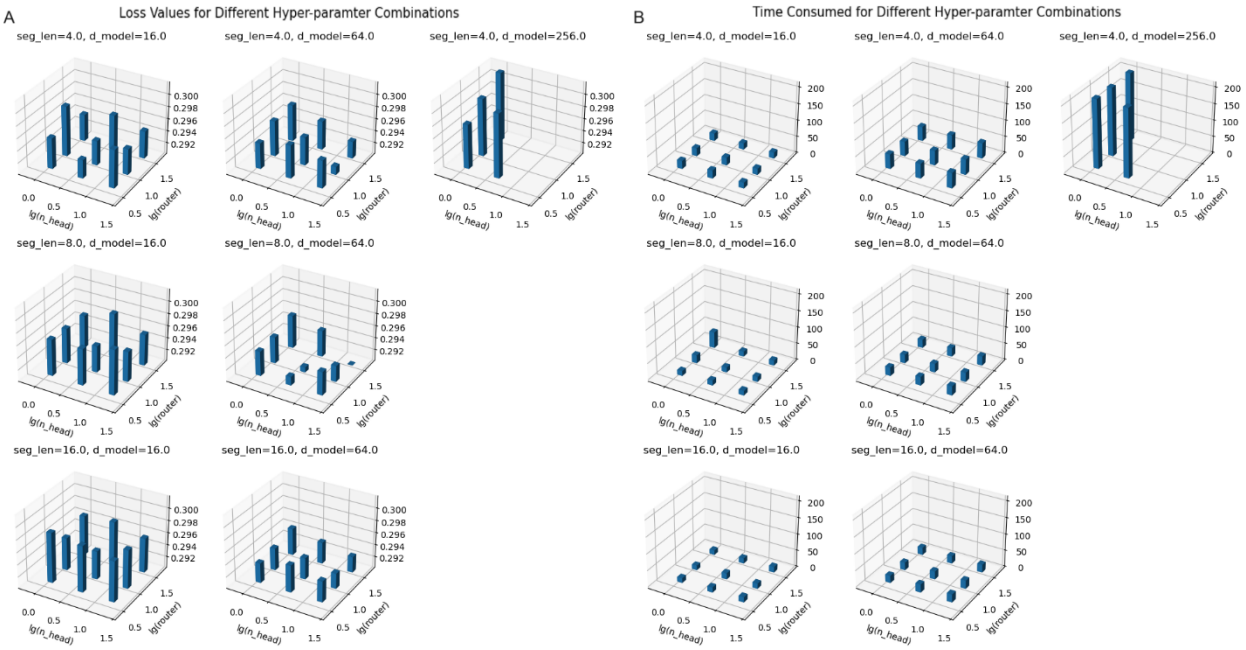

图 8 原始模型的结构超参数测试。A, B 的 x 和 y 轴分别代表注意力头数和中转参数长度的常用对数，A 的 z 轴为损失函数，B 的 z 轴为耗时。A, B 图中分别有一致的坐标尺度。

第四章 讨论

本研究的主要意义在于引入并验证了一种基于单细胞发育轨迹的 GRN 推断模型，加深了对单细胞数据中的细胞状态转变和动态基因调控的理解。本研究提供了一种新的方法框架，通过深度学习技术，包括用 Crossformer、GCN 对基因表达状态进行编码，用线性模型对下一时刻的基因表达编码状态进行预测，以及用两个维度的多层感知机对编码状态进行解码，成功地在 mESC 数据集上构建了该 GRN 推断模型。本研究还并通过从预测的速率推断伪时间并鉴别分析高变基因、从基因表达编码状态构建形式上的 GRN 并与实验值 GRN 比较，以及推断分析基因模块，共同说明其在生物学意义上的有效性。

相比之下，在传统的基因调控网络研究中，大多数模型通常忽视了细胞状态动态变化的意义。这会导致一方面只能提供静态的网络关系，另一方面无法借助细胞状态变化以推断 GRN。与之不同，本研究利用 RNA 速率的概念，使得 GRN 的推断不仅反映了基因间的调控关系，更重要的是揭示了细胞在这些关系的作用下随时间变化。这样意义明确的 GRN 模型的好处在于有客观的评价标准，即 RNA 速率的实际值（通过 scvelo 等软件计算值或其他方法计算值）和预测值的损失函数。另外，意义明确的模型还可以更妥当地用于下游分析，减少被错误理解和应用的可能。

尽管具有启发性，但本研究也存在一定局限性。首先，本研究使用的数据有明显的不足。目前模型主要基于 mESC 单细胞数据集进行训练和测试，数据集的单一性限制其在其他细胞类型或生物体中的应用。未来研究可以通过引入更多种类的单细胞数据来测试和提高模型的泛化能力。此外，本研究使用的 mESC 数据集没有未剪接和剪接后 RNA 表达量数据。尽管有一些结果显示本研究中 RNA 速率计算方法的合理性，但是如果使用更多的、可以用动力学建模方法（比如 scvelo）计算 RNA 速率的数据集可以使实验更严谨，或许可以达到更好的模型训练效果。另外，如何提高本研究中模型对不同数据集的适应性同样不容忽视。不同的单细胞数据集通常具有不同的主要特征，如何调整模型以适应性地选择输入特征并对不同数据集兼容是一个值得探索的方向。不兼容的模型在一个数据集上训练后无法用于其他数据集的预测。

其次，本研究的模型结构特别是编码器结构有待改进。尽管相比于无编码器的模型在损失函数上降低了 25%，但是考虑到多消耗的时间成本，这一性能提升幅度并不大。而且相比于单一编码器的模型，本研究主要使用的双编码器性能提升微乎其微。另外一个证据是当仅使用编码器和解码器对原始基因表达矩阵复原时，损失函数同样较高（结果未展示）。这些共同说明了编码器有较大的改善空间。此外，模型的预测器也较为简单，其他更复杂的如 Neural ODE 等模型值得尝试。

最后，从更广泛的科学研究和技术应用角度来看，本研究所采用的 GRN 推断方法为生物医学研究特别是精准医疗提供了新的展望。例如，通过准确推断出多样化的疾病状态下

的动态 GRN，医学研究人员可以更好地理解疾病的分子机制，从而开发出更有效的治疗策略。在技术层面，将单细胞 RNA 测序与其他类型的单细胞数据（如空间转录组数据、单细胞 ATAC 测序数据、Hi-C 数据）结合更能体现基因调控的动态性与复杂性，或许可以提高模型的预测精度。具体来讲，除了 RNA 表达量相关信息，空间转录组提供的空间与细胞通讯信息，Hi-C 提供的染色质构象数据等，可以共同补充解释无法由其他 RNA 表达量解释的 RNA 表达量变化，拓宽基因表达调控的概念。此外，探索这些 GRN 模型的其他应用场景、如何与多样的单细胞下游分析方法及工具结合，将是推动生物医学研究的一个重要课题。

参考文献

1. Stadhouders, R., Filion, G. J. & Graf, T. Transcription factors and 3D genome conformation in cell-fate decisions. *Nature* **569**, 345-354 (2019).
2. Schlauch, D., Glass, K., Hersh, C. P., Silverman, E. K. & Quackenbush, J. Estimating drivers of cell state transitions using gene regulatory network models. *BMC Syst Biol* **11**, 139 (2017).
3. Morrisey, E. E. *et al.* GATA6 regulates HNF4 and is required for differentiation of visceral endoderm in the mouse embryo. *Genes Dev* **12**, 3579-3590 (1998).
4. Schrode, N., Saiz, N., Di Talia, S. & Hadjantonakis, A.-K. GATA6 levels modulate primitive endoderm cell fate choice and timing in the mouse blastocyst. *Dev Cell* **29**, 454-467 (2014).
5. Van den Berge, K. *et al.* Trajectory-based differential expression analysis for single-cell sequencing data. *Nature Communications* **11**, 1201 (2020).
6. Badia-I-Mompel, P. *et al.* Gene regulatory network inference in the era of single-cell multi-omics. *Nature Reviews. Genetics* **24**, 739-754 (2023).
7. Kim, D. *et al.* Gene regulatory network reconstruction: harnessing the power of single-cell multi-omic data. *NPJ Syst Biol Appl* **9**, 51 (2023).
8. Ma, A. *et al.* Single-cell biological network inference using a heterogeneous graph transformer. *Nature Communications* **14**, 964 (2023).
9. La Manno, G. *et al.* RNA velocity of single cells. *Nature* **560**, 494-498 (2018).
10. Bergen, V., Lange, M., Peidli, S., Wolf, F. A. & Theis, F. J. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology* **38**, 1408-1414 (2020).
11. Qiu, X. Mapping transcriptomic vector fields of single cells. *Cell* **185** (2022).
12. Vaswani, A. et al. Attention is all you need. Advances in Neural Information Processing Systems **30** (2017).
13. Zhang, Y. & Yan, J. Crossformer: transformer utilizing cross-dimension dependency for multivariate time series forecasting. *International Conference on Learning Representations* (2023).
14. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations* (2017).
15. Aibar, S. *et al.* SCENIC: single-cell regulatory network inference and clustering. *Nature Methods* **14**, 1083-1086 (2017).
16. Street, K. *et al.* Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* **19**, 477 (2018).
17. Hayashi, T. *et al.* Single-cell full-length total RNA sequencing uncovers dynamics of recursive splicing and enhancer RNAs. *Nature Communications* **9**, 619 (2018).
18. Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A. & Murali, T. M. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature Methods* **17**, 147-154 (2020).
19. Wang, P. *et al.* Deciphering driver regulators of cell fate decisions from single-cell transcriptomics data with CEFCON. *Nature Communications* **14**, 8459 (2023).
20. Chen, J., Bardes, E. E., Aronow, B. J. & Jegga, A. G. ToppGene Suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Res* **37**, W305-W311 (2009).

21. Dong, J. & Wang, F. GROD: joint inference of gene regulatory networks and data imputation in single-cell RNA sequencing with temporal consideration. *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 275-280 (2023).

附录 A

附录 1

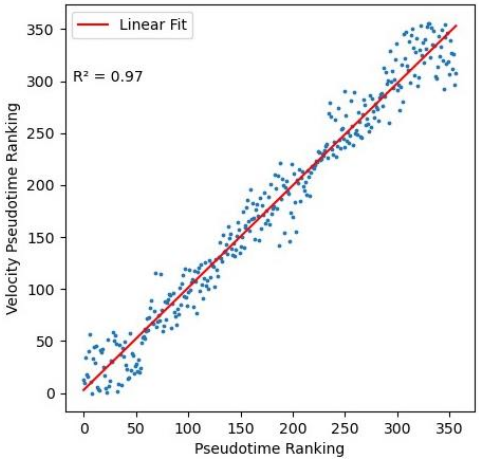

图 A.1 slingshot 伪时间与 scvelo 伪时间比较。 对两种伪时间线性拟合，强线性相关性说明 RNA 速率计算方法的可靠性。

附录 2

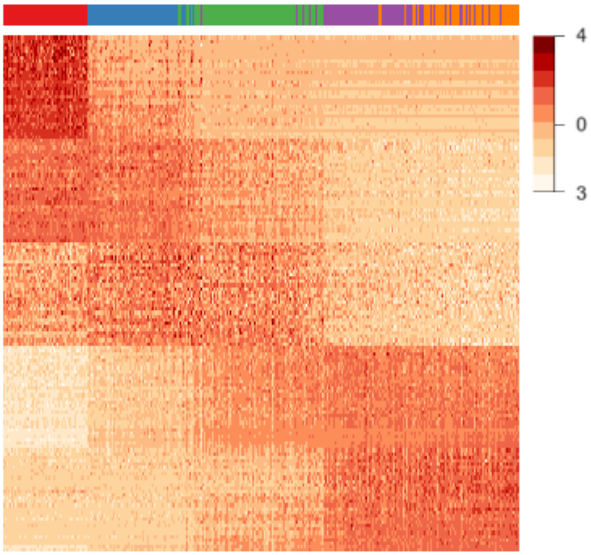

图 A.2 五类高变基因表达热图。图例同图 3C。此图展示了 150 个高变基因，更能看出第 4、5 类基因表达模式区别（第 4 类相比第 5 类在 24h 细胞中表达更强）。

附录 3 用于 GO 富集分析的 5 类高变基因列表

第 1 类：*SEPTIN1, TDGF1, TDGF1-PS1, HSD17B14, IFITM1, ADAM23, PIPOX, IGFBP2, APOBEC3, IRF2BPL, SOX2, TCEA3, ESRRB, LRRC2, SLC7A3, GPA33, TET1, KLF2, TRH, ZFP42, GLI2, PHC1, ILDR1, BMP4, COBL, BNC2, NOTUM, NANOG, RARG, USP28*

第 2 类：*CHCHD10, GSTA4, HMGA1, GM43558, GM38312, GM44206, HMGA1B, RCC2, PARP1, TRIM28, FAM25A, RBPJ, AU018091, TRP53, IFITM2, COL18A1, SNRPN, DPPA4,*

ACTL6A, NUP210, HSP90AA1, RPP25, MSH6, GLDC, JARID2, TFCP2L1, GOT1, SERBP1, NASP, MYBL2

第 3 类：*TBX3, GM13237, TMEM231, SHKBP1, TET2, SNHG1, SLC5A11, PGD, DEK, ULK1, SRSF6, HCFC1, RPL5, SNORA73B, AI506816, SNHG16, NCOA3, ISYNA1, GM24451, TRIM24, ECD, CTCF, LPCAT1, SNHG12, DNMT3A, GABARAPL2, SFPQ, PLRG1, ETL4, SSB*

第 4 类：*COL4A1, LAMC1, LAMB1, LAMA1, GATA4, SERPINH1, P4HB, CALU, KDELR2, DAB2, HDLBP, TAX1BP3, PPIB, PDIA3, ARHGAP29, TXNDC5, CREB3L2, APPBP2, SURF4, TXNDC12, RRBP1, TMED2, COPA, GFPT1, TMEM214, HSPG2, CLTC, VAMP8, CALR, GORASP2*

第 5 类：*SPARC, COL4A2, HS3ST1, SRGN, P4HA1, PTH1R, BMPER, APP, P4HA2, ERP29, RCN3, UBQLN2, HKDC1, CD63, COLGALT1, POFUT2, EFEMP2, ADAM19, NID1, CEMIP, NXF7, CLCN5, PTPN14, TIMP2, BSG, ABHD5, MYH9, ANK, PLOD3, LMF2*

附录 4

表 A.4 不同软件在预测 mESC 数据集的 GRN 的 AUROC 值

GROD	DeepSEM	NGM	CUTS	LEAP	PIDC	SINCERITIES	SCRIBE	SCODE	GRNBOOST2	SINGE
0.518	0.504	0.500	0.500	0.507	0.501	0.520	0.527	0.503	0.519	0.497

致谢

在本研究工作中，我得到了许多人的支持和帮助，我在此向他们表示衷心的感激。我要特别感谢我的导师王劲卓老师，他在我的毕业设计期间提供了细致的指导和无限的耐心。我也非常感谢同实验室的彭睿师兄和徐帅师兄在参考工作的搜集、研究框架的构建和数据集的选择上给予我的巨大的帮助和宝贵的建议。同时，我也要感谢实验室内的其他同学在研究过程中的讨论和意见。此外，我非常感激我的家人和朋友，他们在物质和精神上给予了我持续的支持。我也要对北京大学提供的诸如学术期刊、学术软件和计算资源等丰富的学术资源表示感谢，这些资源为我的研究提供了坚实的后盾。最后，我感谢自己的努力和坚持，以及对生物信息学领域的热情，这使我能够克服难关，顺利完成研究。

北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：

日期：　　年　　月　　日

学位论文使用授权说明

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文；

论文作者签名：　　导师签名：

日期：　　年　　月　　日