

Inferring structural and dynamical properties of gene networks from data with deep learning

Feng Chen^{1,2} and Chunhe Li^{1,2,3,*}

¹Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai 200433, China,

²Shanghai Center for Mathematical Sciences, Fudan University, Shanghai 200433, China and ³School of Mathematical Sciences, Fudan University, Shanghai 200433, China

Received April 18, 2022; Revised July 22, 2022; Editorial Decision August 15, 2022; Accepted August 24, 2022

ABSTRACT

The reconstruction of gene regulatory networks (GRNs) from data is vital in systems biology. Although different approaches have been proposed to infer causality from data, some challenges remain, such as how to accurately infer the direction and type of interactions, how to deal with complex network involving multiple feedbacks, as well as how to infer causality between variables from real-world data, especially single cell data. Here, we tackle these problems by deep neural networks (DNNs). The underlying regulatory network for different systems (gene regulations, ecology, diseases, development) can be successfully reconstructed from trained DNN models. We show that DNN is superior to existing approaches including Boolean network, Random Forest and partial cross mapping for network inference. Further, by interrogating the ensemble DNN model trained from single cell data from dynamical system perspective, we are able to unravel complex cell fate dynamics during preimplantation development. We also propose a data-driven approach to quantify the energy landscape for gene regulatory systems, by combining DNN with the partial self-consistent mean field approximation (PSCA) approach. We anticipate the proposed method can be applied to other fields to decipher the underlying dynamical mechanisms of systems from data.

INTRODUCTION

Many complex biological processes can be modeled as dynamical systems through regulatory networks. For example, through dynamical network modeling, one can study and analyze the underlying mechanisms of cell fate decisions such as stem cell development (1–5) and cancerization (6–9). So, the reconstruction of gene regulatory networks (GRNs) and corresponding dynamical models from

biological data is essential to understanding cellular functions. Many computational tools have been proposed to infer the structure of GRNs (directed or undirected). However, the challenge remains (10). For example, with the approaches from correlation and mutual information it's hard to identify the direction of the interactions. Approaches based on the Random Forest (11) have the potential to infer the direction of interactions, but have difficulties to determine the type of interactions (i.e. activation or inhibition). More recent approaches such as LogicNet algorithm which based on Boolean network can reconstruct GRNs with directed and signed edges (12), however, whether this approach can be applied to more complex networks with feedback loops remains unclarified. Also, these approaches share a common limitation which is that they cannot distinguish direct interactions from indirect interactions. If two nodes that are connected indirectly through other nodes or a network, a direct interaction may be detected from these approaches. Therefore, recently many researchers focus on how to eliminate indirect effects and have proposed methods such as partial cross mapping (PCM) (13) and part mutual information (PMI) (14).

On the other hand, how to quantify the transition dynamics of gene regulatory systems is a substantial question in systems biology, and the energy landscape theory provides an possible route (1,15–20). For example, in our previous work, we have developed a self-consistent mean field approximation (PSCA) approach to calculate the steady state probability distribution and potential energy landscape (21), and a model-based dimension reduction approach of the landscape (DRL) for high-dimensional gene regulatory systems (7). Nevertheless, it remains challenging to develop approaches to quantify the landscape through combining model-driven and data-driven approaches.

Deep learning has made breakthroughs in many fields (22). Recently, researchers have used convolutional neural networks (23) and graph neural networks (24) to reconstruct GRNs from data, but these approaches are limited to inferring the direction of the interactions and cannot obtain the type of interactions. Recently, Shen et al. have used recurrent neural network (RNN) to infer the structure

*To whom correspondence should be addressed. Tel: +86 02131243871; Email: chunheli@fudan.edu.cn

of gene regulatory networks successfully, in several biological systems with different functions including adaptation, controlled oscillation, and pattern formation (25). This work emphasizes the potential effectiveness of deep neural network (DNN) used for gene network reconstruction. In biology, multistable and oscillation phenomena, which are closely related to nonlinear dynamics, play vital roles in cell fate decisions processes. Therefore, whether the DNN can be generalized to infer the structure of multistable and oscillation systems, especially for high-dimensional systems remain to be studied. On the other hand, the reconstruction of gene networks from single cell data is another challenging problem (26–28). Inspired by previous studies on DNN and the development of neural ordinary differential equations (neural ODEs) (25,29–32), we seek to apply DNN to multistable and oscillation systems for the network reconstruction. Especially, we consider the ensemble of multiple models to extract more information from the data to address the unique challenges posed by single-cell data, such as cell-to-cell variabilities, short time series length, and high sparsity. This is not only beneficial for reducing computational complexity, but also allows for better modeling of system dynamics, resulting in more accurate causal inferences. Thus, the network structure and the dynamics that are both of practical interest for revealing cell fate decision mechanism, can be obtained in one framework.

In this work, we introduce the DNN approach for inferring the structure of gene regulatory networks, and test this approach in different systems. We validate the effectiveness of the DNN in gene regulatory networks with different complexities (dimensions) both for multistable and oscillation systems. For benchmark, the DNN is compared with other methods including Random Forest, Boolean network (33,34) and PCM. We find that at different noise level (characterized by diffusion coefficient D), the DNN performs much better than other methods in both accuracy and robustness. We also test this network reconstruction approach for the real-world data from food chain and cardiovascular disease, for which the networks inferred by DNN are well supported by prior knowledge or previous studies.

To further test how the DNN works for real biological data, especially for single cell data, we study an important example for cell fate decisions, the mouse preimplantation development, which begins at fertilization and proceeds from the 1-cell fertilized zygote to the 64-cell blastocyst. In this developmental process, the trophectoderm (TE), primitive endoderm (PE) and epiblast (EPI) constitute three different cell types of the embryo (35,36). Although many insights have been gained from previous research, it remains to be fully clarified for the regulatory mechanisms leading to the formation of different phenotypes and their state transitions. Here, a major challenge is how to reconstruct the gene network and corresponding dynamics based on single cell data (37,38), which is paramount for revealing cell fate decision mechanisms. We tackle this issue by applying the ensemble learning to the single cell data for mouse embryo development. We not only accurately reconstruct a small-scale gene regulatory network based on single cell data, the causal inference of which is supported by a Hill-function model, but also effectively model a high-dimensional dy-

namical system for mouse embryo development. Based on this ensemble model, we successfully reconstruct the cell lineage of development, and quantify corresponding kinetic transition path in terms of gene expression levels. We also discover critical intermediate cell states in cell fate transition process, which have been shown to play critical roles in development (39). Finally, by combing the DNN model with the PSCA approach, we propose an approach for combining data-driven and model-driven strategy to quantify the energy landscape of gene networks.

Our work proposes a general data-driven approach to infer the structure and dynamics of regulatory networks, especially for multistable and oscillation systems in biology, and facilitates the mechanistic understanding of the cell fate decisions in preimplantation development.

MATERIALS AND METHODS

Deep neural network

The neural network model we use contains two hidden layers, each with 32 nodes. The number of neurons in the input layer and output layer varies for different systems. For example, in the MISA model (Figure 1C), the number of input neurons is four, namely x_1 , x_2 , a and b (a and b are parameters, characterizing the intensity of regulation), and the number of output neurons is two, i.e. f_1 and f_2 . In the four-dimensional system (Figure 2A), the number of input neurons is six, which are x_1 , x_2 , x_3 , x_4 , a and b , and the number of output neurons is four, which are f_1 , f_2 , f_3 and f_4 . The activation function is chosen as the ReLU function (except for the output layer, where the Sigmoid function is used to limit the state value to 0–1). The neural network model is trained through standard gradient optimization (Adam optimizer in Pytorch, learning rate is 0.0001), and the minimum batch size is set to 128. The loss function is the average value of the absolute difference between the prediction data and the training data (the more detailed parameters of each DNN model and some suggestions on the choice of these values can be found in Supplementary Text and Supplementary Table S1).

The trained DNN characterizes a function with input ($x_1, x_2, \dots, x_n, \theta_1, \theta_2, \dots, \theta_m$) and output (f_1, f_2, \dots, f_n). We denote the input as \mathbf{x} , the connection weights among the four layers as \mathbf{w}_1 , \mathbf{w}_2 , and \mathbf{w}_3 , and the corresponding bias terms are \mathbf{b}_1 , \mathbf{b}_2 , and \mathbf{b}_3 , respectively. Then the output \mathbf{f} can be expressed as:

$$\mathbf{f} = \text{Sigmoid}\{\mathbf{w}_3^T \cdot \text{ReLU}[\mathbf{w}_2^T \cdot \text{ReLU}(\mathbf{w}_1^T \cdot \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2] + \mathbf{b}_3\}. \quad (1)$$

It can be seen that f_i is a function of input \mathbf{x} , denoted as H_i :

$$f_i = H_i(x_1, x_2, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_n, \theta_1, \theta_2, \dots, \theta_m). \quad (2)$$

Then the link removal operation (e.g. removing the regulation link from X_j to X_i) can be realized by changing the input of the function H_i by setting $X_j = 0$, which leads to:

$$\hat{f}_i = \hat{H}_i(x_1, x_2, \dots, x_{j-1}, 0, x_{j+1}, \dots, x_n, \theta_1, \theta_2, \dots, \theta_m). \quad (3)$$

Here, \hat{f}_i is independent of the variable X_j , so the output dimensions ($f_1, f_2, \dots, f_{i-1}, \hat{f}_i, f_{i+1}, \dots, f_n$) simulate removing the regulation of X_j on X_i .

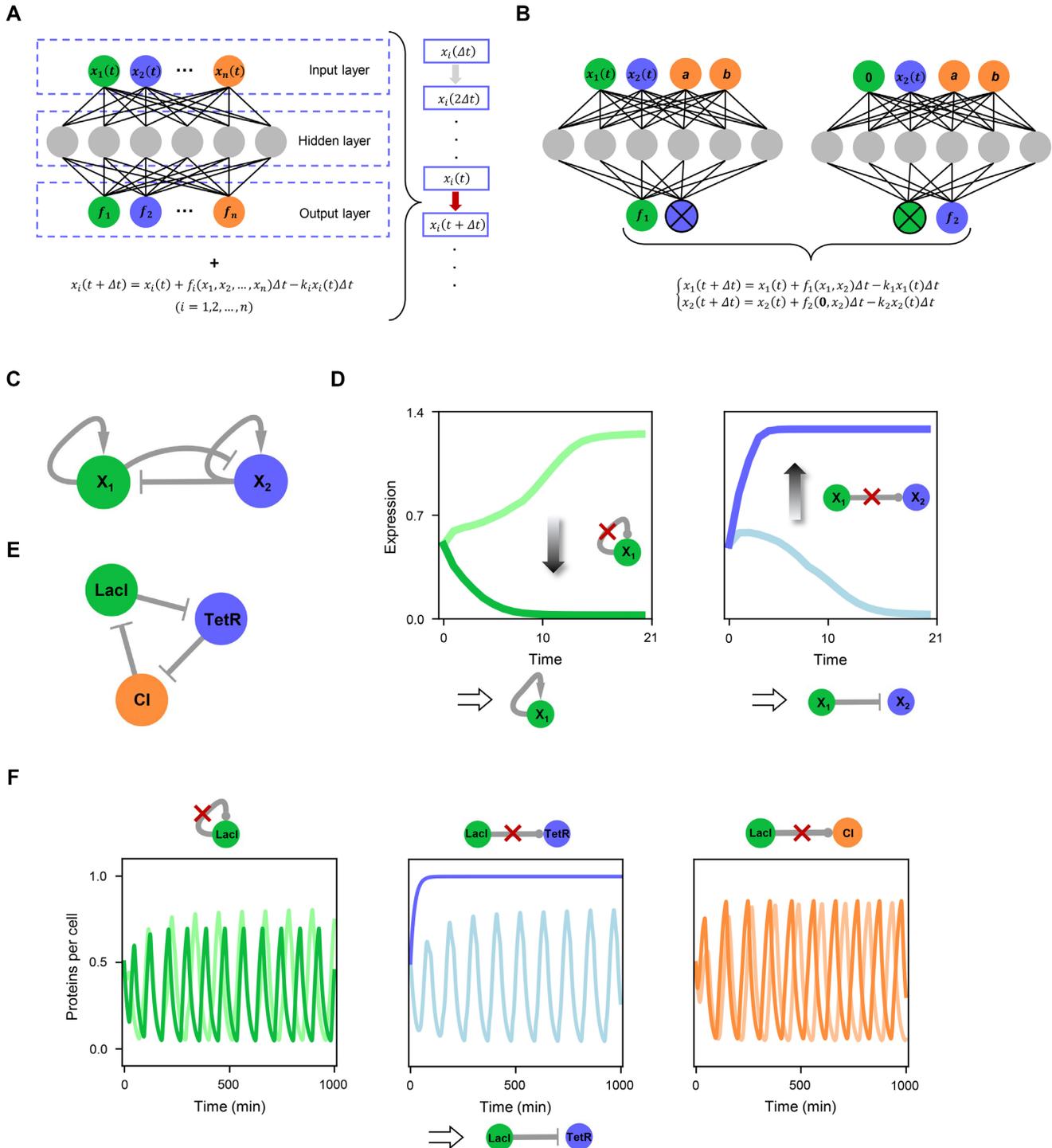


Figure 1. General DNN framework and its application on MISA model and oscillation system. (A) The deep neural network framework. The input is the gene expression of X_i ($i = 1, 2, \dots, n$) at time t , the output is f_i , and then the formula is used to calculate the gene expression of X_i at time $t + \Delta t$. In this loop, $x_i(t + \Delta t)$ is introduced into the model to obtain $x_i(t + 2\Delta t)$. (B) The edge removal strategy for the reconstruction of network structure. Setting the gene expression level of X_1 to zero will eliminate the direct regulation of gene X_1 on gene X_2 . (C) The structure diagram of the MISA model. The green and blue round dots indicate genes, the arrows indicate positive regulation, and the blunt arrows indicate negative regulation. Here, genes X_1 and X_2 are symmetrical, activating themselves and inhibiting each other. (D) Reconstruction of the network structure when the diffusion constant is 0.0004. The parameters are set as $a = 0.5$, $b = 0.8$. In this case the system has two stable states. The dark blue (green) and light blue (green) lines indicate the trajectory of gene X_2 (X_1) after and before edge removal, individually. (E) The oscillation network for repressilator. The oscillation network is a cyclic negative-feedback loop composed of three genes. (F) Inference of the interaction regarding gene $LacI$ in the oscillation network. If we delete a non-existent interaction (for example, the regulation of $LacI$ on CI), the target variable (CI) still maintains oscillation, while if we remove an existing interaction (the regulation of $LacI$ on $TetR$), the target variable ($TetR$) no longer exhibits an oscillation pattern.

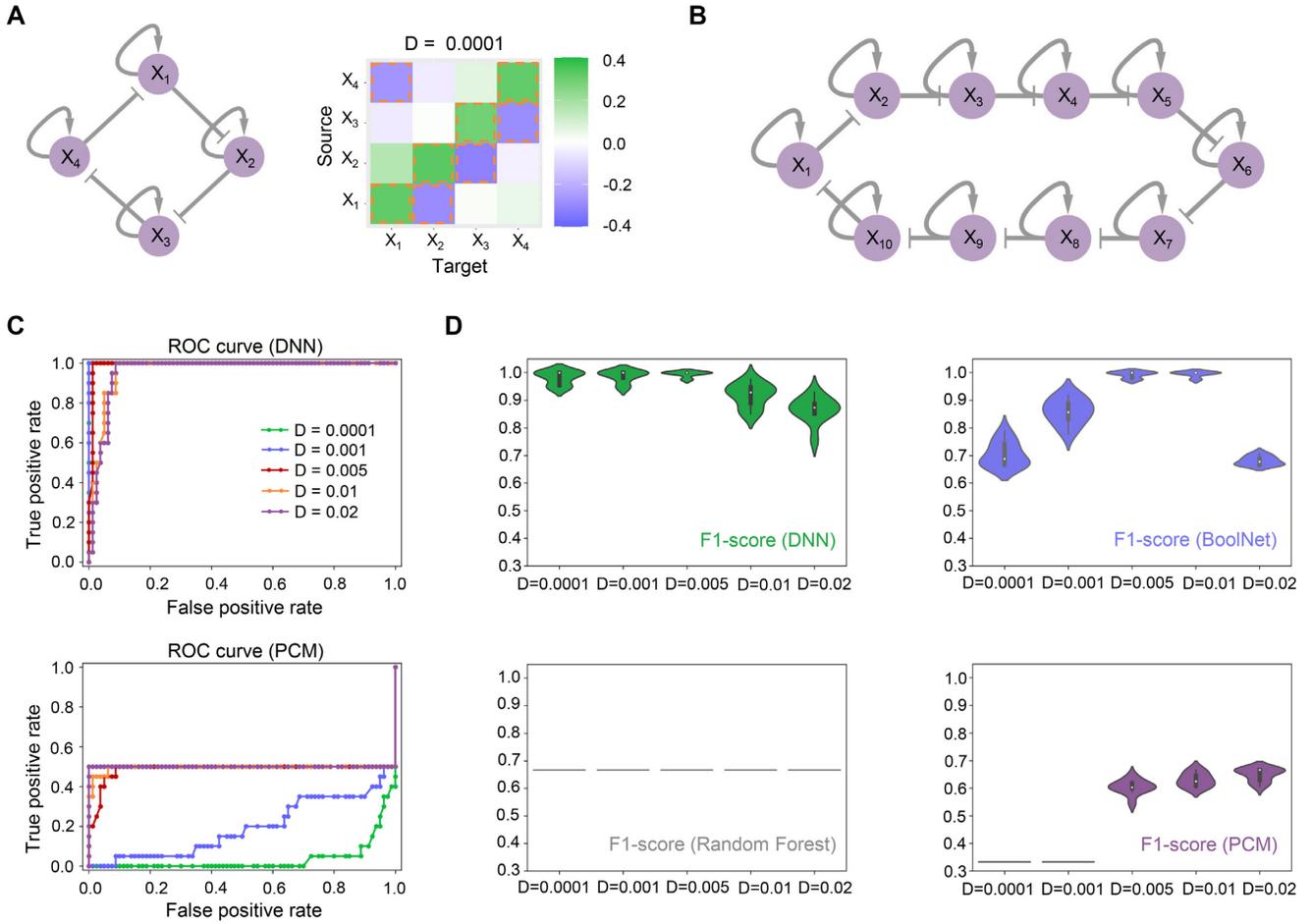


Figure 2. The comparison of different methods for network inference. (A) Structure diagram of the four-dimensional gene network system, and the prediction result of DNN for network inference under the noise level $D = 0.0001$ (right panel), where the vertical axis represents the source of regulation, and the horizontal axis represents the target of regulation. Here, green boxes (blue) indicate positive (negative) regulations and the darker (lighter) the color, the stronger (weaker) the regulation is. The yellow boxes represent the predicted interaction, which are fully consistent with the network structure on the left. (B) Structure diagram of the 10D gene network system. (C) ROC curves of DNN and PCM at different noise levels. (D) F1-score statistics of DNN, BoolNet, Random Forest and PCM under different noise levels (10 trials, the data set used by each trial contains 3600 time series).

Estimation of the diffusion coefficient D

Consider a n -dimensional Langevin equation:

$$\frac{dx(t)}{dt} = F(x(t)) + \Gamma(t), \quad (4)$$

where $F(x(t))$ represents the driving force of the system and $\Gamma(t) = (\Gamma_1(t), \Gamma_2(t), \dots, \Gamma_n(t))^T$ is n -dimensional independent Gaussian white noise, satisfying $E[\Gamma_i(t)] = 0$ and $E[\Gamma_i(t)\Gamma_j(t')] = 2D\delta_{ij}\delta_0(t - t')$ for arbitrary t and t' . Only when $i = j$, $\delta_{ij} = 1$, otherwise $\delta_{ij} = 0$. $\delta_0(t - t')$ is Dirac Delta function. So the Equation (4) can be transformed into:

$$dx(t) = F(x(t))dt + \sqrt{2D}dW(t), \quad (5)$$

where $W(t)$ is the n -dimensional standard Brownian motion. With the Itô formula, Equation (5) can be written as:

$$dx(t) = F(x(t))dt + \sqrt{2D}\sqrt{dt}Z, \quad (6)$$

where $Z = (Z_1, Z_2, \dots, Z_n)$ and $Z_i \sim N(0, 1)$ ($i = 1, 2, \dots, n$). Then we can estimate the diffusion coefficient by calculating the mean squared displacement from data (40). Specifically,

supposing the n -dimensional time series data $x_{t_1}, x_{t_2}, \dots, x_{t_m}$ follow the stochastic differential Equation (6) with time step Δt , we have:

$$x_{t_{i+1}} = x_{t_i} + \Delta t F(x_{t_i}) + \sqrt{2D\Delta t}\xi, \quad (i = 1, 2, \dots, m) \quad (7)$$

where ξ is a standard normal random vector. With x_{k,t_i} representing the k th component of x_{t_i} , the Equation (7) implies that $\frac{x_{k,t_{i+1}} - x_{k,t_i}}{\sqrt{2D\Delta t}} - \sqrt{\frac{\Delta t}{2D}}F(x_{t_i})$ is a standard normal random variable for all i ($i = 1, 2, \dots, m - 1$). Assuming the time step Δt is infinitesimal, $\frac{x_{k,t_{i+1}} - x_{k,t_i}}{\sqrt{2D\Delta t}}$ can be considered as a standard normal random variable for all i ($i = 1, 2, \dots, m - 1$). Using the sample variance to approximate the true variance, we can obtain:

$$\frac{1}{2\hat{D}_k\Delta t(m-1)} \sum_{i=1}^{m-1} (x_{k,t_{i+1}} - x_{k,t_i})^2 = 1, \quad (8)$$

$$\hat{D}_k = \frac{1}{2\Delta t(m-1)} \sum_{i=1}^{m-1} (x_{k,t_{i+1}} - x_{k,t_i})^2, \quad (9)$$

where \hat{D}_k is the estimation of D . We can get n estimations of D ($\hat{D}_k, k = 1, 2, \dots, n$). Thus, we can take the average of \hat{D}_k as the final estimation of D :

$$\hat{D} = \frac{1}{n} \sum_{k=1}^n \hat{D}_k. \quad (10)$$

DNN-PSCA for calculating energy landscape

The dynamical system described by Eq. (5) is composed of n -coupled differential equations. To obtain the energy landscape, one way is to solve diffusion equations describing the probability evolution and obtain the steady state probability distribution. Previously, we have developed a self-consistent mean field approach (1,15), where we split the probability into the products of individual ones: $P(x_1, x_2, \dots, x_n, t) \sim \prod_i^n P(x_i, t)$. Following a weak noise approximation, the probability distribution can be approximated by a Gaussian distribution (41,42):

$$P(\mathbf{x}, t) = \frac{1}{(2\pi)^{n/2} |\sigma(t)|^{1/2}} \times \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{x}^s(t))^T \sigma^{-1}(t) (\mathbf{x} - \mathbf{x}^s(t)) \right\}, \quad (11)$$

where, $\mathbf{x}^s(t)$ and $\sigma(t)$ are the first two moments of the Gaussian distribution, i.e., the mean and the covariance, which can be calculated by (41,42):

$$\frac{d\mathbf{x}^s(t)}{dt} = F(\mathbf{x}^s(t)), \quad (12)$$

$$\frac{d\sigma(t)}{dt} = \sigma(t)A^T(t) + A(t)\sigma(t) + 2D. \quad (13)$$

Here, $\sigma(t)$ denotes the covariance matrix in time t , and the matrix elements of A is $A_{ij} = \frac{\partial F_i(\mathbf{x})}{\partial x_j(t)}|_{\mathbf{x}=\mathbf{x}^s(t)}$, where \mathbf{x}^s is the steady-state value of the system. With F determined by the DNN model, and diffusion constant D estimated from time series data, we can solve Equations (12) and (13) to obtain the first and second moments, and further calculate the steady state probability distribution from Equation (11).

The probability obtained above corresponds to a basin of attraction. If the system displays multiple steady states, there should be several probability distributions localized at every basin with different covariance. Therefore, the total probability is the weighted sum of all these probability distributions. For example, the MISA model has two attractors when the parameters are chosen as: $a = 0.5, b = 0.8$, so the probability distribution takes the form: $P_{ss}(\mathbf{x}) = w_1 P_1(\mathbf{x}) + w_2 P_2(\mathbf{x})$, where the weighting factors (w_1, w_2) represent the relative size of different basin of attraction, and $w_1 + w_2 = 1$. Finally, once we have the total probability, we can construct the potential landscape by: $U(x) = -\ln P_{ss}(\mathbf{x})$ (1,15). Of note, the difference between PSCA and DNN-PSCA is that PSCA uses an explicit expression for F , whereas DNN-PSCA models F using DNN from the noisy data, which

therefore provides a data-driven approach to calculate the energy landscape for gene networks.

RESULTS

Idea illustration for network inference using DNN

We first illustrate the basic idea of using DNN to infer the structure of regulatory networks. Consider a n -dimensional stochastic differential equation, describing the dynamics of a n -dimensional regulatory network, the i th equation can be described as:

$$\frac{dx_i(t)}{dt} = f_i(x_1(t), x_2(t), \dots, x_n(t)) - k_i x_i(t) + \xi_i(t), \quad (14)$$

where $x_i(t)$ ($i = 1, 2, \dots, n$) represents the state value of the i th variable at time t , f_i is a function accounting for the synthesis of x_i , k_i is the degradation rate, and $\xi_i(t)$ represents external noise, which is assumed as gaussian white noise (43–45).

It follows that the structure information of the network is contained in f . Instead of modeling f with explicit form, which may be hard to comprehend, we use DNN to model the properties of f from the time series data. The DNN model we employ here is composed of one input layer, two hidden layers (each with 32 nodes), and one output layer (using three hidden layers gives similar results, Supplementary Figure S5). Regarding the choice of nonlinear function in DNN, the ReLU function is used for the activation function of the two hidden layers, while the sigmoid function is used for the output layer, limiting the state value to 0–1. For the system described by Equation (14), the number of input nodes is n , corresponding to the state value of n variables at time t , and the number of output nodes is also n , representing f_1, f_2, \dots, f_n . We can obtain the time-evolving trajectory $\hat{\mathbf{x}}$ which includes n variables (Figure 1A) via the following discrete iterative equation:

$$\begin{aligned} \hat{x}_i(t + \Delta t) &= \hat{x}_i(t) + f_i(\hat{x}_1(t), \hat{x}_2(t), \dots, \hat{x}_n(t))\Delta t \\ &\quad - k_i \hat{x}_i(t)\Delta t \quad (i = 1, 2, \dots, n), \end{aligned} \quad (15)$$

where $\hat{x}_i(t_j)$ is the i th element of vector $\hat{\mathbf{x}}(t_j)$ at time t_j . The average of the absolute difference between the model trajectory $\hat{\mathbf{x}}$ and the target trajectory \mathbf{x} (time series data) is used as loss function for training, i.e., $Loss = \frac{\sum_{i=1}^n \sum_{j=1}^T |x_i(t_j) - \hat{x}_i(t_j)|}{nT}$ (T is the total number of time points).

Here, we choose multilayer perceptron model for training the DNN model, which is simple but powerful as we will show later. Generally, we do not know the value of the degradation rate k , so we use the ensemble of multiple models trained at different k values. From Eq. (14), the regulatory relationship between variables can be inferred from the monotonicity of f_i . Specifically, if f_i is a monotonically increasing (decreasing) function on X_j , then X_j has an activation (inhibitory) effect on X_i . In the case of low-dimensional system, DNN can learn the monotonicity of f_i well (Supplementary Figure S1B). However, for more complex system, the learned f_i is not necessarily monotonous, but has a parabolic shape (Supplementary Figures S4B and S5B). So we introduce an edge removal strategy to infer the network structure (25). We assume that there exist interactions between any node pair, and the strength of the interaction

is represented by the network connection weight. By setting the input of certain interaction to zero (Figure 1B, see Materials and Methods for details) for simulating the removal of the corresponding regulation, we can observe the change in the value of relevant variables, which will provide the information about the type of regulations. The remarkable change in the value of relevant variables indicates the existence of regulatory effect. Specifically, if the value of the target variable increases (decreases) significantly after an edge removal operation, the corresponding edge is inferred as a positive (negative) regulation.

Rebuilding the structure of multistable system and oscillation system

To see how the DNN works for inferring the network structure, we first study a simplified two-variable model called mutual inhibited self-activation (MISA) model (Figure 1C). The MISA model consists of mutual inhibition between two opposing fates controlled by two transcription factors X_1 and X_2 , which has been shown to govern cell fate decision and commitment in multiple instances of multipotent stem or progenitor cells (20,46–48). In biological modeling, Hill functions are often used to describe the activation and inhibition regulations:

$$\frac{dx_i}{dt} = \sum_{j=1}^n \frac{A_{ji} \times x_j^{n_h}}{S_{ji}^{n_h} + x_j^{n_h}} + \sum_{j=1}^n \frac{B_{ji} \times S_{ji}^{n_h}}{S_{ji}^{n_h} + x_j^{n_h}} - k_i \times x_i + \xi_i. \quad (16)$$

Here, S_{ji} represents the threshold of the sigmoidal function, n_h is the Hill coefficient which determines the steepness of the sigmoidal function (49), k_i is the degradation rate of x_i , ξ is Gaussian white noise, whose autocorrelation function is $\langle \xi_i(\mathbf{x}, t), \xi_j(\mathbf{x}, 0) \rangle = 2D\delta(t)$, and D is diffusion coefficient matrix. In addition, A_{ji} is the activation constant, which measures the strength of the activation of X_j on X_i , and B_{ji} is the inhibition constant, which measures the strength of the inhibition of X_j on X_i . For example, for the MISA model, A_{11} represents the intensity of the activation of the regulatory factor X_1 on itself, and B_{21} represents the intensity of the inhibitory effect of the regulatory factor X_2 on the regulatory factor X_1 . Here, we choose a symmetrical case for the MISA model, namely the parameters are set as: $a = A_{11} = A_{22}$, $b = B_{12} = B_{21}$ and $k = k_1 = k_2$. With fixed parameter values $S = 0.5$, $n_h = 4$, and $k = 1$, as a and b vary, the system will display different behavior of multistable states (20).

In a sense, the structure of a gene regulatory network determines the dynamical trajectory of gene expressions. The artificial neural network framework here seeks to infer the network structure by first learning f from the trajectory change of gene expressions. For the MISA model, the input of the neural network is the gene expression of transcription factors X_1 and X_2 at time t and the parameter a and b , and the output is f_1 and f_2 . The parameters a and b are used as input to the model, so that we can predict the dynamics of the system for changing a and b .

Considering the real data is noisy, here we first generate stochastic trajectory (each time series has 15 data points and diffusion constant D is 0.0004) from stochastic ODEs

(Eq. 16) for the training of DNN. Results suggest that the DNN displays a good performance for replicating data generated by the ODE model. Essentially, for the final steady state value of gene expression under different parameters, the stable point value predicted by the DNN and corresponding weight of reaching each stable point are consistent with the simulation results from original ODE model (Supplementary Figure S1A). These results suggest that the DNN model can model the dynamical characteristics of multistable systems. Further, by interrogating the DNN model trained from the data, we can infer causality between variables (see Materials and Methods). For example, by setting the gene expression value of X_1 at time t to zero (Figure 1B), we can simulate the expression trajectory of X_2 for removing the regulation from X_1 to X_2 , by calculating f_2 . As shown in the right side of Figure 1D, the expression value of gene X_2 is significantly increased after removing the regulation of X_1 on X_2 (blue curves), so it can be concluded that gene X_1 has a negative regulatory effect on gene X_2 . With the parameter $a = 0.5$ and $b = 0.8$, the system has two stable steady states. Under this set of parameters, we can correctly infer all regulatory effects from both X_1 (including the self-activation of X_1 and the inhibition of X_1 on X_2 , Figure 1D), and X_2 (Supplementary Figure S2). When the system displays one, three or four stable states (corresponding to different parameter values), we can still correctly rebuild the structure of the network (Supplementary Figure S2). These results support that the DNN can accurately infer the network structure of multistable MISA model.

We further test how this approach works in an oscillation system. We choose the repressilator as an example (50). As shown in Figure 1E, LacI inhibits the transcription of gene TetR, TetR inhibits the expression of gene CI, and CI inhibits the expression of LacI. The negative feedback loop will lead to an oscillation behavior. The dynamics of the system is described by six coupled first-order differential equations:

$$\begin{aligned} \frac{dm_i}{dt} &= -m_i + \frac{\alpha}{1 + p_j^{n_h}} + \alpha_0, \quad i = LacI, TetR, CI \\ \frac{dp_i}{dt} &= -\beta(p_i - m_i), \quad j = CI, LacI, TetR \end{aligned} \quad (17)$$

where p_i is concentration of the repressor-protein, and m_i is the concentration of corresponding mRNA. The number of protein copies per cell produced from a given promoter type during continuous growth is α_0 , in the presence of saturating amounts of repressor (owing to the ‘leakiness’ of the promoter), and $\alpha + \alpha_0$ in its absence. β denotes the ratio of the protein decay rate to the mRNA decay rate, and n_h is a Hill coefficient. We add noise to the protein concentration, and the noise level is $D = 0.004$. Our results show that the DNN can not only predict temporal oscillation, but also obtain the potential network structure behind the time series data (Figure 1F). If we delete a non-existent interaction (e.g. the regulation of LacI on CI), the target variable (CI) still maintains oscillation, while if we remove an existing interaction (the regulation of LacI on TetR), the target variable (TetR) no longer exhibits an oscillation pattern, indicating that the DNN can model this oscillation system well and has the potential to reconstruct the network for an oscillation system (Figure 1F, also see Supplementary Fig-

ure S3 for similar results considering the regulations from TetR and CI).

DNN performs better than other methods for network inference

To further evaluate the effectiveness of the DNN, we make comparisons between DNN and other network inference methods. Here, we define $C_{ij} = x_j^s - x_{i,j}^s$, where $x_{i,j}^s$ (x_j^s) is the steady state value of X_j after (before) removing the edge from X_i to X_j . So, C_{ij} provides a quantitative measure for the amount of change in the steady state value of the variable X_j after knocking out the regulation of X_i on X_j . For better accuracy, in the following work, we use the steady state change C_{ij} as the major measure for network structure inference, and the change of gene expression trajectory is used for an auxiliary verification. Specifically, when the variable trajectory before the interaction is removed is always above (or below) the variable trajectory after the interaction is removed, the interaction is inferred as a positive (negative) regulation.

We further apply DNN to a four-dimensional gene network (Figure 2 A), which is also modeled using Hill functions (Equation 16). This network contains four variables (denoted as X_i , $i = 1, 2, 3, 4$) and eight interactions, including the self-activation of each variable and four inhibitory interactions, forming a feedback loop. When the parameter values are chosen as: $a = 0.35$, $b = 0.3$, $S = 0.2$, $n_h = 4$, and $k = 1$, the system displays seven stable points. We first generate simulation data based on this four-dimensional dynamical model (each time series has 15 data points), and then train DNN to get f with these simulation data. In this way, we actually obtain a dynamical system model expressed by DNN, although not with explicit form of f functions. We find that this DNN model is able to replicate all the results from original ODE models. For example, the values of stable points predicted by the DNN model and corresponding weight of reaching each stable point are in line with the simulation results from original ODE model (Supplementary Figures S4A and S5A). Based on the DNN model, using the edge-removal approach, we can further infer the network structure. As shown in Figure 2 A (right), the positive (negative) value indicates activation (inhibition), and the greater (smaller) the absolute value, the stronger (weaker) the regulation is predicted. Here, the vertical axis represents the source of regulation, and the horizontal axis represents the target of regulation. For example, when the noise level $D = 0.0001$, the DNN can accurately predict the network structure (see Supplementary Figure S6 for other noise level D). It is worth noting that the prediction of interaction depends on the choice of the threshold. However, the values of C_{ij} are linearly separable when noise level D is not large, so the selection of the threshold has a minor effect for the inference of the interaction (Figure 2 A, Supplementary Figures S4C, S5C and S6).

Next, we apply the DNN to a more complex system, which is a ten-dimensional gene regulatory network. We further compare DNN with other methods, including traditional Random Forest (11) and Boolean network (33,34), as well as the latest PCM (13) (Figure 2 B). We run 10 tri-

als under each different noise level, and the data set used by each trial contains 3600 time series and each with 20 data points (except for the time series used by PCM for which each time series contains 200 data points). When the parameter values are chosen as: $a = 0.5$, $b = 0.8$, $S = 0.2$, $n_h = 4$, and $k = 1$, the system possesses over 100 stable points. Even for such a complex system, the DNN method can still model the system dynamics and infer the causality well from small amounts of data (see Supplementary Figure S7 for one trial result). The true positive (TP) is the number of predicted interactions matching the true structure, the false positive (FP) is the number of interactions predicted but not present in the true structure, the false negative (FN) is the number of unpredicted interactions in the true structure, and the true negative (TN) is the number of unpredicted interactions matching the true structure. It is worth noting that other methods only need to match the interaction with the correct structure, regardless of the type of interaction, while for DNN the interaction type (activation or inhibition) must also match. The true positive rate (TPR) is defined as $\frac{TP}{TP+FN}$, the false positive rate (FPR) as $\frac{FP}{TN+FP}$ and $F1$ -score as $\frac{2*TP}{2*TP+FN+FP}$. We can further get the receiver operating characteristic (ROC) curve (Figure 2C). With the increase of noise level, the DNN method still maintains a high accuracy rate in causality inference even with only a small amount of data. Meanwhile, the PCM does not give good enough performance at different noise levels (Figure 2 C).

Since this ten-dimensional gene network is a multistable system, there exist multiple stable attractors in this system. This may explain why the correlation-based PCM does not perform well when the noise level is small, since correlation-based methods usually require oscillating data. We perform statistical analysis on the results of all experiments via $F1$ -score (Figure 2D). Consistent with the ROC curve, $F1$ -score shows that PCM does not perform well under small noise, and for all noise considered, the DNN displays highest accuracy for the network inference (Figure 2 D). Only at $D = 0.005$ and $D = 0.01$, BoolNet performs closely as DNN does, but we need to emphasize that BoolNet cannot identify the type of interaction. Additionally, when the noise level increases to $D = 0.02$, the prediction performance of BoolNet drops rapidly, while the performance of DNN does not decrease significantly, indicating that DNN has better robustness for structure inference. In general, the DNN shows best performance in the accuracy and robustness of network structure inference among all methods considered. It's important to point out that the DNN can not only predict the type of interaction, but also predict the self-regulation, which is usually not available for other methods mentioned above.

Reconstructing networks from real-world data

DNN is shown to perform well for network inference for the simulated data. We hope to see how it works on structure inference from real-world data. We first consider a food chain network composed of three organisms: *Picophytoplankton* (P), *Rotifers* (R) and *Cyclopooids* (C) with the predator-prey relations (Figure 3A). The oscillatory population data come

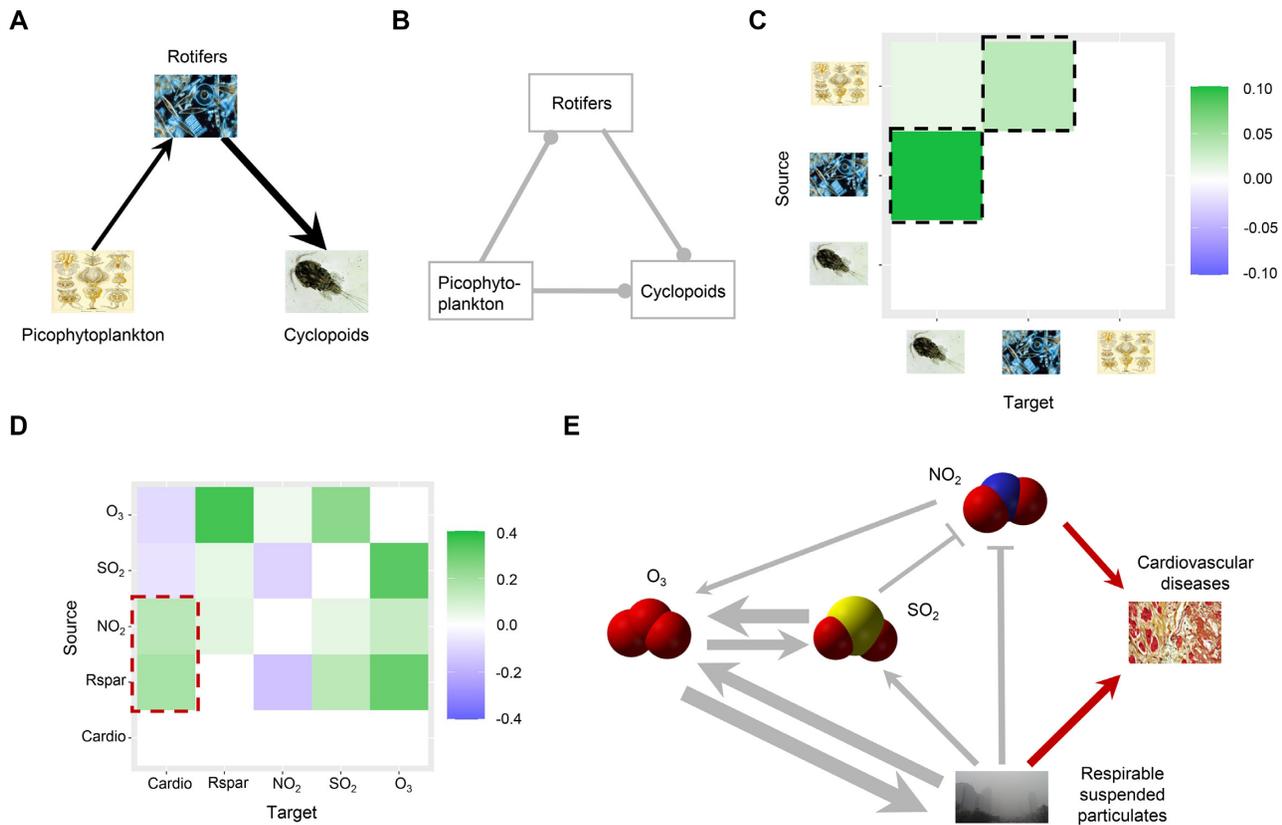


Figure 3. Reconstructing networks from real-world data. (A) A food chain network composed of three types of plankton, where the thickness of the arrows represents the food preferences of the species and the direction of each black arrow represents the interaction between prey and predator. (B) The prior structure used in DNN. Arrows ending with a dot characterize the interaction with directions but without the interaction type. (C) The interaction predicted by DNN, where the square marked by the black box are consistent with the black arrow in (A). (D) The results of all interactions between air pollutants and cardiovascular diseases predicted by DNN. (E) The reconstructed network based on (D), where the thickness of the arrow indicates the intensity of interaction. The red arrows highlight the causal interactions on cardiovascular diseases, which correspond to the red boxes in (D).

from a plankton community isolated from the Baltic Sea which is cultured for more than eight years under constant external conditions (13,51,52). For network inference, a critical issue is how to distinguish the direct causal links from the indirect ones, i.e., how to avoid misidentifying indirect causations as direct ones (13). We seek to test whether DNN can distinguish the direct causal interaction (P to R and R to C), from indirect causal interaction (i.e. the link from *Picophytoplankton* to *Cyclopooids*). We do this by first introducing the prior structure with both direct and indirect interactions among R, P and C (Figure 3 B). Then we train DNN to perform the network structure inference. It is worth noting that each time series we use in this example has only 10 data points (see Supplementary Text and Supplementary Table S1 for parameter settings). Remarkably, the DNN not only successfully predicts the promoting effect of *Picophytoplankton* on *Rotifers* and the promoting effect of *Rotifers* on *Cyclopooids*, but also eliminates the link from P to C in prior structure, that is, the DNN predicts that there is no direct causal interaction from *Picophytoplankton* to *Cyclopooids* (Figure 3C), which is consistent with the ground truth structure (Figure 3A). These results indicate that the DNN model has good ability to distinguish direct interactions from indirect interactions. Additionally, the DNN can also model the food preferences of the species

(the black square in Figure 3C). Here, deeper color corresponds to stronger interaction, which is consistent with the biological knowledge (the thickness of the arrows in Figure 3A represents the food preferences) (51).

Our second real-world example is from the data of hospitalization records of air pollution and cardiovascular diseases in Hong Kong from 1994 to 1997 (53–55). In this case, we also introduce a prior structure, that is, the number of people suffering from cardiovascular disease has no effect on pollutants, and pollutants have little effects on themselves. According to the prediction of the DNN model (each time series we use in this example has 20 data points, and see Supplementary Text and Supplementary Table S1 for parameter settings), only two pollutants, namely nitrogen dioxide and respiratory suspended, are found to be the main cause of cardiovascular disease (Figure 3D and E). Neither sulfur dioxide nor ozone is identified as the cause of the disease, which is in line with previous studies (13,56,57). Due to the influence of other factors (such as temperature, humidity, wind speed, etc.), the causal relations of pollutants warrant further verifications. In summary, the DNN displays great predicting ability for causality inference for real-world data, with a great performance in distinguishing direct causal interactions from indirect causal interactions.

Reconstructing the developmental network from single cell data

Inferring the structure of gene regulatory networks from single cell gene expression data is a challenging problem in computational systems biology. We seek to tackle this problem using DNN. Here we have two major purposes. The first one is to test how the DNN performs in network inference based on single cell data. The second one is to explore the underlying mechanism of cell fate decisions in development based on single cell data. We obtain the data set describing the expression of 48 genes during preimplantation development of mice from the 1-cell stage to the 64-cell stage (35). During this period, the cells make two fate decisions. One is the differentiation of cells (16-cell stage) into two different cell lineages: trophoblast (TE) and internal cell mass (ICM). The other is the differentiation of ICM cells (32-cell stage) into primitive endoderm (PE) and ectoderm (EPI). Our first step is using DNN to infer the network structure of mouse development. We first focus on a small-scale network composed of four genes, NANOG, GATA6, OCT4 and CDX2, by using the regulatory relationship in the literature (58) as a benchmark structure.

Here, a major challenge is that for single cell data we do not have well-defined time series data. What we have is total 442 cells (data points) at 7 time points. To generate suitable time series data for DNN training, we randomly combine the normalized data from different time points without exploiting developmental lineage, while the reconstruction of developmental lineage is regarded as a measure of model accuracy which will be described in the next section. We divide the data set as three data sets for training (70%), validating (15%) and testing (15%), respectively. The regulatory relationship between genes at different stages is not static, but changes over time (34). Considering the heterogeneity among cells and the influence of noise, we consider the ensemble of several models for improving the performance of causal inference. We train 15 models each with different k values ($k = 0.1, 0.2, \dots, 1.5$). Different approaches have been proposed for model ensemble (59,60). Here, we use majority voting strategy to ensemble the 15 models. We denote the classifiers as h_1, h_2, \dots, h_n , respectively, and $h_i(x)$ represents the classification result of the classifier h_i on the interaction x . The set of $h_i(x)$ value is $\{1, -1, 0\}$, where 1 (-1) means that the interaction x exists and is an activation (inhibition), and 0 means that the interaction x does not exist. Then, we have:

$$H(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^n |h_i(x)| > \lambda n \text{ and } \sum_{i=1}^n h_i(x) \geq 0, \\ -1, & \text{if } \sum_{i=1}^n |h_i(x)| > \lambda n \text{ and } \sum_{i=1}^n h_i(x) < 0, \\ 0, & \text{otherwise,} \end{cases} \quad (18)$$

where λ is 0.5 by default, but can be adjusted according to different tasks. For example, in order to reduce the redundancy of the network, we set $\lambda = 0.65$ to remove relatively unnecessary links.

As shown in Figure 4A, the left panel is the benchmark structure, and the middle panel is the structure predicted by the DNN. Among them, black arrows represent the consistent interactions between the two structures and red arrows

represent links in the same direction but different type, indicating a high degree of similarity. We need to emphasize two points. The first one is that the benchmark structure (Figure 4A, left panel) is by collecting experimental evidences from previous studies, while the structure predicted by the ensemble model is completely inferred from single cell data, i.e., the two structures are totally from different types of source. This could be a potential reason for the inconsistent links between the predicted structure and the benchmark structure. The second one is that for the structure reconstruction, here the ensemble DNN model predicts both the interaction directions and the interaction types (activation or repression), which are important for the gene network dynamics.

To evaluate whether our inferred network structure is biologically implementable, here we model the regulation network dynamics by using Hill functions to characterize gene regulations. Specifically, we aim to verify whether the model based on Hill function can match the single cell data. As shown in the right panel of Figure 4A, the predicted network can be successfully converted to a Hill function model (Equation 16) via searching for parameters through Bayesian optimization (see Supplementary Text for details), and the corresponding parameters of the Hill function are $a = 0.8787$, $b = 0.0248$, $S = 0.6111$, $n_h = 4.871$ and $k = 0.4104$. Here, the landscape is derived from the corresponding Hill function model, and the points are single cell data (Figure 4A). One of the major challenges in gene network inference is extensive redundancy of regulatory interactions in real-world networks. Therefore, we raise λ to 0.65 to remove some relatively unnecessary links. The de-redundant regulation network is also successfully transformed into a Hill function model (Supplementary Figure S8).

In this sense, we consider the DNN performing very well in this task, indicating that DNN is a very promising way for the reconstruction of small-scale gene networks based on single cell data. For the whole 48-dimensional network, it is hard to rebuild the whole gene network due to limited number of single cell data. Alternatively, we resort to modeling the system dynamics with DNN, and with that we can explore the important questions usually only available with dynamical equations in hand, such as the mechanisms of cell fate determination.

Rebuilding the dynamics of developmental network from single cell data

To study the dynamical mechanisms of cell fate decision in mouse development, we employ the energy landscape theory, which has been proposed to study stochastic dynamics of gene regulatory network in different biological systems (1,7,15). The potential energy of the system can be calculated via $U(x) = -\ln(p_{ss}(x))$, where $p_{ss}(x)$ is the steady state probability density of the gene expression variable x . Specifically, we use the data of all 48 genes to obtain an ensemble model via DNN. With the increase of k , the validation loss of the model first decreases and then increases (Supplementary Figure S9A), so we select several models with relatively small validation loss (the threshold is chosen as 0.12) for ensemble ($k = 0.3-1.15$, with interval 0.05). For calculating the probability distribution $p_{ss}(x)$ and the en-

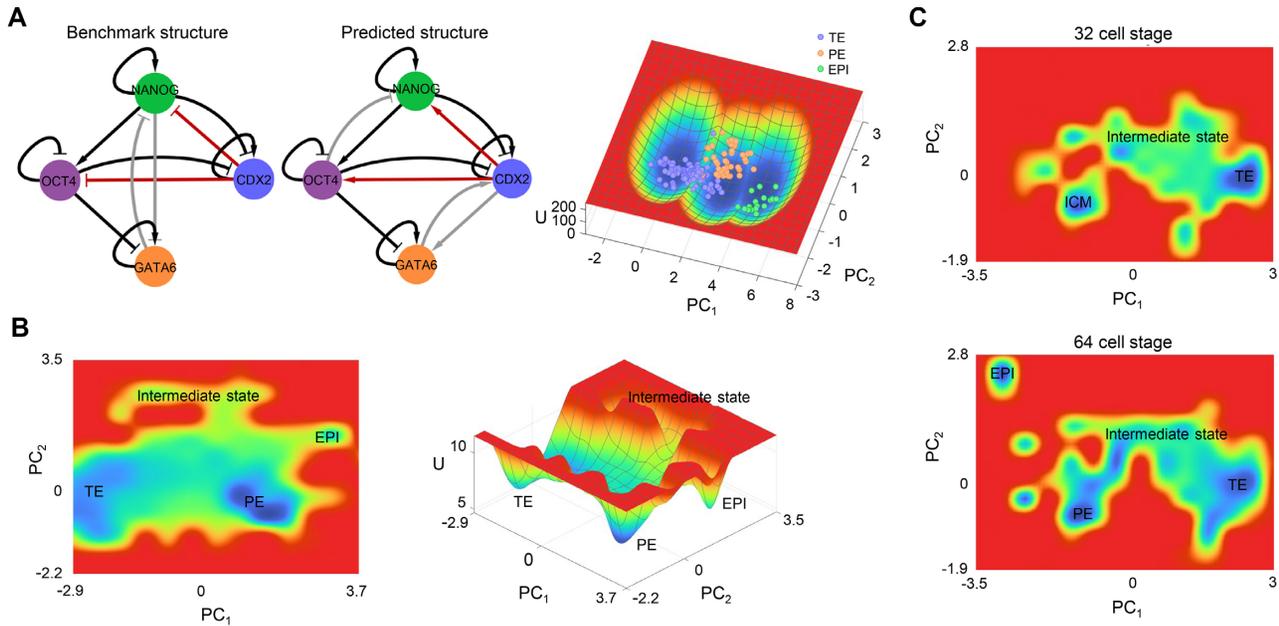


Figure 4. Inference of structure and energy landscape from single cell data. (A) The reconstruction of a small-scale gene networks from single-cell data. The left panel is the benchmark structure, and the middle panel is the structure predicted by the DNN. Black arrows denote the consistent interaction of the two structures, red arrows denote the links in the same direction but different type. The right panel shows the validation of the predicted structure, as its corresponding Hill function model closely matches the single cell data, where the landscape is derived from the corresponding Hill function model, and the points are single cell data. (B) The two-dimensional and three-dimensional energy landscape for the 48-dimensional model of mouse development constructed from single-cell data (all stages) via DNN. Four attractors (characterizing cell types) including EPI, TE, PE and intermediate state emerge on the landscape. (C) The two-dimensional landscapes at 32-cell and 64-cell stage modeled by DNN.

ergy landscape $U(x)$, principal component analysis (PCA) is used to reduce the dimensionality of the system. Then, by collecting the statistics of data points in two-dimensional plane, we acquire the probability distribution for all cell stage (Figure 4B).

By making comparisons with experimental data (35), we can correspond the attractors on the landscape to cell phenotypes. Interestingly, except the TE, PE and EPI phenotype, the DNN model also identify an intermediate cell state (Figure 4B). The identified cell types are supported by the coefficient of PC coordinates in reduced dimensions (Supplementary Figure S9C). In PC_1 , the coefficients of the four markers of TE (Dppa1, Id2, Krt8 and Tspan8) are negative, and the coefficients of the markers of PE (Gata4 and Pdgfra) and EPI (Bmp4, Fgf4, Klf2, Nanog and Sox2) are all positive, which means that PC_1 is a suitable coordinate for distinguishing TE from PE and EPI. In PC_2 , the coefficients of PE markers (Creb312, Gata4 and Pdgfra) are negative, the coefficients of TE markers (Cdx2, Dppa1, Id2, Krt8 and Tspan8) are close to zero, and the coefficients of EPI markers (Fgf4, Klf2, Nanog and Sox2) are very positive, which means that PC_2 is a suitable coordinate for distinguishing PE from EPI. The DNN model also identifies an intermediate state, which is a transitional state which cells tend to go through for cell fate transitions. Specifically, the intermediate state identified displays following characteristics: (i) the expression levels of Dppa1, Id2, Krt8 and Tspan8 (markers of TE cells) in intermediate cells are lower than the expression levels of these genes in TE cells, and higher than the expression levels of these genes in PE and EPI cells. (ii) The expression levels of Fgf4, Klf2 and Nanog (markers of

EPI cells) in intermediate cells are lower than the expression levels of these genes in EPI cells, and higher than the expression levels of these genes in TE and PE cells. (iii) The expression level of Gata4 (marker gene of PE cells) in intermediate cells is lower than the expression level of this gene in PE cells, and higher than the expression level of this gene in TE and EPI cells. Therefore, the intermediate state cells exhibit transitional characteristics, which is also supported by the correlation matrix of all cells (Supplementary Figure S9B).

We further quantify the energy landscape of mouse development at different cell stages. To better simulate the temporal changes of developmental process, we use the experimental data to constrain the initial condition of the established DNN model. Specifically, the initial condition for DNN model is generated by adding Gaussian white noise to the single cell data in 1-cell stage. With these initial conditions and the established DNN model, we can recalculate the temporal trajectory of the mouse developmental system. Similarly, we classify the cells and calculate the coefficients of PC_1 and PC_2 as well as the correlation matrix of the cells for both 32-cell stage and 64-cell stage (Supplementary Figure S10A and B). We obtain the energy landscape at different stages with different cell types marked on the landscape (Figure 4C). Through analysis at the gene expression level, we find that the DNN model reconstructs the developmental lineage well (Supplementary Tables S2 and S3, see Supplementary Text for the characteristics of intermediate states at the 32-cell stage and 64-cell stage), that is, (i) the 16-cell stage cells differentiate into 32-cell stage ICM and TE cells, and (ii) the 32-cell stage ICM cells differentiate into

EPI and PE cells at 64-cell stage, while TE cells do not differentiate.

Identifying the transition path and critical genes in development

With the information about the cell lineage, we manage to generate a better data set to retrain DNN for a more accurate model. Specifically, to consider the missing information between different cell stages, we perform a second-order spline interpolation to add 10 more time points. So, the new data set includes total 13 data points. Using this data set, we train a total of ten models with different k (0.05–0.95, with interval 0.1). Benefitting from the lineage information inferred above, here we combine the validation loss and the clustering indicator Davies-Bouldin Index (DBI) (61) to evaluate the model performance. Considering the data set $\{y_1, y_2, \dots, y_s\}$, with the clustering results described as $C = \{C_1, C_2, \dots, C_k\}$, the DBI is defined as:

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left(\frac{avg(C_i) + avg(C_j)}{d_{cen}(C_i, C_j)} \right), \quad (19)$$

where $d_{cen}(C_i, C_j) = dist(\mu_i, \mu_j)$, μ_i represents the center point of cluster C_i , defined as $\mu_i = \frac{1}{|C_i|} \sum_{y_m \in C_i} y_m$, and $dist(\cdot, \cdot)$ is a measure of the distance between two samples (the Euclidean distance is used here). So, $d_{cen}(C_i, C_j)$ is a measure of the distance between the central points of different clusters. $avg(C_i)$ is a measure of the distance of data points within a cluster, defined as $avg(C_i) = \frac{2}{|C_i|(|C_i|-1)} \sum_{y_m, y_n \in C_i} dist(y_m, y_n)$, so the smaller the DBI, the better the classification is.

Specifically, for each integration model, k -means is used to divide the cells of the 64-cell stage into three clusters to calculate the DBI (Figure 5 A). Here we have ten models for integration. The model 1 is a single model obtained at $k = 0.05$, model 2 is an ensemble model for $k = 0.05$ and $k = 0.15$, and so on. Compared with the other eight ensemble models, the DBI of model 4 and model 5 are lower, indicating that these two ensemble models are more likely consistent with the prior knowledge of lineage information. By further validation of the model for differentiation data from 32-cell to 64-cell stage (Figure 5 B), we find that the model 4 is more in line with the cell lineage tree, that is, ICM cells at 32-cell stage differentiate into EPI and PE cells at 64-cell stage (see Supplementary Figure S11 for the result of model 5). Based on above analysis, we choose model 4 for following analysis. We use this DNN model to simulate the developmental trajectory of 12 000 cells from 16-cell stage to 64-cell stage. We further quantify the kinetic transition paths of cells as the developmental path (Figure 5C), where each path is the average of thousands of corresponding trajectories. The transition paths provide a quantitative description of the dynamical transition process in development, which end at three phenotypes (EPI, TE and PE).

Since we treat the DNN model as a dynamical model, we ask whether it can model biological perturbations, such as knockout of genes. So we perform knockout experiments on the DNN model. We knock out different genes one at a time in the DNN model, observing the changes in cell types at 64-

cell stage to uncover critical genes that govern cell differentiation. Specifically, we remove the connection between one gene and all other genes to achieve the effect of knocking out the corresponding gene. K-means is used to cluster the 64-cell stage cells from the model into two and three clusters, individually, to calculate DBI . If $DBI_2 < DBI_3$ (DBI_2 (DBI_3) represents the DBI calculated when the data is divided into two (three) clusters), the data set tends to be divided into two clusters. The greater the difference between the two values (DBI_2 and DBI_3), the greater the degree of confidence is. Conversely, if $DBI_3 < DBI_2$, the data set tends to be divided into three clusters (for the original model 4 at 64-cell stage, $DBI_3 < DBI_2$). To quantify the results for the total 48 knocking-out experiments, we define the following indicator:

$$Score_{DBI} = \frac{DBI_2 - DBI_3}{DBI_3}. \quad (20)$$

If $Score_{DBI} < 0$, the simulated 64-cell stage cells after gene knockout tend to be divided into two clusters, which means that knocking out corresponding gene may result in the loss of cell cluster, since cells in 64-cell stage should be divided into three clusters, namely PE, TE and EPI. The greater the absolute value of the $Score_{DBI}$, the greater the degree of the confidence is.

To further improve the credibility of DBI, we make verification by analyzing the results in reduced PC dimensions. Specifically, we first select the 11 genes with the largest absolute value of $Score_{DBI}$ among the genes with negative $Score_{DBI}$ (Figure 5D), and then scrutinize the knockout results for these genes in reduced PCA coordinates, which leaves us nine genes whose knockout results in loss of cell cluster (Supplementary Figure S12). Finally, from each set of data of knockout experiments, the cell fate change after knockout is inferred from the gene expression level. Knockout of the genes *Pou5f1*, *Sall4*, *Gata3*, *Nanog* and *Fgf4* will all result in the loss of EPI. This embodies the critical roles of *Pou5f1*, *Sall4* and *Gata3* on stem cell differentiation, which is consistent with experimental studies (62,63). Both *Nanog* and *Fgf4* are markers of EPI (35). Knockout of the gene *Krt8* results in the loss of TE, which is consistent with the fact that *Krt8* is a marker of TE (35). For the remaining three genes, knockout of the genes *Pecam1* and *Msx2* both will lead to the loss of EPI, and knockout of the gene *Msc* leads to the loss of PE and EPI, which are both consistent with biological knowledge. So, these results support that the DNN model is able to replicate major perturbation results in development, and can be treated as a reasonable dynamical model for modeling mouse development.

Quantifying the energy landscape with DNN

The fact that the DNN can successfully describe a dynamical model inspires us to use DNN for other purpose from dynamical model perspective. How to quantify the energy landscape for gene regulatory networks is a challenging problem (15). One way of quantifying the landscape for high-dimensional gene networks is the PSCA method (1,49), which requires an explicit model as well as the value of diffusion coefficient D , quantifying the noise level. In above sections, we have shown that with DNN we

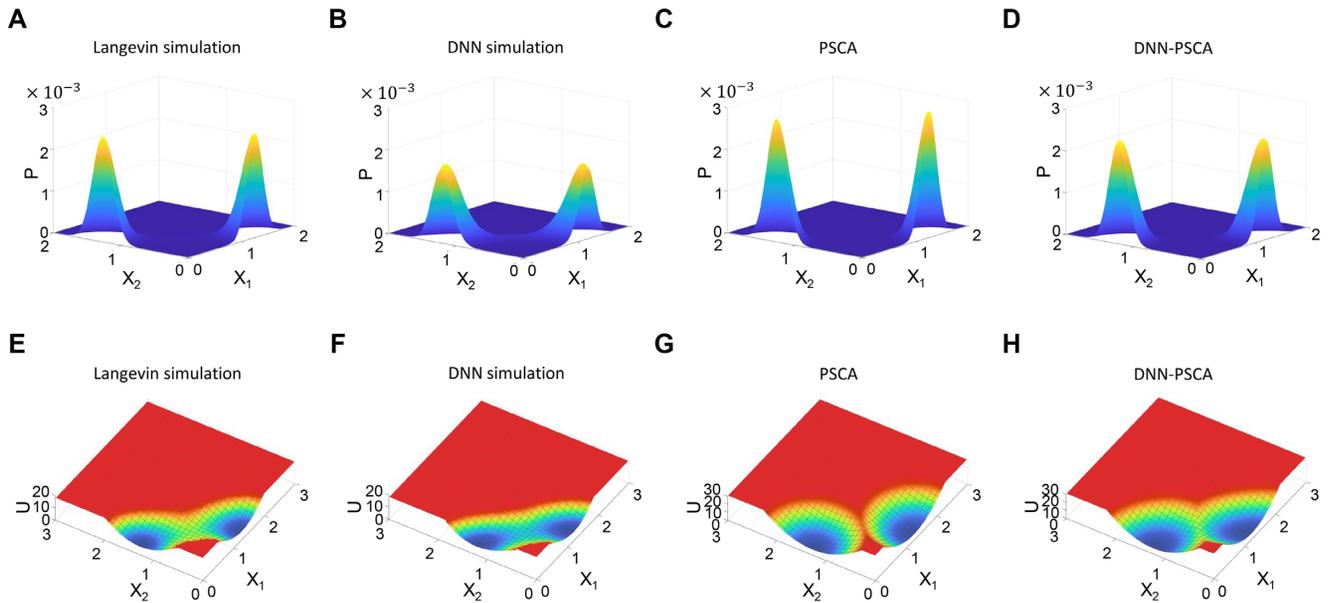


Figure 6. Comparisons among different methods for calculating probability distributions and corresponding energy landscapes. (A–D) The distribution calculated by Langevin simulation, DNN simulation, PSCA and DNN-PSCA. (E–H) The potential landscapes corresponding to (A–D). The distribution calculated from PSCA based on the explicit model is closest to that from Langevin simulation ($d_{L_2} = 0.0053$, $d_{KL} = 0.1121$), followed by DNN-PSCA ($d_{L_2} = 0.0113$, $d_{KL} = 0.1426$) and DNN simulation ($d_{L_2} = 0.0154$, $d_{KL} = 0.2871$).

tion calculated from PSCA (DNN simulation or DNN-PSCA). L_2 measures the distance between two distributions via Euclidean distance. When the deviation of two distributions increases, d_{KL} will also increase, with $d_{KL} = 0$ meaning two equal distributions.

The comparison results (Supplementary Table S4) show that both from L_2 distance and K–L divergence, that the probability distribution calculated from PSCA is most similar to that from Langevin simulation ($d_{L_2} = 0.0053$, $d_{KL} = 0.1121$), while DNN-PSCA also gives a good performance for estimating the probability distribution ($d_{L_2} = 0.0113$, $d_{KL} = 0.1426$). It should be noticed that the PSCA method require an explicit form of a dynamical model, while the DNN-PSCA method is only based on the time-series data. These results demonstrate that the DNN may provide an effective data-driven approach for quantifying the energy landscape.

DISCUSSION

The reconstruction of gene regulatory networks is critical to understanding the underlying mechanisms of cellular processes. Gene regulatory networks can be constructed by mining the literatures, which may be time-consuming and inefficient, whereas reverse engineering provides a challenging but fascinating way for inferring gene regulation networks. Many approaches have been developed to reconstruct gene regulatory networks from the data, however, it remains challenging to accurately infer the direction and type of interactions, especially when the network is involved with self-regulations and feedbacks. It is also vital to develop an approach to distinguish direct causal interactions from indirect causal interactions (13). In this work, we seek to address these issues with DNN. We illustrate the effec-

tiveness of DNN in network inference with various examples.

DNN displays multiple advantages for network inference and simulating unidentified dynamical systems. Firstly, the DNN displays good accuracy and robustness in network inference for various systems. In our first four examples (Figures 1 and 2), we study two completely different systems modeled by differential equations, i.e., multistable systems and oscillatory systems. In the next three examples (Figures 3 and 4A), the networks inferred by DNN with only a small amount of real data are well supported by the prior knowledge and previous studies. In the example of ten-dimensional gene regulatory network, we compare DNN with other methods, including Boolean network, Random Forest, and the latest PCM. The DNN outperforms these commonly used approaches in terms of accuracy and robustness for network inference (Figure 2D). Secondly, DNN can simulate the system dynamics, which is not available for other network inference approaches. So one can treat and analyze the DNN model from a dynamical system perspective. Thirdly, DNN is able to predict both the interaction direction and interaction type, which is critical to studying the dynamical mechanism of regulatory networks. Also, in the example of food chain and cardiovascular disease, DNN displays great ability for distinguishing direct causal interactions from indirect causal interactions. Finally, DNN is able to deal with single cell data for both network inference and simulating the dynamics of cell fate decision systems, which offers a new way to bridge data-driven and model-based approaches on cell-fate transitions at single-cell level.

An important issue is whether DNN can model biological processes from a small amount of data to explore the underlying mechanisms of cell fate decisions. Although DNN

shows good performance in reconstructing small-scale gene regulatory network from single cell data, inferring high-dimensional networks from a small amount of data warrants further study. By studying a high-dimensional system of mouse development modeled via DNN, we discover the existence of intermediate cell states which play a critical role in development, and successfully reconstruct the cell lineages by simulating the temporal changes of gene expression in development. We further quantify the developmental path in terms of gene expression based on DNN model. Importantly, by treating the DNN model as a dynamic model, we can simulate biological perturbations, such as knockout of genes. We perform single gene knockout experiments through DNN model to explore the key genes controlling cell differentiation, and obtain results consistent with experiments.

We further propose that DNN provides a data-driven approach to quantify the energy landscape, which is crucial for understanding the mechanisms of cell fate decisions. The Waddington landscape is a classic metaphor for describing cellular development. Recently, the quantitative framework for constructing energy landscape of gene regulatory networks has been developed from model-driven perspective (e.g. (1,15,20)) and data-driven perspective (e.g. (18,19,65,66)). Based on the deterministic system obtained from DNN and the estimated diffusion coefficient, we propose the DNN-PSCA approach, which offers an approach for combining data-driven and model-driven strategy to quantify the energy landscape for gene networks. Of note, our method showed reasonable computational efficiency. For example, in the MISA task, the learner contains 1348 synaptic weights, and it takes about 7 minutes to train one epoch, on a laptop computer, with Inter Core i7 and a memory size of 16GB, under Windows 10.

Certain points need to be emphasized. Firstly, it is known that the problem of causal inference of networks may have no unique solution, and different biophysical processes may lead to similar time series. An example is that many models discovered from optimization can predict data equally well, but fail to match the correct hypothesis (67). To mitigate this problem, one possible way is to identify models with correct interpretation by comparing model features across multiple data samples (67). In our work, we use the idea of ensemble learning, by integrating several DNNs with different hyperparameters, to capture more comprehensive information from the data. Secondly, in this work, we use a feed-forward neural network structure. It is interesting to see that this simple network structure can model GRNs with feedback loops, due to a hidden feedback loop structure in our computational framework coming from the iteration procedure when we use f to calculate gene variable x . In future work, other neural network approaches can be absorbed to current framework to acquire better performance of learning, such as exploiting prior knowledge (network structure in this work) (68), and in-memory computing (69). Moreover, from the application side, we showed that the DNN can model the oscillation pattern well from a synthetic gene network. We also anticipate that our approach can be used to reveal oscillation patterns from single cell data in more realistic systems (15,70).

In summary, the DNN developed in this work provides a general data-driven approach for the reconstruction of gene regulatory networks and quantification of stochastic dynamics of high-dimensional gene regulatory systems. Our results promote the mechanistic understanding of cell fate decisions and roles of intermediate cell states in development.

DATA AVAILABILITY

The source code to reproduce the results of this study is freely available on GitHub (https://github.com/ChenFeng87/network_inference).

SUPPLEMENTARY DATA

Supplementary Data are available at NARGAB Online.

FUNDING

National Key R&D Program of China [2019YFA0709502 to C.L.]; National Natural Science Foundation of China [12171102].

Conflict of interest statement. None declared.

REFERENCES

- Li, C. and Wang, J. (2013) Quantifying cell fate decisions for differentiation and reprogramming of a human stem cell network: landscape and biological paths. *PLoS Comput. Biol.*, **9**, e1003165.
- MacArthur, B.D., Ma'ayan, A. and Lemischka, I.R. (2009) Systems biology of stem cell fate and cellular reprogramming. *Nat. Rev. Mol. Cell Biol.*, **10**, 672–681.
- Collombet, S., van Oevelen, C., Sardina Ortega, J.L., Abou-Jaoudé, W., Di Stefano, B., Thomas-Chollier, M., Graf, T. and Thieffry, D. (2017) Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proc. Natl. Acad. Sci. U.S.A.*, **114**, 5792–5799.
- Zhang, B. and Wolynes, P.G. (2014) Stem cell differentiation as a many-body problem. *Proc. Natl. Acad. Sci. U.S.A.*, **111**, 10185–10190.
- Lin, Y.T., Hufton, P.G., Lee, E.J. and Potoyan, D.A. (2018) A stochastic and dynamical view of pluripotency in mouse embryonic stem cells. *PLoS Comput. Biol.*, **14**, e1006000.
- Tripathi, S., Levine, H. and Jolly, M.K. (2020) The physics of cellular decision making during epithelial-mesenchymal transition. *Annu. Rev. Biophys.*, **49**, 1–18.
- Kang, X. and Li, C. (2021) A dimension reduction approach for energy landscape: identifying intermediate states in EMT metastasis network. *Adv. Sci.*, **8**, 2003133.
- Huang, S., Ernberg, I. and Kauffman, S. (2009) Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. *Semin. Cell Dev. Biol.*, **20**, 869–876.
- Samaga, R., Saez-Rodriguez, J., Alexopoulos, L.G., Sorger, P.K. and Klami, S. (2009) The logic of EGFR/ErbB signaling: theoretical properties and analysis of high-throughput data. *PLoS Comput. Biol.*, **5**, e1000438.
- Marbach, D., Prill, R.J., Schaffter, T., Mattiussi, C., Floreano, D. and Stolovitzky, G. (2010) Revealing strengths and weaknesses of methods for gene network inference. *Proc. Natl. Acad. Sci. U.S.A.*, **107**, 6286–6291.
- Huynh-Thu, V.A., Irrthum, A., Wehenkel, L. and Geurts, P. (2010) Inferring regulatory networks from expression data using tree-based methods. *PLoS One*, **5**, e12776.
- Malekpour, A., Alizad-Rahvar, A.R. and Sadeghi, M. (2020) LogicNet: probabilistic continuous logics in reconstructing gene regulatory networks. *BMC Bioinformatics*, **21**, 318.
- Leng, S., Ma, H., Kurths, J., Lai, Y.-C., Lin, W., Aihara, K. and Chen, L. (2020) Partial cross mapping eliminates indirect causal influences. *Nat. Commun.*, **11**, 2632.

14. Zhao, J., Zhou, Y., Zhang, X. and Chen, L. (2016) Part mutual information for quantifying direct associations in networks. *Proc. Natl. Acad. Sci. U.S.A.*, **113**, 5130–5135.
15. Li, C. and Wang, J. (2014) Landscape and flux reveal a new global view and physical quantification of mammalian cell cycle. *Proc. Natl. Acad. Sci. U.S.A.*, **111**, 14130–14135.
16. Ge, H. and Qian, H. (2016) Mesoscopic kinetic basis of macroscopic chemical thermodynamics: a mathematical theory. *Phys. Rev. E*, **94**, 052150.
17. Ao, P. (2008) Emerging of stochastic dynamical equalities and steady state thermodynamics from Darwinian dynamics. *Commun. Theor. Phys.*, **49**, 1073–1090.
18. Zhou, P., Tomasi, C., Li, T. and Nie, Q. (2021) Dissecting transition cells from single-cell transcriptome data through multiscale stochastic dynamics. *Nat. Commun.*, **12**, 5609.
19. Jiang, Q., Zhang, S. and Wan, L. (2022) Dynamic inference of cell developmental complex energy landscape from time series single-cell transcriptomic data. *PLoS Comput. Biol.*, **18**, e1009821.
20. Wang, J., Xu, L., Wang, E. and Huang, S. (2010) The potential landscape of genetic circuits imposes the arrow of time in stem cell differentiation. *Biophys. J.*, **99**, 29–39.
21. Kang, X., Wang, J. and Li, C. (2019) Exposing the underlying relationship of cancer metastasis to metabolism and epithelial-mesenchymal transitions. *iScience*, **21**, 754–772.
22. Lecun, Y., Bengio, Y. and Hinton, G. (2015) Deep learning. *Nature*, **521**, 436–444.
23. Yuan, Y. and Bar-Joseph, Z. (2019) Deep learning for inferring gene relationships from single-cell expression data. *Proc. Natl. Acad. Sci. U.S.A.*, **116**, 27151–27158.
24. Zhang, Z., Zhao, Y., Liu, J., Wang, S., Tao, R., Xin, R. and Zhang, J. (2019) A general deep learning framework for network reconstruction and dynamics learning. *Appl. Network Sci.*, **4**, 110.
25. Shen, J., Liu, F., Tu, Y. and Tang, C. (2021) Finding gene network topologies for given biological function with recurrent neural network. *Nat. Commun.*, **12**, 3125.
26. Michael, P.H. Stumpf. (2021) Inferring better gene regulation networks from single cell data. *Curr. Opin. Syst. Biol.*, **27**, 100342.
27. Moignard, V., Woodhouse, S., Haghverdi, L., Lilly, A.J., Tanaka, Y., Wilkinson, A.C., Buettner, F., Macaulay, I.C., Jewaid, W., Diamanti, E. et al. (2015) Decoding the regulatory network of early blood development from single-cell gene expression measurements. *Nat. Biotechnol.*, **33**, 269–276.
28. Chen, S. and Mar, J.C. (2018) Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data. *BMC Bioinformatics*, **19**, 232.
29. Chen, T.Q., Rubanova, Y., Bettencourt, J. and Duvenaud, D.K. (2018) Neural ordinary differential equations. *Adv. Neural Inf. Process. Syst.*, **31**, 6571–6583.
30. Roesch, E., Rackauckas, C. and Stumpf, M.P.H. (2021) Collocation based training of neural ordinary differential equations. *Stat. Appl. Genet. Mol. Biol.*, **20**, 37–49.
31. Baker, R.E., Peña, J.-M., Jayamohan, J. and Jérusalem, A. (2018) Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biol. Lett.*, **14**, 20170660.
32. Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A. and Edelman, A. (2020) Universal differential equations for scientific machine learning. arXiv doi: <https://arxiv.org/abs/2001.04385>, 02 November 2021, preprint: not peer reviewed.
33. Müssel, C., Hopfensitz, M. and Kestler, H.A. (2010) BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, **26**, 1378–1380.
34. Chen, H., Guo, J., Mishra, S.K., Robson, P., Niranjana, M. and Zheng, J. (2015) Single-cell transcriptional analysis to uncover regulatory circuits driving cell fate decisions in early mouse development. *Bioinformatics*, **31**, 1060–1066.
35. Guo, G., Huss, M., Tong, G.Q., Wang, C., Li, Sun, L., Clarke, N.D. and Robson, P. (2010) Resolution of cell fate decisions revealed by single-cell gene expression analysis from zygote to blastocyst. *Dev. Cell*, **18**, 675–685.
36. Rossant, J. and Tam, P.P. (2009) Blastocyst lineage formation, early embryonic asymmetries and axis patterning in the mouse. *Development*, **136**, 701–713.
37. Xue, Z., Huang, K., Cai, C., Cai, L., Jiang, C., Feng, Y., Liu, Z., Zeng, Q., Chen, L., Sun, Y.E. et al. (2013) Genetic programs in human and mouse early embryos revealed by single-cell RNA sequencing. *Nature*, **500**, 593–597.
38. Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B.B., Siddiqui, A. et al. (2009) mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods*, **6**, 377–382.
39. MacLean, A., Hong, T. and Nie, Q. (2018) Exploring intermediate cell states through the lens of single cells. *Curr. Opin. Syst. Biol.*, **9**, 32–41.
40. Sisan, D.R., Halter, M., Hubbard, J.B. and Plant, A.L. (2012) Predicting rates of cell state change caused by stochastic fluctuations using a data-driven landscape model. *Proc. Natl. Acad. Sci. U.S.A.*, **109**, 19262–19267.
41. Hu, G. (1994) In: *Stochastic Forces and Nonlinear Systems*. Shanghai Scientific and Technological Education Press, Shanghai, pp. 68–74.
42. Van Kampen, N. (1992) In: *Stochastic Processes in Chemistry and Physics*. 1st edn. North Holland, Amsterdam, pp. 120–127.
43. Balázsi, G., van Oudenaarden, A. and Collins, J.J. (2011) Cellular decision making and biological noise: from microbes to mammals. *Cell*, **144**, 910–925.
44. Hasty, J., Pradines, J., Dolnik, M. and Collins, J.J. (2000) Noise-based switches and amplifiers for gene expression. *Proc. Natl. Acad. Sci. U.S.A.*, **97**, 2075–2080.
45. Thattai, M. and van Oudenaarden, A. (2001) Intrinsic noise in gene regulatory networks. *Proc. Natl. Acad. Sci. U.S.A.*, **98**, 8614–8619.
46. Huang, S., Guo, Y.P., May, G. and Enver, T. (2007) Bifurcation dynamics in lineage-commitment in bipotent progenitor cells. *Dev. Biol.*, **305**, 695–713.
47. Roeder, I. and Glauche, I. (2006) Towards an understanding of lineage specification in hematopoietic stem cells: a mathematical model for the interaction of transcription factors GATA-1 and PU.1. *J. Theor. Biol.*, **241**, 852–865.
48. Chickarmane, V. and Peterson, C. (2008) A computational model for understanding stem cell, trophectoderm and endoderm lineage determination. *PLoS One*, **3**, e3478.
49. Li, C. and Balazsi, G. (2018) A landscape view on the interplay between EMT and cancer metastasis. *npj. Syst. Biol. Appl.*, **4**, 34.
50. Elowitz, M. and Liebler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature*, **403**, 335–338.
51. Benincà, E., Huisman, J., Heerkloss, R., Jöhnk, K.D., Branco, P., Van Nes, E.H., Scheffer, M. and Ellner, S.P. (2008) Chaos in a long-term experiment with a plankton community. *Nature*, **451**, 822–825.
52. Benincà, E., Jöhnk, K.D., Heerkloss, R. and Huisman, J. (2010) Coupled predator–prey oscillations in a chaotic food web. *Ecol. Lett.*, **12**, 1367–1378.
53. Lee, B.J., Kim, B. and Lee, K. (2014) Air pollution exposure and cardiovascular disease. *Toxicol Res*, **30**, 71–75.
54. Wong, T.W., Lau, T.S., Yu, T.S., Neller, A., Wong, S.L., Tam, W. and Pang, S.W. (1999) Air pollution and hospital admissions for respiratory and cardiovascular diseases in Hong Kong. *Occup. Environ. Med.*, **56**, 679–683.
55. Fan, J. and Zhang, W. (1999) Statistical estimation in varying coefficient models. *Ann. Statist.*, **27**, 1491–1518.
56. Milojevic, A., Wilkinson, P., Armstrong, B., Bhaskaran, K., Smeeth, L. and Hajat, S. (2014) Short-term effects of air pollution on a range of cardiovascular events in England and Wales: case-crossover analysis of the MINAP database, hospital admissions and mortality. *Heart*, **100**, 1093–1098.
57. Ma, H., Leng, S., Tao, C., Ying, X., Kurths, J., Lai, Y.-C. and Lin, W. (2017) Detection of time delays and directional interactions based on time series from complex dynamical systems. *Phys. Rev. E*, **96**, 012221.
58. Li, C. and Wang, J. (2013) Quantifying Waddington landscapes and paths of non-adiabatic cell fate decisions for differentiation, reprogramming and transdifferentiation. *J. R. Soc. Interface*, **10**, 20130787.
59. Dietterich, T.G. (2000a) Ensemble methods in machine learning. In: *International Workshop on Multiple Classifier Systems*. Springer, Berlin, Heidelberg, pp. 1–15.
60. Zhou, Z.H. (2012) In: *Ensemble Methods: Foundations and Algorithms*. CRC Press.
61. Maulik, U. and Bandyopadhyay, S. (2002) Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Patt. Anal. Mach. Int.*, **24**, 1650–1654.

62. Gunne-Braden,A., Sullivan,A., Gharibi,B., Sheriff,R.S.M., Maity,A., Wang,Y.-F., Edwards,A., Jiang,M., Howell,M., Goldstone,R. *et al.* (2020) GATA3 mediates a fast, irreversible commitment to BMP4-driven differentiation in human embryonic stem cells. *Cell Stem. Cell.*, **26**, 693–706.
63. Zhang,J., Tam,W.-L., Tong,G.Q., Wu,Q., Chan,H.-Y., Soh,B.-S., Lou,Y., Yang,J., Ma,Y., Chai,L. *et al.* (2006) Sall4 modulates embryonic stem cell pluripotency and early embryonic development by the transcriptional regulation of Pou5f1. *Nat. Cell Biol.*, **8**, 1114–1123.
64. Rubner,Y., Tomasi,C. and Guibas,L.J. (2000) The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, **40**, 99–121.
65. Chen,Z., An,S., Bai,X., Gong,F., Ma,L. and Wan,L. (2019) DensityPath: an algorithm to visualize and reconstruct cell state-transition path on density landscape for single-cell RNA sequencing data. *Bioinformatics*, **35**, 2593–2601.
66. Guo,J. and Zheng,J. (2017) HopLand: single-cell pseudotime recovery using continuous Hopfield network-based modeling of Waddington's epigenetic landscape. *Bioinformatics*, **33**, i102–i109.
67. Genkin,M. and Engel,T.A. (2020) Moving beyond generalization to accurate interpretation of flexible models. *Nat. Mach. Int.*, **2**, 674–683.
68. Roychowdhury,S., Diligenti,M. and Gori,M. (2021) Regularizing deep networks with prior knowledge: a constraint-based approach. *Knowledge-Based Syst.*, **222**, 106989.
69. Joshi,V., Le Gallo,M., Haefeli,S., Boybat,I., Nandakumar,S.R., Piveteau,C., Dazzi,M., Rajendran,B., Sebastian,A. and Eleftheriou,E. (2020) Accurate deep neural network inference using computational phase-change memory. *Nat. Commun.*, **11**, 2473.
70. An,S., Ma,L. and Wan,L. (2019) TSEE: an elastic embedding method to visualize the dynamic gene expression patterns of time series single-cell RNA sequencing data. *BMC Genom.*, **20**, 224.