OXFORD

Briefings in Bioinformatics, 2025, **26(2)**, bbaf089
https://doi.org/10.1093/bib/bbaf089
Problem Solving Protocol

Inferring gene regulatory networks from time-series scRNA-seq data via GRANGER causal recurrent autoencoders

Liang Chen[1], Madison Dautle[2], Ruoying Gao[1], Shaoqiang Zhang[1,*], Yong Chen[2,*]

[1]College of Computer and Information Engineering, Tianjin Normal University, 393 Binshui W Ave, Tianjin, Tianjin 300387, China
[2]Department of Biological and Biomedical Sciences, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028, United States
[*]Corresponding author. Shaoqiang Zhang, Tel: +86 186 4900 6931. Fax: +86 22-23766303. E-mail: zhangshaoqiang@tjnu.edu.cn; Yong Chen, Tel: +1 856 256 4500. ext. 53589. Fax: +1 856 256 4921. E-mail: chenyong@rowan.edu

Abstract

The development of single-cell RNA sequencing (scRNA-seq) technology provides valuable data resources for inferring gene regulatory networks (GRNs), enabling deeper insights into cellular mechanisms and diseases. While many methods exist for inferring GRNs from static scRNA-seq data, current approaches face challenges in accurately handling time-series scRNA-seq data due to high noise levels and data sparsity. The temporal dimension introduces additional complexity by requiring models to capture dynamic changes, increasing sensitivity to noise, and exacerbating data sparsity across time points. In this study, we introduce GRANGER, an unsupervised deep learning-based method that integrates multiple advanced techniques, including a recurrent variational autoencoder, GRANGER causality, sparsity-inducing penalties, and negative binomial (NB)-based loss functions, to infer GRNs. GRANGER was evaluated using multiple popular benchmarking datasets, where it demonstrated superior performance compared to eight well-known GRN inference methods. The integration of a NB-based loss function and sparsity-inducing penalties in GRANGER significantly enhanced its capacity to address dropout noise and sparsity in scRNA-seq data. Additionally, GRANGER exhibited robustness against high levels of dropout noise. We applied GRANGER to scRNA-seq data from the whole mouse brain obtained through the BRAIN Initiative project and identified GRNs for five transcription regulators: *E2f7*, *Gbx1*, *Sox10*, *Prox1*, and *Onecut2*, which play crucial roles in diverse brain cell types. The inferred GRNs not only recalled many known regulatory relationships but also revealed sets of novel regulatory interactions with functional potential. These findings demonstrate that GRANGER is a highly effective tool for real-world applications in discovering novel gene regulatory relationships.

Keywords: scRNA-seq; gene regulatory network; unsupervised learning; GRANGER causality; recurrent variational autoencoder

Introduction

Deciphering gene regulatory networks (GRNs) is crucial for understanding the complex biological functions of cells, revealing how precise spatiotemporal regulation of genes maintains functions in normal cells and dysfunctions in disease states [1–3]. scRNA-seq is a powerful technique for studying cellular dynamics, enabling measurement of subtle differences in gene expression between cells. This is especially useful for study of those participating in cellular processes such as cell differentiation or cell cycle progression [4]. By analyzing gene expression profiles, we can effectively capture the dynamic dependencies between genes, facilitating the reverse engineering of GRNs. Consequently, these inferred networks provide crucial biological information for the relationships between transcription factors (TFs) and their target genes [5].

Dozens of computational methods to infer GRNs from scRNA-seq data have been proposed, uncovering the potential regulatory dependencies that underlie cellular phenotypes from a genomic perspective [5–7]. Most early methods focused on static scRNA-seq data, including GENIE3 [8], GRNBoost2 [9], SCENIC [10], NLNET [11], PIDC [12], PPCOR [13], and SINCERA [14]. Another type of GRN inference methods utilize time-ordered or pseudotime-ordered gene relationships across different stages of cell development, including LEAP [15], AR1MA1-VBEM [16], SINCERITIES [17], SCIMITAR [18], NME [19], Normi [20], Velorama [21], SINGE, [22] and scTIGER [23]. Most of these approaches begin by using probabilistic or regression models to fit the time or pseudotime series gene expression data and primarily rely on correlation metrics to predict gene regulatory relationships. For example, LEAP [15] uses Pearson's correlation and permutation testing to estimate pairwise co-expression between genes. MMFGRN leverages advanced matrix factorization techniques to integrate multi-modal data to infer regulatory relationships [24]. Additionally, some methods use differential equations to simulate changes of gene expression values over pseudotimed cell orders, including dynGENIE3 [25], Inference Snapshot [26], SCODE [27], SCOUP [28] and GRISLI [29]. Although those methods attempt to incorporate time-series features, they frequently rely on heuristic optimization or linear

Received: October 31, 2024. **Revised:** January 26, 2025. **Accepted:** February 18, 2025
© The Author(s) 2025. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (https://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

assumptions, limiting their ability to model complex, non-linear regulatory relationships.

Recently, the increasing availability of verified TF-gene pairs for specific cell types has facilitated the development of supervised deep learning methods for reconstructing GRNs, including CNNC [30], DeepDRIM [31], dynDeepDRIM [32], GENELink [33], GNNLink [34], TDL [35], SCGRNs [36], STGRNS [37], DeepIMAGER [38], and scGeneRAI [39]. Furthermore, several GRN inference methods, such as scMEGA [40], Dicts [41], LINGER [42], and SCENIC+ [43], utilize not only TF-binding motif enrichments but also other single-cell multi-omics data (e.g., scRNA-seq and scATAC-seq). Although the training accuracies of these supervised deep learning methods are notably high [37, 38], the models trained on partially known TF-gene pairs remain biased due to the absence of a comprehensive global ground truth for the biological processes under investigation. Moreover, the TF-gene pairs used by these supervised methods are primarily derived from specific cell types, limiting the effectiveness of transfer learning when applied to different cell types [31, 38, 44]. As a result, unsupervised methods for inferring GRNs from transcriptomics data remain the dominant approach [45], serving as a crucial strategy for cells with limited availability of omics data.

The BEELINE framework [46] was designed for evaluating dozens of GRN inference tools on scRNA-seq data and found that tools incorporating pseudotime orders of cells showed better accuracy on synthetic datasets, but less accuracy on real datasets. This is because the scRNA-seq has considerable noise and sparsity, making it challenging to accurately infer GRNs from time-series scRNA-seq data [39, 46]. Furthermore, the performance of GRN inference algorithms is often compromised by high rates of false positive regulations [47], especially in many correlation-based methods [10, 15, 27, 48]. To address these limitations, we propose an unsupervised GRN prediction method by employing GRANGER recurrent autoencoders to network gene expression regulations (referred to as GRANGER). GRANGER mainly integrates GRANGER causality [49] into a recurrent variational autoencoder (VAE) [50] to infer GRNs. In addition, a negative binomial (NB) distribution is also incorporated into the loss function to correct for the distribution of gene expression profiles across all cells, as incorporating the NB distribution into deep learning frameworks has been demonstrated to improve both data imputation and the precision of cell clustering [51]. We evaluated GRANGER's inference accuracy using both simulation and real scRNA-seq datasets, and found it demonstrated superior performance compared to other unsupervised GRN inference methods.

Materials and methods

Overview of GRANGER

To infer GRNs from a scRNA-seq dataset of m genes, GRANGER translates the expression profiles of each gene g in all cells into a time-series sequence $X^g = X^g_{1:T} = (X^g_1, X^g_2, \cdots, X^g_T)$, where T represents the number of time points, adhering to the underlying principles of GRANGER causality during time series generation process [49] (Fig. 1A). GRANGER applies a causal recurrent VAE to all sequences $X^1, \ldots, X^g, \ldots, X^m$ to predict regulatory relationships among genes. The causal recurrent VAE in GRANGER consists of an encoder and a multi-head decoder. Each decoder head is responsible for generating X^g, the time-series expression of a gene. The encoder reduces the multivariate time-series of each gene to a lower-dimensional feature space. In the inferred GRN, the edge between two genes is directed, representing a regulatory relationship between a gene (typically a TF) and its target genes (Fig. 1B).

Here the inferred GRN is represented as a GRANGER causal graph [50] $G = (V, E)$, where the set of nodes V consists of the set of time series $X^1, \ldots, X^m$, and each edge $(h, g) \in E$ connecting X^h to X^g, which represents X^h causes X^g in time instant t with a lag, if the past profiles X^h_{t-} of X^h provide statistically significant information about the prediction of X^g_t at time t (note that self-loops are allowed in graph G). The graph $G = (V, E)$ can be characterized by an adjacency matrix $A = (A_{gh})_{m\times m}$,

$$A_{gh} = \begin{cases} 1, & X^h_{t-} \text{ causes } X^g_t, (\text{i.e., edge } (h,g) \in E) \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Therefore, the main objective in GRANGER is to infer the matrix A, each of whose elements A_{gh} represents whether gene h regulates gene g.

Let $X_t = (X^1_t, X^2_t, \cdots, X^m_t)^T \in \mathbb{R}^m$ be the feature vector of all sequences$X^1, \ldots, X^m$ at time t and $X_{a:b} = (X^1_{a:b}, X^2_{a:b}, \cdots, X^m_{a:b})^T$ be the feature matrix of all sequences from time a to time b. As shown in Fig. 1B, a time segment $X_{(t-2l-1):t}$ is extracted from $X_{1:T}$ of the m time-series sequences to set as a batch sample of training data. The first half $X_{(t-2l-1):(t-l-1)}$ of the time segment is taken as the input to encoder, and the second half $X_{(t-l):(t-1)}$of the time segment and the latent space output by encoder are taken as the input of decoder, whose aim is to predict or reconstruct the segment $X_{(t-l):t}$ with time lag l. The decoder contains m heads and the g-th head predicts the g-th sequence X^g, $g = 1, 2, \ldots, m$. A recurrent neural network (RNN) with gated recurrent units (GRUs) [52] is separately employed in the encoder and each head of decoder to form a recurrent VAE.

Data preprocessing and time series generation

The Python package SCANPY (version1.9.3) [53] was used for preprocessing and filtering the raw scRNA-seq data by: (i) removing genes expressed in fewer than 10% of cells, (ii) eliminating low-quality cells that express fewer than three genes or exhibit high mitochondrial gene expression, (iii) normalizing and log-transforming the filtered matrix, and (iv) selecting a subset of highly variable genes for further analysis. If the scRNA-seq data did not contain inherent time-series or pseudotime information, the PAGA method [54] was used through the 'scanpy.tl.paga' function to predict the pseudotime order of cells.

Variational autoencoder based on GRANGER causality

Suppose gene regulatory relationships can be modeled as a GRANGER causal graph G for a set of time series $X^1, \ldots, X^m$. Let $C(X^g)$ be the set of causes (or parents) of X^g in graph G, i.e., $C(X^g) = \{X^h : A_{gh} = 1\}$. At time point t, X^g_t can be predicted by its past profiles X^g_{t-} and the past profiles $C(X^g)_{t-}$ of its cause variables $C(X^g)$. So X^g_t can be represented by a function of X^g_{t-} and $C(X^g)_{t-}$ with additive independent noise ε^g_t. That is,

$$X^g_t = f_g\left(X^g_{t-}, C(X^g)_{t-}\right) + \varepsilon^g_t. \quad (2)$$

To learn the function f_g for each X^g and infer the matrix A, we used a VAE whose encoder and decoder are both composed of RNNs. If we input the first half segment $X_{(t-2l-1):(t-l-1)}$ to the encoder and use the decoder to reconstruct the second half segment $X_{(t-l):t}$,

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

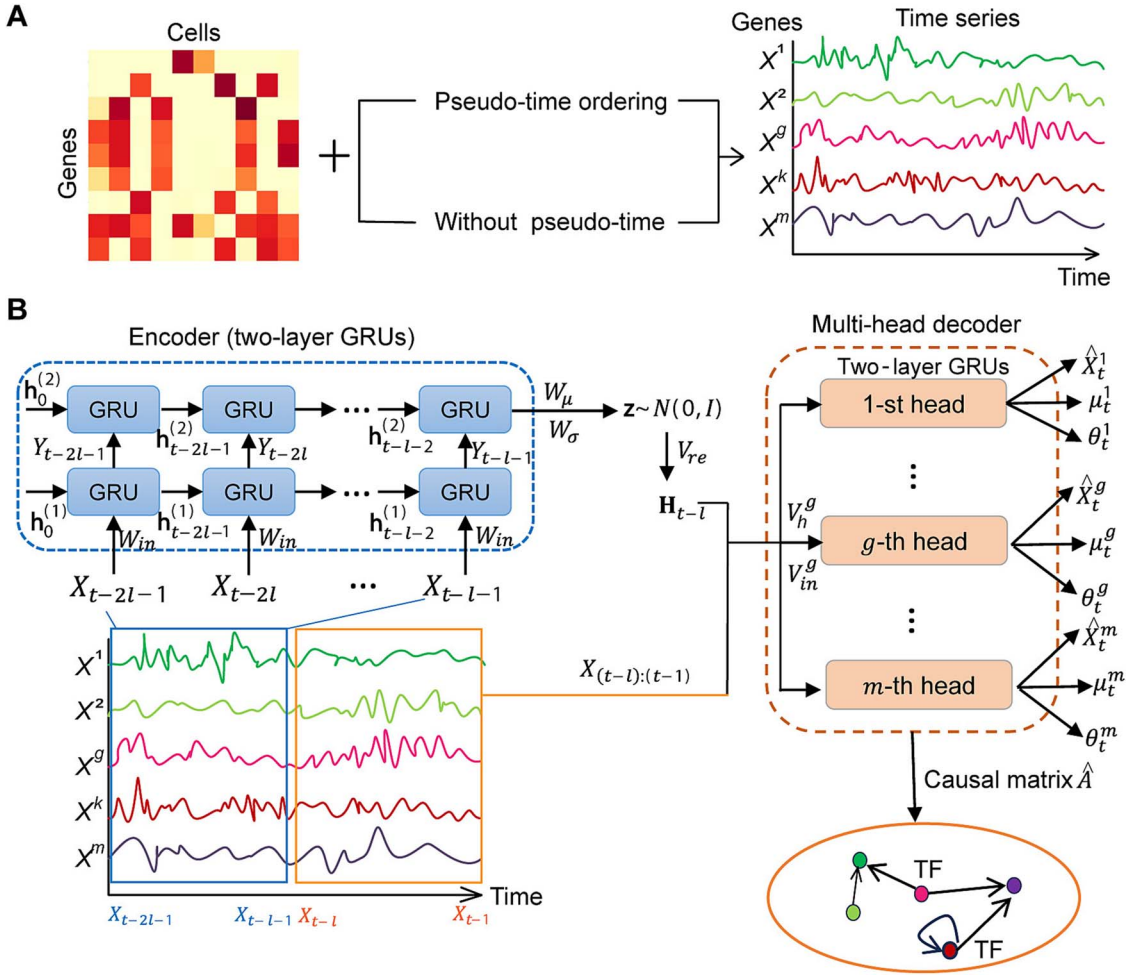

Figure 1. The workflow of GRANGER. (A) For m genes, time-series sequences $\{X^g : g = 1, 2, \ldots, m\}$are generated from the preprocessed gene expression profiles based on cell pseudotime ordering. (B) The pipeline of the recurrent variational autoencoder based on GRANGER causality testing.

the VAE with time lag l is:

$$\hat{X}_{(t-l):t} = D_\varphi\left(X_{(t-l):(t-1)}, E_\psi\left(X_{(t-2l-1):(t-l-1)}\right)\right) + \varepsilon_t, \tag{3}$$

where D_φ and E_ψ represent decoder and encoder with parameter sets φ and ψ, respectively, and ε_t is an additive term without any specific assumption. Furthermore, the decoder has multiple heads, in which the g-th head is used to learn $\hat{X}_t^g = f_g\left(X_{t-}^g, C(X^g)_{t-}\right)$ in Equation (2). Finally, the full vector $\hat{X}_t$ is formed by stacking the output of all m heads. Thus, the VAE based on GRANGER causality can learn the functions $\{f_g : g = 1, 2, \ldots, m\}$.

For $X_\tau \in \mathbb{R}^m$, $\tau = (t - 2l - 1), \ldots, (t - l - 1)$, let $\mathbf{h}_\tau$ be the corresponding hidden state in the encoder. The encoder is formulated by the following equations from (4) to (9), the first four of which are the GRU equations:

$$\mathbf{r}_\tau = \text{Sigmoid}\left(W_r X_\tau + W_{hr}\mathbf{h}_{\tau-1} + b_r\right), \tag{4}$$

$$\mathbf{z}_\tau = \text{Sigmoid}\left(W_z X_\tau + W_{hz}\mathbf{h}_{\tau-1} + b_z\right), \tag{5}$$

$$\tilde{\mathbf{h}}_\tau = \tanh\left(W_{in}X_\tau + \mathbf{r}_\tau \odot (W_h\mathbf{h}_{\tau-1}) + b_h\right), \tag{6}$$

$$\mathbf{h}_\tau = (1 - \mathbf{z}_\tau) \odot \tilde{\mathbf{h}}_\tau + \mathbf{z}_\tau\mathbf{h}_{\tau-1}, \tag{7}$$

$$\mu = W_\mu\mathbf{h}_{t-l-1} + b_\mu, \tag{8}$$

$$\sigma = \exp\left(W_\sigma\mathbf{h}_{t-l-1} + b_\sigma\right), \tag{9}$$

where $\mathbf{r}_\tau$ is the reset gate vector of GRU, $\mathbf{z}_\tau$ is the update gate vector of GRU, $\odot$ represents Hadamard product, and each b denotes the corresponding bias. $\{W_r, W_{hr}, W_z, W_{hz}, W_{in}, W_h, W_\mu, W_\sigma\} \subset \psi$ are weight parameters, where W_{in} and W_h are the weight matrices for input and hidden state in the encoder, respectively, and W_μ and W_σ are the weights to calculate the mean μ and the standard deviation σ of the learned Gaussian distribution.

For $X_\tau \in \mathbb{R}^m$, $\tau = (t - l), \ldots, t$, let $\mathbf{H}_\tau$ denote the corresponding hidden state in the decoder and $\mathbf{H}_\tau^g$ the hidden state of the g-th head. The initial state of the decoder is sampled from the Gaussian distribution which is parameterized with μ and σ. That is,

$$\mathbf{H}_{t-l} = \tanh\left(V_{re}(\mu + \sigma\mathbf{z}) + b_{re}\right), \tag{10}$$

where $\mathbf{z} \sim N(0, I)$, $V_{re}$is weight for reparameterization, and b_{re} is the bias. The decoder is formulated by Equations (11) (18) for the g-th head, the first four of which are the GRU equations:

$$\mathbf{R}_\tau^g = \text{Sigmoid}\left(V_r^g X_\tau^g + V_{hr}^g\mathbf{H}_{\tau-1}^g + b_r^g\right), \tag{11}$$

$$\mathbf{Z}_\tau^g = \text{Sigmoid}\left(V_z^g X_\tau^g + V_{hz}^g\mathbf{H}_{\tau-1}^g + b_r^g\right), \tag{12}$$

$$\tilde{\mathbf{H}}_\tau^g = \tanh\left(V_{in}^g X_{\tau-1} + \mathbf{R}_\tau^g \odot (V_h^g\mathbf{H}_{\tau-1}^g) + b_h^g\right), \tag{13}$$

$$\mathbf{H}_\tau^g = (1 - \mathbf{Z}_\tau^g) \odot \tilde{\mathbf{H}}_\tau^g + \mathbf{Z}_\tau^g\mathbf{H}_{\tau-1}^g, \tag{14}$$

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

$$\hat{A}_g = V_{in}^g, \tag{15}$$

$$\hat{X}_t^g = V_{out}^g \mathbf{H}_t^g + b_{out}^g, \tag{16}$$

$$\mu_t^g = \exp\left(V_\mu^g \mathbf{H}_t^g\right), \tag{17}$$

$$\theta_t^g = \exp\left(V_\theta^g \mathbf{H}_t^g\right), \tag{18}$$

where $\mathbf{R}_\tau$ is the reset gate vector of GRU, $\mathbf{Z}_\tau$ is the update gate vector of GRU and each b denotes the corresponding bias. $\left\{V_r^g, V_{hr}^g, V_z^g, V_{hz}^g, V_{in}^g, V_h^g, V_{out}^g, V_\mu^g, V_\theta^g\right\} \subset \varphi$ are weight parameters, where V_{in}^g, V_h^g *and* V_{out}^g are weight matrices for input, hidden state, and output layer of the g-th head in the encoder, respectively. To capture the characteristics of scRNA-seq data, X_t^g at time install t is assumed to be an NB distribution parameterized with mean μ_t^g and dispersion θ_t^g. Two NB output layers with Equations (17) and (18) were added to the last hidden state of the g-th head $\mathbf{H}_t^g$ to estimate the mean $\mu_t^g$and the dispersion θ_t^g of the learned NB distribution, respectively. V_μ^g and V_θ^g are the learned weights from the last hidden layer of the two NB output layers, respectively. $\hat{A}_g$ is the g-th row of the estimated adjacency matrix $\hat{A}$ of the GRANGER causal graph.

Loss function and network optimization

It is assume that each time-series variable X_t^gfits an NB distribution, i.e., $X_t^g \sim NB\left(\mu_t^g, \theta_t^g\right)$, where μ_t^g is the mean and θ_t^g is the dispersion. That is,

$$P_{NB}\left(X_t^g | \mu_t^g, \theta_t^g\right) = \frac{\Gamma\left(X_t^g + \theta_t^g\right)}{X_t^g!\,\Gamma\left(\theta_t^g\right)}\left(\frac{\theta_t^g}{\theta_t^g + \mu_t^g}\right)^{\theta_t^g}\left(\frac{\mu_t^g}{\theta_t^g + \mu_t^g}\right)^{X_t^g}. \tag{19}$$

Then the NB-based loss function is formulated as the negative log-likelihood of NB distributions:

$$L_{NB}(\psi, \varphi) = -\sum\nolimits_t \sum\nolimits_g \log\left(P_{NB}\left(X_t^g | \mu_t^g, \theta_t^g\right)\right). \tag{20}$$

As the adjacency matrix $\hat{A}$of the predicted GRNs is assumed to be sparse, a sparsity-inducing penalty $R(\hat{A})$ to $\hat{A}$ is appended to the loss function. The final loss function is defined as

$$L(\psi, \varphi) = \sum_{g=1}^{m}\left[\mathbb{E}_{q_\psi\left(\mathbf{z}|X_{(t-2l-1):(t-l-1)}\right)} \log p_\varphi\left(X_{(t-l):t}^g | X_{(t-l):(t-1)}, \mathbf{z}\right)\right]$$
$$- KL\left(q_\psi\left(\mathbf{z}|X_{(t-2l-1):(t-l-1)}\right) \| p(\mathbf{z})\right) + \lambda_{NB} L_{NB}(\psi, \varphi) + \lambda_A R\left(\hat{A}\right), \tag{21}$$

where the first term is the expected negative reconstruction error quantifying how well the model can reconstruct $X_{(t-l):t}^g$ for each g and the second term is the relative Kullback–Leibler (KL) divergence between the learned Gaussian distribution in the encoder and the prior distribution $\mathbf{z} \sim N(0, 1)$. The first term in Equation (21) is estimated by computing the cross-entropy between the training sample distribution and the reported probabilities. We used ℓ_1 norm to compute $R(\cdot)$, because ℓ_1 norm is easier to obtain sparse solutions than $\ell_2$norm and helps reduce the risk of overfitting.

GRANGER was trained by minimizing the loss function in Equation (21) with the proximal gradient descent (PGD) on the weights of input layers $V_{in}^g$and stochastic gradient descent (SGD) on all other parameters [55]. A two-stage training strategy is proposed: (i) we first trained the model by computing the gradients of the convex part of $L(\psi, \varphi)$ that is the first and second term in Equation (21), updating φ and ψ except V_{in}using SGD and updating V_{in} using PGD; (ii) we fixed all zero elements in V_{in} based on obtained causal matrix $\hat{A}$, and continue training the model with SGD on $L(\psi, \varphi)$. In practice, we used the iterative shrinkage thresholding algorithm (ISTA) with constant step size to do PGD [56]. To further improve the accuracy of predicted GRNs, two-layer GRUs are used in the recurrent VAE, where the input Y_t of the second layer is the hidden state $\mathbf{h}_t^{(1)}$ of the first layer.

Benchmark datasets and case study for GRANGER evaluation

To fairly evaluate the performance of GRANGER, we utilized the six synthetic and four curated datasets provided by the GRN evaluation framework BEELINE [46]. The six synthetic datasets simulate a linear trajectory (dyn-LI), a cyclical trajectory (dyn-CY), a long linear trajectory (dyn-LL), a bifurcating trajectory (dyn-BF), a trifurcating trajectory (dyn-TF), and a bifurcating converging trajectory (dyn-BFC). The four curated datasets focus on hematopoietic stem cell differentiation (HSC), mammalian cortical area development (mCAD), ventral spinal cord development (VSC) and gonadal sex determination (GSD). Each of the six synthetic datasets contains five sub-datasets with different numbers of cells (100, 200, 500, 2000, and 5000 cells), and each sub-dataset consists of 10 samples. Each of the four curated datasets contains 2000 cells and 10 samples. For each of curated datasets, BEELINE also provides samples containing 50% and 70% dropout zeros. All these benchmark datasets contain pseudotime-ordered information that were predicted by Slingshot [57] and preprocessed by the BEELINE framework. In addition, we conducted performance validation on four real scRNA-seq datasets, each separately sequenced from human embryonic stem cells (hESC), mouse dendritic cells (mDC), human mature hepatocytes (hHEP), and erythroid-lineage mouse hematopoietic stem cells (mHSC-E). We predicted the GRNs for the TFs with corresponding GRNs provided by BEELINE, including hESC (eight TFs and 310 targeted genes), hHEP (two TFs and 378 targeted genes), mDC (two TFs and 31 targeted genes), and mHSC-E (six TFs and 459 targeted genes).

As a real application, GRANGER was applied to predict GRNs from whole mouse brain data obtained from the Allen Brain Cell Atlas [58]. We downloaded the hippocampal formation (HPF) scRNA-seq data from the batch that was generated by 10Xv2 chemistry. We focused on Glutamatergic neurons (Glut) in the CA1 and prosubiculum (ProS) in HPF, labeled as CA1-ProS-Glut_1, which has 1055 sequenced neurons and predicted GRNs for five TFs, *Sox10*, *Onecut2*, *Gbx1*, *E2f7,* and *Prox1*. The data filtering and preprocessing of 1055 neurons was performed by using SCANPY (version1.9.3) [53]. PAGA was used to generate pseudotime series for the data [54]. For each TF, the top 20 highly correlated genes and top 100 highly variable genes were selected for GRN prediction by GRANGER. The functional enrichment of all genes in the merged GRN network were performed by using the PANTHER database [59–61].

Methods comparison and evaluation metrics

We compared GRANGER with popular unsupervised algorithms: GRNBoost2, SINCERITIES, PIDC, PPCOR, SCODE, GENIE3, and SINGE, which have shown high performance in validations within the comprehensive BEELINE framework [46] and other benchmarking studies [6, 7, 62]. We also included Normi, a recently published unsupervised algorithm that claims outstanding performance. The technical details of these eight methods are listed in Table 1 and Table S1. To ensure the consistency of

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

Table 1. The unsupervised GRN inference methods used in this study.

Method	Category	Directed?	Signed? (+/− regulation)	Pseudotime ordered?
GRANGER	GRANGER causality & Deep learning	Y	Y	Y
GRNBoost2	Random forest	Y	N	N
PIDC	Mutual Information	N	N	N
PPCOR	Correlation	N	Y	N
SCODE	Ordinary differential equations & Regression	Y	Y	Y
SINCERITIES	Regression	Y	Y	Y
Normi	Non-redundant mutual information	Y	N	Y
GENIE3	Regression tree-based ensemble	Y	N	N
SINGE	GRANGER causality & Regression	Y	N	Y

evaluating all methods, directionality, sign, and self-loops of gene–gene interactions were disregarded to measure performance in an identical manner for all methods.

The weights of the recurrent VAE in GRANGER were updated using the Adam optimizer in PyTorch [63], with a learning rate of 10^{-3}. The default batch size and epoch number were set to 8 and 500, respectively. The parameters for GRNBoost2, PIDC, PPCOR, SCODE, SINCERITIES, GENIE3, and SINGE were same as those used in the BEELINE framework (https://github.com/Murali-group/Beeline/tree/master/Algorithms). The parameters of Normi were also set to be default. All experiments were conducted on a Linux workstation equipped with an Nvidia GTX4090 GPU. We evaluated GRANGER and the other methods by calculating the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC) of each sample. As GRN inference is performed on datasets with imbalanced positive and negative samples, we primarily rely on the AUPRC metric, which is recommended for evaluating such tasks [64]. For real datasets (hESC, mDC, hHEP, and mHSC-E), we calculated the AUPRC ratio as the AUPRC divided by the expected AUPRC of a random predictor, providing a normalized measure of performance relative to randomness. The early precision ratio (EPR) was calculated as the fraction of true positives among the top-k predicted edges compared to a random predictor.

Results

The NB-based loss function improves the performance of GRANGER causal recurrent VAE

We systematically designed the loss function of GRANGER, by including not only the similar reconstruction loss and KL divergence loss as traditional VAE, but also an NB-based loss $L_{NB}(\psi, \varphi)$ and a spare-inducing penalty $R(\hat{A})$. The NB distribution has been employed to address the dropout events in scRNA-seq data [51, 65–67], and recent studies suggest that the NB model is more appropriate than other models for scRNA-seq data sequenced by droplet-based platforms [68, 69].

We evaluated the impact of the NB-based loss hyperparameter λ_{NB} in four datasets (dyn-CY, dyn-LI, HSC, and mCAD), each containing 2000 cells. We tested different values of λ_{NB}, ranging from 0 to 5, and assessed the performance of GRANGERs using the AUROC and AUPRC metrics. Across all datasets, we observed that when λ_{NB} was set to 1, GRANGER consistently achieved the highest AUROC and AUPRC scores compared to other values of λ_{NB} (Fig. S1). Specifically, in the dyn-CY dataset, the AUPRC reached around 0.90 when $\lambda_{NB}$was 1, demonstrating its effectiveness in improving predictive accuracy. Meanwhile, the AUPRC of the dyn-LI dataset was even higher at 0.98. In contrast, when λ_{NB} was set to 0 (i.e., without the NB-based loss), both AUROC and AUPRC scores were significantly lower, indicating the importance of including the NB-based loss to handle missing data and improve GRN prediction accuracy. Similarly, the trend was observed across the parameter testing of the HSC and mCAD datasets. The results confirm that including the NB-based loss consistently improves GRANGER's ability to infer GRNs.

Proper sparsity-inducing penalty improves the performance of GRANGER

We tested the performance of GRANGER with different values of the scale factor λ_A on four samples of 2000 cells from different datasets, fixing the other two hyperparameters with $l = 200$ cells and $\lambda_{NB} = 1$. We ran GRANGER on the four datasets with different λ_A settings ranging from 0.1 to 1. We found that the AUROC and AUPRC scores reached their peak when λ_A was set to values between 0.2 and 0.4 (Fig. S2). For example, in the dyn-CY dataset, $\lambda_A = 0.4$ resulted in the best performance, with an AUROC score of ~ 0.84 and an AUPRC score around 0.9. Similarly, for the dyn-LI dataset, the optimal performance occurred with $\lambda_A = 0.3$, where both AUROC and AUPRC scores were consistently high (AUROC $\sim$0.83 and AUPRC $\sim$0.98). In contrast, when λ_A was increased to 0.8 or higher, the model showed a significant drop in performance, with some experiments failing to complete within the allocated 500 epochs. This degradation is likely due to the model becoming overly sparse, leading to the learned weights of the model tending to be smaller or even close to zero. Based on these results, we set the default value of λ_A to 0.3 for subsequent experiments, as it provided superior results across multiple datasets.

Optimizing time lag and GRU structure of GRANGER

Besides the hyperparameters λ_{NB} and λ_A, the time lag l of each training sample also has a significant impact on causality inference. We tested how an appropriate division of time lag l could help the model to improve GRANGER causality inference. We first ran GRANGER on four samples from four datasets (dyn-CY, dyn-LI, HSC, and mCAD), each containing 2000 cells with settings of $\lambda_{NB} = 1, \lambda_A = 0.3$,and different time lag values, $l = 20, 50, 100, 200, 300, 400, 500, and\ 1000$. Among all time lags, the AUPRC and AUROC scores where $l = 200$ rank in the top two, consistently outperforming most other time lags in the four samples (Fig. 2A). For example, in the dyn-CY-2000 dataset, the highest AUROC and AUPRC scores (around 0.9 and 0.86, respectively) were observed when the time lag was set to 200. Similar trends were seen in dyn-LI-2000 and mCAD-2000, with the best performance observed at a time lag of 200 or 300. In addition, we conducted experiments on samples with different cell numbers for dyn-CY and dyn-LI with varying time lags. As shown in Fig. S3, we found that different optimal settings of l values depend on the

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

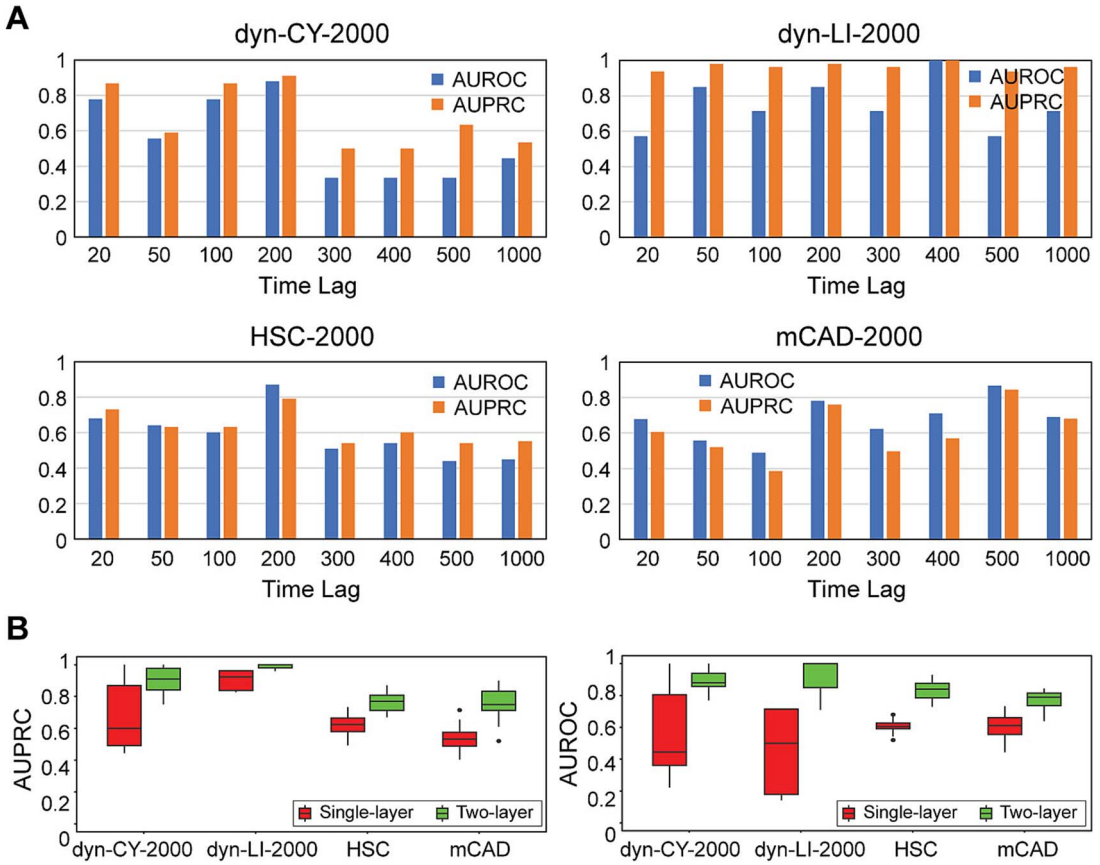

Figure 2. Performance of GRANGER for various time lags and layer structures across samples with different datasets. (A) Performance for various time lag settings for four samples, dyn-CY, dun-LI, HSC, and mCAD, respectively. (B) Box plots of AUROC and AUPRC scores of GRANGER with single-layer or two-layer GRUs on four datasets, dyn-CY, dyn-LI, HSC, and mCAD, each containing 2000 cells.

sample size. For samples containing 100 cells in dyn-CY and dyn-LI datasets, $l = 15$ is a relatively optimal setting; for samples with 500 cells,$l = 200$ is a relatively optimal setting; and for larger samples of 5000 cells, $l = 200 \sim 2,000$ are relatively optimal settings. To reduce computational complexity, we suggest that the setting $l = 200$ for large samples containing 500 or more cells.

In the encoder and each head of the decoder, two-layer GRUs are employed instead of single-lay GRUs. We conducted experiments on both single-layer GRU and two-layer GRUs on four datasets (dyn-CY, dyn-LI, HSC, and mCAD) each containing 2000 cells. For each of the four datasets, the median AUPRC and AUROC scores of GRANGER with two-layer GRUs are significantly higher than those of the model with single-layer GRUs (Fig. 2B). In dyn-CY-2000, For example, the AUPRC score for the two-layer GRU structure reaches around 0.91, whereas the single-layer model achieves only ~ 0.65. Similarly, in dyn-LI-2000, the two-layer GRU achieves an AUPRC score of around 0.95, compared to the single-layer's lower score 0.89. These results support that, compared with single-layer GRUs, two-layer GRUs preserve more contextual information, enabling the model to capture more complex patterns from the input data.

Impact of pseudo-temporal information on GRN prediction accuracy

In GRANGER, we integrated the PAGA method [54] to infer pseudo-time orders for cells. To evaluate the impact of different pseudo-time trajectory inference methods on GRN prediction, we compared the performance of GRANGER using three widely recognized methods, Slingshot [57], PAGA [54], SCORPIUS, [70] and randomly shuffled pseudo-time series (as control) across six datasets (dynCY-2000, dynLI-2000, dynLL-2000, GSD, HSC, and mCAD). As shown in Fig. S4, the results demonstrate that pseudo-time trajectory information generated by Slingshot and PAGA significantly enhances GRN prediction performance, achieving comparable and high AUPRC values across all datasets. SCORPIUS also yielded consistent results, though slightly lower than those achieved by Slingshot and PAGA. In contrast, GRANGER's performance on datasets with randomly shuffled pseudo-time series was slightly reduced, underscoring the importance of accurately inferred pseudo-temporal trajectories for robust GRN inference. Notably, the synthetic datasets (dynCY-2000, dynLI-2000, dynLL-2000) showed a greater dependency on precise pseudo-time information compared to the curated datasets (GSD, HSC, and mCAD), reflecting the complexity of GRNs in real biological systems. Additionally, the minimal differences between Slingshot and PAGA suggest their equivalence in predicting both tree and linear topological trajectories, consistent with findings from previous comparative analysis [71]. Overall, this analysis emphasizes the critical role of pseudo-temporal information in improving GRN inference and provides guidance on selecting suitable trajectory inference methods for scRNA-seq studies.

Comparing performance of GRANGER with other methods

We compared GRANGER with eight unsupervised GRN inference methods (GRNBoost2, PIDC, PPCOR, SCODE, SINCERITIES, GENIE, Normi, and SINGE). We ran GRANGER with the default hyperparameters trained in the previous sections (i.e., $\lambda_{NB} = 1$, $\lambda_A = 0.3$

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

and time lag $l = 200$ for the samples of 2000 cells). For six synthetic datasets (dyn-LI, dyn-CY, dyn-LL, dyn-BF, dyn-BFC, and dyn-TF), the results show that GRANGER consistently outperforms other methods across different datasets and cell numbers (Fig. 3). For example, in the dyn-LI and dyn-CY synthetic datasets, GRANGER achieves the highest median AUROC and AUPRC scores for cell numbers ranging from 100 to 5000 cells. For 100 and 200 cells in dyn-LI, GRANGER's AUROC scores range from 0.85 to 0.9, surpassing other methods, where the AUROC scores for GRNBoost2, PIDC, PPCOR, and SCODE are around 0.6–0.7 and the AUROC scores for SCODE fluctuate around 0.5–0.6. In dyn-CY, GRANGER achieves AUROC scores of 0.85–0.9 across all cell numbers, significantly outperforming methods like SCODE, which yields AUROC scores below 0.6 at smaller cell numbers. GRANGER's AUPRC scores show even greater improvement for all six datasets compared to the other methods. For complex trajectories (e.g., dyn-BF, dyn-TF, and dyn-BFC), GRANGER also maintains superior performance with minimal decline, whereas other methods exhibit a noticeable drop in AUPRC scores.

The performance improvement is most pronounced at smaller cell numbers (100 and 200 cells), where GRANGER demonstrates a significant advantage over competitors. For example, for dyn-LI with 100 cells, GRANGER achieves an AUPRC of ~ 0.95, whereas SCODE shows much lower AUROC values around 0.55. Similarly, in terms of AUPRC, GRANGER outperforms other methods with values approaching 0.98 for small cell sizes, while some methods (i.e., GRNboots2, PIDC, and PPCOR) struggle to reach 0.4. These results highlight GRANGER's reliability in handling smaller datasets, which present greater challenges for accurate GRN inference.

For four curated datasets (HSC, VSC, mCAD, and GSD), GRANGER continues to demonstrate superior performance (Fig. 3, last row). On the HSC and mCAD datasets, GRANGER consistently achieves higher median AUROC and AUPRC scores compared to other methods. While its performance on the GSD dataset is more similar to that of SCODE and SINGE, GRANGER still manages to maintain competitive results. We also evaluated the performance of GRANGER on four real scRNA-seq datasets (hESC, hHEP, mDC, and mHSC-E) using ground truth GRNs derived from ChIP-seq datasets curated by BEELINE. The results are summarized in Tables S2 and S3, highlighting GRANGER's consistent superiority in GRN inference. Table S2 presents the AUPRC ratio values, where GRANGER achieved the highest scores across all datasets, including 1.57 for hESC and 1.91 for hHEP, outperforming other methods. Similarly, Table S3 showcases GRANGER's outstanding EPR values, such as 1.69 for hESC and 1.78 for hHEP. Taken together, these findings demonstrate that GRANGER maintains high predictive accuracy across diverse scRNA-seq datasets and varying cell numbers, highlighting its versatility and reliability in biological data analysis.

Robustness of GRANGER against dropout events

We conducted benchmarking using simulation datasets with different dropout levels to evaluate GRANGER's performance in handling dropout noise. For each of the four curated datasets, we compared GRANGER with eight methods on samples containing 50% and 70% dropout zeros. Figure 4 demonstrates that GRANGER's performance shows minimal degradation at both dropout rates compared to the original data. GRANGER consistently achieves the highest AUPRC scores for different dropout rates compared to the other eight methods. For example, GRANGER achieves AUPRC scores around 0.8 in the HSC dataset and 0.7 in the GSD dataset, whereas the other methods' AUPRC scores are all below 0.5 in the HSC dataset and below 0.3 in the GSD dataset. Although GRANGER performs similarly AUROC scores to other methods in several datasets, it achieves higher AUROC in the HSC and mCAD datasets across different dropout rates. These results show the strong resilience of GRANGER in predicting GRNs from sparse scRNA-seq data, reinforcing its suitability for handling real-world biological data with dropout challenges.

GRANGER detected potential GRNs in mouse brain cells

To access the effectiveness of GRANGER in predicting GRNs in real-world scenarios, we applied it to scRNA-seq datasets of the whole mouse brain [58]. As a case study, we focused on 1055 CA1-ProS-Glut_1 neurons. We specifically predicted GRNs for five TFs, *Sox10*, *Onecut2*, *Gbx1*, *E2f7*, and *Prox1*, which play critical regulatory roles in neurons [72, 73]. GRANGER successfully predicted regulatory interactions for each TF: 46 regulations for *E2f7*, 19 for *Sox10*, 20 for *Gbx1*, 24 for *Prox1*, and 20 for *Onecut2*. These regulated genes are involved in several functional categories, including metabolism, transcription, cell adhesion/structure, the cell cycle, and other biological processes (Fig. 5A, see Table S4 for detailed annotations).

We identified that the GRNs of *Sox10*, *Onecut2*, *Gbx1*, *E2f7*, and *Prox1* form a highly interconnected network comprising 95 genes, including 24 co-regulated genes (marked in pink in Fig. 5A). Several of these co-regulated genes, such as *Fos* [74], *Bdnf* [75, 76], *Ncam2* [77], and *Penk* [78], stand out for their significant regulatory roles in critical cellular functions. These co-regulated genes highlight the interplay of TFs in regulating diverse yet interconnected mechanisms in hippocampal neurons. To further investigate the biological mechanisms, we then performed functional enrichment analysis for these 95 genes and revealed multiple biological processes enriched in critical regulatory functions (Table S5). Notably, nervous system development (GO:0007399) and cell–cell signaling (GO:0007267) were the most significantly enriched processes. The process of negative regulation of neuroblast proliferation (GO:0007406) had an exceptionally high log-fold change of 57.67. Additionally, hormone-related processes such as hormone secretion (GO:0046879) and hormone transport (GO:0009914) were notably enriched.

To validate the predicted regulatory interactions in the five GRNs, we conducted comprehensive literature reviews and analyzed ChIP-seq data to assess binding activities. First, a substantial proportion of the predicted interactions were supported by literature evidence indicating correlations or interactions between the TFs and their targets: 16 out of 24 interactions in the *Prox1* GRN (Table S6), 14 out of 19 in the *Sox10* GRN (Table S7), 14 out of 20 in the *Onecut2* GRN (Table S8), 29 out of 46 in the *E2f7* GRN (Table S9), and 13 out of 20 in the *Gbx1* GRN (Table S10). Second, ChIP-seq data analysis provided additional confirmation of the predictions. ChIP-seq datasets were identified for *Prox1*, *Sox10*, and *Onecut2* in mice; however, no suitable datasets were currently available for *E2f7* and *Gbx1*. Notably, strong binding peaks were detected at critical genomic regions, including promoter regions, 3' UTRs, and introns, for all 24 target genes in the *Prox1* GRN, 19 target genes in the *Sox10* GRN, and 20 target genes in the *Onecut2* GRN. Moreover, these binding peaks frequently co-localized with active regulatory markers such as K4me2, K4me3, K27ac, and ATAC-seq signals. Specifically, co-occupancy with these markers was observed for 20 of 24 targets in the *Prox1* GRN, 13 of 19 in the *Sox10* GRN, and 11 of 20 in the *Onecut2* GRN. An example is shown in Fig. 5B, illustrating *Prox1*'s regulatory activity over the *Limd1* gene, where *Prox1* ChIP-seq data in mouse neurons exhibit

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

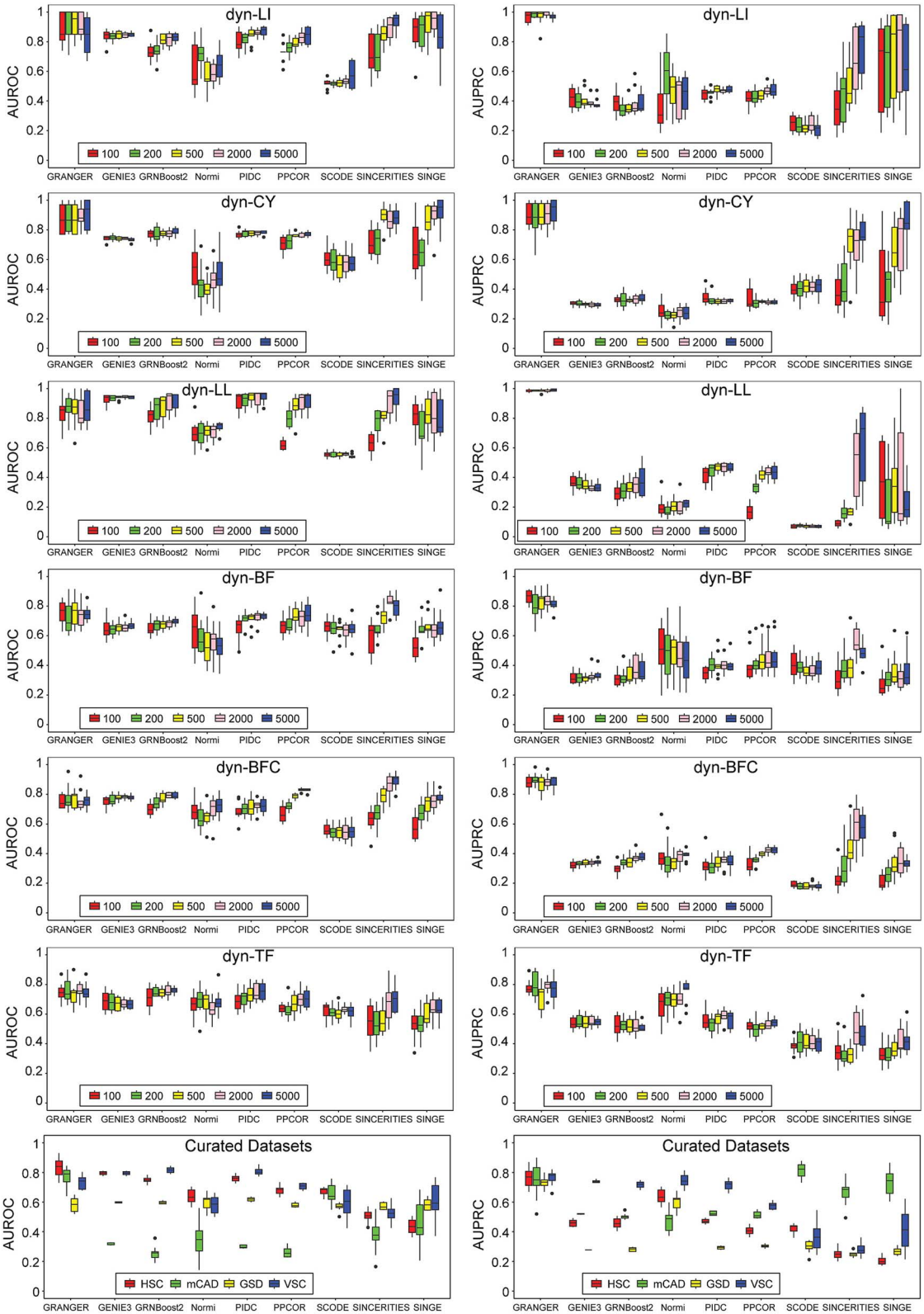

Figure 3. AUPRC and AUROC scores of GRANGER and other eight GRN inference tools on six synthetic datasets and four curated datasets.

strong binding peaks at the *Limd1* promoter region. Furthermore, these peaks overlap with active regulatory markers, including K4me3, K27ac, and ATAC-seq signals, highlighting *Prox1*'s role in directly modulating *Limd1* expression. These results demonstrate the reliability of GRANGER in accurately identifying novel regulatory interactions across diverse TFs with strong support by both literature evidence and ChIP-seq signals.

One of the key insights of GRANGER is that it can effectively detect gene regulatory relationships, even in cases where gene co-expression and correlation are not strong. For example, among

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

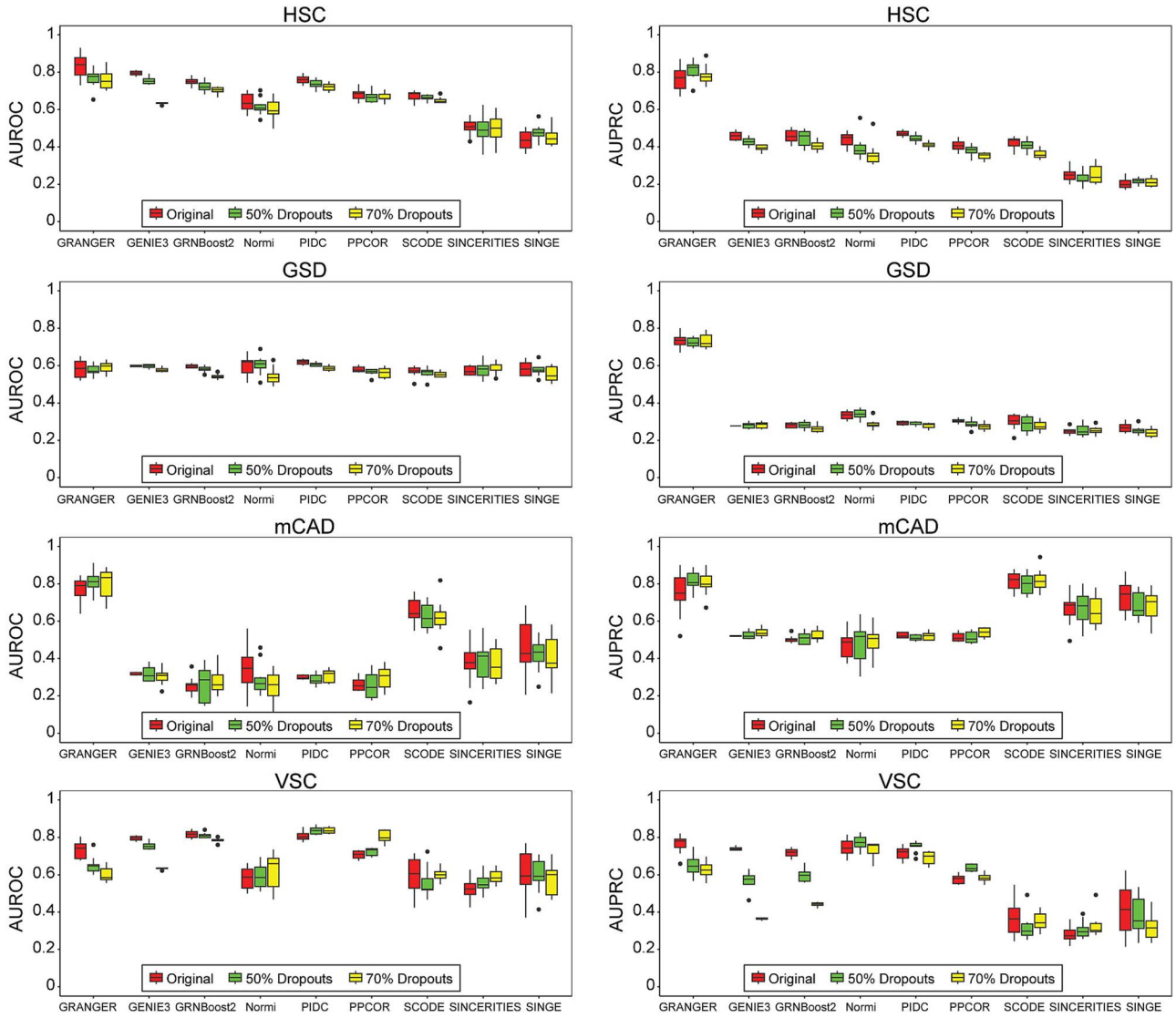

Figure 4. AUPRC and AUROC scores of GRANGER and other eight GRN inference tools on four curated datasets (HSC, GSD, mCAD and VSC) with different dropout rates.

the 46 genes in the *E2f7* GRN, there are four subgroups showing high expression levels and strong correlations within each group, while many other genes exhibit low correlations (Fig. 5C). Similar patterns were observed in the GRNs of *Sox10*, *Onecut2*, *Gbx1*, and *Prox1* (Fig. S5). The ability of GRANGER to infer regulatory interactions without always depending on high co-expression among genes allows it to identify meaningful biological relationships that other correlation-based methods might miss.

Discussion and conclusion

In this study, we introduced GRANGER, an unsupervised deep learning method to infer GRNs from scRNA-seq data. The combination of recurrent VAEs, GRANGER causality and NB-based loss function offers several advantages over traditional GRN inference methods, particularly in terms of capturing temporal dependencies, addressing data sparsity and improving prediction accuracy. One of the notable advantages of using recurrent VAEs is their ability to capture sequential patterns and temporal dynamics in time-series data. Traditional autoencoders compress data into a lower-dimensional space, but recurrent VAEs extend this by using memory cells (such as GRUs) to model time-dependent variations. By leveraging this capability, GRANGER captures gene expression changes over time, which is essential for accurately reconstructing the underlying GRNs in dynamic biological processes. Furthermore, by combining recurrent VAEs with GRANGER causality, GRANGER offers a causal interpretation of gene regulation that is often missing in many correlation-based approaches. Considering that scRNA-seq data are extremely sparse and noisy due to experimental limitations [51, 65–67], GRANGER incorporates an NB-based loss function to well leverage statistical characteristics of the data. This feature, combined with a sparsity-inducing penalty, enables GRANGER to achieve superior performance and robustness against dropout noise over eight competing methods.

We applied GRANGER to scRNA-seq data from the BRAIN Initiative project, specifically targeting the identification of GRNs for five TFs (*E2f7*, *Gbx1*, *Sox10*, *Prox1*, and *Onecut2*) in neurons. GRANGER recalled known regulations but it also identified novel insights that are functional important and could be experimentally validated in future studies. For example, additional ChIP-seq experiments on mouse cell lines or tissues can confirm TF binding signals and provide direct evidence of these TF regulating their

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

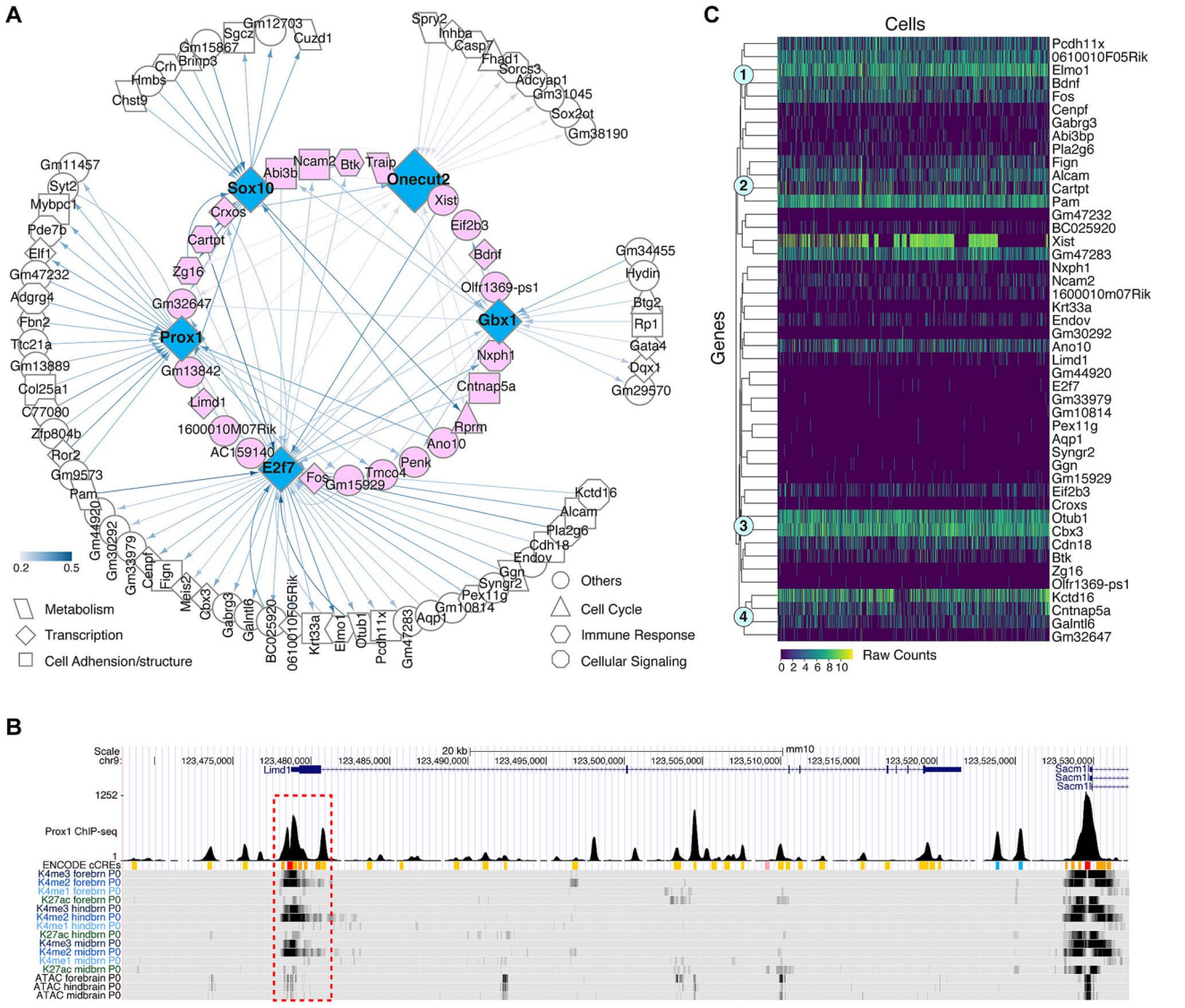

Figure 5. Predicted GRNs for five TFs in mouse CA1-ProS-Glut_1 neurons. (A) Merged GRNs of *Sox10*, *Onecut2*, *Gbx1*, *E2f7* and *Prox1*. Arrows represent regulatory relationships between the TFs and their target genes. The strength of the inferred interactions between the TFs and downstream genes is indicated by the color intensity of the connecting lines. Darker and thicker lines indicate stronger or more confident regulatory relationships. Pink-colored genes are co-regulated by at least two of the five TFs. (B) *Prox1* ChIP-seq reveals strong binding signals at the *Limd1* promoter region. The UCSC genome browser was used for visualization, highlighting *Prox1* binding sites in mouse neurons on the mm10 genome. The ENCODE cCREs track displays promoter regions in red bands, enhancer regions in orange bands, distal enhancer regions in yellow bands, DNase-H3K4me3 signatures in pink and CTCF binding sites in blue bands. The K4me1, K4me2, K4me3, K27ac and ATAC-seq tracks represent histone modifications and open chromatin regions in mouse neurons from the forebrain, hindbrain and midbrain. The red rectangle marks the binding peaks and histone modifications of the *Limd1* promoter. (C) Heatmap shows the expression levels of 46 genes in the *E2f7* GRN. The genes were clustered based on correlations. Four clusters of highly co-expressed genes are marked with light blue circles.

targeted genes under different conditions. Such experimental validations would validate GRANGER's predictions and uncover novel insights for the functional roles of TFs in complex biological systems.

Despite its advantages, GRANGER has certain limitations that need further exploration and improvement. First, the reliance on linear GRANGER causality may not fully capture non-linear dependencies between genes [79]. Integrating non-linear causality measures or hybrid models could enhance GRANGER's ability to detect more complex regulatory interactions. Furthermore, GRANGER can be expanded by incorporating additional omics data, such as epigenomics or proteomics, to build multi-modal models capable of providing a more comprehensive view of gene regulation [45]. We plan to enhance GRANGER's interpretability and accuracy by incorporating attention mechanisms [80], which can dynamically adjust the weighting of gene interactions based on the biological context. In summary, GRANGER provides an unsupervised deep learning framework capable of inferring robust and interpretable GRNs from noisy, high-dimensional scRNA-seq data, offering a powerful tool to discover novel regulatory interactions.

Key Points

- GRANGER is an unsupervised deep learning method that has superior performance in cell-specific GRN inference from scRNA-seq dataset.

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

- GRANGER uses recurrent VAEs and GRANGER causality to infer directional regulatory relationships among genes, offering a more comprehensive view of gene interactions compared to correlation-based methods.
- GRANGER integrates NB-based loss functions and sparsity-inducing penalties to address the sparsity and dropout noise in scRNA-seq data.

Supplementary Data

Supplementary data are available at *Briefings in Bioinformatics* online.

Author contributions

S.Z. and Y.C. initiated the concept and supervised the study. S.Z. and Y.C. designed the methodology. L.C. and R.G. implemented the software. L.C., M.D., Y.C., and S.Z. performed the data analysis. S.Z. and Y.C. drafted and reviewed the paper. All authors read and edited the final manuscript.

Conflict of interest: The authors declare that they have no competing interests.

Funding

This work was supported by the NSF CAREER Award DBI-2239350 for Y.C.; a key project of Natural Science Foundation of Tianjin City (19JCZDJC35100) and the National Science Foundation of China (61572358) to S.Z.

Data availability

Data used in this study are available in Gene Expression Omnibus (GEO) with the accession number GES138852 and BEELINE's benchmark datasets at https://zenodo.org/records/3701939 (Version 3, accessed on 1 June 2024). The scRNA-seq data of the whole mouse brain was selected and downloaded from the Allen Brain Cell Atlas portal (https://alleninstitute.github.io/abc_atlas_access). The code for GRANGER is available at https://github.com/shaoqiangzhang/GRANGER.

References

1. Emmert-Streib F, Dehmer M, Haibe-Kains B. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Front Cell Dev Biol* 2014;**2**:38.
2. van der Wijst MGP, de Vries DH, Brugge H, *et al.* An integrative approach for building personalized gene regulatory networks for precision medicine. *Genome Med* 2018;**10**:96. https://doi.org/10.1186/s13073-018-0608-4.
3. Hekselman I, Yeger-Lotem E. Mechanisms of tissue and cell-type specificity in heritable traits and diseases. *Nat Rev Genet* 2020;**21**:137–50.
4. Fiers M, Minnoye L, Aibar S, *et al.* Mapping gene regulatory networks from single-cell omics data. *Brief Funct Genomics* 2018;**17**:246–54. https://doi.org/10.1093/bfgp/elx046.
5. Kim H, Choi H, Lee D, *et al.* A review on gene regulatory network reconstruction algorithms based on single cell RNA sequencing. *Genes Genomics* 2024;**46**:1–11. https://doi.org/10.1007/s13258-023-01473-8.
6. Stock M, Popp N, Fiorentino J, *et al.* Topological benchmarking of algorithms to infer gene regulatory networks from single-cell RNA-seq data. *Bioinformatics* 2024;**40**:btae267. https://doi.org/10.1093/bioinformatics/btae267.
7. Nguyen H, Tran D, Tran B, *et al.* A comprehensive survey of regulatory network inference methods using single cell RNA sequencing data. *Brief Bioinform* 2021;**22**:bbaa190. https://doi.org/10.1093/bib/bbaa190.
8. Huynh-Thu VA, Irrthum A, Wehenkel L, *et al.* Inferring regulatory networks from expression data using tree-based methods. *PloS One* 2010;**5**:e12776. https://doi.org/10.1371/journal.pone.0012776.
9. Moerman T, Aibar Santos S, Bravo González-Blas C, *et al.* GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics* 2019;**35**:2159–61. https://doi.org/10.1093/bioinformatics/bty916.
10. Aibar S, González-Blas CB, Moerman T, *et al.* SCENIC: single-cell regulatory network inference and clustering. *Nat Methods* 2017;**14**:1083–6. https://doi.org/10.1038/nmeth.4463.
11. Liu H, Li P, Zhu M, *et al.* Nonlinear network reconstruction from gene expression data using marginal dependencies measured by DCOL. *PloS One* 2016;**11**:e0158247. https://doi.org/10.1371/journal.pone.0158247.
12. Chan TE, Stumpf MPH, Babtie AC. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell Syst* 2017;**5**:251–267.e3.
13. Kim S. ppcor: An R package for a fast calculation to semi-partial correlation coefficients. *Commun Stat Appl Methods* 2015;**22**:665–74. https://doi.org/10.5351/CSAM.2015.22.6.665.
14. Guo M, Wang H, Potter SS, *et al.* SINCERA: a pipeline for single-cell RNA-Seq profiling analysis. *PLoS Comput Biol* 2015;**11**:e1004575.
15. Specht AT, Li J. LEAP: constructing gene co-expression networks for single-cell RNA-sequencing data using pseudotime ordering. *Bioinformatics* 2017;**33**:764–6.
16. Sanchez-Castillo M, *et al.* A Bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data. *Bioinformatics* 2018;**34**:964–70.
17. Papili Gao N, Ud-Dean SMM, Gandrillon O, *et al.* SINCERITIES: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics* 2018;**34**:258–66. https://doi.org/10.1093/bioinformatics/btx575.
18. Cordero P, Stuart JM. Tracing co-regulatory network dynamics in noisy, single-cell transcriptome trajectories. *Pac Symp Biocomput* 2017;**22**:576–87.
19. Li L, Xia R, Chen W, *et al.* Single-cell causal network inferred by cross-mapping entropy. *Brief in bioinform* 2023; **24**:bbad281. http://dx.doi.org/10.1093/bib/bbad281.
20. Zeng Y, He Y, Zheng R, *et al.* Inferring single-cell gene regulatory network by non-redundant mutual information. *Brief Bioinform* 2023;**24**:bbad326. https://doi.org/10.1093/bib/bbad326.
21. Singh R, Wu AP, Mudide A, *et al.* Causal gene regulatory analysis with RNA velocity reveals an interplay between slow and fast transcription factors. *Cell Syst* 2024;**15**:462–474.e5. https://doi.org/10.1016/j.cels.2024.04.005.
22. Deshpande A, Chu LF, Stewart R, *et al.* Network inference with GRANGER causality ensembles on single-cell transcriptomics. *Cell Rep* 2022;**38**:110333. https://doi.org/10.1016/j.celrep.2022.110333.
23. Dautle M, Zhang S, Chen Y. scTIGER: a deep-learning method for inferring gene regulatory networks from case versus control scRNA-seq datasets. *Int J Mol Sci* 2023;**24**:13339. https://doi.org/10.3390/ijms241713339.

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

24. He W, Tang J, Zou Q, *et al.* MMFGRN: a multi-source multi-model fusion method for gene regulatory network reconstruction. *Brief Bioinform* 2021;**22**:bbab166. https://doi.org/10.1093/bib/bbab166.
25. Huynh-Thu VA, Geurts P. dynGENIE3: dynamical GENIE3 for the inference of gene networks from time series expression data. *Sci Rep* 2018;**8**:3384.
26. Ocone A, Haghverdi L, Mueller NS, *et al.* Reconstructing gene regulatory dynamics from high-dimensional single-cell snapshot data. *Bioinformatics* 2015;**31**:i89–96. https://doi.org/10.1093/bioinformatics/btv257.
27. Matsumoto H, Kiryu H, Furusawa C, *et al.* SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation. *Bioinformatics* 2017;**33**:2314–21. https://doi.org/10.1093/bioinformatics/btx194.
28. Matsumoto H, Kiryu H. SCOUP: a probabilistic model based on the Ornstein-Uhlenbeck process to analyze single-cell expression data during differentiation. *BMC Bioinformatics* 2016;**17**:232. https://doi.org/10.1186/s12859-016-1109-3.
29. Aubin-Frankowski PC, Vert JP. Gene regulation inference from single-cell RNA-seq data with linear differential equations and velocity inference. *Bioinformatics* 2020;**36**:4774–80.
30. Yuan Y, Bar-Joseph Z. Deep learning for inferring gene relationships from single-cell expression data. *Proc Natl Acad Sci USA* 2019;**116**:27151–8.
31. Chen J, Cheong CW, Lan L, *et al.* DeepDRIM: a deep neural network to reconstruct cell-type-specific gene regulatory network using single-cell RNA-seq data. *Brief Bioinform* 2021;**22**:bbab325. https://doi.org/10.1093/bib/bbab325.
32. Xu Y, Chen J, Lyu A, *et al.* dynDeepDRIM: a dynamic deep learning model to infer direct regulatory interactions using time-course single-cell gene expression data. *Brief Bioinform* 2022;**23**:bbac424. https://doi.org/10.1093/bib/bbac424.
33. Chen G, Liu ZP. Graph attention network for link prediction of gene regulations from single-cell RNA-sequencing data. *Bioinformatics* 2022;**38**:4522–9.
34. Mao G, Pang Z, Zuo K, *et al.* Predicting gene regulatory links from single-cell RNA-seq data using graph neural networks. *Brief Bioinform* 2023;**24**:bbad414. https://doi.org/10.1093/bib/bbad414.
35. Yuan Y, Bar-Joseph Z. Deep learning of gene relationships from single cell time-course expression data. *Brief Bioinform* 2021;**22**:bbab142. https://doi.org/10.1093/bib/bbab142.
36. Turki T, Taguchi YH. SCGRNs: novel supervised inference of single-cell gene regulatory networks of complex diseases. *Comput Biol Med* 2020;**118**:103656.
37. Xu J, Zhang A, Liu F, *et al.* STGRNS: an interpretable transformer-based method for inferring gene regulatory networks from single-cell transcriptomic data. *Bioinformatics* 2023;**39**:btad165. https://doi.org/10.1093/bioinformatics/btad165.
38. Zhou X, Pan J, Chen L, *et al.* DeepIMAGER: deeply Analyzing gene regulatory networks from scRNA-seq data. *Biomolecules* 2024;**14**:766. https://doi.org/10.3390/biom14070766.
39. Keyl P, Bischoff P, Dernbach G, *et al.* Single-cell gene regulatory network prediction by explainable AI. *Nucleic Acids Res* 2023;**51**:e20. https://doi.org/10.1093/nar/gkac1212.
40. Li Z, Nagai JS, Kuppe C, *et al.* scMEGA: single-cell multi-omic enhancer-based gene regulatory network inference. *Bioinform Adv* 2023;**3**:vbad003. https://doi.org/10.1093/bioadv/vbad003.
41. Wang L, Trasanidis N, Wu T, *et al.* Dictys: dynamic gene regulatory network dissects developmental continuum with single-cell multiomics. *Nat Methods* 2023;**20**:1368–78. https://doi.org/10.1038/s41592-023-01971-3.
42. Yuan Q, Duren Z. Inferring gene regulatory networks from single-cell multiome data using atlas-scale external data. *Nat Biotechnol* 2025;**43**:247–57. https://doi.org/10.1038/s41587-024-02182-7.
43. Bravo González-Blas C, de Winter S, Hulselmans G, *et al.* SCENIC+: single-cell multiomic inference of enhancers and gene regulatory networks. *Nat Methods* 2023;**20**:1355–67. https://doi.org/10.1038/s41592-023-01938-4.
44. Kim D, Tran A, Kim HJ, *et al.* Gene regulatory network reconstruction: harnessing the power of single-cell multiomic data. *NPJ Syst Biol Appl* 2023;**9**:51. https://doi.org/10.1038/s41540-023-00312-6.
45. Badia IMP, *et al.* Gene regulatory network inference in the era of single-cell multi-omics. *Nat Rev Genet* 2023;**24**:739–54. https://doi.org/10.1038/s41576-023-00618-5.
46. Pratapa A, Jalihal AP, Law JN, *et al.* Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nat Methods* 2020;**17**:147–54. https://doi.org/10.1038/s41592-019-0690-6.
47. Kang Y, Thieffry D, Cantini L. Evaluating the reproducibility of single-cell gene regulatory network inference algorithms. *Front Genet* 2021;**12**:617282. https://doi.org/10.3389/fgene.2021.617282.
48. Langfelder P, Horvath S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* 2008;**9**:559. https://doi.org/10.1186/1471-2105-9-559.
49. Granger C. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica* 1969;**37**:424–38. https://doi.org/10.2307/1912791.
50. Li H, Yu S, Principe J. Causal recurrent Variational autoencoder for medical time series generation. *Proceedings of the AAAI Conference on Artificial Intelligence* 2023;**37**:8562–70.
51. Lei T, Chen R, Zhang S, *et al.* Self-supervised deep clustering of single-cell RNA-seq data to hierarchically detect rare cell populations. *Brief Bioinform* 2023;**24**:bbad335. https://doi.org/10.1093/bib/bbad335.
52. Cho K. *et al.* On the properties of neural machine translation: Encoder-decoder approaches. In: *Proceedings of Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, 2014:103–111.
53. Wolf FA, Angerer P, Theis FJ. SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biol* 2018;**19**:15.
54. Wolf FA, Hamey FK, Plass M, *et al.* PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol* 2019;**20**:59. https://doi.org/10.1186/s13059-019-1663-x.
55. Tian Y, Zhang Y, Zhang H. Recent advances in stochastic gradient descent in deep learning. *Mathematics* 2023;**11**:682.
56. Daubechies I, Defrise M, De Mol C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics* 2004;**57**:1413–57.
57. Street K, Risso D, Fletcher RB, *et al.* Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* 2018;**19**:477. https://doi.org/10.1186/s12864-018-4772-0.
58. Yao Z, van Velthoven CTJ, Kunst M, *et al.* A high-resolution transcriptomic and spatial atlas of cell types in the whole mouse brain. *Nature* 2023;**624**:317–32. https://doi.org/10.1038/s41586-023-06812-z.
59. Mi H, Muruganujan A, Huang X, *et al.* Protocol update for large-scale genome and gene function analysis with the PANTHER classification system (v.14.0). *Nat Protoc* 2019;**14**:703–21. https://doi.org/10.1038/s41596-019-0128-8.
60. Mi H, Thomas P. PANTHER pathway: an ontology-based pathway database coupled with data analysis tools. *Methods Mol Biol* 2009;**563**:123–40.

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

61. Thomas PD, Ebert D, Muruganujan A, *et al.* PANTHER: making genome-scale phylogenetics accessible to all. *Protein Sci* 2022;**31**: 8–22. https://doi.org/10.1002/pro.4218.
62. Chen S, Mar JA-O. Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data. *BMC Bioinformatics* **19**:232.
63. Kingma DP, Ba J. Adam: a method for stochastic optimization. *CoRR* 2014; abs/1412.6980.
64. Davis J, Goadrich M. *The relationship between Precision-Recall and ROC curves*, in *Proceedings of the 23rd international conference on Machine learning*. 2006, Association for Computing Machinery: Pittsburgh, Pennsylvania, USA. p. 233–40.
65. Qin L, Zhang G, Zhang S, *et al.* Deep batch integration and denoise of single-cell RNA-seq data. *Adv Sci (Weinh)* 2024; **11**:e2308934. https://doi.org/10.1002/advs.202308934.
66. Andrews TS, Hemberg M. M3Drop: dropout-based feature selection for scRNASeq. *Bioinformatics* 2019;**35**:2865–7.
67. Tang W, Bertaux F, Thomas P, *et al.* bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics* 2020;**36**:1174–81. https://doi.org/10.1093/bioinformatics/btz726.
68. Svensson V. Droplet scRNA-seq is not zero-inflated. *Nat Biotechnol* 2020;**38**:147–50.
69. Chen W, Li Y, Easton J, *et al.* UMI-count modeling and differential expression analysis for single-cell RNA sequencing. *Genome Biol* 2018;**19**:70. https://doi.org/10.1186/s13059-018-1438-9.
70. Cannoodt R, *et al. SCORPIUS Improves Trajectory Inference and Identifies Novel Modules in Dendritic Cell Development*. bioRxiv, 2016;079509. https://doi.org/10.1101/079509.
71. Saelens W, Cannoodt R, Todorov H, *et al.* A comparison of single-cell trajectory inference methods. *Nat Biotechnol* 2019;**37**:547–54. https://doi.org/10.1038/s41587-019-0071-9.
72. Swiss VA, Casaccia P. Cell-context specific role of the E2F/Rb pathway in development and disease. *Glia* 2010;**58**: 377–90.
73. Britsch S, Goerich DE, Riethmacher D, *et al.* The transcription factor Sox10 is a key regulator of peripheral glial development. *Genes Dev* 2001;**15**:66–78. https://doi.org/10.1101/gad.186601.
74. Cruz-Mendoza F, Jauregui-Huerta F, Aguilar-Delgadillo A, *et al.* Immediate early gene c-fos in the brain: focus on glial cells. *Brain Sci* 2022;**12**:687. https://doi.org/10.3390/brainsci12060687.
75. Wang CS, Kavalali ET, Monteggia LM. BDNF signaling in context: from synaptic regulation to psychiatric disorders. *Cell* 2022;**185**: 62–76.
76. Pisani A, Paciello F, del Vecchio V, *et al.* The role of BDNF as a biomarker in cognitive and sensory neurodegeneration. *J Pers Med* 2023;**13**:652. https://doi.org/10.3390/jpm13040652.
77. Ortega-Gasco A, Parcerisas A, Hino K. Regulation of young-adult neurogenesis and neuronal differentiation by neural cell adhesion molecule 2 (NCAM2). *Cereb Cortex* 2023;**33**:10931–48. https://doi.org/10.1093/cercor/bhad340.
78. Le Merrer J, Becker JA, Befort K, *et al.* Reward processing by the opioid system in the brain. *Physiol Rev* 2009;**89**:1379–412. https://doi.org/10.1152/physrev.00005.2009.
79. Shojaie A, Fox EB. GRANGER causality: a review and recent advances. *Annu Rev Stat Appl* 2022;**9**:289–319.
80. Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning. *Neurocomputing* 2021;**452**:48–62.

Downloaded from https://academic.oup.com/bib/article/26/2/bbaf089/8068119 by guest on 25 January 2026

© The Author(s) 2025. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (https://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited.
For commercial re-use, please contact journals.permissions@oup.com
Briefings in Bioinformatics, 2025, **26(2)**, bbaf089
https://doi.org/10.1093/bib/bbaf089
Problem Solving Protocol