BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

BTR: training asynchronous Boolean models using single-cell expression data

Chee Yee Lim[1], Huange Wang[1], Steven Woodhouse[1], Nir Piterman[2], Lorenz Wernisch[3], Jasmin Fisher[4,5] and Berthold Göttgens[1*]

Abstract

Background: Rapid technological innovation for the generation of single-cell genomics data presents new challenges and opportunities for bioinformatics analysis. One such area lies in the development of new ways to train gene regulatory networks. The use of single-cell expression profiling technique allows the profiling of the expression states of hundreds of cells, but these expression states are typically noisier due to the presence of technical artefacts such as drop-outs. While many algorithms exist to infer a gene regulatory network, very few of them are able to harness the extra expression states present in single-cell expression data without getting adversely affected by the substantial technical noise present.

Results: Here we introduce BTR, an algorithm for training asynchronous Boolean models with single-cell expression data using a novel Boolean state space scoring function. BTR is capable of refining existing Boolean models and reconstructing new Boolean models by improving the match between model prediction and expression data. We demonstrate that the Boolean scoring function performed favourably against the BIC scoring function for Bayesian networks. In addition, we show that BTR outperforms many other network inference algorithms in both bulk and single-cell synthetic expression data. Lastly, we introduce two case studies, in which we use BTR to improve published Boolean models in order to generate potentially new biological insights.

Conclusions: BTR provides a novel way to refine or reconstruct Boolean models using single-cell expression data. Boolean model is particularly useful for network reconstruction using single-cell data because it is more robust to the effect of drop-outs. In addition, BTR does not assume any relationship in the expression states among cells, it is useful for reconstructing a gene regulatory network with as few assumptions as possible. Given the simplicity of Boolean models and the rapid adoption of single-cell genomics by biologists, BTR has the potential to make an impact across many fields of biomedical research.

Keywords: Asynchronous Boolean model, Single-cell gene expression, Model learning, Network reconstruction, BOOLEAN scoring function, Executable model

Background

The control of gene expression is tightly regulated by complex gene regulatory networks to achieve cell type specific expression, for example in embryonic [1] and blood development [2]. Moreover, dysregulation of gene expression can lead to disease development, including malignant disease such as leukaemia [3]. A better understanding of gene regulatory networks will therefore not only advance our understanding of fundamental biological processes such as tissue development, but also provide mechanistic insights into disease processes. The earlier versions of high-throughput expression profiling techniques were limited to measuring average gene expression across large pools of cells. By contrast, recent technological improvements have made it possible to perform expression profiling in single cells (See [4] for review). Protocols for the single-cell equivalent of microarray [5], qPCR [6] and RNA sequencing [7] have been developed. One of the key advantages of single cell expression profiling is that it enables the analysis of cells that are rare in number, such as tissue stem cells. In

* Correspondence: bg200@cam.ac.uk
[1]Department of Haematology, Wellcome Trust and MRC Cambridge Stem Cell Institute, Cambridge Institute for Medical Research, University of Cambridge, Hills Road, Cambridge CB2 0XY, UK
Full list of author information is available at the end of the article

addition, obtaining the expression profiles of single cells is very useful for dissecting the heterogeneity within seemingly homogenous cell populations [2, 8–12].

Because single cell analysis commonly reports expression states for hundreds of individual cells, this unique information offers new opportunities for the development of algorithms that can reconstruct gene regulatory networks. Many network inference algorithms are available [13], which are based on regression, correlation, mutual information and Bayesian networks. However, most of these network inference algorithms only generate a network with static representation of gene interactions. In contrast, changes in network dynamics can be described by using dynamic models, which possess different levels of granularity and precision ranging from the simpler Boolean models to more complex differential equation-based models. More complex models such as differential equation-based models offer high precision predictions, and have been used to describe gene regulatory networks [14–17]. However, such models rely on a higher number of parameters which are often difficult to obtain and verify. In contrast, a Boolean model is one of the simplest models that can describe the dynamics of a system without the need of many parameters (For reviews, see [18, 19]). In a Boolean model, each gene can take a value of 0 or 1, which represents the absence or presence of gene expression respectively. The interactions among genes in a Boolean model are described by Boolean operators like AND, OR and NOT, which closely resembles how biologists describe such interactions. Boolean models were first used to study gene regulatory networks by Kauffman in the 1970s, and since then have been used extensively to study different biological systems [20–23].

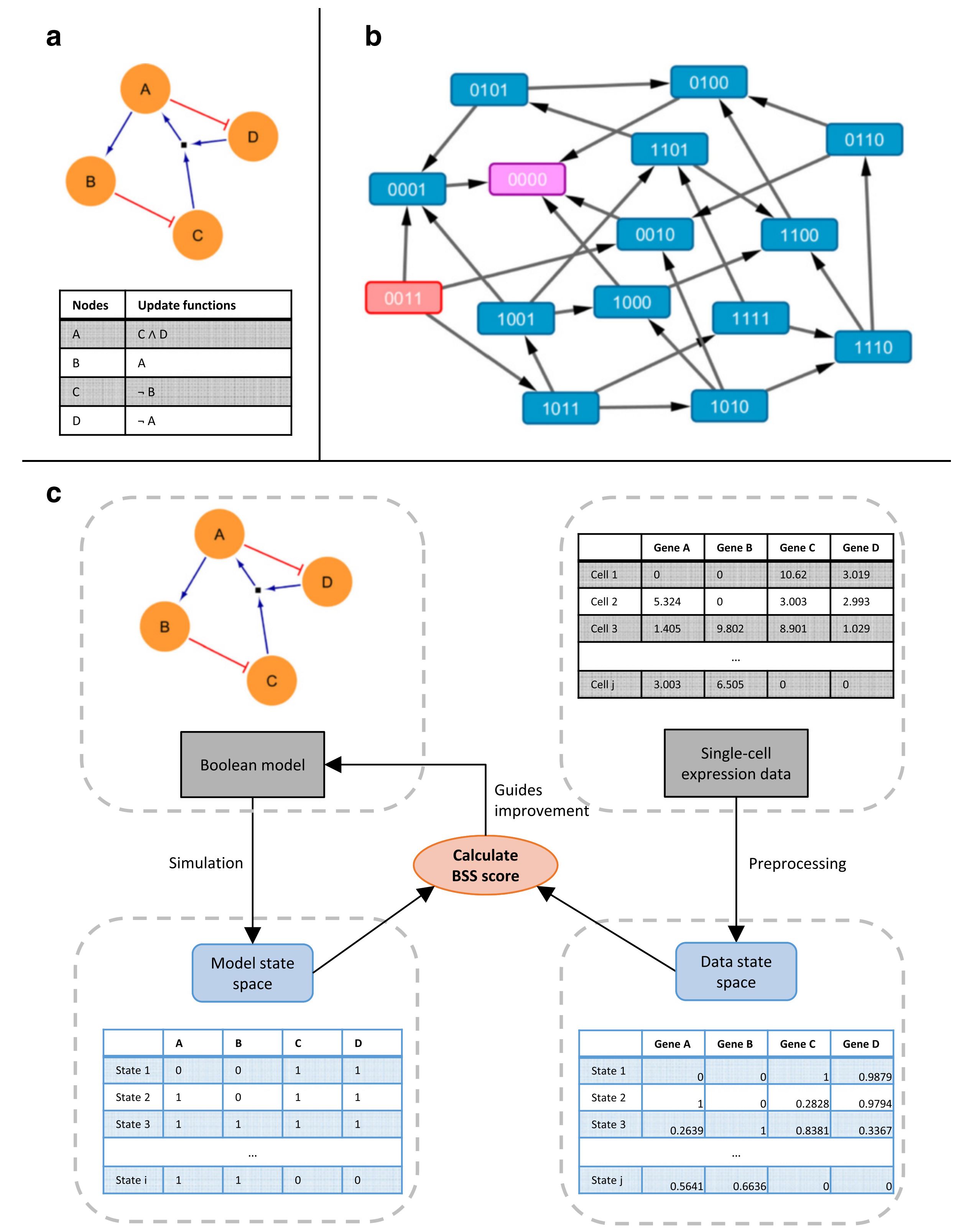
While single-cell expression data offers the advantage of capturing expression profiles at single cell resolution, single-cell expression data are noisier than conventional bulk analysis. The technical noise in single-cell expression data arises due to the low amount of input mRNAs in a single cell. This leads to two major sources of technical noise, which are PCR amplification bias and drop-outs [24]. Drop-outs in particular, which represent false negatives where genes are recorded as not expressed due to the low efficiency of mRNA capture from single cells, represent a substantial portion of the technical noise in single-cell expression data. Therefore, network inference techniques that are robust to the effect of drop-outs are required when reconstructing networks using single-cell expression data. Boolean models are relatively robust to the presence of drop-outs due to the binarisation of expression values. Two recent studies reported algorithms for inferring Boolean models from single-cell expression data [2, 25]. Chen et. al. developed SingCellNet, which uses a genetic algorithm to construct probabilistic Boolean models from expected trajectories through cell states [25]. However, SingCellNet is restricted to small networks with less than 10 genes, and it only determines the network structure and transition probabilities from single-cell expression data. The Boolean rules in SingCellNet are constructed via manual curation from the literature. In another study, SCNS was developed by Moignard et. al. to infer an asynchronous Boolean model by analysing trajectories through a state transition graph [2]. In order to infer a Boolean model using SCNS, a connected state transition graph is required, which can be difficult to obtain from single-cell expression data. This is because the higher the number of genes to be included in SCNS, the more cells will be required to build a connected state transition graph. In addition, SCNS can only infer network structure by using discretised expression data, which not only leads to the loss of information, but also makes SCNS sensitive to the discretisation method used. Lastly, both SingCellNet and SCNS rely on known general trajectories through the cell states, which require single-cell expression data from at least two cell types with known relationships.

Here, we present a model learning algorithm BTR (BoolTraineR), that is able to reconstruct and train asynchronous Boolean models using single-cell expression data. BTR differs from other algorithms described above in that it can infer both network structure and Boolean rules without needing information on trajectories through cell states. We developed a scoring function based on the Boolean framework, which performed favourably in comparison to a scoring function for Bayesian network. We show that BTR outperforms other network inference algorithms when initial networks are supplied. Lastly, we demonstrate the capability of BTR by training Boolean models using single-cell qPCR and RNA-Seq data from haematopoietic studies.

Results and discussion

A framework for scoring Boolean models with single cell expression data

A Boolean model B is made up of n genes $x_1, \ldots, x_n$ and n update functions $f_1, \ldots, f_n : \{0, 1\}^n \rightarrow \{0, 1\}$ each associated with a gene (Fig. 1a). Each gene can take a value $x \in \{0, 1\}$, which represents the absence or presence of gene expressions. Each update function f is expressed in terms of Boolean logic by specifying the relationships among genes $x_1, \ldots, x_n$ using Boolean operators AND ($\wedge$), OR ($\vee$) and NOT ($\neg$). The main difference of asynchronous with other Boolean models is the update scheme used during simulation. An asynchronous Boolean model uses the asynchronous update scheme, which specifies that at most one gene is updated between two consecutive states. Asynchronous updating is critical when modelling developmental systems that generate distinct differentiated cell types from a common progenitor,



Nodes	Update functions
A	C ∧ D
B	A
C	¬ B
D	¬ A

	Gene A	Gene B	Gene C	Gene D
Cell 1	0	0	10.62	3.019
Cell 2	5.324	0	3.003	2.993
Cell 3	1.405	9.802	8.901	1.029
…				
Cell j	3.003	6.505	0	0

	A	B	C	D
State 1	0	0	1	1
State 2	1	0	1	1
State 3	1	1	1	1
…				
State i	1	1	0	0

	Gene A	Gene B	Gene C	Gene D
State 1	0	0	1	0.9879
State 2	1	0	0.2828	0.9794
State 3	0.2639	1	0.8381	0.3367
…				
State j	0.5641	0.6636	0	0

Fig. 1 (See legend on next page.)

(See figure on previous page.)
Fig. 1 Boolean model, asynchronous simulation and the framework underlying BTR. **a** A Boolean model can be expressed graphically in terms of nodes and edges, as well as in tabular form in terms of update functions. Note that the small black node refers to AND interaction. **b** The asynchronous update scheme is best explained with the use of a graph representation of state space, in which each connected state differs in only one node. Starting from the initial state $s_1 = \{0, 0, 1, 1\}$ and evaluated using the update functions in (**a**), asynchronous simulation produces a model state space with 15 states. The initial state is shown in *red node*, while the final steady state is shown in *pink node*. **c** The framework underlying BTR. A Boolean model can be simulated to give a model state space, while a single-cell expression data can be preprocessed to give a data state space. Boolean state space scoring function can then calculate the distance score between the model and data state spaces. Lastly, BTR uses the computed distance score to guide the improvement of the Boolean model through an optimisation process that minimises the distance between model and data state spaces

because synchronous updating generates fully deterministic models and therefore cannot capture the ability of a stem cell to mature into multiple different tissue cells.

A state in a Boolean model B is represented by a Boolean vector $s_t = \{x_{1t}, \ldots, x_{nt}\}$ at simulation step t. States can be generated from an initial state by systematically changing one variable at each step according to the Boolean function associated with that variable. If a state has already been encountered earlier, it is ignored. This results in a directed graph of states as exemplified in Fig. 1b, where any two connected states change in just one variable. When all the states in the directed graph are taken together, they represent a model state space. The initial state used in a simulation can be obtained from the expression values at time = 0 for a time-series expression dataset, or it can be obtained from the expression values of known parental cell types.

Of note, the model state space of an asynchronous Boolean model closely resembles a single-cell expression data. The model state space contains predicted expression states that are dictated by a known gene network that underlies a Boolean model; while the single-cell expression data can be viewed as a data state space which contains observed expression states that are dictated by an unknown gene network. By fine-tuning the network rules underlying the Boolean model, it should be possible to produce a predicted model state space that closely resembles an observed data state space, thereby allowing us to reconstruct the unknown gene network. BTR uses this framework to reconstruct a Boolean model from single-cell expression data (Fig. 1c). In this framework, a Boolean model is represented by its model state space, while a single-cell expression dataset is represented by its data state space. By utilising the novel Boolean state space (BSS) scoring function (See Methods), BTR evaluates how well a particular Boolean model explains the single-cell expression data by scoring the model state space with respect to the data state space. During the model training process, BTR uses a swarming hill climbing strategy to generate minimally modified Boolean models based on an initial Boolean model. These minimally modified Boolean models are then scored using the BSS scoring function, and BTR selects the best scoring Boolean models for the next iteration. By performing this process iteratively, BTR reconstructs the asynchronous Boolean model that can best explain a single-cell expression dataset.

Boolean state space scoring represents a powerful scoring function for Boolean models

How well BTR performs depends heavily on the performance of the BSS scoring function. Among different modelling frameworks, the Bayesian network framework is known to possess several well-established scoring functions that evaluate how well a particular network fits a given dataset. These scoring functions include log-likelihood, Bayesian information criterion (BIC), Bayesian Dirichlet and K2 (See [26, 27] for reviews). Since expression data have continuous values for gene expressions, we have selected the BIC scoring function, which can handle continuous variables, as a scoring function from the Bayesian network framework for comparison purpose.

BSS and BIC scoring functions were evaluated using synthetic data. The true network and expression data in the synthetic data were generated using GeneNetWeaver [28], which is also used in the DREAM5 network inference challenge [13]. In order to simulate the zero-inflated property of single-cell expression data due to the presence of drop-outs, we introduced zero inflation into the synthetic data as described in the Methods section. An ideal scoring function should give an increasing distance score, as the evaluated network becomes increasingly different from the true network. In order to test this, we generated a list of modified networks that are increasingly different from the true network in terms of edges. As Bayesian networks and Boolean frameworks imposed different network structure constraints, the modified networks were generated separately to give a list of modified Bayesian networks and another list of modified Boolean networks. Although the modified Bayesian and Boolean networks are not identical, they possess the same number of differing edges when compared to the true network, ranging from 2 edges up to 40 differing edges. Five independent benchmark data, each with a different true network, true data and modified models, were used in the evaluation of scoring functions.

By evaluating networks using zero-inflated synthetic data, both BSS and BIC scoring functions performed well when acyclic networks are considered (Fig. 2). Both

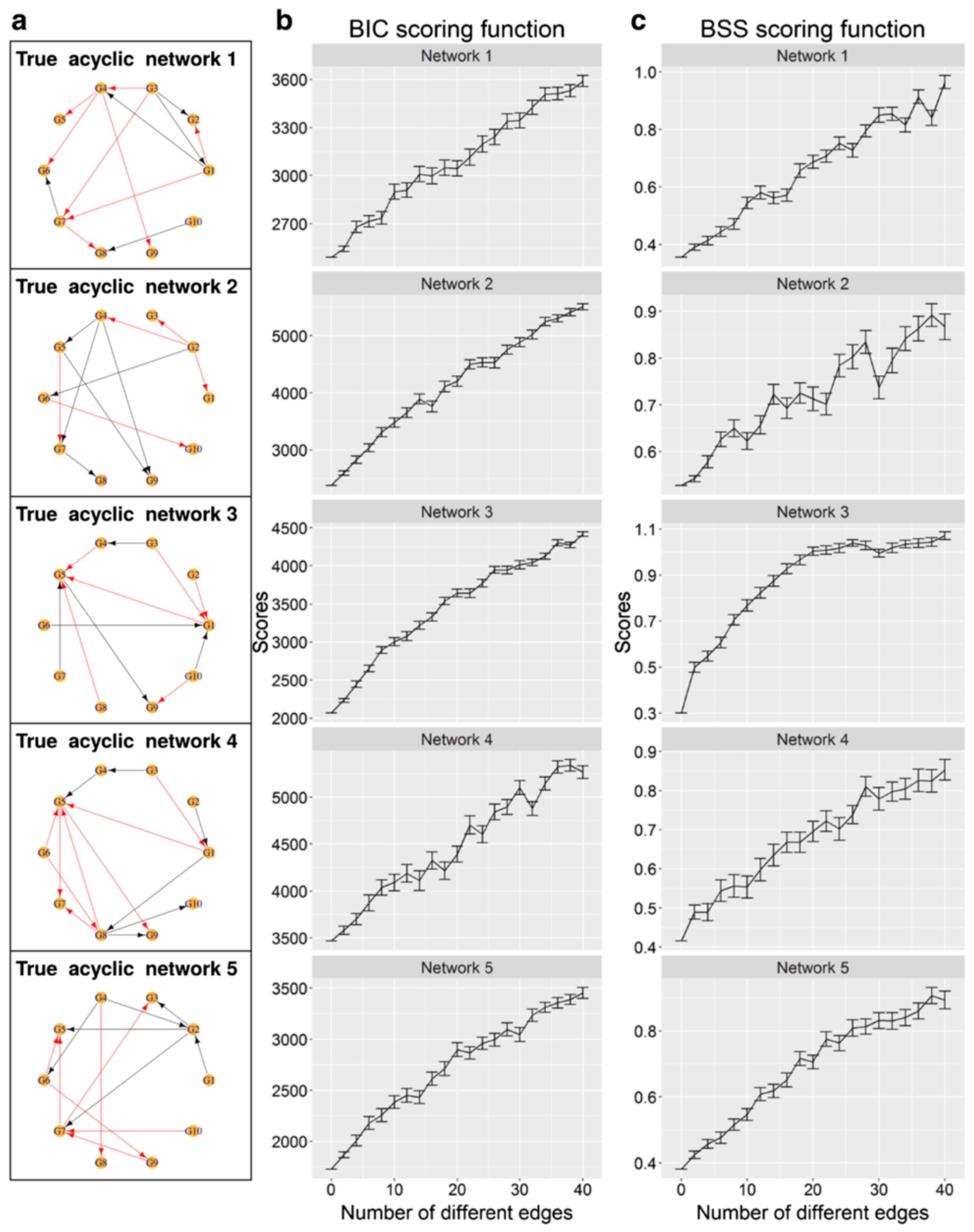


Fig. 2 (See legend on next page.)

(See figure on previous page.)

Fig. 2 BSS scoring function compares favourably with BIC scoring function on acyclic networks. **a** Acyclic networks generated from GeneNetWeaver that are designated as the true acyclic networks. Each node corresponds to a gene. *Black edges* indicate activation interactions, while red edges indicate inhibition interactions. Mean distance scores computed using **b** BIC scoring function and **c** BSS scoring function for modified networks that are increasingly different from the true network in terms of edges using zero-inflated synthetic expression data. The modified networks contain from two edges up to forty different edges when compared with the true network. Each data point is the mean distance score of 100 different random modified networks that contain the same number of different edges with respect to the true network. The error bar is the standard error of the mean

scoring functions were able to give increasing distance scores as the underlying networks become increasingly different from the true network. The BSS scoring function achieves this by considering the input expression data as a data state space, and then computing the distance score by comparing the data state space with the model state space simulated from a given network. It is expected that as a network become increasingly different, its model state space will become increasingly different from the data state space, which is reflected in the distance score as shown in Fig. 2c. To the best of our knowledge, this is the first time a scoring function that is based entirely on the Boolean modelling framework has been demonstrated to give comparable performance with a scoring function for Bayesian networks.

As indicated in the results for Network 2 (Fig. 2c), the BSS scoring function is dependent on the underlying true network structure in certain cases and will work better on distinguishing networks that are very different. However the BSS scoring function has a distinct advantage over scoring functions for Bayesian networks. The Bayesian networks are known to impose relatively strict constraints on permissible network structures, in particular Bayesian networks are not allowed to contain any cyclic network structure. Therefore scoring functions for Bayesian networks cannot be used to evaluate cyclic networks. Cyclic networks are ubiquitous in biological systems, in which cyclic motifs can be present in the form of negative and positive feedback loops. Boolean models on the other hand are allowed to have any number of cyclic motifs in the networks. Therefore, the BSS scoring function can be used to compute scores for cyclic networks. By using another five independent benchmark data with true networks that contain at least one cycle, the distance scores for modified networks were computed (Fig. 3). The distance scores for cyclic networks have more fluctuations compared to acyclic networks due to the presence of cyclic motifs. However, the general trend where the distance scores increase as the underlying networks become increasingly different from the true network was still observed.

We have also evaluated the series of acyclic and cyclic networks using non zero-inflated data (Additional file 1: Figure S1 & Additional file 2: Figure S2). When the results computed with non zero-inflated data are compared to the results computed using zero-inflated data, we can see that zero-inflation has no effect on BIC scores and a small effect on BSS scores that does not affect the general trend (Additional file 3: Figure S3). In summary, the relative mean scores that average across the results of all networks (Fig. 4) show that although the BIC scoring function performs slightly better than the BSS scoring function, the BSS scoring function has the advantage that it can evaluate cyclic networks.

BTR accurately infers the networks underlying synthetic datasets

Next, we compared the network inference performance of BTR with other well-known network inference algorithms. Two search algorithms guided by the BSS Boolean and BIC Bayesian network scoring functions were included in the comparison, indicated as BTR and BIC respectively. The search algorithms used for both scoring functions are both based on hill climbing. The additional network inference algorithms included in the comparison are BestFit [29], ARACNE [30], CLR [31], bc3net [32], GeneNet [33] and Genie3 [34] (See Methods for brief details on the algorithms).

By using the same synthetic networks, as well as both non zero-inflated and zero-inflated synthetic data, we performed network inference using the synthetic expression data alone without any extra information. In contrast to the DREAM5 challenge [13] which also provides perturbed expression data, only a single type of expression data is provided to all the network inference algorithms, which is the wild type time course expression data in steady state. For BTR, besides performing inference with only expression data (indicated as BTR-WO), we also performed inference with both expression data and initial networks (indicated as BTR-WI) to show that BTR is able to use initial networks with known network structure to improve the inference process. The initial networks are generated randomly to contain 18 edges that are different compared with the true networks. The performance of the network inference algorithms is assessed in terms of F-scores [35] (Fig. 5). In order to allow comparisons on the performance across all network inference algorithms tested, we calculated the F-scores based only on the presence or absence of edges, while ignoring any additional information such as the types of edges.

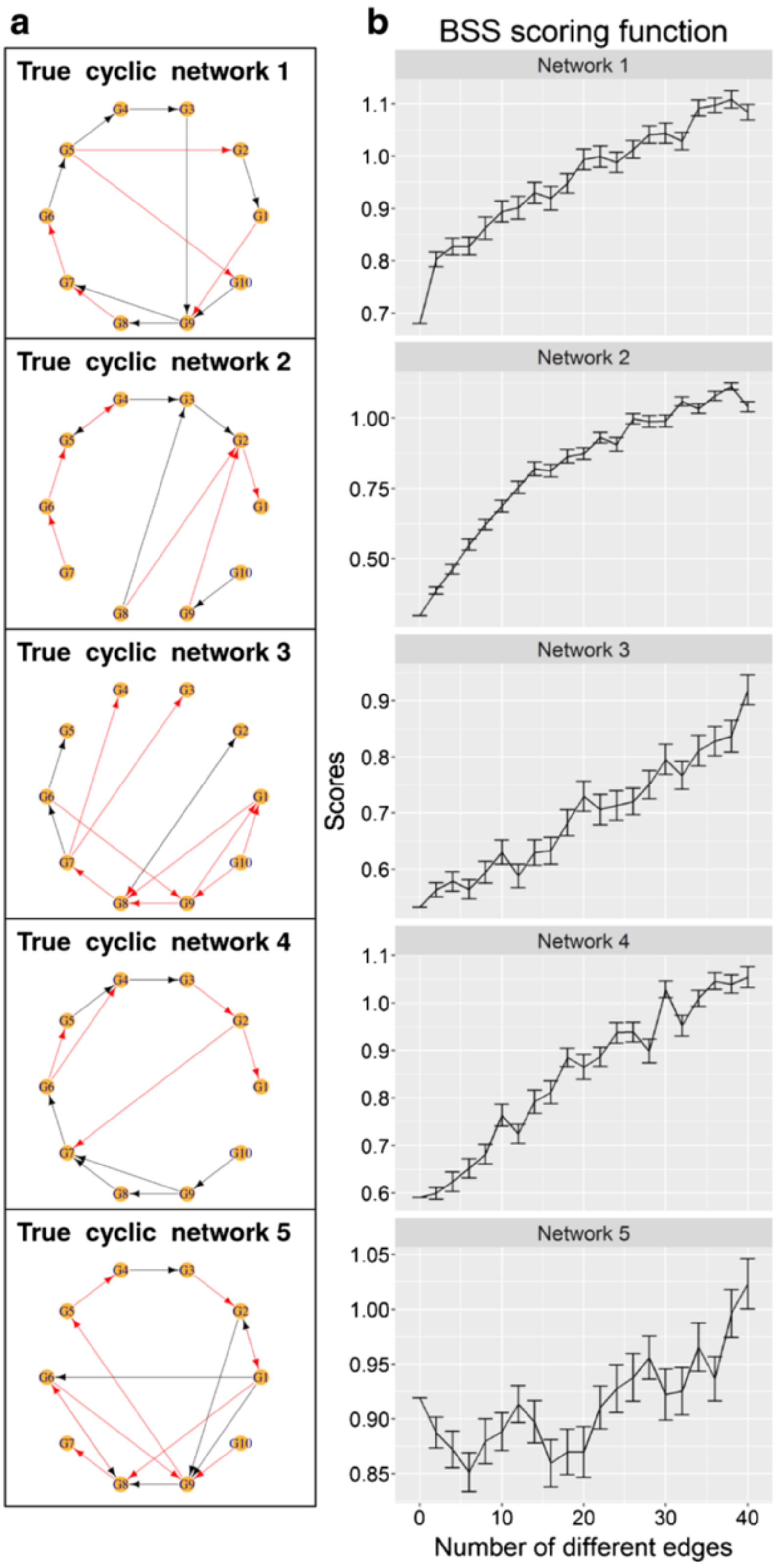


Fig. 3 (See legend on next page.)

(See figure on previous page.)
Fig. 3 BSS scoring function is able to calculate distance scores for cyclic networks. **a** Cyclic networks generated from GeneNetWeaver that are designated as the true cyclic networks. Each node corresponds to a gene. *Black edges* indicate activation interactions, while *red edges* indicate inhibition interactions. **b** Mean distance scores computed using BSS scoring function for modified networks that are increasingly different from the true network in terms of edges using zero-inflated synthetic expression data. The modified networks contain from two edges up to forty different edges when compared with the true network. Each data point is the mean distance score of 100 different random modified networks that contain the same number of different edges with respect to the true network. The error bar is the standard error of the mean

In terms of acyclic networks, the results show that the top inference algorithms using either non zero-inflated or zero-inflated data are BTR-WI, CLR, BIC and BTR-WO. As for cyclic networks, the top inference algorithms differ between using non zero-inflated and zero-inflated data. BTR-WI, BTR-WO, CLR and BC3NET gave the best performance with non zero-inflated data, while BTR-WI, ARACNE, GENIE3 and CLR gave the best performance with zero-inflated data. When all results are taken together, BTR-WI, CLR, BTR-WO and GENIE3 gave the best performance overall. Note that the ranking of network inference algorithms in this study differs from the ranking of the DREAM study because different scoring criteria are used (F-score is used here as opposed to the area under the precision-recall (AUPR) and receiver operating characteristic (AUROC) curves in the DREAM study); and the DREAM study was done using multiple types of synthetic data, such as expression data with gene perturbations. In general, the presence of drop-outs affects the performance of network inference algorithms in different ways (Fig. 5b). In cases such as bc3net and GeneNet, their performance decreases when drop-outs are present, while the impact of drop-outs on the performance of BTR is minimal. Interestingly, the performance of BestFit increases with the presence of drop-outs, possibly due to better binarisation of data due to the information given by drop-outs. As both BTR and BestFit are algorithms for inferring Boolean model, this result provides further support that Boolean models are robust to the presence of drop-outs in single-cell expression data.

When given an initial network as in BTR-WI, the BTR algorithm was able to perform very well in locating the true network. While the performance of the BTR algorithm without an initial network (BTR-WO) is comparable with other inference algorithms, BTR-WO scored less well compared to BTR-WI. This indicates that the greedy hill climbing search strategy implemented in BTR may not be able to traverse the solution space efficiently without any initial information. Taken together, while BTR can be used for reconstructing network models without initial information, BTR performed the best

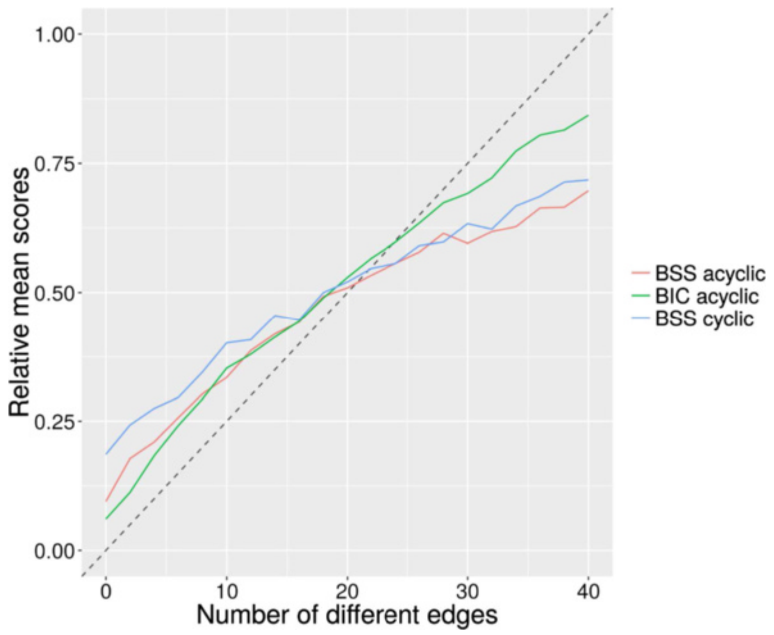


Fig. 4 Summary of BIC and BSS scoring functions. Mean scores have been calculated across all networks (five acyclic networks and five cyclic networks) for BIC and BSS scoring functions calculated using zero-inflated synthetic expression data. All scores have been standardised for comparison purpose, such that the scores range from 0 to 1

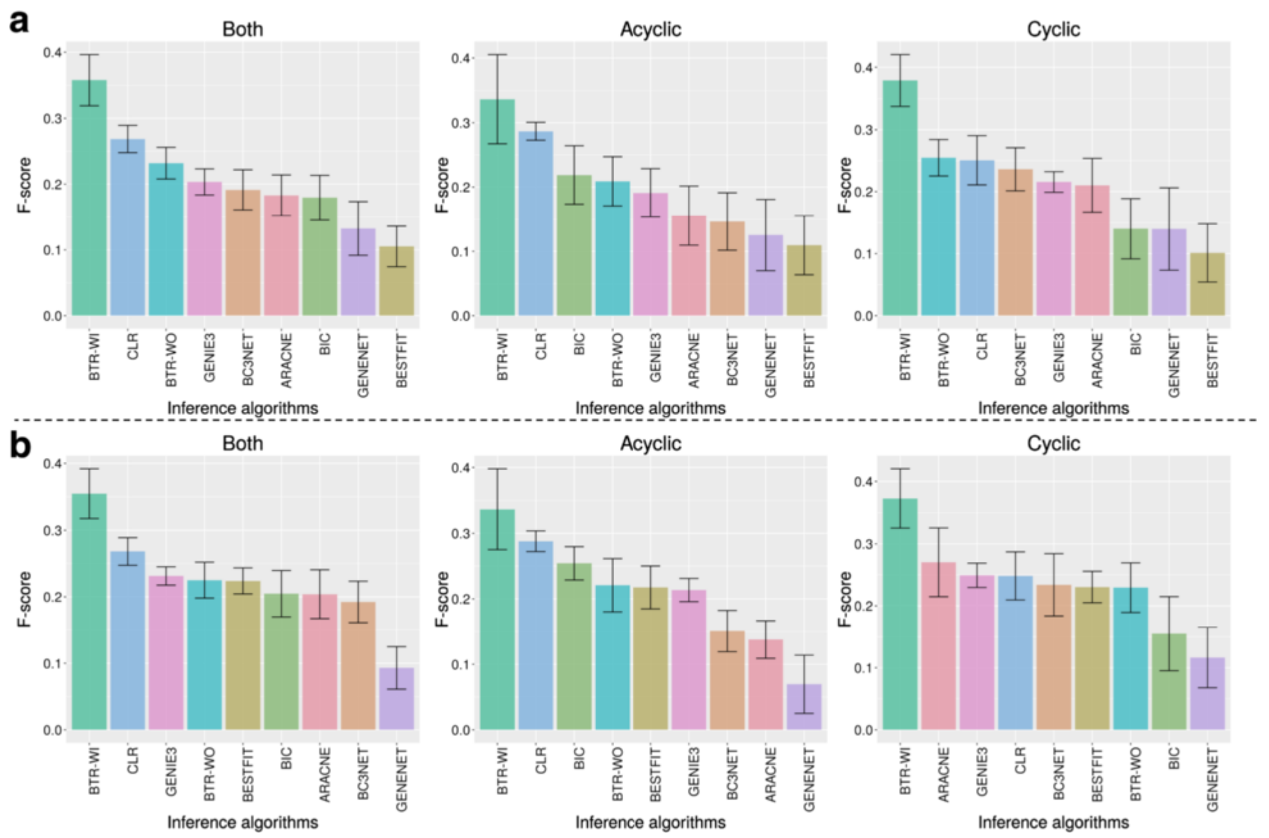


Fig. 5 BTR outperforms other network inference algorithms. Mean F-scores of network inference algorithms inferred using **a** non zero-inflated synthetic data and **b** zero-inflated synthetic data. Ten true synthetic networks (Five each for acyclic and cyclic networks) were used in the assessment of these network inference algorithms. Plots titled 'Both' show the combined results of acyclic and cyclic network inference. The error bar is the standard error of the mean

when it is used to train and improve on existing networks that contain a partially true structure. It is also worth noting that BTR produced a dynamic model with a directed underlying static network, in contrast to most other algorithms such as CLR that only produce an undirected static network.

BTR predicts gene interactions by training haematopoietic Boolean models

We next wanted to apply BTR to biological data to evaluate its utility to biologists. Haematopoiesis research has provided many paradigms for modern biological research, and was one of the first fields to embrace single cell expression profiling [5, 36, 37]. Moreover, literature curated Boolean network models have been reported both for blood stem cell maintenance and blood progenitor differentiation [38, 39]. The single-cell expression data used here includes single-cell qPCR and single-cell RNA-Seq data, which are both obtained from [10]. The two Boolean models will be referred to as the Bonzanni model [39] (Fig. 6a) and the Krumsiek model [38] (Fig. 6c). Both models had been constructed via manual literature curation by the authors of the original papers. The Bonzanni model aimed to capture haematopoietic stem cell (HSC) self-renewal capacity, while the Krumsiek model describes the differentiation process of the erythro-myeloid lineage in haematopoiesis.

We firstly trained the Bonzanni model using single-cell RNA-Seq data collected from HSCs. Compared to the original model, the resulting trained Bonzanni model (Fig. 6b) shows the deletions of ten gene interactions and the additions of thirteen gene interactions (Table 1). The state space of the trained Bonzanni model contains 1486 states when simulated using the initial state used in the original study (Fig. 7a). Of note, there are many densely connected transitional states in the state space, which may be related to the complexity of cell fate decision making processes in multipotent progenitor cells. Steady state analysis performed showed that the steady states of the trained Bonzanni model are almost identical to the steady states of the original Bonzanni model (Fig. 8a), except with the absence of cyclic steady states. The authors suggested that the cyclic steady states in the original Bonzanni model correspond to the self-renewal

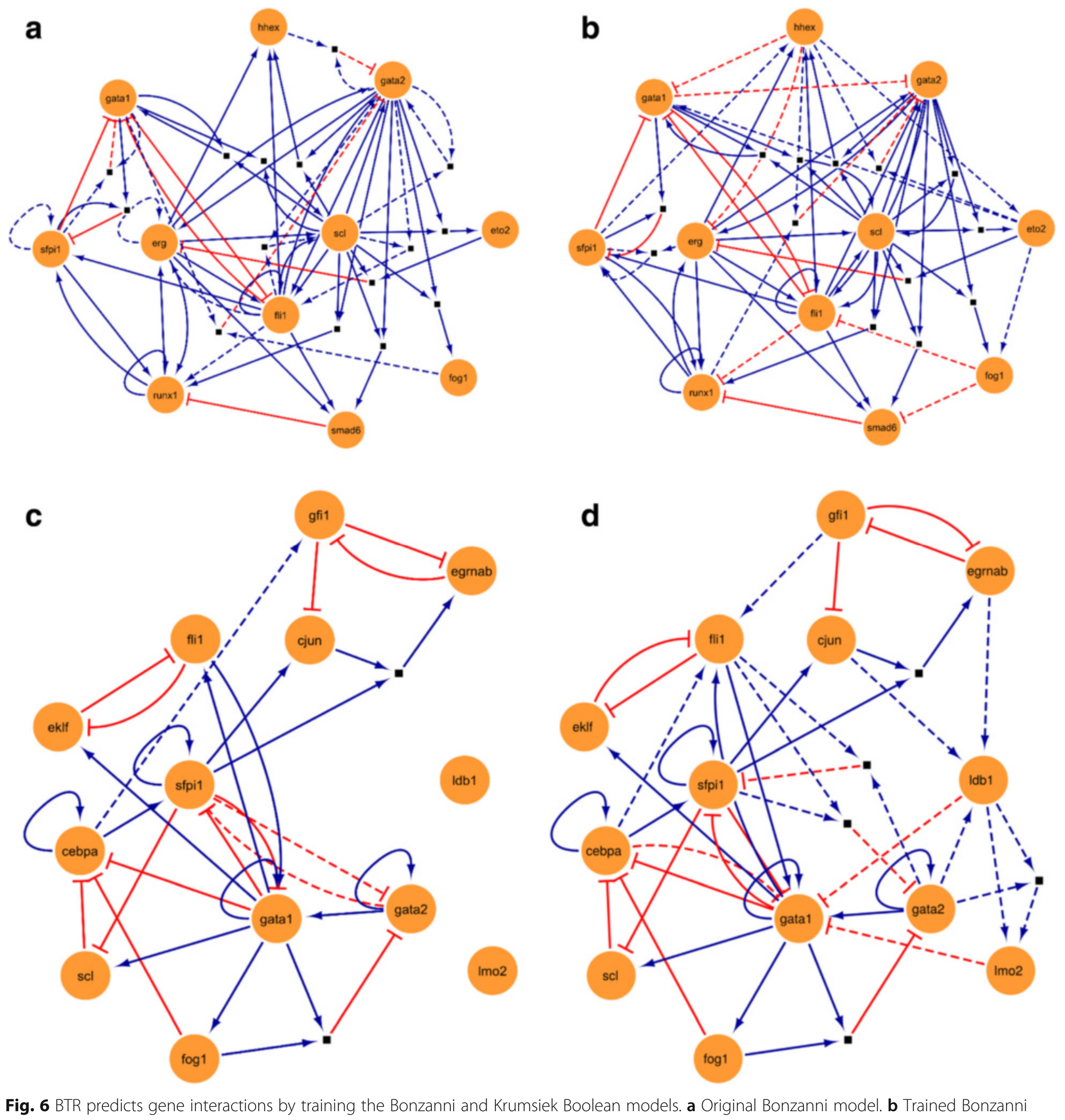


Fig. 6 BTR predicts gene interactions by training the Bonzanni and Krumsiek Boolean models. **a** Original Bonzanni model. **b** Trained Bonzanni model. **c** Original Krumsiek model. **d** Trained Krumsiek model. *Round orange nodes* indicate genes, *square black nodes* indicate AND gates that combine the two input gene interactions. *Blue edges* indicate activation interactions, *red edges* indicate inhibition interactions. *Dashed lines* in the original models indicate edges that are present in the original models, but are removed in the trained models. *Dashed lines* in the trained models indicate edges that are added to the trained models and are not present in the original models

maintenance loop in HSCs, which is not present in our trained model possibly because the number of cells profiled by single-cell RNA-seq is not enough to sufficiently capture the HSC self-renewal expression signature. We then trained the Krumsiek model by using single-cell qPCR data collected from over 450 cells along the erythro-myeloid lineage, which includes common myeloid progenitors, granulocyte-monocyte progenitors and myeloid-erythroid progenitors. In order to demonstrate that BTR can be used in cases where we may want to extend a current Boolean model by adding more genes to it, we have used BTR to train and add two additional genes to the Krumsiek model. The resulting trained Krumsiek model (Fig. 6d) contains three deleted gene interaction and twelve added gene

Table 1 Differences in gene interactions between original and trained Bonzanni and Krumsiek models

Models	Gene interactions	Suggested modifications to original model
Bonzanni	*Gata2* and *Hhex* inhibits *Gata2*	Deletion
	Scl and *Gata2* activates *Scl*	Deletion
	Fli1 and *Gata2* activates *Scl*	Deletion
	Gata2 and *Scl* activates *Fli1*	Deletion
	Fli1 activates *Runx1*	Deletion
	Fli1 activates *Erg*	Deletion
	Erg activates *Erg*	Deletion
	Gata1 and *Fog1* inhibits *Gata2*	Deletion
	Sfpi1 and *Gata1* inhibits *Gata1*	Deletion
	Sfpi1 activates *Sfpi1*	Deletion
	Sfpi1 and *Erg* activates *Sfpi1*	Addition
	Hhex and *Runx1* inhibits *Gata2*	Addition
	Eto2 and *Hhex* inhibits *Gata2*	Addition
	Sfpi1 activates *Hhex*	Addition
	Gata1 inhibits *Gata2*	Addition
	Fog1 inhibits *Smad6*	Addition
	Fog1 inhibits *Fli1*	Addition
	Eto2 activates *Gata1*	Addition
	Eto2 activates *Fog1*	Addition
	Hhex inhibits *Gata1*	Addition
	Hhex activates *Eto2*	Addition
	Hhex inhibits *Erg*	Addition
	Fli1 inhibits *Runx1*	Addition
Krumsiek	*Cebpa* activates *Gfi1*	Deletion
	Gata2 inhibits *Sfpi1*	Deletion
	Sfpi1 inhibits *Gata2*	Deletion
	Gfi1 activates *Fli1*	Activation
	Cebpa activates *Fli1*	Activation
	Fli1 and *Gata2* inhibits *Sfpi1*	Activation
	Fli1 and *Sfpi1* inhibits *Gata2*	Activation
	Cebpa inhibits *Gata1*	Activation
	EgrNab activates *Ldb1*	Activation
	cJun activates *Ldb1*	Activation
	Ldb1 inhibits *Gata1*	Activation
	Gata2 activates *Ldb1*	Activation
	Ldb1 activates *Lmo2*	Activation
	Ldb1 and *Gata2* activates *Lmo2*	Activation
	Lmo2 inhibits *Gata1*	Activation

Other gene interactions that were not modified are not listed in this table. Each gene interaction corresponds to an edge on the network

interactions (Table 1) when compared to the original Krumsiek model. For the two additional genes *Ldb1* and *Lmo2*, BTR has predicted gene interactions among *Ldb1*, *Lmo2*, *Fli1*, *Gata1* and *Gata2*. Previous studies have shown that genome-wide binding profiles for *Lmo2*, *Gata2* and *Fli1* show significant overlaps [40], and that *Ldb1* also occupies nearly all of the binding sites of *Gata2* [41], consistent with a model where these TFs engage in combinatorial interactions. The state space of the trained Krumsiek model contains 21 states when simulated using the initial state used in the original study (Fig. 7b). The two steady states reachable in this state space may correspond well to cell populations that are primed for the erythrocyte and myeloid lineage divergence. When examining the steady states reachable from all possible initial states, the trained Krumsiek model produces additional steady states when compared with the original model due to the addition of two extra genes (Fig. 8b), which may correspond to intermediate cell types along the erythro-myeloid differentiation pathway.

Taken together, the result suggests that both the trained Bonzanni and Krumsiek models have been trained by BTR to predict new gene interactions which give rise to interesting state spaces and steady state properties. Note that the state space of the trained Bonzanni model is substantially larger than the state space of the trained Krumsiek model due to the denser interactions among genes and a lower proportion of inhibitory edges in the trained Bonzanni model (Additional file 4: Figure S5).

Conclusions

We have developed the BTR model learning algorithm for training asynchronous Boolean models using single-cell expression data. The key component in BTR is a novel Boolean state space (BSS) scoring function, which BTR uses to infer a Boolean model through an optimisation process. We have shown that the new BSS scoring function is capable of giving meaningful scores to networks when compared with the BIC scoring function for Bayesian networks. We then showed that when compared to other network reconstruction algorithms, BTR gave the best result when initial networks were provided. In two case studies, we have demonstrated that BTR is capable of suggesting modifications to existing Boolean models based on information from single-cell qPCR and RNA-Seq data. Finally, we anticipate BTR to be a useful addition to the current toolbox for processing and understanding single-cell expression data, as it provides significant new capabilities for regulatory network modelling in a user-friendly way.

Methods

Definitions

A Boolean model B consists of n genes $x_1, \ldots, x_n$ and n update functions $f_1, \ldots, f_n : \{0, 1\}^n \rightarrow \{0, 1\}$, with each f_i being associated with gene x_i (Fig. 1a). Each gene x_i corresponds to a binary variable representing the expression value of the gene, i.e. $x \in \{0, 1\}$. Gene x_i is a target gene when it acts as a response variable and an input gene

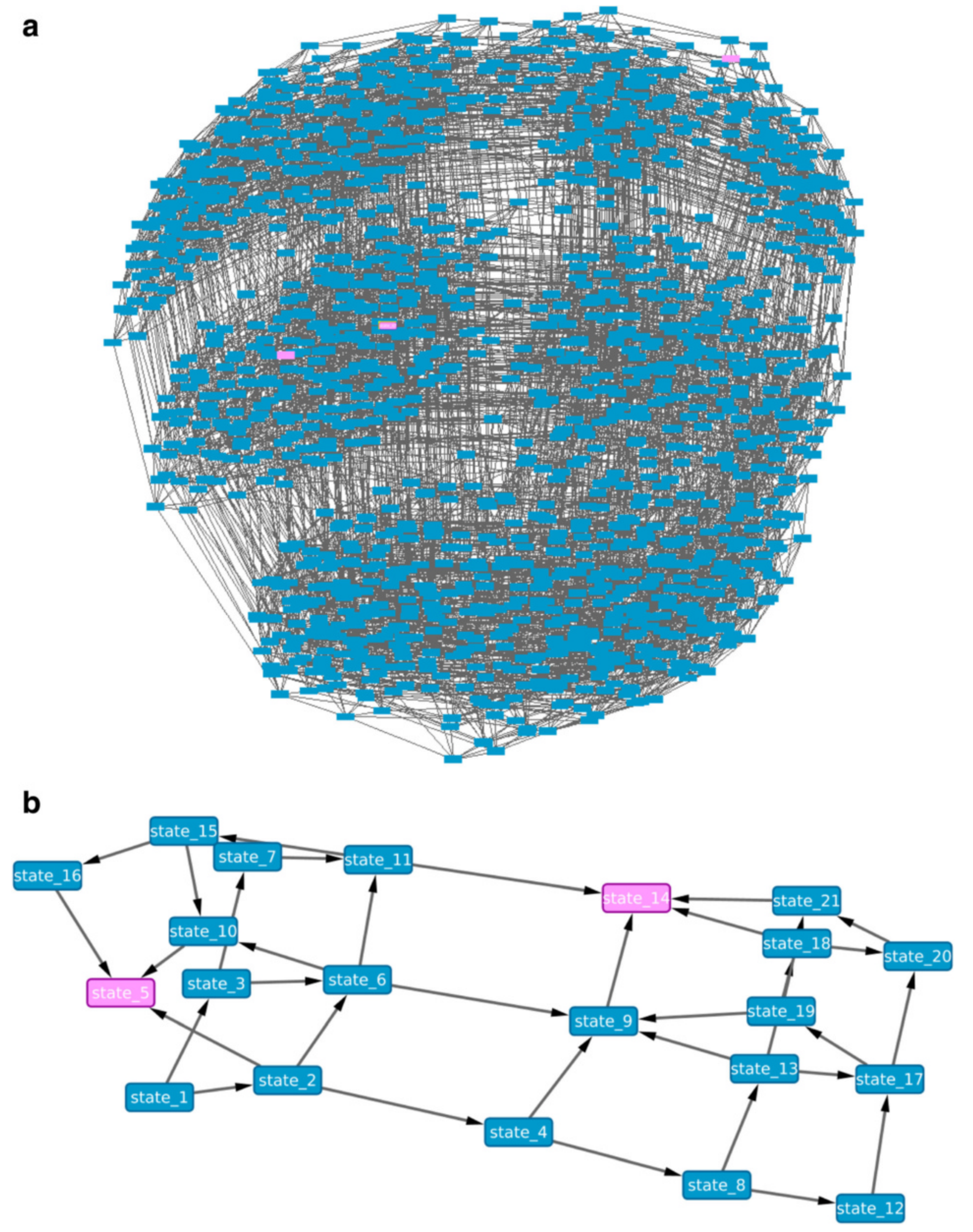
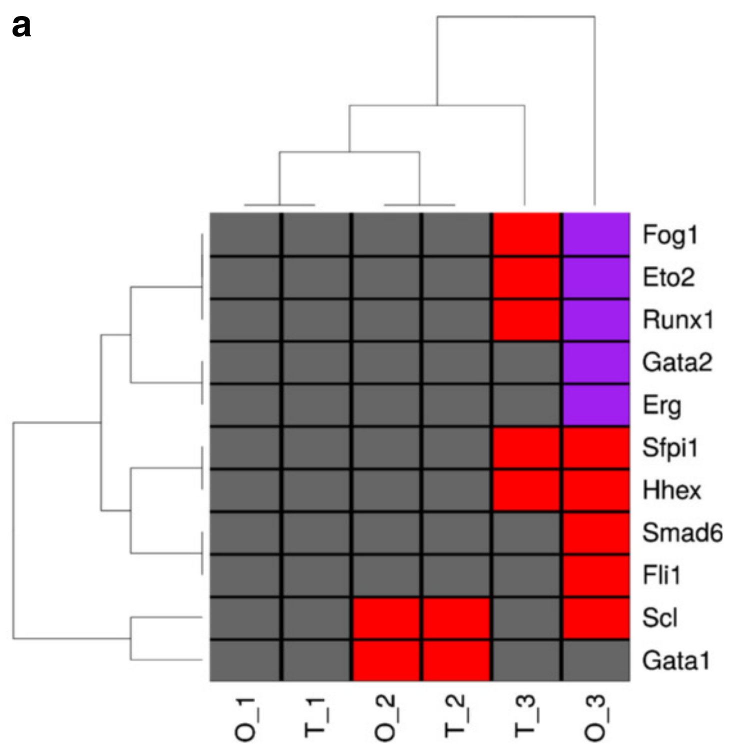
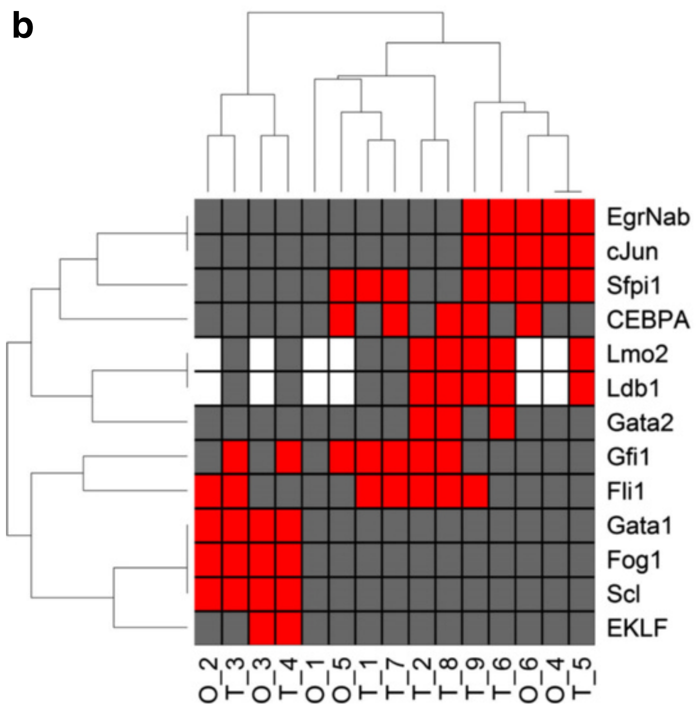


Fig. 7 State spaces for the trained Bonzanni and Krumsiek Boolean models. **a** State space of trained Bonzanni model. **b** State space of trained Krumsiek model. *Blue nodes* represent transitional model states, while *pink nodes* represent steady model states. Each *arrow* indicates transitions among states





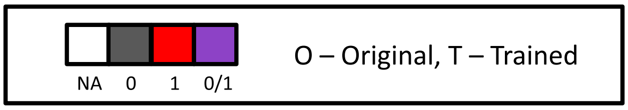


Fig. 8 (See legend on next page.)

(See figure on previous page.)
Fig. 8 Steady states for the Bonzanni and Krumsiek Boolean models. **a** Steady states of Bonzanni models. Both original and trained models contain two point steady states and one cyclic steady state each. **b** Steady states of Krumsiek models. Original model contains six point steady states, while trained model contains nine point steady states. *Black box* indicates expression is absent (i.e. 0), *red box* indicates expression is present (i.e. 1), *purple box* indicates expression can be absent or present (i.e. 0 or 1). *White box* is used to indicate that the additional genes included in trained Krumsiek model that are not present in the original Krumsiek model

when it acts as a predictor variable. Each update function f_i can be evaluated to give a value to a target gene x_i, and is expressed in terms of Boolean logic by specifying the relationships among a subset of the input genes $x_1, \ldots, x_n$ using Boolean operators AND ($\wedge$), OR ($\vee$) and NOT ($\neg$). An update function f_i consists of an activation clause and an inhibition clause in the form of:

$$(\textit{activation clause}) \wedge \neg\ (\textit{inhibition clause})$$

Each clause is individually expressed in disjunctive normal form, $(u_1) \vee (u_2) \vee (u_3) \vee \ldots \vee (u_n)$, where u represents a slot which can either take in a single input gene x_i or a conjunction of two input genes $x_i \wedge x_{i+1}$. An example update function $f_1(s_t)$ for a target gene x_1 with an input state s_t is given below:

$$x_1 = f_1(s_t) = ((x_3 \wedge x_4)) \wedge \neg\ ((x_5) \vee (x_2 \wedge x_9))$$

A few constraints are imposed on the update functions during model learning in BTR. Firstly, the update function allows a conjunction of up to two input genes in each slot u. Secondly, each input gene x_i can only be present in a single update function once, but the same input gene x_i can be present in multiple update functions. Thirdly, a user is able to specify a soft limit on the number of input genes (i.e. in-degree) allowed per update function, where the default in BTR is 6 in-degree per gene. Lastly, by default no self-loop is allowed in BTR.

A model state given by a Boolean model B is represented by a Boolean vector $s_t = \{x_{1t}, \ldots, x_{nt}\}$ at simulation step t. A model state space S represents the set of all model states s_t reachable from an initial model state s_1, i.e. $S = \{s_1, \ldots, s_t\}$. S can be obtained by simulating the model B starting from an initial model state s_1 using the asynchronous update scheme. The asynchronous update scheme specifies that at most one gene is updated between two consecutive states (Fig. 1b). Assuming we have a model state s_t which is not a steady state, there will be i ($i \geq 1$) genes in s_t such that $x_{it} \neq f_i(s_t)$. Therefore at simulation step $t+1$, s_{t+1} would have i possible configurations s^i_{t+1}, where $s^i_{t+1} = \{x_{1t}, \ldots, f_i(s_t), \ldots, x_{nt}\}$. This simulation is repeated until it reaches a steady state. By definition, steady states are a set of states whose destination states also belong to the same set. That is, a steady state may be a single model state s_t, or it may consist of a cyclic sequence of model states $s_t, \ldots, s_{t+j}$.

The single-cell expression data used in this study are each a matrix consisting of n individual genes in the columns and k individual cells in the rows. The expression data are normalised and standardised to give $y_{kn} \in [0, 1]$. A data state $v_k = \{y_1, \ldots, y_n\}$ represents the expression state of cell k for n genes that are observed in the cell. A data state space $V = \{v_1, \ldots, v_k\}$ represents the set of all data states that are observed in an experiment.

BTR model learning

The aim of BTR is to identify a Boolean model B with x_n genes and f_n update functions, that can produce a model state space which closely resembles an independent single-cell expression data (i.e. data state space). Note that model state space and data state space are defined in a similar way, the only difference being that the n genes take continuous values in $[0, 1]$ within a data state, while the n genes take binary values 0 and 1 in a model state. The distance between model and data state spaces is measured by the pairwise distance between pairs of model and data states, as stated in the scoring function (See below). By iteratively modifying an initial Boolean model B_1, the distance between the model and data state spaces can be minimised until a resulting final Boolean model B_f with less distance is obtained.

BTR performs model learning by utilising techniques in discrete optimisation framework. In any optimisation problem, there are two important components, namely a scoring function and a search strategy.

BSS Scoring function in BTR

The scoring function used in BTR is a novel scoring function we developed, termed as Boolean state space (BSS) scoring function. BSS scoring function $g(S, V)$ is a distance function, which consists of a base distance variable and two penalty variables. $g(S, V)$ is given by:

$$g(S, V) = h(S, V) + \lambda_1\varepsilon_1 + \lambda_2\varepsilon_2$$

Where $h(S, V)$ = base distance, ε = penalty variable, λ = constant for penalty variable.

The base distance $h(S, V)$ is given by the following equation. To prevent multiple model states from matching to a single data state, one-to-one matching between model and data states is enforced if the number of data states, N_v, are more than or equal to the number of model states, N_s, i.e. $N_v \geq N_s$. For cases where $N_v < N_s$, one-to-one matching between model and data states is enforced greedily up until the point where every data states have been assigned a matching model state, then non-unique matching will occur for the remaining model states with respect to each corresponding data state with the minimum distance.

$$h(S,\ V) = \frac{\sum_{t=1}^{N_s} min_{k=1}^{Nv}(d(s_t,\ v_k))}{N_s\, n}$$

Where $d(s_t,\ v_k)$ = pairwise distance between each model state s_t and data state v_k $(0 \le d(s_t, v_k) \le 1)$, N_s = number of model states, N_v = number of data states, n = number of genes.

The distance between model state s_t and data state v_k, $d(s_t, v_k)$, is defined as the sum of the absolute differences between values of each gene i in model state s_t and data state v_k.

$$d(s_t,\ v_k) = \sum_{i=1}^{n} |\ x_{ti} - y_{ki}|$$

Where $x_{ti} \in \{0, 1\}$ is the value of gene i in model state s_t and $y_{ki} \in [0, 1]$ is the value of gene i in data state v_k.

The two penalty variables, ε_1 and ε_2, in $g(S, V)$ are used to prevent underfitting and overfitting. ε_1 penalises depending on the proportions of 0 s, p_0, and 1 s, p_1, across all genes and all states in a model state space. The concept of ε_1 is that it penalises complexity in Boolean models by their simulated model state spaces. We have shown that as a Boolean model becomes more complex (i.e. increase in the number of edges), both p_0 and p_1 of its model state space will become closer to 0.5 (See Additional file 5: Figure S4), therefore making ε_1 a good penalty for model complexity.

$$\varepsilon_1 = e^{-a},\ where\ a = \sum_{i \in \{0,\ 1\}} \frac{(p_i - 0.5)^2}{0.5}$$

ε_2 penalises based on the number of input genes present in each of the update function f_i in a Boolean model B, given a specified threshold z_{max}.

$$\varepsilon_2 = \sum_{i=1}^{n} w_i$$

Where w_i the penalty for each update function f_i is given by:

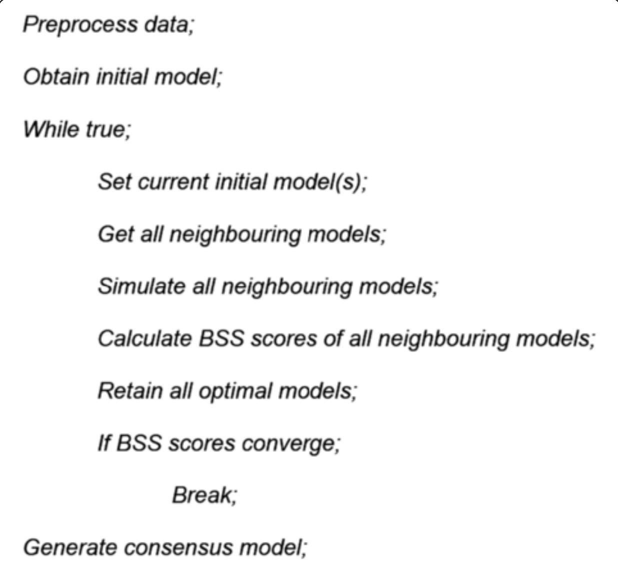
$$w_i = \begin{cases} \frac{z_i - z_{max}}{n}, & if\ z_i > z_{max} \\ 0, & if\ z_i \le z_{max} \end{cases}$$

Where z_i = the number of input genes in update function f_i, z_{max} = the maximum number of input genes allowed per update function. The default z_{max} in BTR is 6, which means that each target gene is encouraged to have not more than 6 input genes.

Search strategy in BTR

A good search strategy is required in optimisation to locate the optimal solutions within a high dimensional and complex solution space. The search strategy in BTR is a form of swarming hill climbing strategy, in which multiple optimal solutions are kept at each search step and the search only ends when the score converges for all of the optimal solutions (Fig. 9). In BTR search algorithm, the search starts from an initial Boolean model, and iteratively explores the neighbourhood of the current Boolean model in the solution space by minimal modification. When no initial model is given to BTR, it will generate a random initial model whose degree distribution satisfies a power-law distribution with a degree exponent $\gamma = 3$.

The minimal modification of a Boolean model is performed by adding or removing a gene from a single update function in the Boolean model. The resulting modified model is then evaluated by the BSS scoring function. By repeating this procedure, BTR is able to explore the solution space and eventually arrives at a more optimal Boolean model. Due to the nature of Boolean models that multiple possible Boolean models can give rise to the exact same simulated state space, BTR usually retains a list of equally optimal Boolean models at the end of the search process. In such cases, a consensus model, whose edges are weighted according to the frequencies of their presence in the list of optimal Boolean models, will be generated. Due to the design of the search strategy, it is more geared towards a local search



```
Preprocess data;
Obtain initial model;
While true;
        Set current initial model(s);
        Get all neighbouring models;
        Simulate all neighbouring models;
        Calculate BSS scores of all neighbouring models;
        Retain all optimal models;
        If BSS scores converge;
                Break;
Generate consensus model;
```

Fig. 9 Pseudocode of the search algorithm in BTR

rather than a global search. Therefore in line with the results shown in Fig. 5, BTR is best used for iteratively improving a gene network with known biological knowledge using an independent set of single-cell expression data.

BTR data processing

BTR is capable of handling all types of expression data, including qPCR and RNA-Seq. Expression data should be processed and normalised before being used in BTR. In BTR, the expression data is further processed in order to facilitate score calculation by the BSS scoring function. Firstly, if the input data is qPCR expression data, it should be inversed such that the gene with a low expression level should have a low value and vice versa. Finally, the expression values for each gene in the data are scaled to continuous values with a range of $0 \leq x \leq 1$.

Calculation of F-score

F-score, which is the harmonic average of precision and recall, represents precision and recall concisely [35], is often used to assess the performance of network inference algorithms. Precision denotes the proportion of edges that are truly present among all edges classified as present, while recall denotes the proportion of edges that are truly present among all correctly classified edges (including both edges that are present and absent) [42]. The calculations were performed on directed adjacency matrix.

Precision is defined as:

$$p = \frac{TP}{TP + FP}$$

Where TP = true positive and FP = false positive.

Recall is defined as:

$$r = \frac{TP}{TP + FN}$$

Where TP = true positive and FN = false negative.

F-score is defined as:

$$F = \frac{2pr}{r + p}$$

Synthetic data

The synthetic data used for comparing scoring functions and network inference algorithms consist of true networks, expression data and lists of modified networks. The true networks and expression data were generated using GeneNetWeaver version 3.13 [28]. The true networks contain 10 genes each and were extracted from the gene network of yeast. Each true network generated by GeneNetWeaver was then categorised into acyclic and cyclic networks. A total of 5 acyclic and 5 cyclic true networks were used in this study. The expression data were generated using ordinary and stochastic differential equations based on the true networks. A single time series expression data with 1000 observations were generated per true network, and the expression data were simulated under steady state wild type condition. A coefficient of 0.05 was used for noise term in the stochastic differential equations. The synthetic expression data as generated by GeneNetWeaver is used as non zero-inflated data. In addition, the synthetic expression data is converted into a zero-inflated data to simulate drop-outs in single-cell expression data by calculating the probability of a reading being a drop-out (i.e. zero value) based on its expression level. The probability of a reading being a drop-out, p_d, is modelled using the following equation:

$$p_d = 2^{-cy}$$

Where c = a constant (in this study, $c = 6$), and y = a reading of the expression level of a particular gene,

The lists of modified networks were generated in R using the bnlearn package [43] for Bayesian networks and the BTR package for Boolean models. The modified networks were generated by modifying the number of edges that differ from the true network, ranging from 2 edges up to 40 differing edges. The modified Bayesian networks and the modified Boolean models were generated separately due to different underlying structural constraints imposed by each framework. In Bayesian framework all networks must be directed acyclic graphs, while Boolean models do not have such restrictions. In contrast, Boolean models require explicit specification of activation and inhibition edges, while Bayesian networks handle activation and inhibition implicitly without modifying the edges. Although the generation of modified Bayesian networks and Boolean models were done separately and therefore they are not identical, all modified networks contain the same number of differing edges (2 to 40 edges) with respect to the true network. Note that the differences in edges for acyclic modified networks are not cumulative, due to difficulties in generating a directed acyclic graph with cumulative edge differences. The differences in edges for cyclic modified networks are also not cumulative to maintain consistency with the acyclic modified networks.

For synthetic data, the initial state used for the simulation of Boolean models is the expression values at time $t = 0$.

Haematopoietic data

Two Boolean models of haematopoiesis were used as initial models for model learning in this study, namely

Krumsiek [39] and Bonzanni models [38]. The update functions of both models were converted into functions with an activation clause and an inhibition clause, in which each of the clauses are individually expressed in disjunctive normal form. Note that one of the nodes (EgrNab) in the Krumsiek model comprises of 3 different genes, Egr-1, Egr-2 and Nab-2. The initial states used in the simulation were obtained from both papers respectively.

A single-cell qPCR data and a single-cell RNA-Seq data, both obtained from Wilson et al. [10], were used for model learning. The single-cell qPCR data contain 44 genes from 1626 cells (992 HSCs, 178 LMPPs, 147 CMPs, 185 GMPs and 124 MEPs), while the single-cell RNA-Seq data are collected from 96 HSCs. The expression data are processed and normalised as described in the original paper.

For Bonzanni and Krumsiek models, the initial states used for the simulation Boolean models are obtained from each paper respectively.

Network inference algorithms and analyses software used

BIC and its associated hill-climbing algorithm are implemented in bnlearn [43]. BestFit [29] is an algorithm for inferring Boolean models under synchronous framework implemented in BoolNet [44]. ARACNE [30] and CLR [31] are inference algorithms for inferring relevance networks based on mutual information. bc3net [32] and GeneNet [33] are inference algorithms based on Bayesian networks, while GENIE3 is a type of tree-based methods [34].

Plots in this study were generated using ggplot2 [45], except network plots that were generated using Cytoscape [46] and heat maps that were generated using gplots [47]. Steady state analysis was performed using genYsis [48], which search for steady states reachable from all possible initial states.

Additional files

Additional file 1: Figure S1. Is a PowerPoint file containing the results of comparing BIC and BSS scoring functions with acyclic networks using non zero-inflated synthetic expression data. (PPTX 491 kb)

Additional file 2: Figure S2. Is a PowerPoint file containing the results of comparing BIC and BSS scoring functions with cyclic networks using non zero-inflated synthetic expression data. (PPTX 366 kb)

Additional file 3: Figure S3. Is a PowerPoint file containing the summary results for both BIC and BSS scoring functions across all networks using non zero-inflated synthetic expression data. (PPTX 213 kb)

Additional file 4: Figure S5. Is a PowerPoint file containing detailed update functions of Boolean models discussed in this study. (PPTX 587 kb)

Additional file 5: Figure S4. Is a PowerPoint file containing a plot that explains and justifies the use of ε_1 as a penalty variable in BSS scoring function. (PPTX 124 kb)

Acknowledgements
Research in the authors' laboratory is supported by Bloodwise, Cancer Research UK, the Biotechnology and Biological Sciences Research Council, NIHR Cambridge Biomedical Research Centre Award, Microsoft Research and core support grants by the Wellcome Trust to the Cambridge Institute for Medical Research and Wellcome Trust - MRC Cambridge Stem Cell Institute. The funding bodies had no role in the design, collection, analysis, and interpretation of data; in the writing of the manuscript; and in the decision to submit the manuscript for publication.

Funding
Research in the authors' laboratory is supported by Bloodwise, Cancer Research UK, the Biotechnology and Biological Sciences Research Council, NIHR Cambridge Biomedical Research Centre Award, Microsoft Research and core support grants by the Wellcome Trust to the Cambridge Institute for Medical Research and Wellcome Trust - MRC Cambridge Stem Cell Institute. The funding bodies had no role in the design, collection, analysis, and interpretation of data; in the writing of the manuscript; and in the decision to submit the manuscript for publication.

Availability of data and materials
The haematopoietic data, which include two Boolean models [38, 39] and the two datasets [10] are included in the BTR package, and are also available in their respective publications. BTR is available as an R package on CRAN and also on Github [https://github.com/cheeyeelim/btr] [49]. All data and scripts that are used to generate results in this paper are available either as part of the BTR package or at [https://github.com/cheeyeelim/btr_resultscripts] [50].

Authors' contributions
CYL and SW conceived the study. CYL developed the algorithm and wrote the paper. LW, HW, SW, BG, NP and JF contributed ideas for the study and revised the paper. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Ethics approval and consent to participate
Not applicable.

Author details
[1]Department of Haematology, Wellcome Trust and MRC Cambridge Stem Cell Institute, Cambridge Institute for Medical Research, University of Cambridge, Hills Road, Cambridge CB2 0XY, UK. [2]Department of Computer Science, University of Leicester, Leicester, UK. [3]Biostatistics Unit, Medical Research Council, Cambridge, UK. [4]Microsoft Research Cambridge, Cambridge, UK. [5]Department of Biochemistry, University of Cambridge, Cambridge, UK.

Received: 21 April 2016 Accepted: 1 September 2016
Published online: 06 September 2016

References
1. Xu H, Schaniel C, Lemischka IR, Ma'ayan A. Toward a complete in silico, multi-layered embryonic stem cell regulatory network. Wiley Interdiscip Rev Syst Biol Med. 2010;2:708–33.
2. Moignard V, Woodhouse S, Haghverdi L, Lilly AJ, Tanaka Y, Wilkinson AC, Buettner F, Macaulay IC, Jawaid W, Diamanti E, Nishikawa S-I, Piterman N, Kouskoff V, Theis FJ, Fisher J, Göttgens B. Decoding the regulatory network of early blood development from single-cell gene expression measurements. Nat Biotechnol. 2015;33:269–76.
3. Suzuki H, Forrest ARR, van Nimwegen E, Daub CO, Balwierz PJ, Irvine KM, Lassmann T, Ravasi T, Hasegawa Y, de Hoon MJL, Katayama S, Schroder K, Carninci P, Tomaru Y, Kanamori-Katayama M, Kubosaki A, Akalin A, Ando Y, Arner E, Asada M, Asahara H, Bailey T, Bajic VB, Bauer D, Beckhouse AG, Bertin N, Björkegren J, Brombacher F, Bulger E, Chalk AM, et al. The transcriptional network that controls growth arrest and differentiation in a human myeloid leukemia cell line. Nat Genet. 2009;41:553–62.
4. Shapiro E, Biezuner T, Linnarsson S. Single-cell sequencing-based technologies will revolutionize whole-organism science. Nat Rev Genet. 2013;14:618–30.

5. Ramos CA, Bowman TA, Boles NC, Merchant AA, Zheng Y, Parra I, Fuqua SAW, Shaw CA, Goodell MA. Evidence for diversity in transcriptional profiles of single hematopoietic stem cells. PLoS Genet. 2006;2:e159.
6. Ståhlberg A, Bengtsson M. Single-cell gene expression profiling using reverse transcription quantitative real-time PCR. Methods. 2010;50:282–8.
7. Tang F, Barbacioru C, Wang Y, Nordman E, Lee C, Xu N, Wang X, Bodeau J, Tuch BB, Siddiqui A, Lao K, Surani MA. mRNA-Seq whole-transcriptome analysis of a single cell. Nat Methods. 2009;6:377–82.
8. Yan L, Yang M, Guo H, Yang L, Wu J, Li R, Liu P, Lian Y, Zheng X, Yan J, Huang J, Li M, Wu X, Wen L, Lao K, Li R, Qiao J, Tang F. Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. Nat Struct Mol Biol. 2013;20:1131–9.
9. Mahata B, Zhang X, Kolodziejczyk AA, Proserpio V, Haim-Vilmovsky L, Taylor AE, Hebenstreit D, Dingler FA, Moignard V, Göttgens B, Arlt W, McKenzie ANJ, Teichmann SA. Single-cell RNA sequencing reveals T helper cells synthesizing steroids de novo to contribute to immune homeostasis. Cell Rep. 2014;7:1130–42.
10. Wilson NK, Kent DG, Buettner F, Shehata M, Macaulay IC, Calero-Nieto FJ, Sánchez Castillo M, Oedekoven CA, Diamanti E, Schulte R, Ponting CP, Voet T, Caldas C, Stingl J, Green AR, Theis FJ, Göttgens B. Combined single-cell functional and gene expression analysis resolves heterogeneity within stem cell populations. Cell Stem Cell. 2015;16:712–24.
11. Buettner F, Natarajan KN, Casale FP, Proserpio V, Scialdone A, Theis FJ, Teichmann SA, Marioni JC, Stegle O. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. Nat Biotechnol. 2015;33:155–60.
12. Usoskin D, Furlan A, Islam S, Abdo H, Lönnerberg P, Lou D, Hjerling-Leffler J, Haeggström J, Kharchenko O, Kharchenko PV, Linnarsson S, Ernfors P. Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. Nat Neurosci. 2014;18:145–53.
13. Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, Allison KR, Kellis M, Collins JJ, Stolovitzky G. Wisdom of crowds for robust gene network inference. Nat Methods. 2012;9:796–804.
14. Davidson EH, Rast JP, Oliveri P, Ransick A, Calestani C, Yuh C-H, Minokawa T, Amore G, Hinman V, Arenas-Mena C, Otim O, Brown CT, Livi CB, Lee PY, Revilla R, Rust AG, Pan Z j, Schilstra MJ, Clarke PJC, Arnone MI, Rowen L, Cameron RA, McClay DR, Hood L, Bolouri H. A genomic regulatory network for development. Science. 2002;295:1669–78.
15. Bolouri H, Davidson EH. Transcriptional regulatory cascades in development: initial rates, not steady state, determine network kinetics. Proc Natl Acad Sci U S A. 2003;100:9371–6.
16. Andrecut M, Halley JD, Winkler DA, Huang S. A general model for binary cell fate decision gene circuits with degeneracy: indeterminacy and switch behavior in the absence of cooperativity. PLoS One. 2011;6:e19358.
17. Li C, Wang J. Quantifying cell fate decisions for differentiation and reprogramming of a human stem cell network: landscape and biological paths. PLoS Comput Biol. 2013;9:e1003165.
18. de Jong H. Modeling and simulation of genetic regulatory systems: a literature review. J Comput Biol. 2002;9:67–103.
19. Fisher J, Henzinger TA. Executable cell biology. Nat Biotech. 2007;25:1239–49.
20. Li F, Long T, Lu Y, Ouyang Q, Tang C. The yeast cell-cycle network is robustly designed. Proc Natl Acad Sci U S A. 2004;101:4781–6.
21. Fauré A, Naldi A, Chaouiya C, Thieffry D. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. Bioinformatics. 2006;22:e124–31.
22. Giacomantonio CE, Goodhill GJ. A Boolean model of the gene regulatory network underlying Mammalian cortical area development. PLoS Comput Biol. 2010;6. doi:10.1371/journal.pcbi.1000936.
23. Dunn S-J, Martello G, Yordanov B, Emmott S, Smith AG. Defining an essential transcription factor program for naïve pluripotency. Science. 2014; 344:1156–60.
24. Kharchenko PV, Silberstein L, Scadden DT. Bayesian approach to single-cell differential expression analysis. Nat Methods. 2014;11:740–2.
25. Chen H, Guo J, Mishra SK, Robson P, Niranjan M, Zheng J. Single-cell transcriptional analysis to uncover regulatory circuits driving cell fate decisions in early mouse development. Bioinformatics. 2014;31:1060–6.
26. Liu Z, Malone B, Yuan C. Empirical evaluation of scoring functions for Bayesian network model selection. BMC Bioinformatics. 2012;13 Suppl 15:S14.
27. Carvalho AM. Scoring functions for learning Bayesian networks. In: INESC-ID Tec. Rep. 2009. p. 54.
28. Schaffter T, Marbach D, Floreano D. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics. 2011;27:2263–70.
29. Lähdesmäki H, Shmulevich I, Yli-Harja O. On learning gene regulatory networks under the boolean network model. Mach Learn. 2003;52:147–67.
30. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, Califano A. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics. 2006;7 Suppl 1:S7.
31. Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins JJ, Gardner TS. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biol. 2007;5:e8.
32. de Matos Simoes R, Emmert-Streib F. Bagging statistical network inference from large-scale gene expression data. PLoS One. 2012;7:e33624.
33. Opgen-Rhein R, Strimmer K. From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. BMC Syst Biol. 2007;1:37.
34. Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P: Inferring regulatory networks from expression data using tree-based methods. PLoS One. 2010;5. doi:10.1371/journal.pone.0012776.
35. Sokolova M, Japkowicz N, Szpakowicz S. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation, vol. 4304. Berlin, Heidelberg: Springer Berlin Heidelberg; 2006 [Lecture Notes in Computer Science].
36. Pina C, Fugazza C, Tipping AJ, Brown J, Soneji S, Teles J, Peterson C, Enver T. Inferring rules of lineage commitment in haematopoiesis. Nat Cell Biol. 2012;14:287–94.
37. Moignard V, Macaulay IC, Swiers G, Buettner F, Schütte J, Calero-Nieto FJ, Kinston S, Joshi A, Hannah R, Theis FJ, Jacobsen SE, de Bruijn MF, Göttgens B. Characterization of transcriptional networks in blood stem and progenitor cells using high-throughput single-cell gene expression analysis. Nat Cell Biol. 2013;15:363–72.
38. Bonzanni N, Garg A, Feenstra KA, Schütte J, Kinston S, Miranda-Saavedra D, Heringa J, Xenarios I, Göttgens B. Hard-wired heterogeneity in blood stem cells revealed using a dynamic regulatory network model. Bioinformatics. 2013;29:i80–8.
39. Krumsiek J, Marr C, Schroeder T, Theis FJ. Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. PLoS One. 2011;6:e22649.
40. Wilson NK, Foster SD, Wang X, Knezevic K, Schütte J, Kaimakis P, Chilarska PM, Kinston S, Ouwehand WH, Dzierzak E, Pimanda JE, de Bruijn MFTR, Göttgens B. Combinatorial transcriptional control in blood stem/progenitor cells: genome-wide analysis of ten major transcriptional regulators. Cell Stem Cell. 2010;7:532–44.
41. Li L, Jothi R, Cui K, Lee JY, Cohen T, Gorivodsky M, Tzchori I, Zhao Y, Hayes SM, Bresnick EH, Zhao K, Westphal H, Love PE. Nuclear adaptor Ldb1 regulates a transcriptional program essential for the maintenance of hematopoietic stem cells. Nat Immunol. 2011;12:129–36.
42. Bockhorst J, Craven M. Markov networks for detecting overlapping elements in sequence data. In Advances in Neural Information Processing Systems (NIPS). Volume 17. Cambridge: MIT Press; 2005.
43. Scutari M. Learning Bayesian networks with the bnlearn R package. J Stat Softw. 2010;35:1–22.
44. Müssel C, Hopfensitz M, Kestler HA. BoolNet–an R package for generation, reconstruction and analysis of Boolean networks. Bioinformatics. 2010;26:1378–80.
45. Wickham H. ggplot2: elegant graphics for data analysis. New York: Springer; 2009.
46. Shannon P. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 2003;13:2498–504.
47. Warnes GR, Bolker B, Bonebakker L, Gentleman R, Liaw WHA, Lumley T, Maechler M, Magnusson A, Moeller S, Schwartz M, Venables B. gplots: various R programming tools for plotting data. 2015.
48. Garg A, Di Cara A, Xenarios I, Mendoza L, De Micheli G. Synchronous versus asynchronous modeling of gene regulatory networks. Bioinformatics. 2008;24:1917–25.
49. BTR R package. [https://github.com/cheeyeelim/btr]. Accessed 4 Sept 2016.
50. BTR result scripts. [https://github.com/cheeyeelim/btr_resultscripts]. Accessed 4 Sept 2016.