Bioinformatics, 31(7), 2015, 1060–1066
doi: 10.1093/bioinformatics/btu777
Advance Access Publication Date: 20 November 2014
Original Paper

OXFORD

Systems biology

Single-cell transcriptional analysis to uncover regulatory circuits driving cell fate decisions in early mouse development

Haifen Chen[1], Jing Guo[1], Shital K. Mishra[1], Paul Robson[2], Mahesan Niranjan[3] and Jie Zheng[1,2,*]

[1]School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore, [2]Genome Institute of Singapore, Biopolis, Singapore 138672, Singapore and [3]School of Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK

*To whom correspondence should be addressed.

Associate Editor: Igor Jurisica

Received on July 14, 2014; revised on November 2, 2014; accepted on November 17, 2014

Abstract

Motivation: Transcriptional regulatory networks controlling cell fate decisions in mammalian embryonic development remain elusive despite a long time of research. The recent emergence of single-cell RNA profiling technology raises hope for new discovery. Although experimental works have obtained intriguing insights into the mouse early development, a holistic and systematic view is still missing. Mathematical models of cell fates tend to be concept-based, not designed to learn from real data. To elucidate the regulatory mechanisms behind cell fate decisions, it is highly desirable to synthesize the data-driven and knowledge-driven modeling approaches.

Results: We propose a novel method that integrates the structure of a cell lineage tree with transcriptional patterns from single-cell data. This method adopts probabilistic Boolean network (PBN) for network modeling, and genetic algorithm as search strategy. Guided by the 'directionality' of cell development along branches of the cell lineage tree, our method is able to accurately infer the regulatory circuits from single-cell gene expression data, in a holistic way. Applied on the single-cell transcriptional data of mouse preimplantation development, our algorithm outperforms conventional methods of network inference. Given the network topology, our method can also identify the operational interactions in the gene regulatory network (GRN), corresponding to specific cell fate determination. This is one of the first attempts to infer GRNs from single-cell transcriptional data, incorporating dynamics of cell development along a cell lineage tree.

Availability and implementation: Implementation of our algorithm is available from the authors upon request.

Contact: zhengjie@ntu.edu.sg

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Understanding gene regulatory networks (GRNs) that control cell fate decisions is of paramount importance in biology. It can provide critical insights into many questions in developmental biology (Oliveri and Davidson, 2004), stem cell (Macarthur *et al.*, 2009), cancer (Huang *et al.*, 2009; Samaga *et al.*, 2009), synthetic biology (Wu *et al.*, 2013), etc. Mouse preimplantation development is a prominent example for the study of cell fate decisions (Guo *et al.*, 2010; Oron and Ivanova, 2012). Starting from the 1-cell fertilized zygote it proceeds to the 64-cell blastocyst, consisting of three

distinct cell types, i.e. trophectoderm (TE), primitive endoderm (PE) and epiblast (EPI) (Rossant and Tam, 2009). For this classic developmental process, many results and insights have been obtained through decades of research. However, the regulatory mechanism for differential gene expression that leads to the formation of the three cell types remains to be elucidated. One hurdle lies in that many analyses were conducted on the average expression of a population of cells in mouse as well as other species (Hoppe *et al.*, 2014; Samsonova *et al.*, 2007), ignoring the heterogeneity among individual cells, which however is critical for the cell lineage formation. Recently, the technology of single-cell RNA profiling promises to overcome this hurdle and provides new insights (Tang *et al.*, 2011).

In 2010, Robson lab generated a single-cell RNA profiling dataset for the early development of mouse embryo (Guo *et al.*, 2010). Using statistical analysis techniques on the dataset, they identified intriguing patterns that shed light on the dynamics of cell fate decisions. The results obtained from this dataset (which we will refer to as 'Robson's data' hereafter) demonstrated the power of single-cell expression analysis. However, the data-driven approach in (Guo *et al.*, 2010) was limited to descriptive statistical relations of gene expression, interpretation of which requires highly specialized knowledge and experiences of researchers. To gain a systematic and holistic understanding of the causal mechanisms, an integrative network model that is 'executable' through computer simulations would be highly desirable.

In conjunction with experimental works, mathematical models have been proposed for the study of cell fate determination. Many models are based on the theory of dynamical systems (Huang, 2010), which represents a cell state by a vector of gene expression levels, $S = [x_1, x_2, \ldots, x_n]$, where x_i is the expression value of the ith gene. Hence, each cell state is a point in the n-dimensional state space, in which a change of the cell state is represented as a movement along a trajectory. Constrained by the architecture of a GRN, cell states tend to move toward stable states called 'attractors'. Kauffman proposed that cell types can be modeled as attractors over 40 years ago, and he pioneered the use of Boolean networks (Kauffman, 1969). Among numerous applications, Boolean networks have been applied to studies of cancer cell death (Calzone *et al.*, 2010; Tournier and Chaves, 2009) and stem cells (Bonzanni *et al.*, 2013; Xu *et al.*, 2014). In addition to the discrete model of Boolean network, ordinary differential equations based approaches that include kinetic models of molecular interactions are popular for cell fate modeling (Andrecut *et al.*, 2011; Li and Wang, 2013). The continuous models can represent the physical processes of gene regulation precisely and quantitatively, and they are amenable to theoretical analysis of the underlying systems (e.g. steady-state analysis). For the inference and analysis of GRNs, various differential equation models have been developed (De Jong, 2002; Kimura *et al.*, 2005). However, differential equation models need accurate estimation of parameter values, which remains a challenge in many cases (Meyer *et al.*, 2014). Machine learning models have also been developed to elucidate gene regulation in cell development. For example, in (Parikh *et al.*, 2011) and (Hashimoto *et al.*, 2012), cell lineage trees were incorporated into machine learning algorithms to infer gene networks and expression programs, respectively.

Reverse engineering of GRN from gene expression data is a challenging problem in computational biology. Although many methods have been proposed (see Hecker *et al.*, 2009, Schlitt and Brazma, 2007, and references therein), they were mostly designed for the average expression in a population of cells, and rely on local dependence among variables (e.g. genes). However, single-cell expression data pose new challenges for GRN inference. The heterogeneity among cells could make the average of expression levels misleading. Moreover, many existing methods of GRN inference require long and dense times series data of RNA profiling, which are often unavailable, especially when the experimental cost is high. Therefore, alternative strategies of GRN inference are needed for single-cell transcriptional data with short and sparse time series.

In this article, we propose a novel approach to the inference of GRN from single-cell expression data. It takes into account the relative positions of cells on the cell lineage tree. Our assumption is that GRN provides the driving force that pushes cells from ancestral states towards descendant states (Huang and Kauffman, 2012). Such 'directionality' of cell development is imposed by the logical constraints of the underlying GRN. From Robson's dataset of mouse early development, we can observe which cell states belong to a certain node in the cell lineage tree. Given a candidate GRN, we can simulate the dynamic trajectories of cells following the regulatory rules encoded in the GRN. If the simulated trajectories are compatible with the directed paths from the root towards leaves of the cell lineage tree, the GRN is likely to be true. Conversely, if a candidate GRN drives cells from descendant states backward to ancestral states, it is likely to be false. To implement this strategy, we choose the framework of Boolean network for its strength in state transition analysis, and genetic algorithms (GAs) for optimization (see Section 2.2 for details). A similar idea for GRN inference based on Boolean network was proposed in (Pal *et al.*, 2005). However, Pal *et al.* assumed that the expression data should be sampled from attractor states, which may be too restrictive for single-cell data. Relying on the concordance of attractor states of the model with states of the data, they did not consider fully the 'directionality' of cell development along branches of a cell lineage tree. Therefore, although the method of Pal *et al.* is very insightful, it is not directly applicable to the single-cell expression data of mouse embryonic development.

Running on the Robson's dataset, our method for GRN structure inference achieved better performance than most of the other methods, validated using a published benchmark GRN constructed manually from experiments (Oron and Ivanova, 2012). Moreover, given the structure of GRN, our method is able to predict operational regulatory interactions responsible for a specific cell lineage formation. Most of the predictions are consistent with literature of experimental observations. To the best of our knowledge, the method proposed here is one of the first attempts to infer GRN from single-cell expression data. Unlike existing methods, it does not rely on local dependence of variables, which is difficult to detect from short and sparse time series data. Instead, it is a holistic approach that utilizes global information about a system (e.g. network stability). As more single-cell data become available in near future, our method is promising to help decipher the mechanisms of cell fate decisions.

2 Methods

2.1 Data

We used the single-cell gene expression data from (Guo *et al.*, 2010), which were generated by high-throughput single-cell qPCR. This dataset describes the expression profiles of 48 genes during the mouse preimplantation development, from 1- to 64-cell stage (see Section S.1 in Supplementary material for more information). Two cell fate decisions are made during this period (see Fig. 1a). The first decision is made when cells transform from 16- to 32-cell stage, where two distinct cell lineages take shape: TE and inner cell mass (ICM). The second decision is made during the transitions from the 32- to 64-cell stage, and two cell types are generated from ICM: PE and EPI. To focus on the cell fate regulation during these two processes of cell lineage formation, we extracted data from the three stages, i.e. 16-, 32- and 64-cell stages.

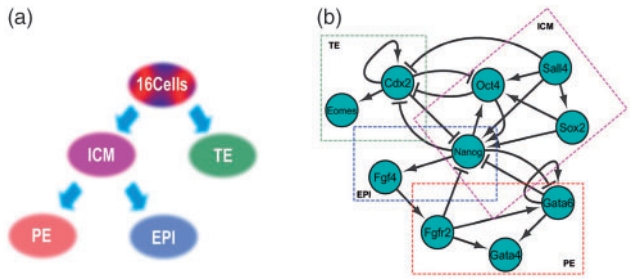

Fig. 1. The structure of cell lineage tree **(a)** and the GRN **(b)** for the formation of three distinct cell lineages: TE, PE and EPI (adapted from Oron and Ivanova, 2012)

As Boolean network was used to model the GRN, we first discretized the continuous gene expression levels into two states: 0 (lowly expressed) and 1 (highly expressed), for the data of three stages (i.e. the 16-, 32- and 64-cell stages) separately. Let $X=\{x_{ij}\}_{n\times m}$ be the gene expression dataset with n genes and m samples. For each gene, by assuming Gaussian distribution, we calculated the mean μ and SD σ, and set the discrete value d_{ij} of the ith gene in the jth sample as

$$d_{ij}=\begin{cases}1, & \text{if } x_{ij} >= \mu_i + w\times\sigma_i\\ 0, & \text{otherwise}\end{cases} \quad (1)$$

where $\mu_i=\frac{\sum_{j=1}^{m} x_{ij}}{m}$, $\sigma_i=\sqrt{\frac{\sum_{j=1}^{m}(x_{ij}-\mu_i)^2}{m-1}}$, $i=1,\ldots,n$, and w is the weight which was set to 0.5 in our experiments (Ding and Peng, 2005; Li *et al.*, 2010).

2.2 Structure inference of GRN

In this section, our goal is to reconstruct the topological structures of GRNs underlying cell fate decisions. The hierarchical structure of the cell lineage tree is used to guide the GRN inference, based on the premise that a true GRN should have simulated trajectories consistent with the temporal order of the developmental process.

To simulate the dynamic trajectories of cell state transitions, we used simple and general rules to update the states of genes (Li *et al.*, 2004), based on the assumption that the behaviors of a cellular system depend more on the topology of the regulatory network and less on the parameters of the interactions (Feiglin *et al.*, 2012; Samaga *et al.*, 2009). Let $g_i(t+1)$ be the state of the ith gene at time $t+1$, $N_i^I(t)$ be the number of inhibitory edges incident to the ith gene at time t, and $N_i^A(t)$ be the number of activating edges at time t. The updating rule for the ith gene is defined as follows:

$$g_i(t+1)=\begin{cases}0, & \text{if } N_i^I(t) > N_i^A(t)\\ 0.5, & \text{if } N_i^I(t) = N_i^A(t)\\ 1, & \text{if } N_i^I(t) < N_i^A(t)\end{cases} \quad (2)$$

Because the state of a gene in a Boolean network can only be 0 or 1, when $N_i^I(t)=N_i^A(t)$, $g_i(t+1)$ will be randomly assigned to 0 or 1 with equal probability.

For a proposed network structure, we generate its state transition graph (STG) by asynchronously updating the states of genes using the rule described in Equation (2). For a particular cell state (i.e. a vector of states of genes), in each step, we randomly pick one gene with equal probability and update its state based on its inputs (i.e. regulators) and the rule in Equation (2), thereby making a transition from this cell state to another. After u updates, we calculate the transition probabilities from this cell state to others. In this way, we obtain an STG (represented by its adjacency matrix A), of which each entry contains the transition probability between cell states. For the ith cell state, we have

$$\sum_{j=1}^{2^n} A_{ij}=1, \quad (3)$$

where n is the number of genes and $i=1,\ldots,2^n$. Here A is a Markov matrix, since the transition probabilities do not change over time. Therefore, by calculating A^l, we can obtain the end states after l time points from any starting state.

Following the 'directionality' of cell development, during ICM → PE + EPI (similarly for 16C → TE + ICM), we expect the states in PE and EPI to be stabler than the ICM states. As such, the 'forward' transitions from ICM to PE or EPI should have higher probabilities than the 'backward' transitions. Therefore, we give bonus to the transitions from ICM to PE or EPI (we also give bonus to self-transitions within each cell type), while impose a penalty on any transition from PE or EPI back to ICM. In addition, we also put penalties on trans-transitions across different cell types at the same stage, e.g. PE → EPI or EPI → PE. Accordingly, the fitness score for structure inference is defined as

$$\begin{aligned}\text{Score}=&\sum_{k=1}^{2}\sum_{j\in W}\sum_{i\in M_k}(p_{ij}\times \text{Bonus}(UpLevel\rightarrow LowLevel))\\ &-\sum_{j\in W}\sum_{k=1}^{2}\sum_{i\in M_k}(p_{ij}\times \text{Penalty}(LowLevel\rightarrow UpLevel))\\ &-\sum_{k=1}^{2}\sum_{j\in M_k}\sum_{i\in M_{k'}}(p_{ij}\times \text{Penalty}(trans\text{-}trasitions)),\end{aligned} \quad (4)$$

where W is the set of indices of cell states belonging to the upper-level cell type; M_k is the set of indices of states in the kth lower-level cell type (here we assume there are two cell types at the lower level, i.e. $k=1$ or 2, although our method can be easily generalized to more than two descendant cell types); p_{ij} is the transition probability from ith to jth state (i.e. p_{ij} is an element of the matrix of STG); $M_{k'}$ is the alternative representation of M_k, which means when $k=1$, $k'=2$ and when $k=2$, $k'=1$.

To learn the optimal GRN structure that best fits the data, a GA is designed to optimize the scores of candidate topologies. GA is often used to solve optimization and search problems, by simulating the process of natural selection, crossover and mutation to reach an advanced species (Holland, 1992; De Jong, 1988). The pseudocode of GRN structure inference based on GA is shown in Algorithm 1. First, we create a population with N candidate individuals (i.e. the initial population). Each GA individual, which corresponds to a network, is a binary string obtained by concatenating the columns of the adjacency matrix of the network. The maximum number of regulators for each gene is K, as a constraint for each individual. Then we calculate the fitness scores based on Equation (4), which are used to select the top N' individuals as candidates for the next GA iteration. In 'crossover', two parents are randomly chosen by Roulette wheel selection based on their scores. A typical 'recombination' procedure is performed to generate their children. In 'mutation', dependent on the mutation rate, we randomly choose chromosomes (i.e. individuals) and randomly decide the mutation sites to be flipped. As such, a new population is generated and the algorithm enters the next iteration till the maximum number of iterations is reached. Since the time complexity of Algorithm 1 is exponential in the number of genes n, our method is currently more applicable to small networks (e.g. $n<10$). We have provided detailed information about the parameters in our algorithm and analyzed their influence on the performance of the algorithm in Section S.2 in Supplementary material.

Algorithm 1. GA-based GRN inference from single-cell transcription data

INPUT: Single-cell gene expression data, population size N, number of selected individuals N', mutation rate μ, permutations per chromosome η, and maximum iteration times *MaxIt*
OUTPUT: A vector P, which consists of regulators of each gene in the network (i.e. P is the representation of a network topology)

$P_0(N) \leftarrow$ initial population
for each $i \in P_0$ **do**
  Calculate *score*(i) using Equation (4)
end for
Select top N' individuals based on their scores and insert them into a queue Q
for $k = 1$ **to** *MaxIt* **do**
  $P_k \leftarrow$ a null set
  // Crossover
  while(size(P_k) $< N$)
    Choose two parents P^i_{k-1} and P^j_{k-1} from Q by Roulette wheel selection
    Generate two children by Recombination(P^i_{k-1}, P^j_{k-1})
    Check and modify children according to the constraint for individual
    Add children to P_k
  end while
  // Mutation
  Select $N \times \mu$ members from P_k
  Invert η randomly selected site for each chromosome
  Check and modify children according to the constraint for individual
  Add new individuals to P_k
  // Scoring
  for each $i \in P_k$ **do**
    Calculate *score*(i) using Equation (4)
  end for
  $Q \leftarrow$ a null set
  // Selection
  Select N' top score individuals from P_k and insert them into Q
  Save a copy of the N' top score individuals to $\mathbb{P}$
end for
Determine P by taking the 'consensus' of high-score networks in $\mathbb{P}$
Return P

2.3 Identifying operational interactions in GRN

2.3.1 Network encoding

In this section, we will demonstrate that the same framework of inference as in the previous section can also be used to identify the operational interactions responsible for a specific cell fate decision, from a given network topology. The GRN is again encoded as an asynchronous PBN $\mathbb{N}$. The states of genes are updated based on their inputs (regulators) and Boolean functions (or rules). Because the incoming edges for each gene are known, we only need to determine the Boolean rules for generating the dynamic behaviors of $\mathbb{N}$. Here the rules were derived based on the knowledge from literature (see Section 3.2 for details).

For a particular network $\mathbb{N}$, the collection of regulatory rules R consists of a set of Boolean functions F and their corresponding probabilities in set P. Thus we have $R = \{F, P\}$, where $F = \{F^1, F^2, ..., F^n\}$, $P = \{P^1, P^2, ..., P^n\}$, and n is the number of genes. For the ith gene, $F^i = \left\{f^i_1, f^i_2, ..., f^i_{L_i}\right\}$ and $P^i = \left\{p^i_1, p^i_2, ..., p^i_{L_i}\right\}$, where $i = 1, 2, ..., n$. Here f^i_j is the jth regulatory rule of the ith gene with probability p^i_j, and L_i is the total number of rules for the ith gene. The probabilities of rules should satisfy

$$\sum_{i=1}^{n}\sum_{j=1}^{L_i} p^i_j = 1. \tag{5}$$

The dynamic behavior of a GRN is not only determined by the Boolean functions in F, but also influenced by the probability parameters in P. Because a rule with a higher probability is more likely to be chosen to update the states of genes, the probabilities of rules affect the dynamics of the network.

2.3.2 Probability inference

We employ GA again to infer the probability vector P. Without loss of generality, we assume that cell fate decision is a bifurcation in the cell lineage tree: $Up \rightarrow Low_1 + Low_2$ (e.g. ICM $\rightarrow$ PE + EPI). Because we aim to identify the operational components for a specific lineage decision (e.g. $Up \rightarrow Low_1$), the GA should maximize the transition probabilities for this lineage, and minimize the transition probabilities in the alternative direction (i.e. $Up \rightarrow Low_2$). Hence, the fitness score is defined as

$$\begin{aligned}\text{Score} = &\sum_{j\in M_1}\sum_{i\in W}(p_{ij} \times \text{Bonus}(Up \rightarrow Low_1)) - \\ &\sum_{j\in M_2}\sum_{i\in W}(p_{ij} \times \text{Penalty}(Up \rightarrow Low_2)),\end{aligned} \tag{6}$$

where M_1 and M_2 are the sets of indices of cell states belonging to the two lower cell types; W is the set of indices of cell states of the upper-level cell type; p_{ij} is the transition probability from the ith state to the jth state.

To calculate the transition probabilities among cell states, we employ an asynchronous PBN to model GRN and obtain the STG by asynchronously updating the activities of nodes. In each step, we randomly select one rule f ($f \in F$) based on its probability p ($p \in P$) and update the corresponding target gene. After the STG is obtained, we map the cell states from real data to the STG. Then a fitness score will be calculated for a particular setting of P using Equation (6). Through GA, the final probability vector P would show which rules are more frequently used for the cells to commit to a specific lineage.

The process of probability inference is very similar to Algorithm 1, except for the codification and constraint of GA individuals, and the calculation of fitness scores. Here the GA individual is a vector of the probabilities of all rules for all genes, and each individual is constrained by Equation (5).

3 Results

3.1 Results of structure inference

To validate our algorithm (named SingCellNet hereafter) for GRN structure inference, we applied SingCellNet to real single-cell expression data and compared output with benchmark networks derived from experimental evidence (Oron and Ivanova, 2012). The benchmark networks for 16C $\rightarrow$ TE + ICM and ICM $\rightarrow$ PE + EPI are shown in Figure 2. Note that the edge *Fgf4* $\rightarrow$ *Fgfr2* in Figure 2b

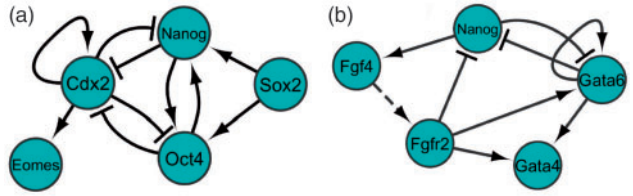

Fig. 2 Benchmark networks for **(a)** 16C → TE + ICM, **(b)** ICM → PE + EPI, (adapted from Oron and Ivanova, 2012). Because the link *Fgf4* → *Fgfr2* is a signal transduction event, it is represented by a dashed line

is a signal transduction event which occurs among cells (i.e. from EPI progenitors to PE progenitors) (Frankenberg *et al.*, 2011). Because we aim to reconstruct GRN within cells rather than across cells, we do not count it when evaluating the performance of GRN inference.

To evaluate the strength of SingCellNet, we applied other methods of GRN inference on the same dataset and compared their performance with our method. These methods include Bayesian network [using WinMine package (Chickering, 2002)], Boolean network [using BoolNet package (Müssel *et al.*, 2010)], Random Forest (Irrthum *et al.*, 2010; Maduranga *et al.*, 2013), LASSO (Fujita *et al.*, 2007), Elastic-Net (Rajapakse and Mundra, 2011) and *Treegl* (Parikh *et al.*, 2011). The results of comparisons are shown in Tables 1 and 2. The true positive (TP) is the number of inferred directed edges matching the benchmark networks in Figure 2; likewise are false positive (FP) and false negative (FN) defined. Precision (Pre) is defined as TP/(TP + FP), sensitivity (Sen) as TP/(TP + FN), and F-measure as 2 ∗ (Pre ∗ Sen)/(Pre + Sen). Figure 3 shows the inferred networks for ICM → PE + EPI by different methods. Additional experimental results of structure inference are provided in Section S.3 of Supplementary material, including the inferred networks for 16C → TE + ICM, and more results about the performance comparison among the methods.

As shown in Table 1, our method SingCellNet can achieve performance close to Random Forest which ranks first in the GRN inference for 16C → TE + ICM. It suggests that SingCellNet is competitive with Random Forest, even although it probably loses some information in data due to discretization. Table 2 shows that SingCellNet outperforms all other methods significantly in the structure inference for ICM → PE + EPI. We notice that SingCellNet performs better in ICM → PE + EPI than in 16C → TE + ICM, while most other methods have lower performance in ICM → PE + EPI. One possible reason could be that the performance of other methods might be affected by the reduced number of samples, as the number of cells in ICM → PE + EPI is 56.75% of that in 16C → TE + ICM.

The good performance of SingCellNet is probably due to its ability to utilize the characteristics of single-cell expression data in the context of cell development. Among cells of the same cell type in the single-cell datasets there is heterogeneity of cellular gene expression. However, there are only two time points with a wide interval in each of the two datasets used here, which provides insufficient information of variation along time points for conventional methods to capture the dependency among variables (i.e. genes). Bayesian network can learn causal relationships among variables and is widely used in GRN inference. However, when applied to the single-cell expression dataset, Bayesian network would assume each cell is an independent sample without considering the heterogeneity among cells and their clustering into cell types. Boolean network is a simple yet effective method for GRN inference. To compare Boolean network with our method, we used the *BestFit* algorithm to represent

Table 1. Performance comparison of structure inference for 16C → TE + ICM

Method	Precision	Sensitivity	F-measure
Random forest	0.6000	0.6000	0.6000
SingCellNet	**0.5000**	**0.6000**	**0.5455**
BoolNet	0.5000	0.4000	0.4444
Treegl	0.7500	0.3000	0.4286
LASSO	0.4000	0.4000	0.4000
Elastic-Net	0.3333	0.4444	0.3636
Bayesian	0.1667	0.1000	0.1250

Table 2. Performance comparison of structure inference for ICM → PE + EPI

Method	Precision	Sensitivity	F-measure
SingCellNet	**0.7500**	**0.7500**	**0.7500**
BoolNet	0.4167	0.6250	0.5000
Treegl	0.3750	0.3750	0.3750
Random forest	0.3000	0.3750	0.3333
Bayesian	0.3333	0.2500	0.2857
LASSO	0.2143	0.3750	0.2727
Elastic-Net	0.1875	0.3750	0.2500

Boolean network approaches. Because this method does not take into account the inherent relationship among cell types, its performance is lower than SingCellNet. Random Forest, LASSO and Elastic-Net are regression-based methods, which are quite effective for GRN inference partly because they do not require data discretization. Similar as Bayesian network, these methods would consider a single cell as an independent sample and build the regression models based on the samples. As a result, they may have high FP rates. Among the six methods, *Treegl* is probably the closest to SingCellNet in that it also considers the cell lineage tree for reverse engineering of gene networks. However, *Treegl* was not designed for single-cell expression data. Moreover, it does not take into account the dynamic 'directionality' of cell development, but instead relies on the similarity among networks.

Compared with other methods, SingCellNet takes advantage of additional information supplied by the single-cell gene expression data. For example, the heterogeneity among individual cells can provide statistical signals of state transitions; the directionality of trajectories along branches of cell lineage tree can help filter out many spurious network topologies. Running on the dataset with short and sparse time series, SingCellNet is less affected by the lack of sufficient signals of local dependence among variables, because it captures the global and qualitative properties of a system (e.g. the stability of a network state).

3.2 Results of identifying operational interactions

To validate our method for probability inference of rules on real expression data from 16- to 64-cell stage, we adapted the network derived from experimental evidence (Oron and Ivanova, 2012) as our benchmark network (see Fig. 1b). By encoding a network with rules and corresponding probabilities, and comparing simulated trajectories of cells with expected paths on the cell lineage tree, our method can infer which part of the network is responsible for a specific cell fate decision. The encoded rules for this network (i.e. Fig. 1b) are provided in S4 of Supplementary material. Applied to

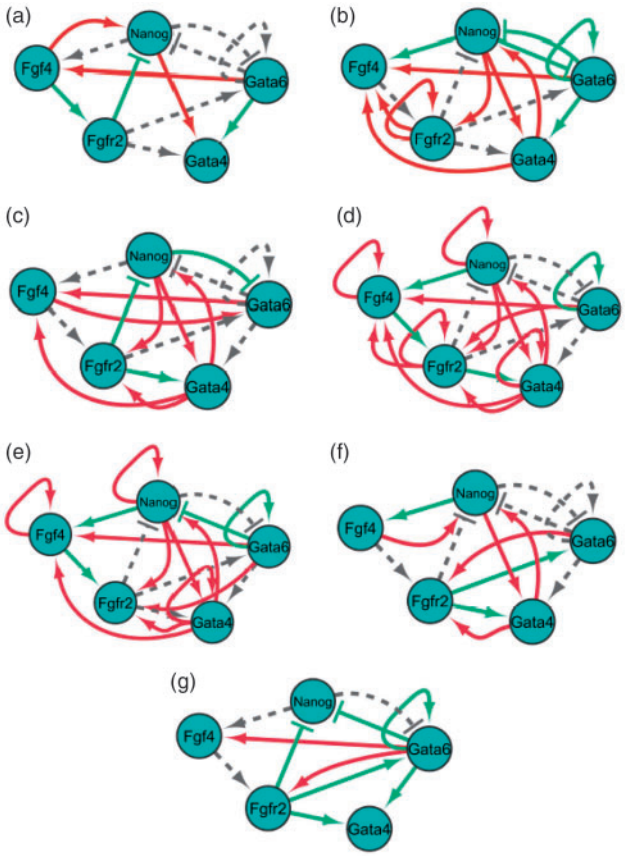

Fig. 3. The inferred networks for ICM →PE + EPI by: **(a)** Bayesian network, **(b)** BoolNet, **(c)** Random Forest, **(d)** Elastic-Net, **(e)** LASSO, **(f)** Treegl and **(g)** SingCellNet, where solid green and red lines denote TPs and FPs respectively, and gray dashed lines denote FNs

Robson's single-cell gene expression data, our method is able to identify the significant rules (i.e. rules with high probabilities) for each lineage formation, shown in Table 3 (see S4 of Supplementary material for more details).

As shown in Table 3, the most significant rule for 16C → TE is 1 → *Cdx2*. Gene *Cdx2* is the main regulator for TE cell type (Takaoka and Hamada, 2012; Yamanaka and Ralston, 2010). Signaling pathways such as the Hippo pathway are known to play important roles in TE/ICM segregation, by activating *Cdx2* in TE while blocking it in ICM (Oron and Ivanova, 2012). Consistent with the observations, our algorithm assigned the highest probability to the rule 1 → *Cdx2* in 16C → TE. The second significant rule *Oct4*|*Nanog* → *Sox2* may suggest that, during the TE formation, *Sox2* needs to be suppressed, which is achieved by the inhibition of its activator *Oct4* or *Nanog* by *Cdx2*. In other words, *Oct4* or *Nanog* passes the inhibition from *Cdx2* to *Sox2*. The active rule for 16C → ICM is *Cdx2*& ∼(*Oct4*|*Nanog*) → *Cdx2*, which shows the inhibition of *Cdx2* by *Oct4* (as well as *Nanog*). It has been shown that the antagonism between *Cdx2* and *Oct4* (as well as *Nanog*) directs and reinforces the cell fate to be either TE or ICM (Oron and Ivanova, 2012; Takaoka and Hamada, 2012; Yamanaka and Ralston, 2010). The target of this rule is *Cdx2*, which indicates that the driving force of ICM formation may be from the suppression of *Cdx2*. Unlike TE formation, where the driving force is from signaling pathways, cells adopt the ICM cell fate mainly by negative regulation of TE regulators (Nishioka *et al.*, 2009; Oron and Ivanova, 2012).

Table 3. Significant rules for lineage decisions

Lineage decisions	Significant rules
16C → TE	1 → *Cdx2*; *Oct4*|*Nanog* → *Sox2*
16C → ICM	*Cdx2*& ∼(*Oct4*|*Nanog*) → *Cdx2*
ICM → PE	*Nanog* → *Fgf4*
ICM → EPI	∼*Nanog*&(*Gata6*|*Fgfr2*) → *Gata6*

For ICM → PE, the rule with the highest probability is *Nanog* → *Fgf4*, which reflects the following aspect of PE/EPI specification. *Fgf4*/*Fgfr2* is a ligand-receptor pair in FGF signaling pathway, which is essential for PE differentiation. First *Fgf4* and *Fgfr2* are both activated. Then *Fgfr2* activates *Gata6* which is the marker gene of the PE cell type. *Nanog* in EPI progenitors activates *Fgf4* (i.e. *Nanog* → *Fgf4*), which would further induce PE-specific genes in PE progenitors through *Fgf4* → *Fgfr2* and complete the program of PE differentiation (Chazaud *et al.*, 2006; Yamanaka *et al.*, 2010; Frankenberg *et al.*, 2011). The significant rule for ICM → EPI inferred by our algorithm is ∼*Nanog*&(*Gata6*|*Fgfr2*) → *Gata6*, which shows the inhibition of *Gata6* from *Nanog*. This is consistent with literature that EPI formation is obtained by the suppression of *Gata6* (Frankenberg *et al.*, 2011; Oron and Ivanova, 2012).

Overall, our method can infer the operational interactions in the GRN for specific cell fate decisions, which are consistent with experimental literature, and can shed new light on mechanism of cell development by analysis of the single-cell data.

4 Discussion

In this article, we have proposed a novel method to uncover regulatory circuits of cell fate decisions from single-cell transcriptional data. We mapped the transcriptional dynamics of cell development to the STG generated by an asynchronous PBN, and defined a score to evaluate how well a candidate network fits the data. To learn the optimal network structure, a GA was employed to search for the network with the most natural cell state transitions according to the cell lineage tree. Applied to real single-cell expression data and validated against benchmark networks from biological experiments, our method outperformed most of the other methods for GRN inference. Through simple adaption, our method could also be used to identify operational interactions in GRN responsible for cell fate decisions.

As one of the first methods for network analysis of single-cell data, our method still needs to address several issues. First, it has been verified only on Robson's dataset, partly due to the dearth of publicly available single-cell data and benchmark networks. Nevertheless, we believe that the strategy of our method can be naturally applied to other single-cell datasets in many biological contexts (e.g. cancer, stem cell). Second, although our method achieved good prediction performance (measured by precision, sensitivity and F-measure), it is currently slower than the other methods, partly because our method was implemented in Matlab, while others were in Java or C++. Also, we used GA, which is known to be computationally intensive. In future, we will speed up our method by parallel computing, and other strategies (e.g. Qian *et al.*, 2010). Third, the procedure of cell lineage commitment is often triggered by external stimulus, e.g. signaling events from neighboring cells, which has not been explicitly included in our current model. In future, this could be modeled as a perturbation that pushes a cell state to a different point in a state space without following any internal regulatory rules.

Funding

This work was supported by MOE AcRF Tier 1 Seed Grant on Complexity [RGC 2/13, M4011101.020], Ministry of Education Singapore.

Conflict of Interest: none declared.

References

Andrecut,M. *et al.* (2011). A general model for binary cell fate decision gene circuits with degeneracy: indeterminacy and switch behavior in the absence of cooperativity. *PLoS One,* **6**, e19358.

Bonzanni,N. *et al.* (2013). Hard-wired heterogeneity in blood stem cells revealed using a dynamic regulatory network model. *Bioinformatics,* **29**, i80–i88.

Calzone,L. *et al.* (2010). Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLoS Comput. Biol.,* **6**, e1000702.

Chazaud,C.*et al.* (2006). Early lineage segregation between epiblast and primitive endoderm in mouse blastocysts through the Grb2-MAPK pathway. *Develop. Cell,* **10**, 615–624.

Chickering,D.M. (2002). *The WinMine Toolkit*. Redmond, WA: Microsoft.

De Jong,H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.,* **9**, 67–103.

De Jong,K. (1988). Learning with genetic algorithms: an overview. *Mach Learn,* **3**, 121–138.

Ding,C. and Peng,H. (2005). Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol.,* **3**, 185–205.

Feiglin,A. *et al.* (2012). Static network structure can be used to model the phenotypic effects of perturbations in regulatory networks. *Bioinformatics,* **28**, 2811–2818.

Frankenberg,S. *et al.* (2011). Primitive endoderm differentiates via a three-step mechanism involving Nanog and RTK signaling. *Develop. Cell,* **21**, 1005–1013.

Fujita,A. *et al.* (2007). Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC Syst. Biol.,* **1**, 39.

Guo,G. *et al.* (2010). Resolution of cell fate decisions revealed by single-cell gene expression analysis from zygote to blastocyst. *Develop. Cell,* **18**, 675–685.

Hashimoto,T. *et al.* (2012). Lineage-based identification of cellular states and expression programs. *Bioinformatics,* **28**, i250–i257.

Hecker,M. *et al.* (2009). Gene regulatory network inference: Data integration in dynamic models—a review. *Biosystems,* **96**, 86–103.

Holland,J.H. (1992). Genetic algorithms. *Sci. Am.,* **267**, 66–72.

Hoppe,P.S. *et al.* (2014). Single-cell technologies sharpen up mammalian stem cell research. *Nat. Cell Biol.,* **16**, 919–927.

Huang,S. (2010). Cell lineage determination in state space: a systems view brings flexibility to dogmatic canonical rules. *PLoS Biol.,* **8**, e1000380.

Huang,S. and Kauffman,S.A. (2012). Complex gene regulatory networks–from structure to biological observables: cell fate determination. In: R.A., Meyers (ed.) *Computational Complexity: Theory, Techniques, and Applications*, Springer, New York, pp. 527–560.

Huang,S. *et al.* (2009). Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. *Semin. Cell Develop. Biol.,* **20**(7), 869–876.

Irrthum,A. *et al.* (2010). Inferring regulatory networks from expression data using tree-based methods. *PLoS One,* **5**, e12776.

Kauffman,S. (1969). Homeostasis and differentiation in random genetic control networks. *Nature,* **224**, 177–178.

Kimura,S. *et al.* (2005). Inference of s-system models of genetic networks using a cooperative coevolutionary algorithm. *Bioinformatics,* **21**, 1154–1163.

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.

Li,F. *et al.* (2004). The yeast cell-cycle network is robustly designed. *Proc. Natl. Acad. Sci. U S A,* **101**, 4781–4786.

Li,J. *et al.* (2010). Negative correlations in collaboration: concepts and algorithms. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM New York, pp. 463–472.

Macarthur,B.D. *et al.* (2009). Systems biology of stem cell fate and cellular reprogramming. *Nat. Rev. Mol. Cell Biol.,* **10**, 672–681.

Maduranga,D. *et al.* (2013). Inferring gene regulatory networks from time-series expressions using Random Forests ensemble. In: *Pattern Recognition in Bioinformatics*. Springer Berlin Heidelberg, pp. 13–22.

Meyer,P. *et al.* (2014). Network topology and parameter estimation: from experimental design methods to gene regulatory network kinetics using a community based approach. *BMC Syst. Biol.,* **8**, 13.

Müssel,C. *et al.* (2010). BoolNet—An R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics,* **26**, 1378–1380.

Nishioka,N. *et al.* (2009). The Hippo signaling pathway components Lats and Yap pattern Tead4 activity to distinguish mouse trophectoderm from inner cell mass. *Develop. Cell,* **16**, 398–410.

Oliveri,P. and Davidson,E.H. (2004). Gene regulatory network controlling embryonic specification in the sea urchin. *Curr. Opin. Genet. Develop.,* **14**, 351–360.

Oron,E. and Ivanova,N. (2012). Cell fate regulation in early mammalian development. *Phys. Biol.,* **9**, 045002.

Pal,R. *et al.* (2005). Generating boolean networks with a prescribed attractor structure. *Bioinformatics,* **21**, 4021–4025.

Parikh,A.P. *et al.* (2011). TREEGL: reverse engineering tree-evolving gene networks underlying developing biological lineages. *Bioinformatics,* **27**, i196–i204.

Qian,X. *et al.* (2010). State reduction for network intervention in probabilistic boolean networks. *Bioinformatics,* **26**, 3098–3104.

Rajapakse,J.C. and Mundra,P.A. (2011). Stability of building gene regulatory networks with sparse autoregressive models. *BMC Bioinformatics,* **12**(Suppl. 13), S17.

Rossant,J. and Tam,P.P. (2009). Blastocyst lineage formation, early embryonic asymmetries and axis patterning in the mouse. *Development,* **136**, 701–713.

Samaga,R. *et al.* (2009). The logic of EGFR/ErbB signaling: theoretical properties and analysis of high-throughput data. *PLoS Comput. Biol.,* **5**, e1000438.

Samsonova,A.A. *et al.* (2007). Prediction of gene expression in embryonic structures of *Drosophila melanogaster*. *PLoS Comput. Biol.,* **3**, e144.

Schlitt,T. and Brazma,A. (2007). Current approaches to gene regulatory network modelling. *BMC Bioinformatics,* **8**(Suppl. 6), S9.

Takaoka,K. and Hamada,H. (2012). Cell fate decisions and axis determination in the early mouse embryo. *Development,* **139**, 3–14.

Tang,F. *et al.* (2011). Development and applications of single-cell transcriptome analysis. *Nat. Methods,* **8**, S6–S11.

Tournier,L. and Chaves,M. (2009). Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics. *J. Theor. Biol.,* **260**, 196–209.

Wu,M. *et al.* (2013). Engineering of regulated stochastic cell fate determination. *Proc. Natl. Acad. Sci. U S A,* **110**, 10610–10615.

Xu,H. *et al.* (2014). Construction and validation of a regulatory network for pluripotency and self-renewal of mouse embryonic stem cells. *PLoS Comput. Biol.,* **10**, e1003777.

Yamanaka,Y. and Ralston,A. (2010). Early embryonic cell fate decisions in the mouse. In: *The Cell Biology of Stem Cells: Advances in Experimental Medicine and Biology,* Springer US, 695, pp. 1–13.

Yamanaka,Y. *et al.* (2010). FGF signal-dependent segregation of primitive endoderm and epiblast in the mouse blastocyst. *Development,* **137**, 715–724.