BMC Bioinformatics

RESEARCH Open Access

A novel procedure for statistical inference and verification of gene regulatory subnetwork

Haijun Gong*, Jakob Klinger, Kevin Damazyn, Xiangrui Li, Shiyang Huang

From The 11th Annual Biotechnology and Bioinformatics Symposium (BIOT-2014)
Provo, UT USA. 11-12 December 2014

Abstract

Background: The reconstruction of gene regulatory network from time course microarray data can help us comprehensively understand the biological system and discover the pathogenesis of cancer and other diseases. But how to correctly and efficiently decifer the gene regulatory network from high-throughput gene expression data is a big challenge due to the relatively small amount of observations and curse of dimensionality. Computational biologists have developed many statistical inference and machine learning algorithms to analyze the microarray data. In the previous studies, the correctness of an inferred regulatory network is manually checked through comparing with public database or an existing model.

Results: In this work, we present a novel procedure to automatically infer and verify gene regulatory networks from time series expression data. The dynamic Bayesian network, a statistical inference algorithm, is at first implemented to infer an optimal network from time series microarray data of S. cerevisiae, then, a weighted symbolic model checker is applied to automatically verify or falsify the inferred network through checking some desired temporal logic formulas abstracted from experiments or public database.

Conclusions: Our studies show that the marriage of statistical inference algorithm with model checking technique provides a more efficient way to automatically infer and verify the gene regulatory network from time series expression data than previous studies.

Introduction

Advancement of DNA microarray technology and next generation sequencing technique have revolutionized the molecular biology, making it possible for biologists to measure and collect thousands of genes' expression levels simultaneously, efficiently and precisely in one experiment. Computational analysis of genome-wide transcriptomics data will help us understand the regulatory components and mechanisms underlying some diseases. These explosively growing amount of highdimensional gene expression data can be divided into two types: static and time series. The static expression data are assumed to be independently and identically distributed (IID), and many statistical inference algorithms [1-8] have been developed to identify key genetic signatures and signaling pathways that are frequently altered in some diseases. Gene regulatory network plays a critical role in the cell's proliferation and differentiation, so, a comprehensive understanding of gene regulatory network (GRN) and regulatory components will help discover some drug targeted genes in cancer and other diseases. Computational biologists have proposed a variety of methods, for example, the Boolean networks [9] and differential equations [10], to study the gene regulatory network. Friedman and other researchers [11,12] developed and applied discrete and continuous Bayesian networks (BN) with linear regression and non-parametric regression to infer gene regulatory networks. The BN approach could identify the causal relationships between different genes to some degree. However, it cannot construct cyclic networks and this method is unable to handle the temporal aspect of time-series data. But the feedback loops (cyclic pathways)

* Correspondence: hgong2@slu.edu
Department of Mathematics and Computer Science, Saint Louis University, 220 N Grand Blvd, 63103 St. Louis, MO, USA

are prevalent in the gene regulatory networks and signaling pathways.

The time series gene expression data can provide abundant information regarding the dynamic and temporal behaviors of biological system, which can not be handled by Bayesian network method. Dynamic Bayesian network (DBN) [13-16] is a promising alternative which has been proposed to construct GRN with feedback loops from time-series expression data. DBN has attracted a lot of attention from numerous bioinformatics researchers, and different DBN based approaches and tools were developed to increase accuracy and reduce computational time. An extended expectation-maximisation (EM) algorithm [17] was proposed to estimate the parameters in the DBN model. However, the DBN method has some limitations, for example, it is very sensitive to the choice of data discretization. Moreover, the deduction of the "activation" or "inhibition" relationship between different genes is not easy and accurate, so, the inferred optimal network might not be a correct one. Recently, Liang *et al.*'s work [18] proposed a network and community identification (NCI) method to infer multiple signed subnetworks from gene expression data by incorporating community structure information.

Without verification or validation, the inferred regulatory networks can not help us correctly understand the mechanism in the cell cycle. Another limitation in the previous studies is, the correctness or verification of the inferred networks is manually checked by comparing with public database (KEGG, GO, GenMAPP, etc) or existing/known models. This verification procedure is only good for small and already-known network "inference" and "verification". However, the signaling pathway or regulatory network is complex due to the excessive number of components and interactions, it is not realistic and efficient to use traditional methods to manually verify or analyze large networks. An intelligent verification technique called Model Checking [19] has been successfully applied for the verification of complex systems, including the hardware (e.g., CPU) and software (aerospace control software) systems. Recently, we applied this technique to study some complex biological networks [20-25]. Model Checking is the process of determining whether or not a given system M satisfies a desired temporal logic formula ψ, denoted by $M \models \psi$. Our previous work proposed and applied different model checking techniques, including statistical model checking [21,22], synchronous symbolic model checking [23-25], asynchronous model checking technique [26] and probabilistic model checking [27], to formally verify some given stochastic, boolean, and discrete-value models of signaling pathways in the cancer cells. The model checkers automatically and exhaustively search the state space to verify some desired temporal logic formula, and it can check up to 10^{100} possible states.

In this work, we proposed a novel inference and verification procedure, which marries the dynamic Bayesian network inference algorithm with a powerful model checking technique, to analyze time course microarray data. We will first briefly introduce the dynamic Bayesian network inference with Java objects (Banjo) [28] method developed in Hartemink's group [29] and apply it to infer optimal gene regulatory networks from time-series expression data. Then, we proposed a novel weighted symbolic model checking technique (weighted SMV) to automatically verify or falsify the inferred weighted networks or models through checking some temporal logic formulas abstracted from experiments. Finally, we apply Banjo and weighted SMV to analyze time-series microarray data and reconstruct gene regulatory subnetwork of yeast.

Methods

Dynamic Bayesian network inference

Probabilistic graphical model describes each node in the network by a random variable, and the directed edge represents a conditional dependence between two variables. Therefore, gene regulatory network can be graphically represented by a joint distribution of all random variables over time. The time series gene expression (microarray) data, which consists of p genes measured at n different time points, can be described by an $n \times p$ matrix $\mathbf{X}$. If $\mathbf{X_i} = (X_{i1},..., X_{ip})^T$ is defined as a random variable vector (at time i), then, $\mathbf{x_i} = (x_{i1},...,x_{ip})^T$ corresponds to the values of a vector of p genes' expression measured at time i = 1, 2,..., n; that is, x_{ij} represents an observation value of the random variable X_{ij} (the jth gene's expression measured at time i). We adopt some conventions used in Kim *et al.*'s work [13].

Since the random variable vector $\mathbf{X_i}$ is time dependent, the dynamic Bayesian network [13-15] assumes the genes' expression levels measured at time i are dependent on those at time i − 1 only which is illustrated in Figure 1. This assumption is also called first-order Markov chain. The joint probability distribution for the $n \times p$ random variables (or n vectors of random variables) can be written as

$$P(\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_n}) = P(\mathbf{X_1})\, P(\mathbf{X_2}|\mathbf{X_1}) ... P(\mathbf{X_n}|\mathbf{X_{n-1}}).$$

We use $Par(X_{ij})$ to denote the gene j (at time i)'s parents (at time i − 1, an immediate previous time point), and also assume each gene (node) at time I is influenced by itself and its parent genes (nodes) at time i − 1 only. Therefore, the conditional probability distribution can be expressed as

$$P(\mathbf{X_i}|\mathbf{X_{i-1}}) = P(X_{i1}|Par(X_{i1}))...P(X_{ip}|Par(X_{ip})).$$

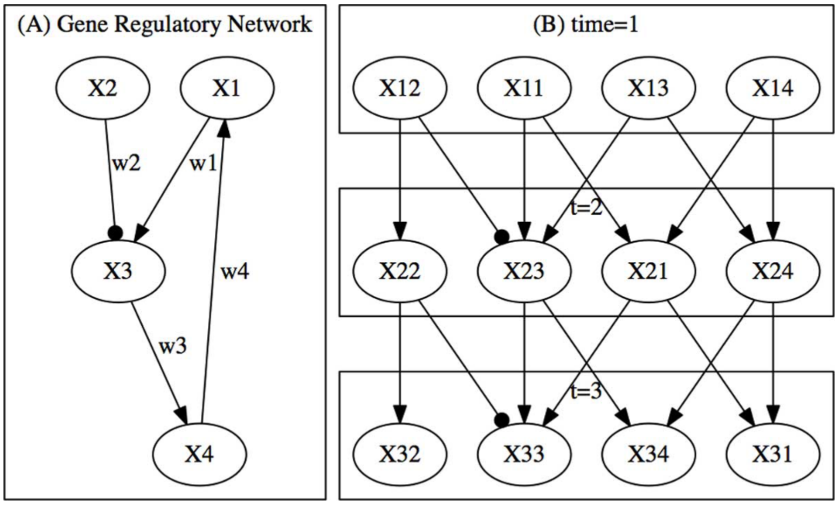


Figure 1 Illustration of gene regulatory network (A) and dynamic Bayesian network (B). The gene regulatory network is composed of a feedback loop. Arrows represent activation, and circlehead arrows denote inhibition. The random variable X_{ij} represents a gene *j* measured at time *i*.

Figure 2 and Figure 3 show the pseudocode and flowchart of the GRN inference and verification with dynamic Bayesian network learning method (implemented by Banjo) and weighted model checking technique (implemented by SMV model checker). First, the time series microarray data **D** are discretized into *k* levels $\{l_1,..., l_k\}(k = 2, 3,...)$ using either quantile (qk) or interval discretization (in) methods [28]. Second, apply a Bayesian Dirichlet equivalence (BDe) scoring metric [30] to evaluate the goodness of each possible network. BDe scoring metric has been widely used as a criterion or score function in the regulatory network learning [13,15]. Then the idea is to find the posterior probability distribution of the possible networks **G**:

$$P(G|\mathbf{D}) = \frac{P(G, \mathbf{D})}{P(\mathbf{D})},$$
$$P(G, \mathbf{D}) = \int P(G, \mathbf{D}, \Theta)d\Theta$$
$$= \pi(G)\int f(\mathbf{D}|G, \Theta)\pi(\Theta|G, \Lambda)d\Theta$$

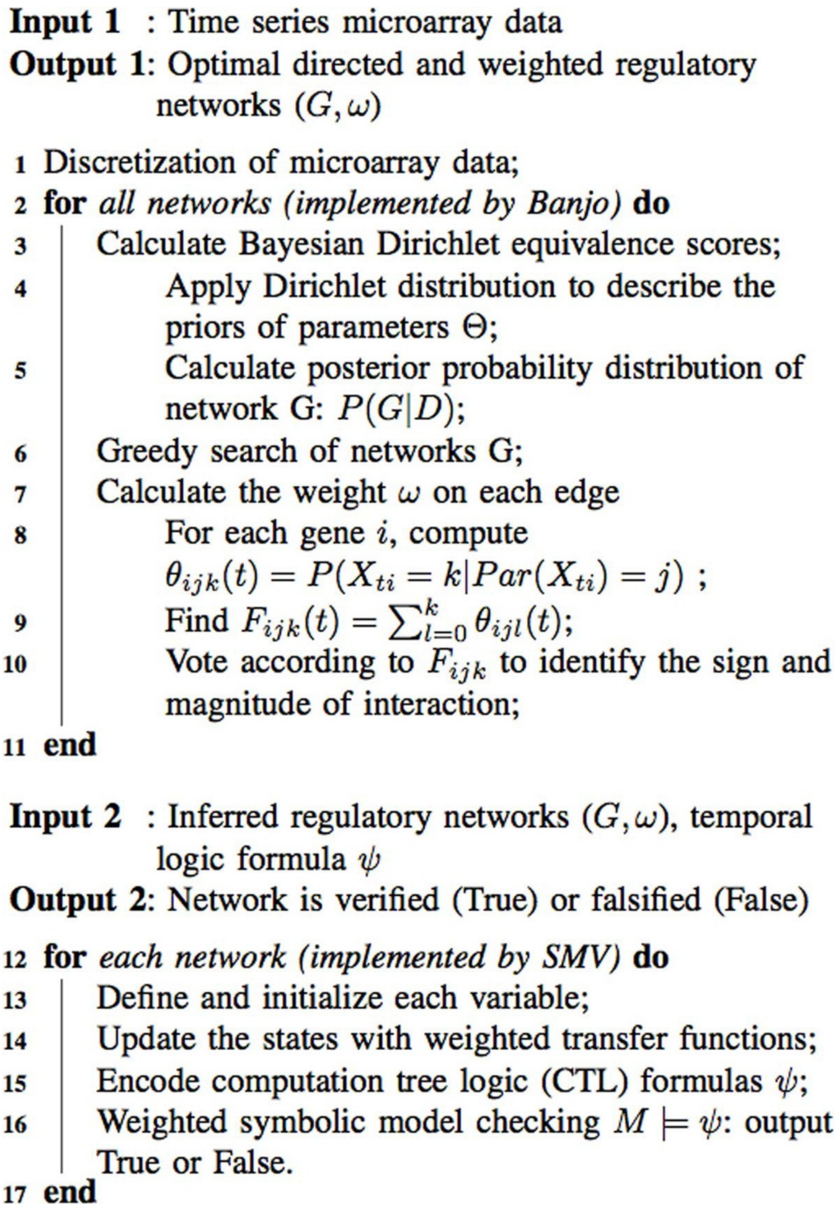
The BDe score function is based on the assumption that the microarray data **D** is a multinomial sample, that is, $\mathbf{D}|\boldsymbol{\Theta} \sim Multinomial(\boldsymbol{\Theta})$. BDe also assumes the parameters Θ are globally and locally independent and the priors of Θ, denoted by $\pi(\Theta|G, \boldsymbol{\Lambda})$, follow Dirichlet distribution with a hyperparameter vector $\boldsymbol{\Lambda}$, that is, $\Theta|G \sim Dirichlet(\boldsymbol{\Lambda})$, which is a conjugate prior of multinomial distribution. The optimal network is selected according to the BDe scores which are dependent on P (G, **D**). Next, search all the possible optimal networks. Banjo allows two different search strategies, including the greedy search and simulated annealing algorithm proposed by Heckerman [31], which can output top *n* directed networks with highest scores, and it can also retain and average some highest scoring networks to produce a weighted consensus network [28].

Bayesian network inference with Java objects (Banjo) [28] can also compute the influence score (weight) [32] on each edge of the inferred optimal network. The value of influence score describes the relative magnitude of interactions, and its sign identifies the activation (a positive value) or inhibition (a negative value) relationship between two nodes (genes). The estimation of influence score [32] is dependent on the values of the conditional probability

$$\theta_{ijk}(\mathrm{t}) = P(X_{ti} = k|Par(X_{ti}) = j),$$

which is the probability that gene X_{ti} takes a value of *k* given its parent gene $Par(X_{ti})$ takes a value of *j*; the cumulative distribution function $F_{ijk}(t) = \sum_{l=0}^{k} \theta_{ijl}(t)$, which describes the probability that gene X_{ti} takes a value less than or equal to *k* given its parent gene takes a value of *j*; and a predefined voting system. If there is a high probability for the gene X_{ti} to take a larger value given its parent's value increases, then, the voting system [32] in the Banjo will increase the positive vote by one; else, the negative vote will increase by one. If the influence score is close to 0, the sign of the edge can not be identified. Banjo can automatically implement the dynamic Bayesian network inference algorithm to search for high-scoring probabilistic graphical models, output the optimal networks and calculate the (signed) influence scores or weights. The interested reader could refer to [28,32] for details.

The dynamic Bayesian network implemented with Banjo can infer the high-scoring gene regulatory networks based on the BDe metrics, however, this algorithm is sensitive to the data discretization methods.



```
Input 1  : Time series microarray data
Output 1: Optimal directed and weighted regulatory
          networks $(G, \omega)$

Discretization of microarray data;
for all networks (implemented by Banjo) do
    Calculate Bayesian Dirichlet equivalence scores;
        Apply Dirichlet distribution to describe the
        priors of parameters $\Theta$;
        Calculate posterior probability distribution of
        network G: $P(G|D)$;
    Greedy search of networks G;
    Calculate the weight $\omega$ on each edge
        For each gene $i$, compute
        $\theta_{ijk}(t) = P(X_{ti} = k|Par(X_{ti}) = j)$ ;
        Find $F_{ijk}(t) = \sum_{l=0}^{k} \theta_{ijl}(t)$;
        Vote according to $F_{ijk}$ to identify the sign and
        magnitude of interaction;
end

Input 2  : Inferred regulatory networks $(G, \omega)$, temporal
          logic formula $\psi$
Output 2: Network is verified (True) or falsified (False)

for each network (implemented by SMV) do
    Define and initialize each variable;
    Update the states with weighted transfer functions;
    Encode computation tree logic (CTL) formulas $\psi$;
    Weighted symbolic model checking $M \models \psi$: output
    True or False.
end
```

Figure 2 Pseudocode of gene regulatory network inference and formal verification. Part I describes the dynamic Bayesian network inference method implemeted by Banjo; part II describes the formal verification implemented by weighted symbolic model checker.

Moreover, in many cases, the inferred optimal network might not be a correct one based on different scoring functions. Which model is closest to the truth in the biological system? Previous studies validate the inferred network through manually comparing with the public database or known models. The manual verification method is not realistic for the large or unknown network verification. The most innovative aspect of the proposed procedure in Figure 2 is the marriage of dynamic Bayesian network inference algorithm with formal verification technique, called weighted symbolic model checking (Part 2 of pseudocode in Figure 2), which can automatically verify the network through checking some temporal logic formulas abstracted from the experiments or public database. Next, we will introduce a powerful model checking technique and apply it to formally verify the inferred regulatory networks.

Weighted symbolic model checking

A network or model can be described as a Kripke structure [19,20] $M = (S, s_0, R, L)$, representing a finite-state concurrent system with the initial state $s_0 \in S$, states transition relation R , and a labeling function L. Given a model or concurrent system, we expect it to satisfy

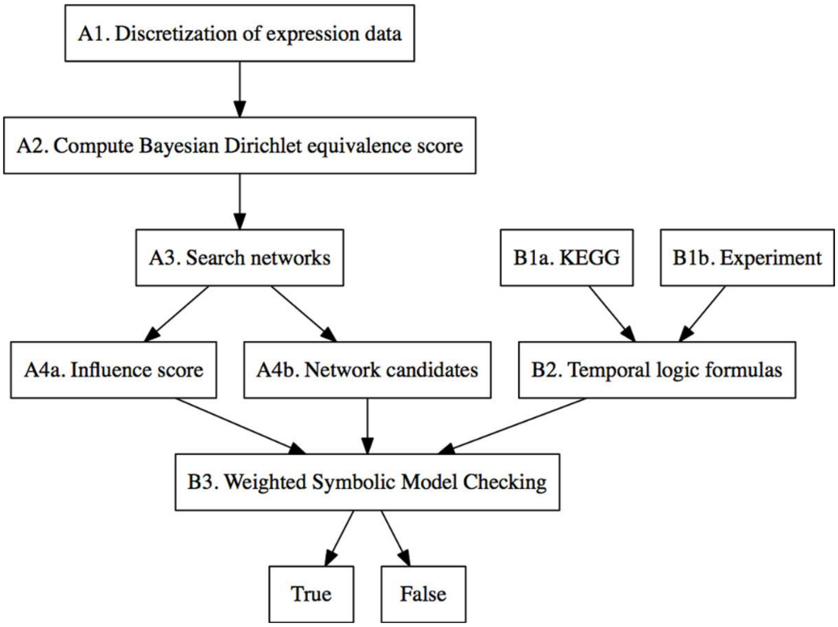

A1. Discretization of expression data
A2. Compute Bayesian Dirichlet equivalence score
A3. Search networks
B1a. KEGG
B1b. Experiment
A4a. Influence score
A4b. Network candidates
B2. Temporal logic formulas
B3. Weighted Symbolic Model Checking
True
False

Figure 3 Flowchart of gene regulatory network inference from time series microarray data and formal verification. The dynamic Bayesian network inference (A1-A4) is implemented by Banjo, and the inferred network's verification (B1-B3) is implemented by the weighted symbolic model verifier (SMV).

some desired property. So, model checking, a formal verification technique [19], is the process of determining whether or not a given model M satisfies the desired property, which is expressed as a temporal logic formula ψ, denoted by $M \models \psi$. During formal verification, model checkers can search the state space of concurrent system exhaustively to find all states that satisfy the formula ψ. If the property is satisfied, model checker will output "True"; else, it will output "False" with a counter-example sequence that falsifies ψ. Model checking of hardware and software systems has been very successful in the past three decades. Recently, we proposed different (probabilistic, statistical, symbolic, synchronous and asynchronous) model checkers to formally investigate the complex signal transduction networks in the cancer cells [20-24,26].

The desired properties describing some existing wet lab experimental results or phenomena are expressed in a high-level, expressive language - Computation Tree Logic (CTL) formula ψ. On the computation tree, the root represents an initial state, the branches and leaves represent all possible sequences of state transitions (paths) from the root [19]. CTL formula ψ is composed of path quantifiers which describes the branching structure in the computation tree: **A** (for all paths), **E** (there exists some path); temporal operators describing properties on a path through the tree: **X** (next time), **F** (in the future), **G** (globally), **U** (until), **R** (release); and Boolean logic connectives (| (or), & (and), → (implies)). In the CTL formula, the temporal operator must be immediately preceded by a path quantifier [19]. Similar to our previous work [19,20,26], we will use (**AX, EX, AG, EG, AF, EF**) to construct CTL formulas for the verification of gene regulatory network. For example, **AG**ϕ means ϕ is globally true on all paths; **EF**ϕ means ϕ holds at some state in the future on some path. More interesting CTL operators and formulas have been discussed in Clarke *et al.*'s book [19].

Given a Kripke structure M , the state formula and path formula are represented by ψ and ϕ respectively in CTL syntax, and a path π is defined as an infinite sequence of states, $\pi = s_0, s_1,...,$ where s_0 is an initial state. We use π_i to denote the suffix of π starting at state s_i, and $M, \pi \models \phi$ denotes the path π satisfies the path formula ϕ. The semantics of CTL have been defined in [19], below (Table 1) we list some semantics that are used in this work:

The interested readers could refer to the book [19] and our recent work [20] for details regarding the syntax and semantics of CTL logic.

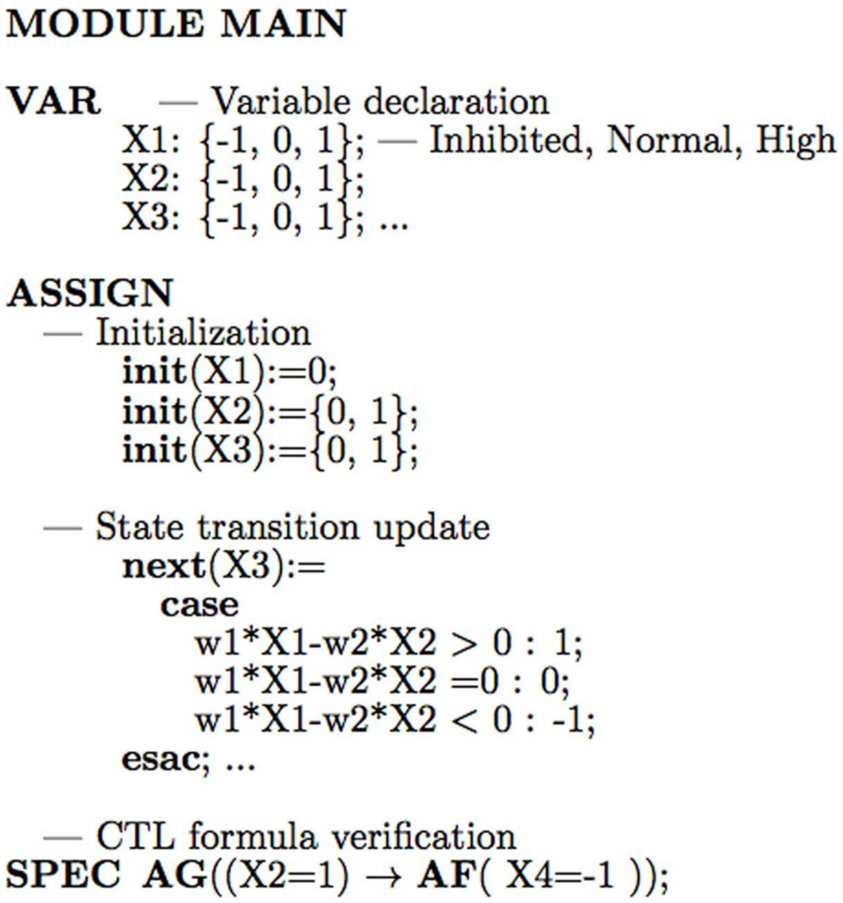
Symbolic Model Verifier (SMV) [33] is a popular formal verification tool encoded by ordered binary decision diagram [34], and the state transition relation is implicitly represented by a Boolean function. SMV can verify (output "True") or falsify (output "False" with a counter-example) a desired CTL formula ψ through automatically and exhaustively searching the state transition system M. Our recent studies [20,23-25,27] proposed

Table 1

$M, s \models ! \psi$	iff $M, s \models \psi$ does not hold;
$M, s \models \psi_1 \& \psi_2$	iff $M, s \models \psi_1$ and $M, s \models \psi_2$;
$M, s \models \psi_1 \mid \psi_2$	iff $M, s \models \psi_1$ or $M, s \models \psi_2$;
$M, \pi \models \mathbf{X}\psi$	iff $M, \pi^1 \models \psi$;
$M, \pi \models \mathbf{F}\psi$	iff there exists a $k \geq 0$, such that $M, \pi^k \models \psi$;
$M, \pi \models \mathbf{G}\psi$	iff for all $k \geq 0$, $M, \pi^k \models \psi$;
$M, s \models \mathbf{A}\phi$	iff for every path π from s, $M, \pi \models \phi$;
$M, s \models \mathbf{E}\phi$	iff there exists a path π from s, such that, $M, \pi \models \phi$,

both synchronous and asynchronous symbolic model checkers to study the Boolean and discrete value models of signaling pathways. These studies are based on the *unweighted* model checking, that is, the interaction between two nodes is represented by an unweighted edge.

Next, we will propose a *weighted* symbolic model checking method (Part 2 of pseudocode in Figure 2) which is an extension of the unweighted model checker. Figure 4 illustrates some weighted SMV model checking code and CTL formulas for the verification of gene regulatory network given in Figure 1. The grammar of SMV code is similar to the unweighted SMV program [20,26], and both start with “MODULE MAIN”. All the variables are declared with the keyword “VAR”, and initialized with “init” under the keyword “ASSIGN”. However, in the weighted SMV code, the state transition update of each variable (e.g., X3) is not only dependent on its parents’ states (e.g., X1, X2), but also influenced by the strength of interactions, that is, the influence



```
MODULE MAIN

VAR    — Variable declaration
       X1: {-1, 0, 1}; — Inhibited, Normal, High
       X2: {-1, 0, 1};
       X3: {-1, 0, 1}; ...

ASSIGN
   — Initialization
       init(X1):=0;
       init(X2):={0, 1};
       init(X3):={0, 1};

   — State transition update
       next(X3):=
          case
             w1*X1-w2*X2 > 0 : 1;
             w1*X1-w2*X2 =0 : 0;
             w1*X1-w2*X2 < 0 : -1;
       esac; ...

   — CTL formula verification
SPEC AG((X2=1) → AF( X4=-1 ));
```

Figure 4 Illustration of weighted symbolic model checking of the regulatory network in Fig. 1. The state transition update is dependent on the modified influence score (weight w_i) calculated by Banjo.

score or weight (e.g., w_1, w_2). The value of influence score calculated by Banjo [32] ranges from −1 to 1, which describes the sign and magnitude of interaction between two genes. Since the weighted SMV model checker does not allow floating point numbers, all the influence scores will be converted to integers (modified weights) first before formal verification. The CTL formula which abstracts the experimental phenomenon or public database is encoded with the keyword "SPEC". For example, the statement "SPEC AG($X2$ = 1 → AF ($X4$ = −1))" means, overexpressed $X2$ will eventually inhibit $X4$'s expression on all paths. The weighted SMV model checker will automatically verify all the CTL formulas (encoded by SPEC), and find the best model which satisfies all or most of the properties based on existing experimental evidence.

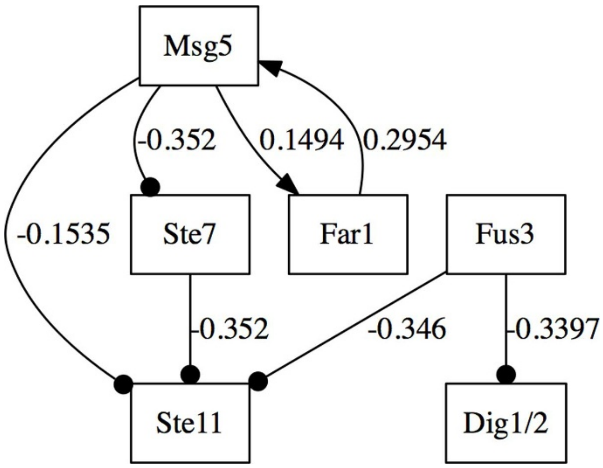


Figure 5 An optimal subnetwork of MAPK pathway inferred by Banjo. The optimal network is inferred based on *i*2 interval discretization method. The directed and circlehead arrows represent activation and inhibition respectively, the value on each edge is influence score or weight describing the interaction between two nodes.

Results and discussion

In this section, we will apply the dynamic Bayesian network inference and weighted symbolic model checking methods proposed in Figure 2 to infer and verify gene regulatory subnetworks from time series microarray data of yeast.

The time series microarray data of *Saccharomyces cerevisiae* collected by Spellman *et al.* [35] has been studied by many researchers using different inference algorithms. The data were measured and collected from the yeast cultures synchronized by three independent methods: alpha factor arrest, elutriation, and arrest of a cdc15 temperature-sensitive mutant, which contain 16, 25 and 14 time points. A full description and complete data sets are available at [36]. The Banjo setting code, microarray data and weighted SMV code developed for this work are available at [37].

We will first infer and verify a small network of MAPK signaling pathway which plays an important role in the cell cycle. We focused on the subnetwok around Fus3 which contains 8 genes (Ste20, Ste11, Ste7, Fus3, Dig1/2, Ste12, Far1, Msg5), while, Dig1/2 denotes the mean value of Dig1 and Dig2 in our analysis. Figure 5 shows an inferred optimal network which is composed of 6 genes based on *i*2 interval discretization (two discrete states) method in Banjo. The weighted symbolic model checker will be applied to formally verify or falsify this optimal network.

In Table 2 we summarize four CTL formulas abstracted from experiments and KEGG that MAPK pathway should satisfy. All the genes can take three possible states: inhibited (−1), normal (0), or activated (1), and they are initially set to be either 0 or −1 with a probability. Formula P1 is checking, if Fus3 is activated, Dig1/2 will be inhibited immediately in the next step (AX) on all paths. Our studies infer and verify that Fus3 is a direct inhibitor of Dig1/2. P2 means, it is globally true (AG) that Ste11 (MAPKKK)'s activation will immediately activate its downstream gene Ste7 (MAPKK) on all paths. P3 and P4 are checking whether or not Msg5 or Ste7's activation will finally inhibit or promote the transcription of Fus3 or Far1, cell cycle regulatory genes, respectively. The weighted SMV verified the formulas P1 and P3, but falsified P2 and P4. That is, the inferred network does not satisfy all the desired properties. So, this optimal network candidate inferred by Banjo is falsified by the weighted SMV model checker, which is also confirmed by the KEGG database. If some property is falsified, SMV model checker will also output a counterexample to demonstrate why this network is wrong, and help us refine the inferred network.

Next, we will apply our methods to infer and verify a cell cycle subnetwork (including the genes: ste20, ste11, ste7, msg5, ste12, dig12, fus3, far1, cdc6, cdc7, cdc20, cdc28, cdc45, cdc46, cdc54, cln1/2, cln3, clb5/6, mcm2/3/6, swi4/6). Partial pathway has been registered in KEGG. Similar to MAPK pathway inference, we will use the mean values for some genes from the same family (e.g., mcm2, mcm3, mcm6) in the data analysis. Figure 6

Table 2. List of CTL formulas related to MAPK pathway in Figure 5 and verification results

	CTL Formula	Result
P1	Fus3 = 1 → AX(Dig1/2 = -1)	True
P2	AG(Ste11 = 1 → AX(Ste7 = 1))	False
P3	Msg5 = 1 → AF(Fus3 = -1 & Far1 = -1 & Dig1/2 = 1)	True
P4	AG((Ste7 = 1 → AF(Fus3 = 1)) & (Fus3 = 1 → AF (Ste7 = 1)))	False

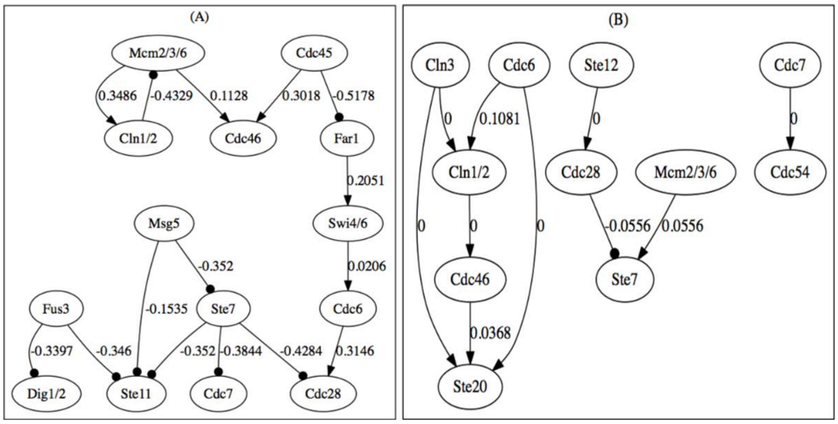


Figure 6 Two optimal subnetworks of cell cycle inferred by Banjo. (A) and (B) are inferred optimal networks based on the *i*2 interval discretization and *q*2 quantile discretization methods respectively.

shows two inferred candidates of “optimal” subnetworks based on interval (*i*2, Figure 6A) and quantile (*q*2, Figure 6B) discretization methods. The difference between these two “optimal” networks demonstrates that this inference algorithm is very sensitive to the choice of data discretization methods. Both figures in Figure 6 are weighted and directed, however, some weights on the Figure 6B are 0s, which means that, Banjo can not identify the signs (activation or inhibition) of these interactions. Next we will apply weighted SMV to verify the optimal network shown in Figure 6A.

Table 3 summarizes the verification results of some desired temporal logic formulas (Q1-Q4 are same as P1-P4 in Table 2) for the inferred *i*2 “optimal” network in Figure 6A. In the yeast, SWI4 regulates the transcription of Cln1 (property Q6), and Cdc28 is a downstream gene regulated by the MAPK pathway (Q7) [38]. SMV model checker verified Q6 but falsified Q7, which indicates a misconnection between Cdc28 and MAPK pathway during the network inference made by Banjo. Property Q9, which describes an oscillation behavior in the yeast, is also verified to be true by SMV model checker. So, 6 out of 9 properties are satisfied by the inferred network. Since the microarray data contains a small number of time points and a lot of measurement noise, we can not expect the inferred “optimal” networks to be completely correct. However, the model checking technique in this work can help identify the best optimal network which satisfies all or most temporal logic properties from all the possible candidates of inferred networks.

Conclusions

A comprehensive understanding of the signaling pathways or gene regulatory networks will advance our knowledge in molecular biology. Network reconstruction from high-dimensional microarray data can help researchers to investigate the crosstalk of different pathways and develop effective multi-gene targeted treatments for some diseases, e.g., cancer and neurodegenerative diseases. Previous studies develop different statistical inference algorithms [13,14,39] to reconstruct gene regulatory network from time series expression data. The validation of inferred networks is implemented manually by comparing with public database or existing models, and, normally, a quantitative comparison is used to evaluate the superiority of a new approach [40,41]. In this work, we proposed a novel

Table 3 CTL formulas related to cell cycle subnetwork in Figure 6A and verification results

	CTL Formula	Result
Q1:	Fus3 = 1 → AX(Dig1/2 = −1)	True
Q2:	AG(Ste11 = 1 → AX(Ste7 = 1))	False
Q3:	Msg5 = 1 → AF(Fus3 < 0 & Far1 < 0 & Dig1/2 = 1)	True
Q4:	AG((Ste7 = 1 → AF(Fus3 = 1)) & (Fus3 = 1 → AF(Ste7 = 1)))	False
Q5:	AG((Swi4/6 = 1 → AF(Cdc6 ≥ 0)))	True
Q6:	AG((Swi4/6 = 1 → AF(Cln12 ≥ 0)))	True
Q7:	AG((Cdc45 = 1 | Msg5 = 1 → EF(Cdc28 = 1 & Mcm2/3/6 ≥ 0)))	False
Q8:	EG((Swi4/6 = 1 → EF(Cdc28 ≤ 0)))	True
Q9:	AG((Mcm2/3/6 = 1 → AF(Cln12 = 1)) & (Cln12 = 1 → AF(Mcm2/3/6 ≤ 0)))	True

procedure, which integrates the dynamic Bayesian network inference algorithm with formal verification technique implemented by Banjo and weighted SMV model checker respectively, to analyze time series gene expression data. The dynamic Bayesian network inference algorithm implemented by Banjo could infer optimal networks of highest scores with directed and weighted edges, however, this method is sensitive to the choice of data discretization methods. The weighted symbolic model checker will exhaustively search the state space to verify or falsify these network candidates through checking some desired temporal logic formulas. Compared with previous studies, the proposed procedure can automatically infer, verify or falsify a biological network based on existing experiments, so it has advantages in the large network inference and verification. The goodness of the verified network will be dependent on not only the learning scores, but also the number of verified temporal logic formulas. One of the key issues in the model checking procedure is the quantity and also quality of the desired temporal logic formulas, which can be abstracted directly from existing experimental results or public database. The more temporal properties we have, the more constrains we can impose on the inferred network candidates. Currently, the inferred regulatory networks are manually encoded into SMV program for model checking. Our future work will build a bioinformatics infrastructure which integrates statistical inference algorithms with different model checkers in a unified framework to automatically infer, encode network candidates into SMV program, and formally verify the inferred gene regulatory networks to select the best models.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

HG proposed the study, HG, JK, KD prepared the computational code, HG, JK, KD, XL, SH analyzed the results, HG wrote the manuscript. All authors approved the final manuscript.

Acknowledgements

HG would like to thank Dr. Hartemink for answering some questions related to Banjo. This work was partially supported by HG's new faculty start-up grant and President Research Fund award (230152) from the Saint Louis University.

Declarations

Publication of this article was funded by the Saint Louis University to HG. This article has been published as part of *BMC Bioinformatics* Volume 16 Supplement 7, 2015: Selected articles from The 11th Annual Biotechnology and Bioinformatics Symposium (BIOT-2014): Bioinformatics. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/16/S7.

Published: 23 April 2015

References

1. Statnikov A, Aliferis CF, Tsamardinos I, Hardin D, Levy S: A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics* 2005, **21**:631-643.
2. Luan Y, Li H: Group additive regression models for genomic data analysis. *Biostatistics* 2008, **9**:100-113.
3. Wu TT, Chen YF, Hastie T, Sobel E, Lange K: Genomewide association analysis by lasso penalized logistic regression. *Bioinformatics* 2009, **25**:714-721.
4. Ma S, Song X, Huang J: Supervised group lasso with applications to microarray data analysis. *BMC Bioinformatics* 2007, **8**:60-76.
5. Wu TT, Wang S: Doubly regularized cox regression for high-dimensional survival data with group structures. *Statisstics and Its Interface* 2013, **6**:175-186.
6. Yuan M, Lin Y: Model selection and estimation in regression with grouped variables. *J R Statist Soc B* 2006, **68**:49-67.
7. Wu TT, Gong H, Clarke EM: A transcriptome analysis by lasso penalized cox regression for pancreatic cancer survival. *Journal of Bioinformatics and Computational Biology* 2011, **9**:63.
8. Gong H, Wu TT, Clarke EM: Pathway-gene identification for pancreatic cancer survival via doubly regularized cox regression. *BMC Systems Biology* 2014, **8.**
9. Akutsu T, Miyano S, Kuhara S: Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics* 2000, **16**:727-734.
10. Chen T, He H, Church G: Moeling gene expression with differential equations. *Pacific Symposium on Biocomputing* 1999, 29-40.
11. Friedman N, Linial M, Nachman I, Pe'er D: Using bn to analyze expression data. *J Comp Biol* 2000, **7**:601-620.
12. Imoto S, Goto T, Miyano S: Estimation of genetic networks and functional structures between genes by using bn and nonparametric regression. *Pacific symposium on Biocomputing* 2002.
13. Kim S, Imoto S, Miyano S: Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in Bioinformatics* 2003, **4**:228-235.
14. Friedman N, Murphy K, Russell S: Learning the structure of dynamic probabilistic networks. *Prpceedings of the 14th conference on the uncertainty in artificial intelligence* 1998.
15. Ong I, Glasner J, Page D: Modelling regulatoruypathways in e. coli from time series expression profiles. *Bioinformatics* 2002, **18**:241-248.
16. Kim S, Imoto S, Miyano S: Dynamic bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *BioSystems* 2004, **75**:57-65.
17. Perrin B, Ralaivola L, Mazurie A, *et al*: Gene networks inference using dynamic bayesian networks. *Bioinformatics* 2003, **74**:138-148.
18. Liang X, Xia Z, Zhang L, Wu F: Inference of gene regulatory subnetworks from time course gene expression data. *BMC Bioinformatics* 2012, **13**:3.
19. Clarke EM, Grumberg O, Peled DA: *Model Checking* MIT Press; 1999.
20. Gong H: Analysis of intercellular signal transduction in the tumor microenvironment. *BMC Systems Biology* 2013, **7(Suppl 3)**:S5.
21. Gong H, Zuliani P, Komuravelli A, Faeder JR, Clarke EM: Analysis and verification of the HMGB1 signaling pathway. *BMC Bioinformatics* 2010, **11(Suppl 7)**:S10.
22. Gong H, Zuliani P, Komuravelli A, Faeder J, Clarke E: Computational modeling and verification of signaling pathways in cancer. *Proceedings of Algebraic and Numeric Biology, LNCS* 2012, **6479.**
23. Gong H, Wang Q, Zuliani P, Lotze MT, Faeder JR, Clarke EM: Symbolic model checking of the signaling pathway in pancreatic cancer. *Proceedings of the International Conference on Bioinformatics and Computational Biology (BICoB)* 2011.
24. Gong H, Zuliani P, Clarke E: Model checking of a diabetes-cancer model. *3rd International Symposium on Computational Models for Life Sciences* 2011.
25. Gong H, Wang Q, Zuliani P, Clarke E: Formal analysis for logical models of pancreatic cancer. *50th IEEE Conference on Decision and Control and European Control Conference* 2011.
26. Gong H, Feng L: Computational analysis of the roles of er-golgi network in the cell cycle. *BMC Systems Biology* 2014, **8(Suppl 4)**:S3.
27. Gong H, Feng L: Probabilistic verification of er stress-induced signaling pathways. *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine* 2014.
28. Sladeczek J, Hartemink A, Robinson J: Banjo: Bayesian network inference with java objects. *User Guide* 2005.
29. Banjo Software. [http://www.cs.duke.edu/~amink/software/banjo/].
30. Heckerman D, Geiger D, Chickering D: Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 1995, **20(3)**.

31. Heckerman D: **A tutorial on learning with bayesian networks.** *Technical Report MSR-TR-95-06, Microsoft Research* 1996.
32. Yu J, Smith V, Wang P, Hartemink A, Jarvis E: **Advances to bayesian network inference for generating causal networks from observational biological data.** *Bioinformatics* 2004, **20**:3594-3603.
33. McMillan KL: *PhD Thesis: Symbolic Model Checking an Approach to the State Explosion Problem* Carnegie Mellon University; 1992.
34. Bryant RE: **Graph-based algorithms for boolean function manipulation.** *IEEE Tran. on Computers* 1986, **35(8)**:677-691.
35. Spellman P, Sherlock G, Zhang M, *et al*: **Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hydridization.** *Mol Biol Cell* 1998, **9**:3273-3297.
36. **S. Cerevisiae Expression Data by Spellman.** [http://downloads.yeastgenome.org/expression/microarray/].
37. **Computer Code.** [http://cs.slu.edu/~gong/Banjocode.zip].
38. Chai L, Mohamad M, *et al*: **A dynamic bayesian network-based model for inferring gene regulatory networks from gene expression data.** *International Journal of Bio-Science and Bio-Technology* 2014, **6**:41-52.
39. Novikov E, Barillot E: **Regulatory network reconstruction using an integral additive model with flexible kernel functions.** *BMC Systems Biology* 2008, **2**:8.
40. R RS, Ventura D, Prince J: **Controlling for confounding variables in ms-omics protocol: why modularity matters.** *Brief Bioinform* 2014, **15(5)**:768-70.
41. Smith1 R, Ventura D, Prince JT: **Novel algorithms and the benefits of comparative validation.** *Bioinformatics* 2013, **29**:1583-1585.

doi:10.1186/1471-2105-16-S7-S7
Cite this article as: Gong *et al.*: A novel procedure for statistical inference and verification of gene regulatory subnetwork. *BMC Bioinformatics* 2015 **16**(Suppl 7):S7.