Bioinformatics, 39(5), 2023, btad256
https://doi.org/10.1093/bioinformatics/btad256
Advance Access Publication Date: 20 April 2023
Original Paper

Gene expression

LogBTF: gene regulatory network inference using Boolean threshold network model from single-cell gene expression data

Lingyu Li [1,2], Liangjie Sun[2], Guangyi Chen[1], Chi-Wing Wong[2], Wai-Ki Ching[2,*], Zhi-Ping Liu [1,*]

[1]Department of Biomedical Engineering, School of Control Science and Engineering, Shandong University, Jinan 250061, China
[2]Advanced Modeling and Applied Computing Laboratory, Department of Mathematics, The University of Hong Kong, Hong Kong, China

*Corresponding authors. Zhi-Ping Liu, Department of Biomedical Engineering, School of Control Science and Engineering, Shandong University, Jinan 250061, China. E-mail: zpliu@sdu.edu.cn; Wai-Ki Ching, Advanced Modeling and Applied Computing Laboratory, Department of Mathematics, The University of Hong Kong, Hong Kong, China. E-mail: wching@hku.hk
Associate Editor: Alfonso Valencia

Received 28 November 2022; revised 25 February 2023; accepted 13 April 2023

Abstract

Motivation: From a systematic perspective, it is crucial to infer and analyze gene regulatory network (GRN) from high-throughput single-cell RNA sequencing data. However, most existing GRN inference methods mainly focus on the network topology, only few of them consider how to explicitly describe the updated logic rules of regulation in GRNs to obtain their dynamics. Moreover, some inference methods also fail to deal with the over-fitting problem caused by the noise in time series data.

Results: In this article, we propose a novel embedded Boolean threshold network method called LogBTF, which effectively infers GRN by integrating regularized logistic regression and Boolean threshold function. First, the continuous gene expression values are converted into Boolean values and the elastic net regression model is adopted to fit the binarized time series data. Then, the estimated regression coefficients are applied to represent the unknown Boolean threshold function of the candidate Boolean threshold network as the dynamical equations. To overcome the multi-collinearity and over-fitting problems, a new and effective approach is designed to optimize the network topology by adding a perturbation design matrix to the input data and thereafter setting sufficiently small elements of the output coefficient vector to zeros. In addition, the cross-validation procedure is implemented into the Boolean threshold network model framework to strengthen the inference capability. Finally, extensive experiments on one simulated Boolean value dataset, dozens of simulation datasets, and three real single-cell RNA sequencing datasets demonstrate that the LogBTF method can infer GRNs from time series data more accurately than some other alternative methods for GRN inference.

Availability and implementation: The source data and code are available at https://github.com/zpliulab/LogBTF.

1 Introduction

With the tremendous progress of advanced technology and the improvement of sensitivity of cell analysis, single-cell RNA sequencing (scRNA-seq) data have brought unprecedented challenges and opportunities for deciphering the regulatory relationship among genes (Chen and Liu 2022; Luo et al. 2022). One challenge of scRNA-seq data analytics is its "dropout" (Qiu 2020). Because of the dropouts, scRNA-seq data are extremely sparse (excessive zero counts) so as to only capture a little information about the transcriptome of each cell (Qiu 2020). Despite this, the analysis for scRNA-seq data promotes the development of gene regulatory network (GRN) inference methods to help us explore the mechanisms underlying various biological processes (Papili Gao et al. 2018; Algabri et al. 2022) and expectedly develop efficient therapies to treat and cure diseases (Aalto et al. 2020). Recently, GRN inference from time series data has gained more and more attention (Aalto et al. 2020). In particular, the studies of scRNA-seq provide an ordering of cells involved in a dynamical process (such as differentiation) through inferred pseudo-time associated with each cell (Zhang et al. 2021). Furthermore, the pseudo-time can be regarded as temporal information for single-cell gene expression profiles, which offers a unique opportunity to infer GRNs (Aubin-Frankowski and Vert 2020).

From the systems biology point of view, inferring GRN plays an extremely crucial role in revealing underlying regulatory mechanisms to uncover potential genes (Akutsu et al. 2000; Zhang et al. 2012; Liu et al. 2021). A vast number of network inference/reconstruction methods have been widely developed to infer GRN using transcriptomic profiles. To the best of our knowledge, the existing network inference/reconstruction methods can be categorized into the following groups according to their principles (Liu 2015): regression-based method [multiple regression model (Zhang et al. 2010), SINCERITIES (Papili Gao et al. 2018), and GNIPLR (Zhang et al. 2021)], tree-based method [GENIE3 (Huynh-Thu et al. 2010)], stability selection method [TIGRESS (Haury et al. 2012)], correlation-based method [ARACNE (Margolin et al. 2006) and CLR (Faith et al. 2007)], knowledge-based method [RegNetwork (Liu et al. 2015)], ordinary differential equation method [linear ODE (Wu et al. 2014), SCODE (Matsumoto et al. 2017), and GRISLI (Aubin-Frankowski and Vert 2020)], Bayesian-based method [SSMs (Beal et al. 2005), Vireo (Huang et al. 2019), and NAE (Wang et al. 2022)], Boolean network (BN) model method [ATEN (Shi et al. 2020) and GAPORE (Liu et al. 2021)], and deep learning model [DeepDRIM (Chen et al. 2021), dynDeepDRIM (Xu et al. 2022), and DeepSEM (Shu et al. 2021)].

Generally speaking, there are four levels of inferring GRNs (Liu 2015). (i) Is there a regulatory relationship? (ii) Who is the regulator, and who is the target? (iii) Whether the regulatory relationship is activating or inhibiting? (iv) How strong is this regulatory relationship? Some GRN reconstruction methods (e.g. gene co-expression network) can infer a network with undirected edges, which only reflect the associations among genes and build the fundamental architecture of these regulations, i.e. at the level i. Some methods can identify the regulatory direction (level ii) but cannot obtain extra regulatory information, such as regulatory mechanism (level iii), and relative regulatory strength (level iv).

Several data-driven methods have been applied to infer GRNs (Luo et al. 2022). For example, the accurate cellular network reconstruction algorithm (ARACNE) method assumes that correlation analysis can be used to discover regulatory interactions between genes by considering a time gap between gene expression values (Margolin et al. 2006). However, such statistical dependence between the studied genes does not necessarily capture the regulatory relationships (Aubin-Frankowski and Vert 2020). The context likelihood of relatedness (CLR) method reduces the connections from a complete graph by discarding false connections through comparisons of pairwise mutual information scores with a background correction of mutual information scores (Faith et al. 2007). The CLR method effectively saves computational costs, but it only evaluates pairwise interactions (Barman and Kwon 2017). Later, by performing feature selection problems, several ensemble methods are proposed to provide a set of solutions rather than a single solution. For example, the gene network inference with the ensemble of trees (GENIE3) method selects a set of features for extra trees or random forests learning models (Huynh-Thu et al. 2010), and it is regarded as an effective algorithm for reconstructing GRN on both scRNA-seq and bulk RNA-seq gene expression data (Chen et al. 2021). The trustful inference of gene regulation using stability selection (TIGRESS) method conducts feature selection using the least angle regression integrated with a stability selection (Haury et al. 2012). Although these two methods show satisfactory performances, they are only intended to infer the regulatory structure without considering regulatory rules or functions. Also, the inferred GRN may include many false positives and indirect regulations since they are only based on relevance (Aibar et al. 2017). What's more, single-cell regulatory network inference and clustering (SCENIC) is a typical computational method to infer GRNs and cell types from scRNA-seq data (Aibar et al. 2017). However, it still establishes all possible regulatory relationships using GENIE3 in its first step. Therefore, ones have to develop a novel approach to infer the regulatory rules for the prediction of network dynamics.

The BN was first introduced by Kauffman (1969) to qualitatively describe gene regulatory interactions to account for a variety of complex biological processes. Over the past few decades, BNs have been attracting the great attention of many researchers (Akutsu et al. 1999; Sun et al. 2021; Mori and Akutsu 2022; Sun and Ching 2023). BNs are not only a fundamental model for genetic systems that identify network structures from a systematic perspective (Liu 2015), but also a powerful framework for studying and modeling the dynamics of GRNs (Shi et al. 2020). In the face of tens of thousands of such high-dimensional gene expression time series data, traditional ordinary differential equation models face very intractable difficulties in solving and inferring the network topology, especially sufficient storage space and expensive computing cost. In contrast, BNs faithfully reproduce the states obtained with more realistic continuum reaction kinetics models by simplifying regulatory interactions between genes using discrete variables (Font-Clos et al. 2021). Furthermore, although the major obstacle in inferring GRNs is the dropouts of data (Aalto et al. 2020), Qiu (2020) has illustrated that the dropout pattern in scRNA-seq data is an extremely useful signal by binarizing the count matrix, i.e. setting all non-zero observations into 1 and all dropouts are still 0, and pointed out that recognizing the utility of dropouts suggests an alternative direction for developing computational algorithms for scRNA-seq data.

In this article, we demonstrate an application of the Boolean threshold network model conceived to integrate time series single-cell data into a **Log**istic regression estimation-based **B**oolean **T**hreshold **F**unction (called LogBTF method) to infer GRN by synchronous evolution. First, LogBTF embeds the coefficients estimated by regularized logistic regression into the Boolean threshold network model, which perfectly controls the in-degree of nodes and addresses the problem of over-fitting. Second, LogBTF employs the knowledge of the perturbation design to optimize the inferred network topology, which successfully handles the multi-collinearity problem caused by binarized gene expression data. Third, LogBTF is a kind of interpretable network inference method that could comprehensively output more detailed information about regulatory relationships, such as regulator or target, activation or inhibition, and relative regulatory strength. Moreover, numerous experiments conducted on artificial Boolean datasets, simulated single-cell datasets, and real scRNA-seq datasets demonstrate the effectiveness and efficiency of the LogBTF method. Lastly, the comparison study with eight existing GRN inference methods shows that the LogBTF method unearths potential regulatory relationships and obtains better inference performances simultaneously.

2 Materials and methods

2.1 Boolean threshold network model

2.1.1 Boolean threshold network

A BN model works on a directed graph network. It consists of a set of nodes representing the elements of a system, and the state of each node is quantified as 0 (false/not expressed) or 1 (true/expressed). At each discrete time, the state of each node is updated by the state of its neighbor (the node pointing to it) at the previous moment through a rule called Boolean function. Therefore, the edges in the BN represent the regulatory relationships between elements. Generally speaking, the Boolean function is expressed as a statement that acts on the inputs through a logical function using logical operators NOT, AND, OR, etc.; clearly, this statement also returns a False/True state. Hence, suppose N is the total number of genes, the updating scheme of $x_i,\ i \in [1, N]$ can be described as follows:

$$x_i(t+1) = f_i(x_{i_1}(t), x_{i_2}(t), \ldots, x_{i_{k_i}}(t)), \tag{1}$$

where f_i is a Boolean function, $x_i(t+1)$ is the state of the target node x_i at the time point $t+1$, and $x_{i_1}(t), x_{i_2}(t), \ldots, x_{i_{k_i}}(t)$ are the states of the regulatory nodes $x_{i_1}, x_{i_2}, \ldots, x_{i_{k_i}}$ at the time point t. Here, $i_1, i_2, \ldots, i_{k_i} \in [1, N]$ and k_i is the in-degree of the node x_i, denoting the number of regulatory nodes of the target node x_i.

BNs with threshold functions are called Boolean threshold networks, which is a special subset of BNs (Melkman et al. 2018; Cheng et al. 2021). Here the threshold function is a special kind of

Boolean function, which can be calculated by a linear threshold unit. It is defined as follows (Anthony 2001):

DEFINITION 2.1. *Let N Boolean inputs be $x_1, x_2, \ldots, x_N$. A threshold function f on $x_1, x_2, \ldots, x_N$ has the form:*

$$f(x_1, x_2, \ldots, x_N) = \begin{cases} 1, & w_1 l_1 + w_2 l_2 + \cdots + w_N l_N \geq \theta, \\ 0, & \text{otherwise}, \end{cases} \quad (2)$$

where l_i is either x_i or $\overline{x_i}$ for $i \in [1, N]$, $(w_1, w_2, \ldots, w_N) \in \mathbb{R}^N$ is the weight-vector, and $\theta \in \mathbb{R}$ is the threshold.

Compared with BNs, Boolean threshold networks can easily be implemented and are very suitable for representing regulatory networks (Bornholdt 2008; Zañudo et al. 2011). Here, Boolean threshold network model is employed to infer the complex dynamics of GRN. The updating scheme of x_i, $i \in [1, N]$ can be rewritten as follows:

$$\begin{aligned} x_i(t+1) &= f_i(x_{i_1}(t), x_{i_2}(t), \ldots, x_{i_{k_i}}(t)) \\ &= \begin{cases} 1, & w_{i_1} l_{i_1} + w_{i_2} l_{i_2} + \cdots + w_{i_{k_i}} l_{i_{k_i}} \geq \theta_i, \\ 0, & \text{otherwise}. \end{cases} \end{aligned} \quad (3)$$

For each l_j, $j = i_1, i_2, \ldots, i_{k_i}$, if $l_j = x_j$, the regulatory node x_j promotes the target node x_i; if $l_j = \overline{x_j}$, the regulatory node x_j inhibits the target node x_i.

2.1.2 Logistic regression model

The logistic regression model is applied to estimate parameters of the above Boolean threshold network model using time series gene expression data. Let T be the total number of time points. For each node x_i, $i \in [1, N]$, there are $T-1$ observations $(\boldsymbol{X}^i(j), y^i(j))$, $i \in [1, N], j \in [1, T-1]$, which are independent and identical distributed. Let $\mathcal{D}_i = \{(\boldsymbol{X}^i(1), y^i(1)), (\boldsymbol{X}^i(2), y^i(2)), \ldots, (\boldsymbol{X}^i(T-1), y^i(T-1))\}$ where $\boldsymbol{X}^i(j) = (x_1(j), x_2(j), \ldots, x_N(j), 1)^\top \in \mathbb{R}^{N+1}$, $x_1(j), x_2(j), \ldots, x_N(j)$ represent the states of nodes (genes) in the j-th observation (or at the time point j), $j \in [1, T-1]$, and $y^i(j)$ is the state of the node x_i at the time point $j+1$, and the value is either 0 or 1, i.e. $y^i(j) = x_i(j+1)$.

The logistic regression is considered as follows:

$$\begin{aligned} \overset{ji}{\pi} &= \Pr\left(y^i(j) | \boldsymbol{X}^i(j); \boldsymbol{\theta}_i\right) \\ &= f\left(\boldsymbol{\theta}_i^\top \boldsymbol{X}^i(j)\right) \\ &= \frac{\exp\left(\boldsymbol{\theta}_i^\top \boldsymbol{X}^i(j)\right)}{1 + \exp\left(\boldsymbol{\theta}_i^\top \boldsymbol{X}^i(j)\right)}, \quad i \in [1, N], j \in [1, T-1], \end{aligned} \quad (4)$$

where $\boldsymbol{\theta}_i = (\theta_1^i, \theta_2^i, \ldots, \theta_N^i, \theta_0^i)^\top$, θ_k^i ($k \in [1, N]$) are the unknown coefficients to be estimated, and θ_0^i is the intercept to be estimated, $f(\cdot)$ is the logistic function used to predict y for any input of class labels (0 or 1) (James et al. 2013). By applying the logit transformation to Equation (4), we have

$$\begin{aligned} \text{logit}(\pi_j^i) &= \log\left(\frac{\pi_j^i}{1 - \pi_j^i}\right) \\ &= \theta_1^i x_1(j) + \theta_2^i x_2(j) + \cdots + \theta_N^i x_N(j) + \theta_0^i. \end{aligned} \quad (5)$$

It can be seen that the logistic regression model (4) has a logit that is linear in $\boldsymbol{X}^i(j)$.

2.1.3 Estimation of the regression coefficients

The coefficient vector $\boldsymbol{\theta}_i$ in Equation (4) is unknown. Here, we use the maximum likelihood approach and a regularized technique to fit the model (4) containing N variables (James et al. 2013). Since $y^i(j)$ follows the Bernoulli distribution, its probability function is given by $\Pr(y^i(j)) = (\pi_j^i)^{y^i(j)}(1 - \pi_j^i)^{1-y^i(j)}$ for $i \in [1, N]$, then we have the likelihood function as follows:

$$\mathcal{L}(\pi_j^i) = \prod_{j=1}^{N} \Pr(y^i(j)) = \prod_{j=1}^{N} (\pi_j^i)^{y^i(j)}(1 - \pi_j^i)^{1-y^i(j)}. \quad (6)$$

The corresponding log-likelihood function is a function of the regression coefficient vector $\boldsymbol{\theta}_i$ given by

$$\mathcal{L}(\boldsymbol{\theta}_i | \mathcal{D}_i) = \sum_{j=1}^{N} \{y^i(j) \cdot \log(\pi_j^i) + [1 - y^i(j)] \cdot \log(1 - \pi_j^i)\}. \quad (7)$$

Clearly, by minimizing the negative of Equation (7), we can estimate the regression coefficient vector $\boldsymbol{\theta}_i$. To avoid over-fitting, we add the elastic net penalty term (Zou and Hastie 2005) to Equation (7), and solve the regression coefficient vector $\boldsymbol{\theta}_i$ according to the following regularized logistic regression model (Li and Liu 2020, 2022):

$$\boldsymbol{\theta}_i = \arg\min\{-\mathcal{L}(\boldsymbol{\theta}_i | \mathcal{D}_i) + \lambda[\alpha \|\boldsymbol{\theta}_i\|_1 + (1 - \alpha)\|\boldsymbol{\theta}_i\|_2^2]\}. \quad (8)$$

Here $\|\boldsymbol{\theta}_i\|_1 = \sum_{k=1}^{N} |\theta_k^i|$ and $\|\boldsymbol{\theta}_i\|_2^2 = \sum_{k=1}^{N} {\theta_k^i}^2$ represent the L_1-norm and the square of L_2-norm, respectively. Here $\lambda > 0$ is a tuning parameter used to balance the negative log-likelihood term and the elastic net penalty term, $\alpha \in [0, 1]$ is used to shrink the estimated coefficient $\boldsymbol{\theta}_i$ to control the sparsity of inferred network, i.e. the in-degree of the node x_i.

2.1.4 Aggregation of Boolean threshold function with regression coefficients

In this work, we adopt a synchronous update mode, i.e. all nodes evolve simultaneously at consecutive time points. Here, $\boldsymbol{\theta}_i$ is obtained from Equation (8), the corresponding update scheme of x_i in the form of Boolean threshold function can also be obtained. The details are as follows:

$$\text{Update scheme} := \begin{cases} w_k = \theta_k^i \text{ and } l_k = x_k, & \text{if } \theta_k^i > 0, \\ w_k = 0, & \text{if } \theta_k^i = 0, \\ w_k = -\theta_k^i \text{ and } l_k = \overline{x_k}, & \text{if } \theta_k^i < 0. \end{cases} \quad (9)$$

Then, the Boolean threshold function of x_i is given as follows:

$$\begin{aligned} x_i(t+1) &= f_i(x_1(t), x_2(t), \ldots, x_N(t)) \\ &= \begin{cases} 1, & w_1 l_1 + w_2 l_2 + \cdots + w_N l_N \geq \theta_i, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \quad (10)$$

where $\theta_i = -\theta_0^i - \sum_{k \in \{k | k \in [1,N], \theta_k^i < 0\}} \theta_k^i$. Based on that, we have the following result.

THEOREM 2.1. *The inequality $w_1 l_1 + w_2 l_2 + \cdots + w_N l_N \geq \theta_i$ in Equation (10) is equivalent to $\theta_1^i x_1(j) + \theta_2^i x_2(j) + \cdots + \theta_N^i x_N(j) + \theta_0^i \geq 0$ in Equation (5) under the given update scheme shown in Equation (9).*

The proof of Theorem 2.1 can be available in Supplementary Equations (S1)–(S7). Therefore, we can use logistic regression model to estimate the parameters of Boolean threshold network model from the given dataset $\mathcal{D}$. In this article, we call this aggregation strategy the LogBTF method and its novelty lies in the use of logistic regression and Boolean threshold function to construct a Boolean threshold network model. The framework of the LogBTF method for inferring GRN from single-cell gene expression data is shown in Fig. 1. Especially, the consistency between Boolean trajectory generated by the inferred network and the binarized time series gene expression data is characterized by the dynamical accuracy (DyAcc) metric, while the consistency between inferred GRN and the ground-truth GRN is characterized by the structural accuracy (StAcc) metric.

2.2 Datasets

First, an artificial Boolean value dataset with a ground-truth regulatory network is generated to evaluate the validity and accuracy of

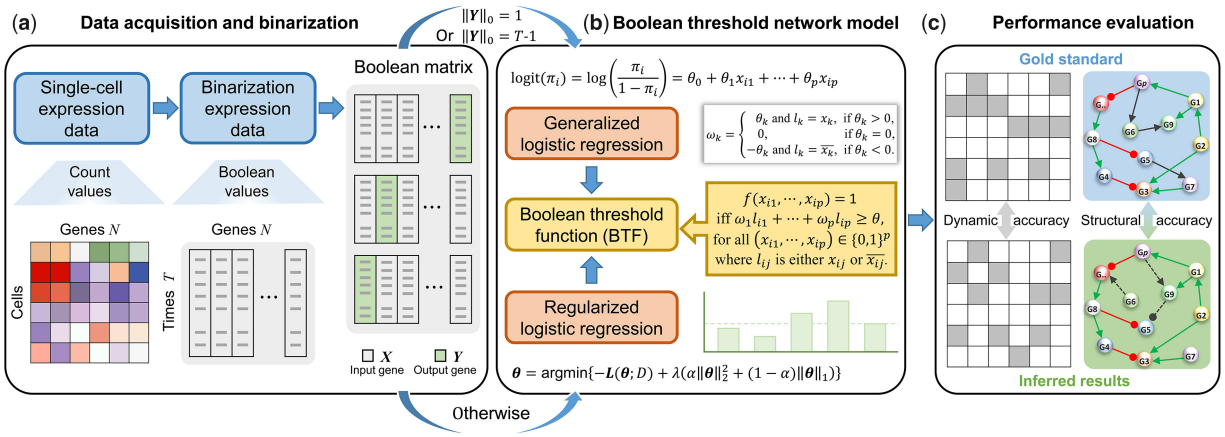

Figure 1. The framework of LogBTF for GRN inference from single-cell gene expression data, where $\|Y\|_0 = \sum_{j=1}^{T-1} 1(y_j \neq 0)$ is the L_0-norm. (**a**) For single-cell gene expression data with corresponding pseudo-time series information, binarizing the count values into Boolean values (0 or 1). (**b**) Based on the binarized gene expression data, Boolean threshold network model is constructed by aggregating Boolean threshold function with logistic regression coefficients. (**c**) The inferred gene expression state and the reconstructed GRN are, respectively, compared with the original states and gold standard to evaluate the inference performance.

our proposed method. Then, considering that simulated single-cell gene expression data are the promising alternative approach for mimicking real data with their statistical properties and underlying biological relationships (Dibaeinia and Sinha 2020), so 15 simulated single-cell datasets from GeneNetWeaver (GNW) (Schaffter et al. 2011) and ten simulated single-cell datasets from SERGIO (Dibaeinia and Sinha 2020), guided by corresponding source networks (gold standard), are used in our experiments. Finally, three real scRNA-seq datasets mined from existing literature are also applied in our work. The details of all datasets are illustrated in Supplementary Table S1 and Supplementary Figs. S1–S3, in which SIGN is used to characterize the gold standard whether with signed edges (SIGN = 1, activation or inhibition) or not (SIGN = 0).

2.3 Data preprocessing

LogBTF method requires the state of each gene to be quantified as 0 (false/not expressed) or 1 (true/expressed). For the simulated or real single-cell expression data, no imputation is required, all dropouts are set to 0 and all non-zero counts are set to 1 regardless of the expression level (Qiu 2020). Moreover, for the bulk expression data additionally shown in the Supplementary Materials, the continuous gene expression values need to be pro-preprocessed by binarization. For each gene, its expression data at all time points can be seen as a one-dimensional vector, data binarization is the process of converting continuous data attribute values into finite interval sets, that is, 0 or 1 in Boolean modeling. The detailed implementation is presented in Supplementary Equation (S8).

2.4 Optimization of network topology

To overcome the multi-collinearity problem and detect reliable regulatory relationships, we propose an optimization strategy based on knowledge of the perturbation design matrix (Seçilmiş et al. 2022) as follows: first, a normal distribution matrix Σ with mean $\mu = 0$ and variance σ^2 (enough small) is generated, and the dimension of Σ is the same as the dimension of binarized input matrix $\boldsymbol{X}$. Then, a new perturbation input matrix $\hat{\boldsymbol{X}} \triangleq \boldsymbol{X} + \Sigma$ is obtained by adding the binarized input matrix and the random generated matrix. Based on the newly obtained input matrix $\hat{\boldsymbol{X}}$, we apply the LogBTF method to estimate the regression coefficient $\hat{\boldsymbol{\theta}}_i$ of the i-th target gene as follows:

$$\hat{\boldsymbol{\theta}}_i = (\hat{\theta}_1^i, \hat{\theta}_2^i, \ldots, \hat{\theta}_N^i, \hat{\theta}_0^i)^\top, \quad i \in [1, N]. \tag{11}$$

Using the above strategy, the multi-collinearity problem is solved but the estimated value $\hat{\boldsymbol{\theta}}_i$ is noisy, and if the resulting regression coefficients are used directly for GRN inference, some redundant edges are generated. To further optimize the inferred network topology, we first remove the influence of the added random tiny perturbation matrix Σ by setting the coefficient whose absolute value is less than the given σ value to 0. Then the remaining no-zero elements in the coefficient $\hat{\boldsymbol{\theta}}_i$ are carried out to infer the potential GRN. Finally, we normalize the regulatory coefficient $\hat{\boldsymbol{\theta}}_i$ of each gene by

$$\hat{\boldsymbol{\theta}}_i \leftarrow \frac{\hat{\boldsymbol{\theta}}_i}{\|\hat{\boldsymbol{\theta}}_i\|_\infty}, \tag{12}$$

where $\|\hat{\boldsymbol{\theta}}_i\|_\infty = \max_{1 \leq k \leq N} |\hat{\theta}_k^i|$ is the L_∞-norm. Thus, the contribution of all regulatory genes to the given target gene has a standard scale, with the strongest regulatory relationship being assigned 1, and the weakest being 0.

2.5 Parameter selection and performance evaluation

On one hand, the selection of optimal tuning parameters for our method is vital, the specific process can be found in Supplementary Equations (S9) and (S10). On the other hand, we define the DyAcc, StAcc, recall (Recal), precision (Pre), false positive rates (FPRs), *F*-measure, and the area under the ROC curve (AUC) (Bradley 1997) to evaluate the inferring performance of LogBTF and other eight inferring methods. At the same time, we assess the performance of LogBTF by evaluating the areas under the receiver operating characteristic (AUROC) (Hanley and McNeil 1982) and the precision–recall curve (AUPR). All corresponding definitions and calculation formulas of the above evaluation metrics and the differences between LogBTF and eight alternative GRN inference methods can be found in Supplementary Equations (S11)–(S13) and Supplementary Table S2.

3 Results

3.1 Simulated Boolean data

For ease of exposition, let $[w_1 l_1 + w_2 l_2 + \cdots + w_N l_N \geq \theta_i]$ denote the Boolean threshold function defined by

$$= \begin{cases} [w_1 l_1 + w_2 l_2 + \cdots + w_N l_N \geq \theta_i] \\ 1, \quad w_1 l_1 + w_2 l_2 + \cdots + w_N l_N \geq \theta_i, \\ 0, \quad \text{otherwise.} \end{cases} \tag{13}$$

According to Theorem 2.1, we find that $w_1 l_1 + w_2 l_2 + \cdots + w_N l_N \geq \theta_i$ in Equation (10) is equivalent to $\theta_1^i x_1(j) + \theta_2^i x_2(j) + \cdots + \theta_N^i x_N(j) + \theta_0^i \geq 0$ in Equation (5). We can therefore determine the coefficients $w_1, w_2, \ldots, w_N$ and θ_i in Equation (13) by estimating

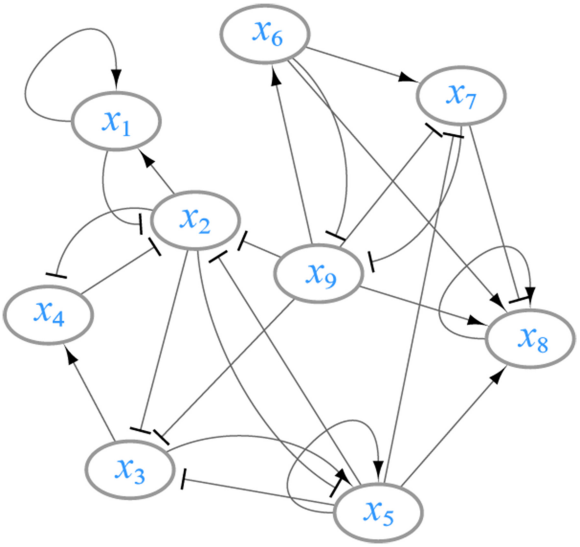

Figure 2. The synthetic Boolean threshold network, where each node x_i has a state value of 1 (expressed) or 0 (not expressed). The directed links with "→" represent a positive regulation (activating) from x_i to x_j, and "⊣" represent a negative regulation (inhibiting).

the regression coefficients. Specifically, $w_k = \theta_k^i$ if $\theta_k^i > 0$; $w_k = 0$ if $\theta_k^i = 0$; $w_k = -\theta_k^i$ if $\theta_k^i < 0$ and $\theta_i = -\theta_0^i - \sum_{k\in\{k|k\in[1,N],\theta_k^i<0\}} \theta_k^i$.

First, we generated a set of synthetic expression data. Here, we first construct a BN with nine nodes as follows:

$$\begin{aligned}
x_1 :&\ [x_1 + x_2 \geq 1],\\
x_2 :&\ [\overline{x_1} + \overline{x_4} + \overline{x_5} + \overline{x_9} \geq 4],\\
x_3 :&\ [\overline{x_2} + \overline{x_5} + \overline{x_9} \geq 3],\\
x_4 :&\ [\overline{x_2} + x_3 \geq 2],\\
x_5 :&\ [\overline{x_2} + x_3 + x_5 \geq 3],\\
x_6 :&\ [x_9 \geq 1],\\
x_7 :&\ [\overline{x_5} + 2x_6 + \overline{x_9} \geq 2],\\
x_8 :&\ [x_5 + x_6 + 5\overline{x_7} + 4x_8 + x_9 \geq 5],\\
x_9 :&\ [\overline{x_6} + \overline{x_7} \geq 2].
\end{aligned} \tag{14}$$

Clearly, there are 2^9 initial states. For each initial state, according to the Boolean threshold function in Equation (13), the state of each node at the next time point is obtained. Therefore, for each node x_i with $i \in [1, 9]$, there are 512 observations in total. The Boolean threshold network corresponding to the BN in Equations (14) is shown in Fig. 2.

In the following, we use all possible initial states to form a state matrix at time t, and set this matrix as the input data $\boldsymbol{X}$. The state of the node at time $t+1$ is set as output $\boldsymbol{y}$. In this way, the input state matrix $\boldsymbol{X}$ is a full-rank matrix, which avoids the multi-collinearity problem between variables for coefficient estimation. Considering that the node size is only nine, it does not need to penalize the coefficients, so we employ the generalized linear regression (un-penalized logistic regression) model by setting $\lambda = 0$ in Equation (8). In this case, we obtain the inferred Boolean threshold network as shown in Supplementary Equation (S14). Concerning the threshold network structure in Fig. 2, the inferred BN in Equation (S14) and BN in Equation (14) are equivalent, which indicates that the LogBTF method is effective and efficient for inferring GRN from time series data.

In order to test the robustness of the LogBTF method, we investigate the tolerance of the model to data changes. Namely, we introduce noise into the simulated time series data by randomly flipping the state of each node with the probability of δ for $\delta \in \{0\%, 1\%, 3\%, 5\%\}$, respectively. As described in Table 1, all metrics remain stable when the noise increases from 1% to 5%, especially for Pre and FPR indexes. AUROC decreases from 0.920 (at $\delta = 1\%$) to 0.880 (at $\delta = 5\%$). The trend of AUPR under different noise levels is similar to that of AUROC. Remarkably, LogBTF can infer the correct topology of the idealized regulatory network under 0% noise. When the noise ratio increases to 5%, LogBTF can still infer the topology with StAcc = 0.926, Recal = 0.760, and F-measure = 0.864. In conclusion, it is robust and stable for LogBTF to adopt the Boolean threshold network model to improve the inference performance. Especially, we also explore the inference accuracy of the LogBTF method by randomly sampling a certain percentage of data from 512 observations, and the related results with discussions are given in Supplementary Fig. S4.

Table 1. The performance of LogBTF inferring the Boolean network with nine nodes under different noise levels.

Noise	AUROC	AUPR	StAcc	Recal	Pre	FPR	F-measure
$\delta = 0\%$	1.000	1.000	1.000	1.000	1.000	0.000	1.000
$\delta = 1\%$	0.920	0.943	0.951	0.840	1.000	0.000	0.913
$\delta = 3\%$	0.900	0.928	0.938	0.800	1.000	0.000	0.889
$\delta = 5\%$	0.880	0.913	0.926	0.760	1.000	0.000	0.864

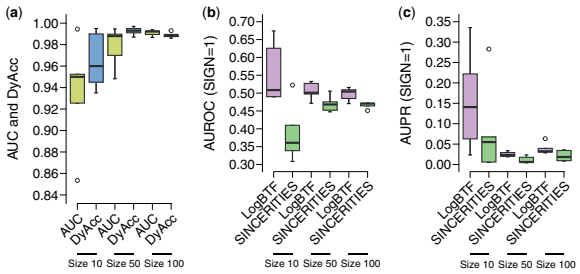

Figure 3. The prediction results of LogBTF method on simulated single-cell datasets in the case of SNGN = 1. (**a**) The AUC and DyAcc values of LogBTF method. (**b**) The AUROC and (**c**) AUPR comparison of LogBTF and SINCERITIES methods on three types of simulated datasets with different sizes.

3.2 Simulated single-cell data by GNW

3.2.1 Case 1: SIGN = 1

In this experiment, we further simulated dropout datasets to investigate the GRN inference applicability of LogBTF to zero-inflated single-cell gene expression data (Chan et al. 2017). Namely, we induced dropout events at ∼20% − 25% ratio for the data generated from GNW. For each sample, the expression values lower than the given threshold for each gene would be recorded as 0 according to a Binomial probability of 50% (Chen and Mar 2018). The expression value distributions and ground truth networks of all datasets are shown in Supplementary Figs. S1 and S2. We predict the expression state of each gene during all time points to investigate the performance of LogBTF in terms of predictive AUC value. Figure 3(a) shows that the means of the AUC value and DyAcc value on 15 simulated single-cell datasets are >0.92.

Considering that the SINCERITIES method is also functional in evaluating the inferred network from the aspect of activating and inhibitive regulations between regulator and target genes, we compare the AUROC and AUPR indexes of LogBTF with SINCERITIES under the situation of SIGN = 1. Figure 3(b) shows the results of the AUROC value comparing the LogBTF with the SINCERITIES method on three types of datasets of different sizes. We can see that our proposed LogBTF method performs better than SINCERITIES for all different network sizes. Also, there seems to be a trend that when the number of genes in the network increases, the difference between these two methods gradually decreases, under the premise that LogBTF is better than SINCERITIES. From Fig. 3(c), one can see that the AUPR values of LogBTF are significantly larger than SINCERITIES when the network size is small. While as the network size increases, our LogBTF method still shows strong superiority when compared with SINCERITIES, both from single experimental results and mean value. In addition, we also show the StAcc results in Supplementary Fig. S5, the larger the number of nodes in the network, the higher the StAcc value of our method.

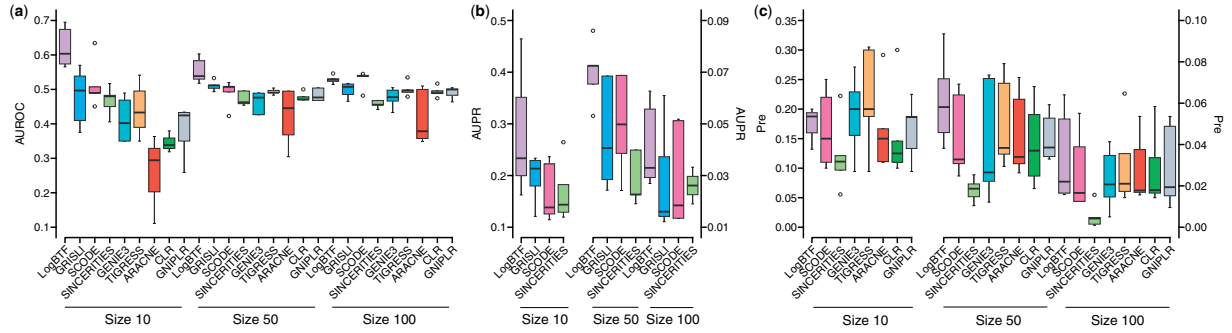

Figure 4. The comparing results of LogBTF with the other methods on three type datasets with different gene sizes from simulated single-cell data in the case of SNGN = 0. (**a**) The AUROC values. (**b**) The AUPR value. (**c**) The Pre value.

3.2.2 Case 2: SIGN = 0

When omitting the activation or inhibition functions, we only study the regulatory relationships and the regulator/target roles among genes, i.e. SIGN = 0. In this case, the signature of regression or correlation coefficient does not work, which means that all coefficients can be taken as the measure like weights. Thus these nine GRN inference methods have a standard benchmark for comparisons. For more reliable performance validation, we further compare LogBTF with eight methods, including SINCERITIES that we just discussed, GRISLI, SCODE, GENIE3, TIGRESS, ARACNE, CLR, and GNIPLR. The comparing results of AUROC, AUPR, and Pre indexes on all available datasets are shown in Fig. 4, where each boxplot describes the statistical results from five independent runs with five simulated datasets under three different network size groups. The center line denotes the median, the lower and upper hinges, respectively, represent the 25th and 75th percentiles, the vertical lines express the 1.5 times interquartile range, and the dots outside the vertical lines are the outliers.

Figure 4(a) displays that our proposed LogBTF method is significantly superior to the other eight comparable methods in terms of AUROC evaluation with size 10 and size 50. In contrast, the performance of LogBTF is less satisfactory in the network with node size 100, where the SCODE achieves almost perfect inference. And, compared to all methods, ARACNE obtains the worst AUROC results on all simulated single-cell datasets. Although AUROC values are the classical choice for comparing methods, the AUPR value is more relevant for evaluating the network comparison (Chen and Mar 2018). Figure 4(b) shows the AUPR values among four inference methods specially developed for single-cell data, where the LogBTF method achieves the best results regardless of the network size of the dataset. In particular, when the size of network nodes is relatively large (such as size 50), our method has the largest mean AUPR and the smallest standard deviation.

The problem of GRN inference is a sparse prediction problem, which has a relatively low positive rate. In this case, Pre is a more valuable index because it measures the proportion of correctly inferred edges (Chen and Mar 2018). Figure 4(c) gives the Pre values comparing the inferred network relationship with sourced gold standards. Here we do not compare GRISLI method considering its source code does not output the Pre value metric. In the five datasets with size 10, the LogBTF method achieves lower (mean) Pre values than GENE3 and TIGRESS methods, but its standard deviation is smaller than theirs. As for the case of size 50 and size 100, compared with seven alternative methods, the mean Pre value of LogBTF is the highest. Especially, SINCERITIES obtains the worst Pre values among all methods on all 15 datasets, although whose standard deviation is the smallest.

All experimental results on datasets from GWN, including other estimation criteria (such as Recal, FPR, and *F*-measure), are available in Supplementary Tables S3–S6. They reflect the stability and applicability of the LogBTF method for different networks. Additionally, in order to study the applicability of our GRN inference method to bulk gene expression data, we also apply our LogBTF method to the synthetic bulk RNA-seq data from GNW (Schaffter et al. 2011). The results and discussion are available in Supplementary Figs. S6 and S7 and Supplementary Table S7. All results reflect the universality and applicability of the LogBTF method for large-scale networks.

3.3 Simulated single-cell data by SERGIO

To further verify the inference performance of LogBTF, ten single-cell gene expression datasets generated by SERGIO simulator (Dibaeinia and Sinha 2020) are employed in the numerical experiments as well. Especially, as a simulator for single-cell expression data guided by given GRNs, SERGIO can generate data that includes technical noise, outliers, and "dropout" and convert the data to Unique Molecular Identifier counts. In this part, we simulated the single-cell expression profiles of 20 and 100 genes with 5 kinds of cell numbers (10, 20, 30, 40, and 50) in each cell type. For the datasets with network size 20, we set the "dropout" parameters (i.e. percentile) >80, while setting the "dropout" parameters >50 for the datasets with network size 20. Thus, a total of ten simulation datasets are obtained, and the two prior networks (gold standard) are shown in Supplementary Fig. S3, respectively.

Next, we also conduct experiments using LogBTF method and eight other GRN inference methods on these ten simulated single-cell data. Previous experiments have demonstrated the inference performance of LogBTF on directed networks, so we will not repeat them here. Here we only comprehensively compare the network inference performances of different GRN inference methods in the case of SIGN = 0. Figure 5(a) shows that LogBTF achieves the best overall inference performance in terms of AUROC values regardless of network size 20 or 100. It also depicts that SINCERITIE ranks the second for the size 20 network and GRISLI ranks the second for the size 100 network. Moreover, consistent with the trend on the simulated dataset generated by GNW, the larger the network size is, the smaller the standard deviation of all methods will be.

In particular, compared with three popular methods (GRISLI, SCODE, and SINCERITIE) for inferring GRN from single-cell gene expression data, Fig. 5(b) gives the comparing results of AUPR values, which demonstrates that LogBTF outperforms GRISLI and SINCERITIE, but not better/competitive than SCODE on the datasets of size 20 network. As for the Precision ("Pre") metric, just like the experiments in the last subsection, Fig. 5(c) illustrates that the proposed LogBTF method achieves higher inference accuracy than the other seven methods. In contrast, SINCERITIE performed poorly in terms of precision. As shown in Fig. 5, although the single-cell data generated by the SERGIO simulator has large dropouts, LogBTF is also effective in GRN inference and performs better than the other methods.

3.4 Real scRNA-seq data

To evaluate the performance of the LogBTF method, we apply it to three retrieved real single-cell gene expression datasets. Considering that the known gold standard networks of Matsumoto and hHEP datasets do not underlie the activation or inhibition information, so

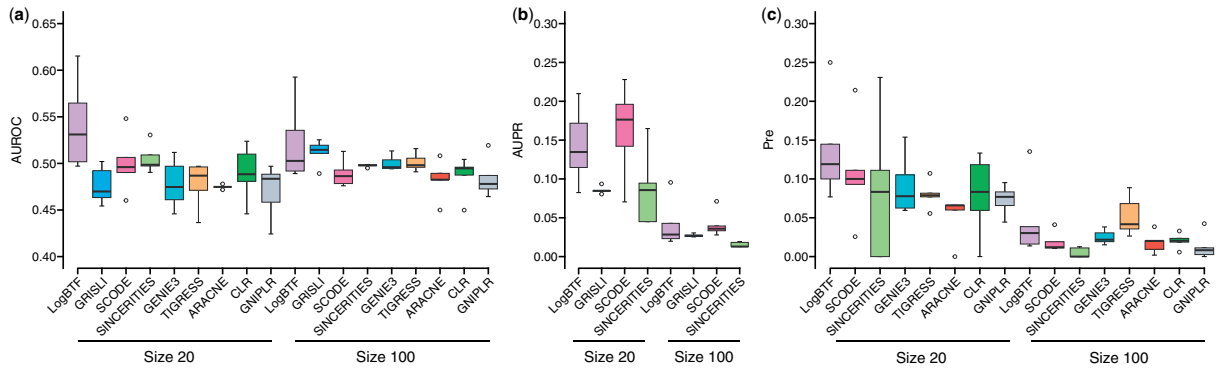

Figure 5. The comparison results of LogBTF with the other methods on two types of single-cell simulation datasets from SERGIO in the case of SNGN = 0. (**a**) The AUROC values. (**b**) The AUPR value. (**c**) The Pre value.

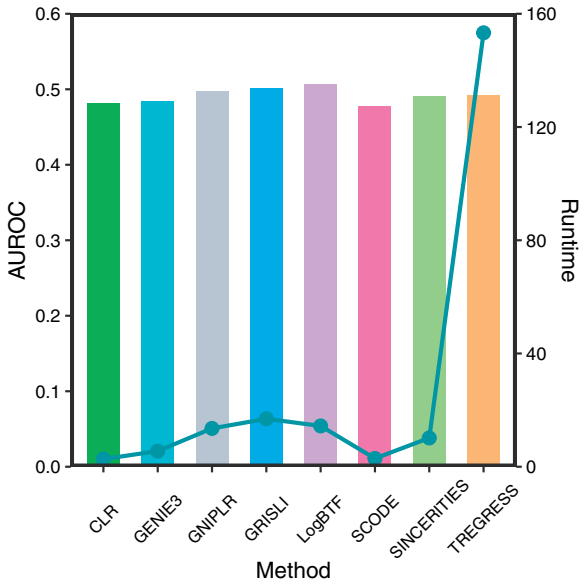

Figure 6. The performance comparisons among different GRN inference methods on Matsumoto dataset.

we only investigate the performances in the case of SIGN = 0. Figure 6 shows the AUROC comparison result of LogBTF with the other seven methods on Matsumoto dataset, where LogBTF method behaves with higher AUROC values than those of the other seven methods and GRISLI shows a second-best AUROC. We remark that ARACNE performs extremely worst AUROC value (zero), here we do not show it in Fig. 6. So, ARACNE looks not appropriate in the case of scRNA-seq data with the pseudo-time, as mentioned in the prior work that it is not applied to the time course data (Cantone et al. 2009).

In addition, Fig. 6 also counts the runtimes, where the computation time of LogBTF is significantly smaller than GRISLI and TIGRESS. In theory, LogBTF and SINCERITIES have the same computational complexity, but we design a program to calculate more comprehensive indicators (including mean AUC and DyAcc) and consider the time evolution/update rules (in the form of Boolean threshold function equations) in each inference process, so the actual running time of LogBTF is almost double that of SINCERITIES. In particular, we note that TIGRESS is not suitable for large-scale network inference analysis due to the expensive computational time, though it can also obtain satisfactory inference results. Additionally, similar comparisons on hHEP dataset are given in Supplementary Fig. S8 and Supplementary Table S8. All the experiments are conducted on a workstation with two Xeon Gold 6226R CPUs and 256G of RAM.

Finally, we apply the LogBTF method to the LMPP scRNA-seq dataset with 31 genes and 531 pseudo-time points. As a result, LogBTF method infers 306 regulatory relationships with DyAcc = 0.676, taking 1.156 min. The inferred GRN shown in Supplementary Fig. S9 is represented by 31 genes with 166 activating and 140 inhibiting regulatory relationships. From the existing literature (Hamey et al. 2017), we found 72 regulatory relationships have been verified, with 40 activations and 21 inhibitions accurately inferred. Furthermore, we perform the network ontology analysis (Wang et al. 2011) to enrich the network biological significance of the regulatory relationship in the inferred GRN. The results can be found in Supplementary Table S9, which shows that the GRN inferred by the LogBTF method provides a reference for elaborating molecular regulatory mechanisms in biology.

4 Conclusions

In this study, we proposed the Boolean threshold network framework, named LogBTF, by aggregating regularized logistic regression with Boolean threshold function. LogBTF is a *de novo* GRN inference method, and it is also a logic model with explainability of the regulatory relationships among genes. First, we applied the logistic regression model to estimate the parameters of the Boolean threshold network model from given gene expression data and proved their equivalence in theory. Second, we designed an optimization strategy to handle the multi-collinearity problem caused by binarized data and applied a cross-validation procedure to select the optimal tuning parameters. Finally, we conducted extensive experiments with the single-cell gene expression datasets of simulated, *in silico* and real and compared the performance of our method with those of eight well-known existing inference methods. In particular, our method significantly outperformed them in terms of AUROC, AUPR, Precision, and other evaluation criteria. Apparently, these results indicate that the proposed approach is a promising tool for accurate regulatory networks from time series single-cell gene expression data. Although the LogBTF method increases computation cost by utilizing cross-validation to choose the optimal parameters, it is possible to reduce the running time by parallel implementation, which will be included in our future study. Another future direction is to infer GRNs by constructing an asynchronously updated Boolean threshold network model and then comparing its results on the network dynamics with that of the synchronous update model.

Acknowledgments

The authors would like to thank the editors and anonymous reviewers for their valuable comments and suggestions which greatly improved the article. They also thank the members of our lab at Shandong University and AMACL lab at the University of Hong Kong for their assistance in the project, and Dr. Jianming Zeng at the University of Macau for sharing related references.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest statement

None declared.

Funding

This work was partially supported by the National Key Research and Development Program of China [grant number 2020YFA0712402]; National Natural Science Foundation of China (NSFC) [grant number 61973190]; Hong Kong RGC GRF [grant number 17301519]; the Innovation Method Fund of China (Ministry of Science and Technology of China) [grant number 2018IM020200]; the Scholarship under Shandong University's Exchange Program; the Fundamental Research Funds for the Central Universities [grant number 2022JC008]; and the Program of Qilu Young Scholars of Shandong University.

References

Aalto A, Viitasaari L, Ilmonen P *et al.* Gene regulatory network inference from sparsely sampled noisy data. *Nat Commun* 2020;**11**:3493. https://doi.org/10.1038/s41467-020-17217-1.

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.

Akutsu T, Miyano S, Kuhara S. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pac Symp Biocomput* 1999;**4**:17–28. https://doi.org/10.1142/9789814447300_0003.

Akutsu T, Miyano S, Kuhara S. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics* 2000;**16**:727–34. https://doi.org/10.1093/bioinformatics/16.8.727 .

Algabri YA, Li L, Liu ZP. scGENA: a single-cell gene coexpression network analysis framework for clustering cell types and revealing biological mechanisms. *Bioengineering* 2022;**9**:353. https://doi.org/10.3390/bioengineering9080353 .

Anthony M. *Discrete Mathematics of Neural Networks: Selected Topics*. Philadelphia, PA: Society for Industrial and Applied Athematics, 2001.

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. https://doi.org/10.1093/bioinformatics/btaa576.

Barman S, Kwon YK. A novel mutual information-based Boolean network inference method from time-series gene expression data. *PLoS ONE* 2017;**12**:e0171097. https://doi.org/10.1371/journal.pone.0171097.

Beal MJ, Falciani F, Ghahramani Z *et al.* A bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics* 2005;**21**:349–56. https://doi.org/10.1093/bioinformatics/bti014.

Bornholdt S. Boolean network models of cellular regulation: prospects and limitations. *JR Soc. Interface* 2008;**5**:S85–94.

Bradley AP. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn* 1997;**30**:1145–59. https://doi.org/10.1016/S0031-3203(96)00142-2.

Cantone I, Marucci L, Iorio F *et al.* A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell* 2009;**137**:172–81. https://doi.org/10.1016/j.cell.2009.01.055.

Chan TE, Stumpf MPH, Babtie AC *et al.* Gene regulatory network inference from single-cell data using multivariate information measures. *Cell Syst* 2017;**5**:251–67.e3. https://doi.org/10.1016/j.cels.2017.08.014.

Chen G, Liu ZP. Graph attention network for link prediction of gene regulations from single-cell RNA-sequencing data. *Bioinformatics* 2022;**38**:4522–9. https://doi.org/10.1093/bioinformatics/btac559.

Chen J, Cheong J, 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.

Chen S, Mar JC. Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data. *BMC Bioinf* 2018;**19**:1–21. https://doi.org/10.1186/s12859-018-2217-z.

Cheng X, Ching WK, Guo S *et al.* Discrimination of attractors with noisy nodes in Boolean networks. *Automatica* 2021;**130**:109630. https://doi.org/10.1016/j.automatica.2021.109630.

Dibaeinia P, Sinha S. SERGIO: a single-cell expression simulator guided by gene regulatory networks. *Cell Syst* 2020;**11**:252–71.e11. https://doi.org/10.1016/j.cels.2020.08.003.

Faith JJ, Hayete B, Thaden JT *et al.* Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol* 2007;**5**:e8. https://doi.org/10.1371/journal.pbio.0050008.

Font-Clos F, Zapperi S, La Porta CAM *et al.* Classification of triple-negative breast cancers through a Boolean network model of the epithelial–mesenchymal transition. *Cell Syst* 2021;**12**:457–62.e4. https://doi.org/10.1016/j.cels.2021.04.007.

Hamey FK, Nestorowa S, Kinston SJ *et al.* Reconstructing blood stem cell regulatory network models from single-cell molecular profiles. *Proc Natl Acad Sci USA* 2017;**114**:5822–9. https://doi.org/10.1073/pnas.1610609114.

Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 1982;**143**:29–36. https://doi.org/10.1148/radiology.143.1.7063747.

Haury AC, Mordelet F, Vera-Licona P *et al.* TIGRESS: trustful inference of gene regulation using stability selection. *BMC Syst Biol* 2012;**6**:1–17. https://doi.org/10.1186/1752-0509-6-145.

Huang Y, McCarthy DJ, Stegle O. Vireo: bayesian demultiplexing of pooled single-cell RNA-seq data without genotype reference. *Genome Biol* 2019;**20**:12. https://doi.org/10.1186/s13059-019-1865-2 .

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.

James G, Witten D, Hastie T *et al. An Introduction to Statistical Learning*. New York, NY: Springer, 2013.

Kauffman SA. Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol* 1969;**22**:437–67.

Li L, Liu ZP. Biomarker discovery for predicting spontaneous preterm birth from gene expression data by regularized logistic regression. *Comput Struct Biotechnol J* 2020;**18**:3434–46. https://doi.org/10.1016/j.csbj.2020.10.028.

Li L, Liu ZP. A connected network-regularized logistic regression model for feature selection. *Appl Intell* 2022;**52**:11672–702. https://doi.org/10.1007/s10489-021-02877-3.

Liu X, Wang Y, Shi N *et al.* GAPORE: Boolean network inference using a genetic algorithm with novel polynomial representation and encoding scheme. *Knowl.-Based Syst* 2021;**228**:107277. https://doi.org/10.1016/j.knosys.2021.107277.

Liu ZP. Reverse engineering of genome-wide gene regulatory networks from gene expression data. *Curr Genomics* 2015;**16**:3–22. https://doi.org/10.2174/1389202915666141110210634.

Liu ZP, Wu C, Miao H *et al.* RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database* 2015;**2015**:bav095. https://doi.org/10.1093/database/bav095.

Luo Q, Yu Y, Lan X. SIGNET: single-cell RNA-seq-based gene regulatory network prediction using multiple-layer perceptron bagging. *Brief Bioinform* 2022;**23**:bbab547. https://doi.org/10.1093/bib/bbab547 .

Margolin AA, Nemenman I, Basso K *et al.* ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinf* 2006;**7**:1–15. https://doi.org/10.1186/1471-2105-7-S1-S7.

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.

Melkman AA, Cheng X, Ching WK *et al.* Identifying a probabilistic boolean threshold network from samples. *IEEE Trans Neural Netw Learn Syst* 2018;**29**:869–81. https://doi.org/10.1109/TNNLS.2017.2648039.

Mori T, Akutsu T. Attractor detection and enumeration algorithms for Boolean networks. *Comput Struct Biotechnol J* 2022;**20**:2512–20. https://doi.org/10.1016/j.csbj.2022.05.027.

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.

Qiu P. Embracing the dropouts in single-cell RNA-seq analysis. *Nat Commun* 2020;**11**:1169. https://doi.org/10.1038/s41467-020-14976-9.

Schaffter T, Marbach D, Floreano D. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods.

Bioinformatics 2011;**27**:2263–70. https://doi.org/10.1093/bioinformatics/btr373.

Seçilmiş D, Hillerton T, Tjärnberg A *et al*. Knowledge of the perturbation design is essential for accurate gene regulatory network inference. *Sci Rep* 2022;**12**:16531. https://doi.org/10.1038/s41598-022-19005-x.

Shi N, Zhu Z, Tang K *et al*. ATEN: and/or tree ensemble for inferring accurate Boolean network topology and dynamics. *Bioinformatics* 2020;**36**:578–85. https://doi.org/10.1093/bioinformatics/btz563.

Shu H, Zhou J, Lian Q *et al*. Modeling gene regulatory networks using neural network architectures. *Nat Comput Sci* 2021;**1**:491–501. https://doi.org/10.1038/s43588-021-00099-8.

Sun L, Ching WK. Stabilization and reconstruction of sampled-data boolean control networks under noisy sampling interval. *IEEE Trans Automat Contr* 2023;**68**:2444–51. https://doi.org/10.1109/TAC.2022.3173942.

Sun L, Ching WK, Lu J. Stabilization of aperiodic sampled-data boolean control networks: a delay approach. *IEEE Trans Automat Contr* 2021;**66**:5606–11. https://doi.org/10.1109/TAC.2021.3055191 .

Wang C, Xu S, Liu ZP *et al*. Evaluating gene regulatory network activity from dynamic expression data by regularized constraint programming. *IEEE J Biomed Health Inform* 2022;**26**:5738–49. https://doi.org/10.1109/JBHI.2022.3199243.

Wang J, Huang Q, Liu ZP *et al*. NOA: a novel network ontology analysis method. *Nucleic Acids Res* 2011;**39**:e87. https://doi.org/10.1093/nar/gkr251.

Wu S, Liu ZP, Qiu X *et al*. Modeling genome-wide dynamic regulatory network in mouse lungs with influenza infection using high-dimensional ordinary differential equations. *PLoS ONE* 2014;**9**:e95276. https://doi.org/10.1371/journal.pone.0095276.

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.

Zañudo JG, Aldana M, Martínez-Mekler G. Boolean threshold networks: virtues and limitations for biological modeling. In: Niiranen S. and Ribeiro A. (eds.) *Information Processing and Biological Systems*. Berlin Heidelberg, Germany: Springer, 2011, 113–51.

Zhang SQ, Ching WK, Tsing NK *et al*. A new multiple regression approach for the construction of genetic regulatory networks. *Artif Intell Med* 2010;**48**:153–60. https://doi.org/10.1016/j.artmed.2009.11.001.

Zhang X, Zhao XM, He K *et al*. Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. *Bioinformatics* 2012;**28**:98–104. https://doi.org/10.1093/bioinformatics/btr626.

Zhang Y, Chang X, Liu X *et al*. Inference of gene regulatory networks using pseudo-time series data. *Bioinformatics* 2021;**37**:2423–31. https://doi.org/10.1093/bioinformatics/btab099.

Zou H, Hastie T. Regularization and variable selection via the elastic net. *J R Stat Soc Ser B-Stat Methodol* 2005;**67**:301–20. https://doi.org/10.1111/j.1467-9868.2005.00503.x.