Inference and Verification of Probabilistic Graphical Models from High-Dimensional Data

Yinjiao Ma[1], Kevin Damazyn[2], Jakob Klinger[2], and Haijun Gong[2(✉)]

[1] Department of Biostatistics, Saint Louis University, St. Louis, MO, USA
[2] Department of Mathematics and Comptuer Science, Saint Louis University, 220 N Grand Blvd, St. Louis, MO, USA
hgong2@slu.edu

Abstract. Probabilistic graphical modelling technique has been widely used to infer the causal relations in the network from high-dimensional data. One of the most challenging biological questions is the inference and verification of biological network, for example, gene regulatory network and signaling pathway, from high-dimensional omics data. Conditionally dependent genes and undirected network can be inferred from the independently and identically distributed static data, while the time series data can help reconstruct a directed network which is more important to our understanding of the complex biological system. Due to the curse of dimensionality and network sparsity, statistical inference algorithm alone is not efficient and realistic to infer and verify large networks. In this work, we propose a novel technique, which applies the dimensionality reduction, network inference and formal verification methods together to reconstruct some regulatory networks from the static and time-series microarray data. A graphical lasso algorithm is first applied to learn the structure of Gaussian graphical models from static data and infer some conditionally dependent genes. Then, an extended dynamic Bayesian network method is applied to reconstruct some weighted and directed networks from the time series data of selected genes, and also generate symbolic model verification code for model checking. Finally, we apply this technique to reconstruct and verify some regulatory networks in yeast and prostate cancer in response to stress and irradiation respectively for illustration.

Keywords: Dimensionality reduction · Gaussian graphical model · Graphical lasso · Dynamic Bayesian network · Model checking · Microarray · Prostate cancer

1 Introduction

High-dimensional data, including the static and time-series types, provide abundant and important information to help us understand the dynamic and temporal properties in the complex system. One of the most challenging biological questions is the inference and verification of complex biological network, for example,

N. Ashish and J.-L. Ambite (Eds.): DILS 2015, LNBI 9162, pp. 223–239, 2015.
DOI: 10.1007/978-3-319-21843-4_18

gene regulatory network and signaling pathway, from high-dimensional omics data. Correctly deciphering the gene regulatory networks can elucidate some fundamental biological processes in the cell and pathogenesis of some diseases. A number of machine learning and statistical inference algorithms [18,26,28], including the graphical model methods [5,6,16], have been proposed to study the gene regulatory networks (GRN) from microarray data, where, each node represents a variable, and the edge connecting two nodes indicates a possible causal relationship.

Since the chemical reactions in the regulatory network are stochastic processes, probabilistic graphical model methods have been widely used to describe the conditional dependence between two random variables in the network. Two nodes in the graph are connected if and only if they are conditionally dependent given the other variables. Gaussian graphical model (GGM) has attracted a lot of attention from computational biologists to learn network structures from static microarray data. In the Gaussian graphical models, the random variable vector X follows a multivariate Gaussian distribution with mean-vector μ and covariance matrix Σ. In the undirected graph, a missing edge implies a conditional independence between two random variables given the rest. So, the problem of inferring a GGM is equivalent to estimating an inverse covariance matrix Σ^{-1}, where the non-zero off-diagnol element indicates the existence of an edge between two nodes. Due to a large number of covariates p (e.g., genes) and insufficient observations ($n << p$), different optimization techniques [2,20], e.g., graphical lasso regularization algorithm [4], have been proposed to estimate the inverse covariance matrix and increase its sparsity through maximizing the L1-penalized log-lihelihood function. However, these techniques could only infer an undirected network, while other important information, such as the positions of genes in the pathway, upstream or downstream, activation or inhibition relationship can not be inferred.

Dynamic Bayesian network (DBN) [6,16,17,22] is a promising learning technique that can reconstruct a directed gene regulatory network with feedback loops from time-series data. Expectation-maximisation algorithm [24] can estimate the parameters in the model. DBN method is based on the first-order Markov chain, and different DBN-based softwares have been developed to increase inference accuracy and reduce computational time. However, most of these softwares can not either infer the "activation" or "inhibition" relationship between different genes in the directed network, or estimate the interaction strength. Moreover, the inferred networks are sensitive to the data discretization policies. Bayesian network inference with Java objects, called Banjo [28] which is based on DBN, can infer an optimal directed and weighted network through calculating an signed (activation or inhibition) influence score for each edge.

Another important aspect in the gene regulatory network learning is the model validation. Previous studies focus on the development of novel inference algorithms to learn a statistically optimal network, and the inferred networks are manually compared with existing database or known models, which is not realistic in the large network verification. Our work has proposed and applied a formal verification technique, called model checking, alone to study some given

signaling pathways [10,11,13,14,19] in the cancer cells. Model checking [3] can automatically and exhaustively search the state space to determine whether or not a given system satisfies some desired temporal logic formula. Recently, we proposed a novel procedure [9] to apply the dynamic Bayesian network algorithm with model checking technique to infer and verify a subnetwork from time series data of yeast. In that work, we have to manually select a subset of genes to reconstruct a subnetwork, and also manually prepare the formal verification code to do model checking for each network, which is not realistic for the verification of multiple large networks.

The goal of this work is to integrate the dimensionality reduction, network inference and model checking methods to reconstruct and verify gene regulatory network from high-dimensional static and time-series microarray data. A graphical lasso algorithm is first applied to learn an undirected Gaussian graphical model and identify some conditionally dependent variables from static data. Then, a dynamic Bayesian network inference method (modified Banjo) is applied to reconstruct some directed and weighted regulatory network candidates, and also automatically generate formal verification code for each model. Finally, model checker is applied to automatically verify the inferred networks. We illustrate this technique to reconstruct and verify gene regulatory networks from the microarray data of yeast and prostate cancer.

2 Statistical Learning and Verification Methods

2.1 Dimensionality Reduction with Graphical Lasso

In this section, we assume the observations measuring the expression levels of genes in the static microarray data are independent and follow the Gaussian distribution, that is, the random vectors $\mathbf{X_1}, \ldots, \mathbf{X_n} \sim N(\mu, \Sigma_p)$, where $\mu \in R^p$ is a mean vector (p denotes the number of features or genes) and Σ_p is $p \times p$ covariance matrix. In this work, without loss of generality, we assume $\mu = 0$. The precision matrix $\Theta = \Sigma_p^{-1}$ has been used to represent the conditional independence and dependence relationship among the variables [4] in the Gaussian graphical models (GGM). A non-zero off-diagonal element ($\theta_{ij} \neq 0$) in the precision matrix indicates a dependence between two covariates, while, $\theta_{ij} = 0$ implies a conditional independence between two variables given the rest.

Inference of the GGM is equivalent to estimating the elements in an unknown precision matrix Θ, which are taken as random variables instead of fixed parameters. Then, the problem is to optimize the log-likelihood function, that is, the log of a joint probability density function which is expressed as

$$\begin{aligned} l(\mathbf{X_1}, \ldots, \mathbf{X_n}, \Theta) &= logP(\mathbf{X_1}, \ldots, \mathbf{X_n}|\Theta) + logP(\Theta) \\ &\propto \frac{n}{2} \log\det\Theta - \frac{n}{2}\mathrm{tr}(\Theta\mathbf{S}) + logP(\Theta), \end{aligned} \tag{1}$$

where $\mathbf{S}$ is an observed or empirical covariance matrix of the data, and it is written as

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{X_i} - \bar{X})(\mathbf{X_i} - \bar{X})^T.$$

If we assume the random variables Θ follow Laplacian distribution, that is, the prior

$$\theta_{ij} \sim \frac{\lambda}{2} exp(-\lambda|\theta_{ij}|),$$

then, the likelihood function in Eq. 1 can be written as

$$l(\mathbf{X_1}, \ldots, \mathbf{X_n}, \Theta) \propto \frac{n}{2}\log\det\Theta - \frac{n}{2}\mathrm{tr}(\Theta\mathbf{S}) - \frac{n}{2}\lambda||\Theta||_1, \tag{2}$$

Equation 2 is equivalent to the graphical lasso method which builds undirected graphs by panalizing the off-diagonal elements of Θ with an $L1$ norm proposed by Friedman et al. [4]. The optimization problem is expressed as

$$\underset{\Theta}{\text{maximize}}\ l(\Theta) = \underset{\Theta}{\text{maximize}}\{\log\det\Theta - \mathrm{tr}(\mathbf{S}\Theta) - \lambda||\Theta||_1\}, \tag{3}$$

where λ is a nonnegative tuning parameter controling the sparsity of the matrix, and $||\Theta||_1 = \sum_{ij}|\theta_{ij}|$. That is, we need solve the following equation

$$\frac{\partial}{\partial\Theta} l(\Theta) = \Theta^{-1} - \mathbf{S} - \lambda \cdot sign(\Theta) = 0. \tag{4}$$

Algorithm 1 describes the graphical lasso based on the block-coordinate descent method which can estimate the sparse precision matrix Θ. We will briefly discuss the procedure to solve Eq. 4 proposed in [4]. Each matrix, including $\mathbf{W} = \Theta^{-1}$, Θ, and $\mathbf{S}$, will be partitioned as following:

$$\begin{pmatrix} \mathbf{W_{11}} & \mathbf{w_{12}} \\ \mathbf{w_{12}^T} & w_{22} \end{pmatrix}, \begin{pmatrix} \Theta_{11} & \theta_{\mathbf{12}} \\ \theta_{\mathbf{12}}^{\mathbf{T}} & \theta_{22} \end{pmatrix}, \begin{pmatrix} \mathbf{S_{11}} & \mathbf{s_{12}} \\ \mathbf{s_{12}^T} & s_{22} \end{pmatrix},$$

where the sizes of ($\mathbf{W_{11}}$, Θ_{11}, $\mathbf{S_{11}}$) and ($\mathbf{w_{12}}$, $\theta_{\mathbf{12}}$ and $\mathbf{s_{12}}$) are $(p-1)\times(p-1)$, $(p-1)\times 1$ respectively, w_{22}, θ_{22} and s_{22} are scalars.

Algorithm 1. Graphical lasso based on a block-coordinate descent method

Input: Omics Data, $\mathbf{S}, \lambda$
Output: $\mathbf{\Theta}$, conditionally dependent variables
Initialization: $\mathbf{W} = \Theta^{-1} = \mathbf{S} + \lambda\mathbf{I}$
while $\mathbf{W}$ *is not converged* **do**
  Partition of matrix $\mathbf{W}$ into blocks
  Apply block-coordiate descent approach to solve L1 lasso penalized problem.
  Update $\mathbf{w_{12}}$
end
Calculate $\theta_{22} = 1/(w_{22} - \mathbf{w_{12}^T W_{11}^{-1} w_{12}})$;
Calculate $\theta_{\mathbf{12}} = -\theta_{22}\mathbf{W_{11}^{-1}w_{12}}$.
return $\mathbf{\Theta}$;
output conditionally dependent variables.

Then, the elements in the precision matrix Θ can be expressed as $\theta_{\mathbf{12}} = -\theta_{22}\mathbf{W_{11}^{-1}w_{12}}$, and $\theta_{22} = 1/(w_{22} - \mathbf{w_{12}^T W_{11}^{-1} w_{12}})$, where $\mathbf{w_{12}} = -\mathbf{W_{11}}\theta_{\mathbf{12}}/\theta_{22}$. Graphical lasso method (implemented by the R package GLASSOPATH) applies $L1$ lasso algorithm based on a block-coordinate descent method to estimate the sparse precision matrix.

We can apply the graphical lasso algorithm to infer a sparse undirected probabilistic graphical model which is composed of conditionally-dependent genes. However, biologists are more interested in the directed regulatory network which contains not only the conditional dependence information, but also the activation or inhibition relationship between two genes. Next we will introduce an extended dynamic Bayesian network inference algorithm which can reconstruct directed regulatory networks from the time series data and automatically generate formal verification code for model checking.

2.2 Directed Network Inference

In the time series microarray data, the expression levels of p genes at n different time points can be described by the random vectors $\mathbf{X_1}, \ldots, \mathbf{X_n}$, where $\mathbf{X_i} = (X_{i1}, ..., X_{ip})^T$ is defined as the p random variables (e.g., genes) measured at time i, and x_{ij} represents an observation value (expression level) of the random variable X_{ij}. Dynamic Bayesian network (DBN) [6,16,22] has been used to reconstruct a directed network, where each edge can be either activation or inhibition relationship between two nodes. This method is based on the first-order Markov chain assumption that, each random variable at time i is dependent on its parents at time $i-1$ only. Therefore, a directed network can be graphically represented by a joint distribution of n random vectors over time [16], which is expressed as $P(\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_n}) = P(\mathbf{X_1})P(\mathbf{X_2}|\mathbf{X_1}) \times \ldots \times P(\mathbf{X_n}|\mathbf{X_{n-1}})$, and $P(\mathbf{X_i}|\mathbf{X_{i-1}}) = P(X_{i1}|Par(X_{i1})) \times ... \times P(X_{ip}|Par(X_{ip}))$, where $Par(X_{ij})$ represents the gene j's parents at time $i-1$ [9,16].

In this work, we use the $n \times p$ matrix $\mathbf{X}$ to represent the time-series microarray data which consists of p genes measured at n different time points, and also it is discrete. The goodness of a network is evaluated by the likelihood-equivalence Bayesian Dirichlet (BDe) scoring function proposed in Heckerman et al's work and used by many researchers in the network inference studies [9,15,16,22]. This work will apply the Bayesian network inference with Java objects [28], which is a network learning software, to calculate BDe scores. The idea is to maximize the posterior probability distribution of the network G conditional on the microarray data $\mathbf{X}$, which is written as

$$P(G|\mathbf{X}) \propto P(G, \mathbf{X}) = \int P(G, \mathbf{X}, \mathbf{\Theta}) d\mathbf{\Theta} = P(G) \int P(\mathbf{X}|G, \mathbf{\Theta}) P(\Theta|G) d\mathbf{\Theta},$$

where, $P(G)$ is the prior of the network G, which can be chosen in different ways, for example, Friedman chose $P(G)$ based on the minimal description length (MDL) encoding of G. The BDe score function is based on the following assumptions [15]:

1. The data $\mathbf{X}$ is a multinomial sample dependent on the parameters $\mathbf{\Theta}$, that is, $\mathbf{X}|\mathbf{\Theta} \sim Multinomial(\mathbf{\Theta})$;
2. The parameters in $\mathbf{\Theta}$ are globally and locally independent;
3. Given a network G, the parameters in $\mathbf{\Theta}$ follows Dirichlet distribution with a hyperparameter vector α, that is, $\mathbf{\Theta}|G \sim Dirichlet(\alpha)$. The Dirichlet function has been given in [15,16].
4. Two directed acyclic networks G_1, G_2 are equivalent if they encode the same joint probability distribution;
5. If the network G_1 is equivalent to G_2, the distribution function of $\mathbf{\Theta}$ will be same in both networks.

The BDe scores for all possible networks will be calculated by the Bayesian network inference with Java objects [28], then, a greedy searching or simulated annealing algorithm proposed by Heckerman will be used to find optimal networks. Learning the activation and inhibition relationship between two genes will help us comprehensively understand the mechanism underlying the gene regulatory network. The influence score proposed in Yu et al.'s work [28] can be used to identify the activation and inhibition relationship and interaction strength. A positive influence score indicates an activation event, while a negative value corresponds to an inhibition event between two nodes. If the influence score is close to 0, the sign can not be identified based on the current time series data. The Bayesian network inference with Java objects [28] used a voting system and the value of a cumulative distribution function to estimate the influence score

$$G_{ijk}(t) = \sum_{l=0}^{k} \omega_{ijl}(t) = \sum_{l=0}^{k} P(X_{ti} = l \,|Par(X_{ti}) = j). \tag{5}$$

$G_{ijk}(t)$ describes the probability that, at time t, gene X_{ti} takes a (discrete) value less than or equal to k given its parent gene takes a value of j, where, $\omega_{ijk}(t)$ 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 interested reader could refer to [28] for details. Algorithm 2 shows the procedure of dynamic Bayesian network inference based on BDe metrics and influence score estimation. We modified the Banjo code to search and output top n high-scoring directed networks with influence scores, and automatically generate the weighted SMV formal verification code for each network to do model checking.

Model verification is another aspect to studying the gene regulatory network due to the complexity of biological system. How to verify or falsify the network candidates inferred by the DBN? Recently, we proposed a weighted symbolic model verification (SMV) technique [9] to formally verify the networks. However, in that work [9], the regulatory subnetworks are inferred from a given subset of genes, and they are manually encoded into SMV program for model checking, which is not realistic and efficient to encode multiple large networks.

Besides the dimensionality reduction method is first used to select the conditionally dependent genes intead of manually selecting a subset of genes, another novelty in this work is that, the extended dynamic Bayesian network inference

Algorithm 2. Directed graph inference based on dynamic Bayesian network method

Input : Conditionally-dependent variables selected by Algorithm 1;
Time series data $\mathbf{X}$
Output: Directed networks of top n BDe scores;
Symbolic model verification (SMV) code for each network.

Data Discretization;
for *each network* G **do**
  Evaluate the goodness of a network ;
    Data $\mathbf{X}|\mathbf{\Theta} \sim Multinomial(\mathbf{\Theta})$;
    $\mathbf{\Theta}|G \sim Dirichlet(\alpha)$;
    Calculate BDe score based on $P(G|\mathbf{X})$;
  Network searching;
    Network sorting based on BDe scores;
    Greedy search or simulated annealing algorithm;
  Estimate influence score;
    for *each edge* $(X_i, Par(X_i))$ **do**
      Compute $\omega_{ijk}(t) = P(X_{ti} = k|Par(X_{ti}) = j)$;
      Estimate $G_{ijk}(t) = \sum_{l=0}^{k} \omega_{ijl}(t)$;
      Identify the sign and magnitude of interaction using a vote
    end
  Output top n networks;
  Generate weighted SMV code for each model.
end

with Java object method can automatically encode all the inferred network candidates into weighted SMV program for model checking. Next, we will introduce the weighted symbolic model checking technique used for network verification which has been discussed in our recent work [9].

2.3 Network Verification

An inferred network might be trustable only if it is consistent with the existing experiment or knowledge. Previous studies manually compared the inferred network with existing database or known models. Our recent studies [9,14] have demonstrated the power of formal verification technique in the biological studies. Model checking [3] is a powerful and automatic formal verification method, it can check whether or not a given model M satisfies a desired temporal logic formula ψ, denoted by $M \models \psi$. Different model checkers have been developed and successfully applied to verify the hardware and software systems in the past thirty years.

We have discussed the model checking technique in our recent work [9,14], for completeness, we will review some fundamental concepts and algorithms in this work. In formal verification studies, a model is described as a Kripke structure [3,12] $M = (S, s_0, R, L)$ with the initial state $s_0 \in S$, states transition relation R, and a labeling function L. SMV [21] is one of the most popular symbolic model checking tools that are encoded by ordered binary decision diagram

(OBDD) [1]. During model checking process, SMV model checker will automatically and exhaustively search the state transition system M to verify some desired property ψ which is expressed as a computation tree logic (CTL) formula. CTL formula is composed of Boolean logic connectives, temporal operators describing some property on a path and path quantifiers describing the branching structure in the computation tree. Table 1 lists these operators and the corresponding meanings. For example, $\mathbf{AG}(\mathbf{AF})\phi$ means ϕ is globally (finally) true on all paths; $\mathbf{EG}(\mathbf{EF})\phi$ means ϕ will be true always (in the future) on some path. In this work, most CTL formulas are constructed using these 6 operators: $\mathbf{AX}, \mathbf{EX}, \mathbf{AG}, \mathbf{EG}, \mathbf{AF}, \mathbf{EF}$. The interested readers could refer to [3] for more interesting operators. In the CTL syntax, the state formula and path formula are represented by ψ and ϕ respectively, and an infinite sequence of states or a path is denoted by π. The interested readers could refer to [3] for the syntax and semantics of CTL logic, and CTL formulas. After verification, the SMV model checker will output either "True" if the property is satisfied, or "False" with a counter example.

Table 1. Boolean logic connective, temporal operator and path quantifier in CTL formula.

Operators	!	|	&	→	X	F	G	U	A	E
Meaning	not	or	and	implies	neXt	Future	Globally	Until	All paths	There Exists some path

Algorithm 3 (Part I) presents the weighted symbolic model checking pseudocode of SMV program that can be automatically genetated by the Algorithm 2 for each network, which is an extension of the unweighted model checking method. Similar to the unweighted model checking code, the program should start with "MODULE MAIN", and all the variables are defined and initialized by "VAR" and "init" respectively under the keyword "ASSIGN". The difference is, the state transition update for each variable is not only dependent on its parents' states, but also the *integral* influence score (only integers or Boolean values are allowed in the symbolic model checking), which is calculated by Algorithm 2. SMV model checker will automatically verify all the CTL formulas (encoded by the keyword "SPEC") to find the best models (which could be more than one candidates) satisfying all or most of the properties proposed or desired by the investigators. Part II shows the SMV algorithm based on OBDD data structure [1]. A Boolean function is applied to describe the transition relation between states implicitly. Detailed symbolic model checking algorithm and weighted SMV code have been discussed in [3,9,21].

3 Applications

In this section, we will apply the proposed integrative methods in the Algorithms 1–3 to analyze the static and time series microarray data of yeast and prostate cancer. The graphical lasso method is first applied for dimensionality reduction,

Algorithm 3. Weighted symbolic model checking pseudocode and SMV algorithm

Part I: Weighted symbolic model checking pseudocode

Input 1 : Inferred regulatory networks M by Algorithm 2;
Temporal logic formula ψ
Output 1: True or False

for *each network* **do**
Variable declaration by "VAR";
Variable initialization by "init";
State update with the weighted transfer functions by "next";
CTL formula specification by "SPEC";
$M \models \psi$: output True or False.
end

Part II: SMV Algorithm [3]

Input 2: A model M; desired CTL formulas f, g
Check: Take a CTL formula as its argument
Return: OBDD for the set of states that satisfy a given temporal logic formula.
Output 2: A set of states of M, which satisfy the formula.

- if f is an atomic proposition v_i: return $\mathbf{Check}(f) = v_i$;
- if $\neg f$: return $\mathbf{Check}(\neg f) = \neg\mathbf{Check}(f)$;
- if $f \vee g$: return $\mathbf{Check}(f) \vee \mathbf{Check}(g)$;
- if $\mathbf{EX}f$: return $\mathbf{Check}(\mathbf{EX}(\mathbf{Check}(f)))$;
- if $\mathbf{E}[f\mathbf{U}g]$: return $\mathbf{Check}(\mathbf{EU}(\mathbf{Check}(f), \mathbf{Check}(g)))$;
- if $\mathbf{EG}f$: return $\mathbf{Check}(\mathbf{EG}(\mathbf{Check}(f)))$.

infer an undirected Gaussian graph model from the static data, and identify a subset of genes that are conditionally dependent. Then the dynamic Bayesian network inference method is applied to reconstruct some directed networks from the time series data of conditionally dependent genes, which will be verified by the symbolic model checking technique. The graphical lasso, modified Banjo code, and weighted SMV code developed for this work are available at http://cs.slu.edu/~gong/Research/DILS.zip.

3.1 Yeast Data Analysis

The static microarray data (Accession No: GSE19213) of the yeast studies the transcription factor Yap1 which mediates an adaptive response to oxidative stress (e.g., H_2O_2 or thiol-reactive chemicals) by regulating some protective genes [23]. For illustration, only the top around 5000 differently expressed genes (between treatment and control group) in the wild-type strain treated with H_2O_2 will be used in our studies. For simplicity, the expression levels among different probes that map to the same gene were averaged to a single value in our data analysis.

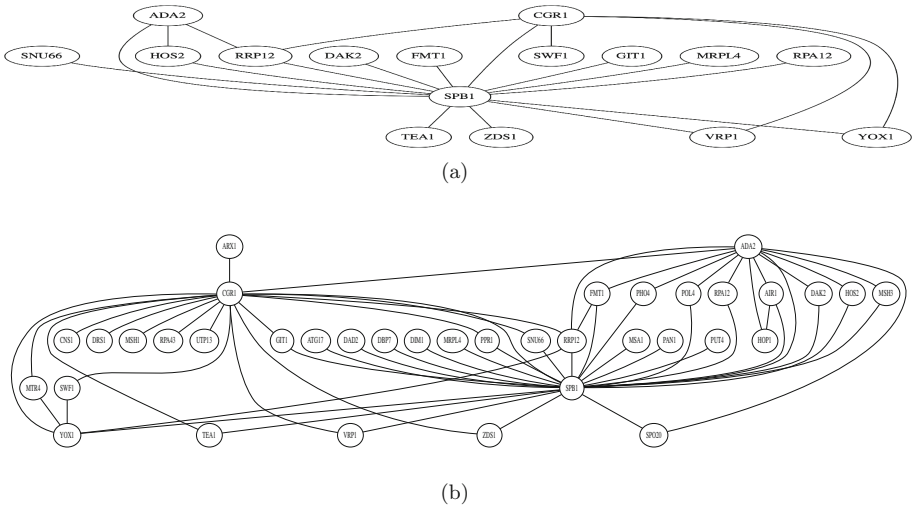

Fig. 1. Conditionally dependent genes and undirected Gaussian graph models inferred from static microarray data of yeast with different values of λ: (a) $\lambda = 0.41$, (b) $\lambda = 0.30$.

Graphical lasso method is applied with different λ values in order to infer the precision matrices ranging from a dense matrix to a sparse one, it is known that a large λ value will result in a sparse matrix. Figure 1 shows two figures with $\lambda = 0.41$ (16 genes) and $\lambda = 0.30$ (37 genes) for demonstration. More figures with a wide range of λ value are given in the online supplementary files. It is apparent that Fig. 1(a) is a subnetwork of Fig. 1(b), SPB1 and CGR1 are highly connected genes, which are also called hub genes. Goel et al's work [8] has confirmed the hub protein SPB1's important role in rRNA processing and ribosome biogenesis.

Then, the modified dynamic Bayesian network inference with Java Object is applied to reconstruct directed regulatory networks of high-BDe scores from time series microarray data and automatically output the encoded symbolic model

Table 2. Proposed CTL formulas for the network verification of yeast

	CTL formula
Property 1	$A(CGR1 = 1 \rightarrow AX(RRP12 = 1))$
Property 2	$EG(TEA1 = 1 \rightarrow EF(SNU66 \leq 0))$
Property 3	$AG(RRP12 = 1 \rightarrow AF(SNU66 = -1))$
Property 4	$AG(CGR1 = 1 \rightarrow AF(MRPL4 < 0 \ \& \ HOS2 < 0 \ \& \ VRP1 = 1))$
Property 5	$EG((SWF1 = 1 \mid FMT1 = 1 \rightarrow EF(RPA12 = 0 \ \& \ GIT1 \geq 0)))$
Property 6	$AG((SNU66 = 1 \rightarrow AF(TEA1 = 1)) \ \&(TEA1 = 1 \rightarrow AF(SNU66 \leq 0)))$

verification (SMV) program for all the network candidates. The time series data (Access No: GSE62120) measure the expression levels of yeast in response to the oxidative stress (H_2O_2) with 11 time points. Only the conditionally dependent genes identified by the graphical lasso in Fig. 1 will be used for the directed network reconstruction. Since the Banjo performs well with a small number of genes, in this work, we select 16 genes for network construction. The goal is to find a network that might regulate the oxidative stress in the yeast.

Figure 2 demonstrates some directed and weighted regulatory network candidates (of top two BDe scores) based on the genes selected in the GGM with $\lambda = 0.41$ and two different discretization policies $q2$ (a–b) and $i2$ (c–d). The solid lines with arrows represent an activation event, while the circle-head arrows represent inhibition processes. The integers on the directed edges represent the

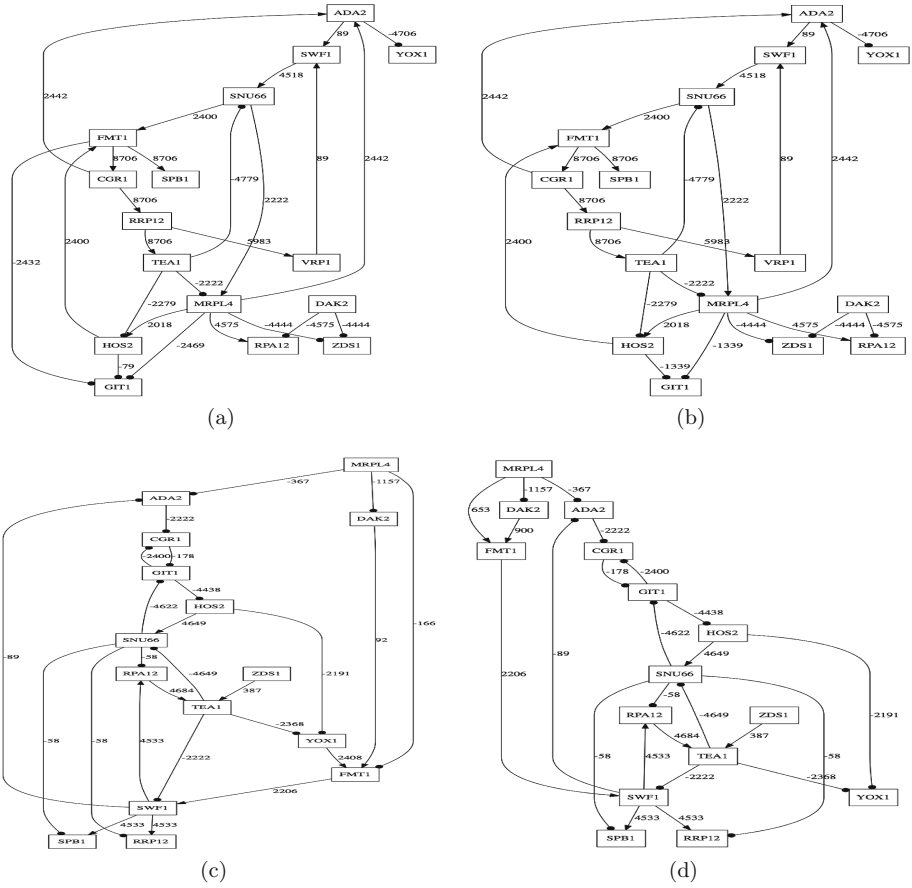

Fig. 2. Top two directed and weighted gene regulatory network candidates of yeast based on $q2$ (a–b) and $i2$ (c–d) discretization policies with $\lambda = 0.41$. Solid lines with arrows represent an activation event, circle-head arrows represent inhibition processes. The values on the directed edges represent the influence scores.

modified influence scores or weights, which will be used for the weighted symbolic model checking. All these network candidates take the MRPL4 gene as a hub, which plays an important role in the protein synthesis within the mitochondrion.

Figures 1 and 2 demonstrate that, given different values of λ, graphical lasso and dynamic Bayesian network inference methods could generate many statistically optimal undirected and directed network candidates. Compared with our previous work, the novelty of this method is that, the extended dynamic Bayesian network inference method, a modified Banjo, could automatically generate SMV verification code for each network candidate which will be used for model checking. Then, next step, we apply SMV model checker to verify or falsify these inferred networks through checking some putative properties that we defined in the Table 2.

Table 2 summarizes some putative CTL formulas that we assume the inferred networks should satisfy. Each gene or variable can take three possible values: $-1, 0, 1$, which denote inhibited, normal and activated, and the initial state is set to be either 0 or -1. Property 1 indicates that RRP12 might be a downstream gene of CGR1, that is, CGR1's activation will immediately activate RRP12 in the neXt step. Property 2 means, there exist a path on which TEA1's activation will finally inhibit SNU66's expression, while Property 3 means, for all paths, it is globally true that RRP12's activation will finally inhibit SNU66's activity. Property 4 and 5 are similar to property 3 and 2 respectively. Property 5 describes a negative feedback loop between SNU66 and TEA1.

Next, symbolic model checker will automatically verify or falsify all the 4 inferred networks shown in Fig. 2, and output either "True" or "False" if some property is satisfied or not respectively. Table 3 summarizes the verification results of these putative properties in 4 different models. Our methods do not intend to infer and verify only one statistically optimal network (as other researchers' work), however, the model checker will search and find one or a pool of "best-fit" networks from a number of inferred network candidates which satisfies all or most of desired temporal logic properties. Moreover, new evidence or future studies can continue to refine the "optimal" network pool with more properties. In our examples, the inferred networks in Fig. 2 (a–b), which satisfy 4 putative properties, might be better than those in Fig. 2 (c–d). But more temporal properties from the wet lab experiments will be needed to identify a really best-fit network candidate.

Table 3. Network verification results of yeast

	Property 1	Property 2	Property 3	Property 4	Property 5	Property 6
Model a	True	True	True	False	False	True
Model b	True	True	True	False	False	True
Model c	True	True	False	False	False	True
Model d	True	True	False	False	False	True

3.2 Prostate Cancer Data Analysis

The static microarray data of prostate cancer [25] contains 639 tumor samples, including 270 African-American and 369 European American patients, where 517 genes linked with prostate cancer were measured by 1,507 probes (Gene Expression Omnibus accession number GSE41969). The expression levels among different probes that map to the same gene were averaged to a single value for the data analysis. The time series data (GSE770) studied the androgen-independent LNCaP C4-2 human prostate adenocarcinoma cells following irradiation, the RNA was extracted from cells at 1, 2, 4, 6, 8, 12, 16, 20 and 24 h after irradiation, and the untreated control sample was labeled 0. Our goal is to find some networks that might be associated with the prostate cancer in response to irradiation.

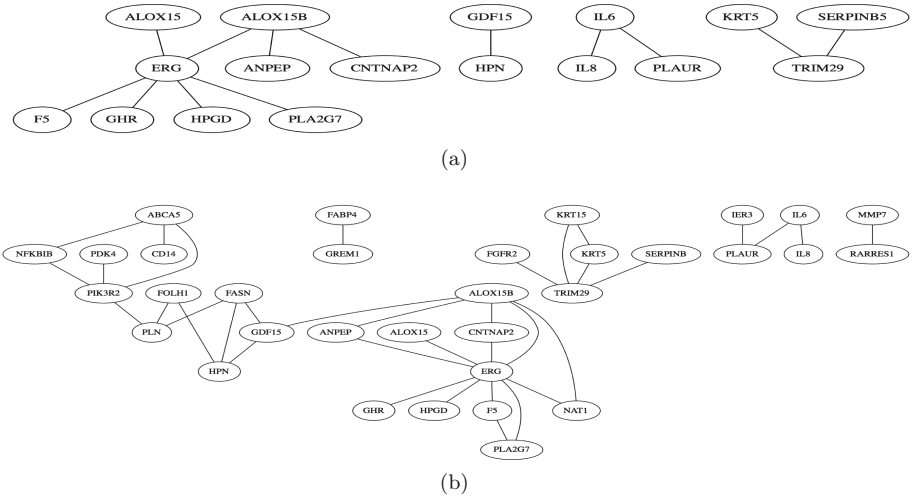

Fig. 3. Conditionally dependent genes and undirected Gaussian graph models inferred from static microarray data of prostate cancer with different values of λ: (a) $\lambda = 0.39$, (b) $\lambda = 0.32$.

Similar to the yeast data analysis, we first select some conditionally dependent genes and infer undirected Gaussian graphical models given different values of λ which are shown in the Fig. 3 ((a) $\lambda = 0.39$, (b) $\lambda = 0.32$). The highly connected genes, including ERG, ALOX15B, TRIM29 have been found to be deregulated in the prostate cancer [7]. ERG activation, one of the most common oncogenic alterations, is present in 50–70 % of prostate tumors; especially, TRIM29 can negatively regulates p53 via inhibition of Tip60 [27], and it is overexpressed in lung, bladder, pancreatic and endometrial cancers, but opposite in prostate cancer.

Figure 4 shows four optimal directed and weighted network candidates of androgen-independent prostate cancer in response to irradiation based on the

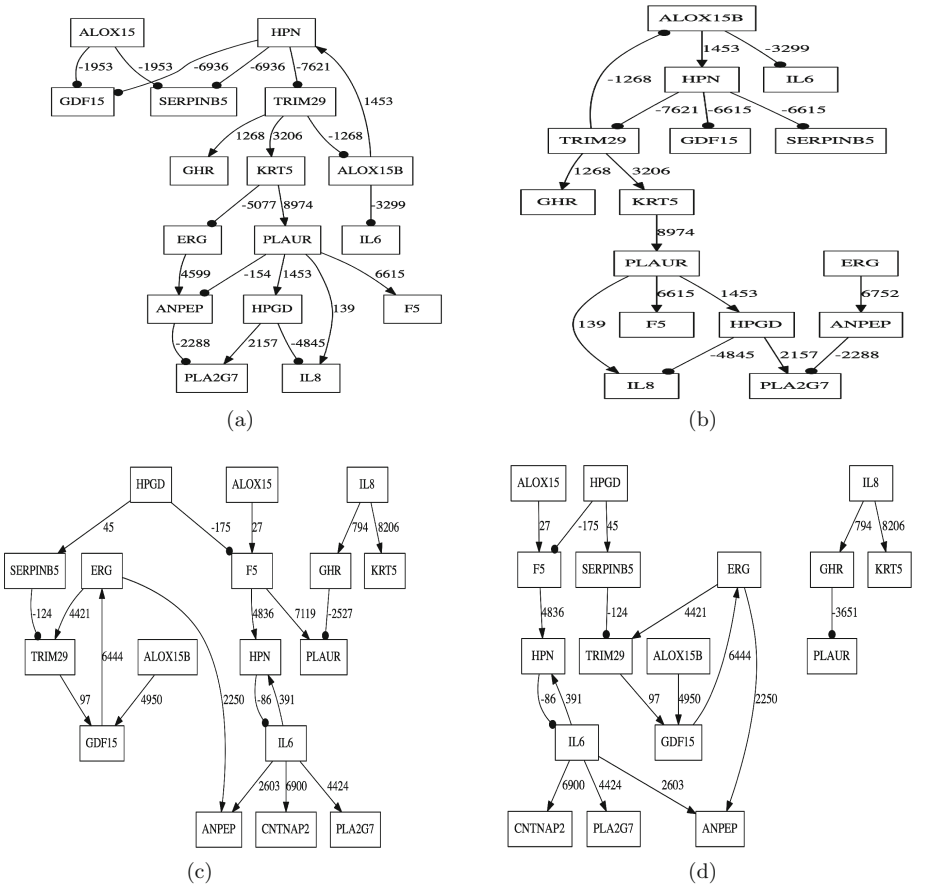

Fig. 4. Top two directed and weighted gene regulatory network candidates of prostate cancer based on $q2$ (a–b) and $i2$ (c–d) discretization policies with $\lambda = 0.39$.

conditionally dependent genes selected by GGM with $\lambda = 0.39$ and two different discretization policies $q2$ (a–b) and $i2$ (c–d) using the modified Banjo software. Besides the verification of desired properties, the model checker could also predict some properties that the future experiments can test. We proposed two predictions which describe two possible feedback loops related to the highly connected gene TRIM29. Prediction 1 incorporates two inhibition events, while Prediction 2 is composed of activation events only. The difference of these two predictions could be observed easily from the Fig. 4, which is a small network, but difficult in the large model. However, the model checker could easily and automatically find this difference in different models through checking the following two properties using the generated SMV formal verification code:

Prediction 1: AG((HPN1 $= 1 \rightarrow$ AF(TRIM29 ≤ 0)) & (TRIM29 $\leq 0 \rightarrow$ AF(ALOX15B ≥ 0)) & (ALOX15B $\geq 0 \rightarrow$ AF(HPN ≥ 0))).

Prediction 2: A((ERG = 1 → AF(TRIM29 = 1)) & (TRIM29 = 1 → AF(GDF15 = 1)) & (GDF15 = 1 → AF(ERG = 1))).

Figure 4(a–b) satisfy both predictions, but Fig. 4(c–d) only satisfy the Prediction 2. The future experiments could help validate these predictions, and help refine the inferred models of prostate cancer cells after irradiation.

4 Conclusions

Correct learning and efficient verification of complex biological networks from high dimensional data is a challening job in systems biology. In this work, we proposed an integrative technique, which incorporates the graphical lasso, dynamic Bayesian network inference algorithm and weighted symbolic model checker, to analyze both static and time series microarray data of yeast and prostate cancer in response to oxidative stress and irradiation respectively. The graphical lasso first identified some conditionally dependent genes and inferred undirected networks from static microarray data that are associated with the oxidative stress or irradiation; then, DBN is applied to reconstruct the directed and weighted networks from time series data using different data discretization policies and automatically generate formal verification code for model checking. Compared with other researchers' work which learns only one statistically optimal network, this proposed method can both infer and verify several optimal networks that satisfy some desired temporal properties. This method is universal and applicable to any type of static and time series data, which can help us investigate the biological networks implicated in the pathogenesis of some diseases.

Our studies found the Bayesian network inference with Java objects [28] method is very sensitive to the data discretization policies, and it could learn and generate reasonably-connected networks only if the number of genes are not very large. However the gene regulatory network is large in fact and the model checking technique is powerful in the verification of large networks. Our future work will develop new learning algorithms which can handle the inference of large number of variables and take advantage of the verification power in model checking. Moreover, we will develop a GUI version to make the network learning and verification easy and convenient.

5 Contribution

HG proposed the project, YM wrote the glasso code and analyzed the microarray data, KD and JK modified the Banjo code to infer directed network and automatically generate verification code.

Acknowledgment. This work was partially supported by HG's new faculty start-up grant and President Research Fund award (230152) from the Saint Louis University.

References

1. Bryant, R.: Graph-based algorithms for boolean function manipulation. IEEE Trans. Comput. **35**(8), 677–691 (1986)
2. Celik, S., Logsdon, B., Lee, S.: Efficient dimensionality reduction for high-dimensional network estimation. JMLR **32** (2014)
3. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press, Cambridge (1999)
4. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics **9**(3), 432–441 (2008)
5. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using BN to analyze expression data. J. Comp. Biol. **7**, 601–620 (2000)
6. Friedman, N., Murphy, K., Russell, S.: Learning the structure of dynamic probabilistic networks. In: Proceedings of the 14th Conference on the Uncertainty in Artificial Intelligence (1998)
7. Furusato, B., Tan, S., et al.: ERG oncoprotein expression in prostate cancer: clonal progression of ERG-positive tumor cells and potential for ERG-based stratification. Prostate Cancer Prostatic Dis. **13**, 228–237 (2010)
8. Goel, A., Wilkins, M.R.: Dynamic hubs show competitive and static hubs non-competitive regulation of their interaction partners. PLoS One **7**(10), e48209 (2012)
9. Gong, H., Klinger, J., Damazyn, K., Li, X., Huang, S.: A novel procedure for statistical inference and verification of gene regulatory subnetwork. BMC Bioinformatics **16**(Suppl 7), S7 (2015)
10. Gong, H., Zuliani, P., Komuravelli, A., Faeder, J.R., Clarke, E.M.: Computational modeling and verification of signaling pathways in cancer. In: Horimoto, K., Nakatsui, M., Popov, N. (eds.) ANB 2010. LNCS, vol. 6479, pp. 117–135. Springer, Heidelberg (2012)
11. Gong, H., Zuliani, P., Komuravelli, A., Faeder, J.R., Clarke, E.M.: Analysis and verification of the HMGB1 signaling pathway. BMC Bioinformatics **11**(Supp 7), S10 (2010)
12. Gong, H.: Analysis of intercellular signal transduction in the tumor microenvironment. BMC Syst. Biol. **7**, S5 (2013)
13. Gong, H., Feng, L.: Computational analysis of the roles of ER-Golgi network in the cell cycle. BMC Syst. Biol. **8**, S4 (2014)
14. Gong, H., Feng, L.: Probabilistic verification of ER stress-induced signaling pathways. In: Proceedings of IEEE International Conference on Bioinformatics and Biomedicine (2014)
15. Heckerman, D., Geiger, D., Chickering, D.: Learning bayesian networks: the combination of knowledge and statistical data. Mach. Learn. **20**(3), 197–243 (1995)
16. Kim, S., Imoto, S., Miyano, S.: Inferring gene networks from time series microarray data using dynamic Bayesian networks. Briefings Bioinf. **4**, 228–235 (2003)
17. 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 **75**, 57–65 (2004)
18. Liang, X., Xia, Z., Zhang, L., Wu, F.: Inference of gene regulatory subnetworks from time course gene expression data. BMC Bioinformatics **13**, S3 (2012)
19. Ma, Y., Feng, L., Guo, Y., Gong, H.: Statistical analysis and probabilistic verification of stress-induced signaling pathways. Int. J. Data Min. Bioinf. (2015)
20. Mazumder, R., Hastie, T.: The graphical lasso: new isights and alternatives. Electron. J. Stat. **6**, 2125 (2012)

21. McMillan, K.L.: Ph.D thesis: Symbolic model checking - an approach to the state explosion problem. Carnegie Mellon University (1992)
22. Ong, I., Glasner, J., Page, D.: Modelling regulatoruypathways in E. coli from time series expression profiles. Bioinformatics **18**, S241–S248 (2002)
23. Ouyang, X., Tran, Q., Goodwin, S., Wible, R., Sutter, C., Sutter, T.: Yap1 activation by H2o2 or thiol-reactive chemicals elicits distinct adaptive gene responses. Free Radic. Biol. Med. **50**, 1–13 (2011)
24. Perrin, B., Ralaivola, L., Mazurie, A., et al.: Gene networks inference using dynamic bayesian networks. Bioinformatics **74**, i138–i148 (2003)
25. Powell, I., Dyson, G., et al.: Genes associated with prostate cancer are differentially expressed in African American and European American men. Cancer Epidemiol. Biomark. Prev. **22**, 891–897 (2013)
26. Rob Smith, R., Ventura, D., Prince, J.: Controlling for confounding variables in MS-omics protocol: why modularity matters. Brief Bioinform. **15**(5), 768–770 (2014)
27. Shoa, T., Tsukiyama, T., et al.: Trim29 negatively regulates p53 via inhibition of Tip60. Mol. Cell Res. **1813**, 1245–1253 (2011)
28. Yu, J., Smith, V., Wang, P., Hartemink, A., Jarvis, E.: Advances to bayesian network inference for generating causal networks from observational biological data. Bioinformatics **20**, 3594–3603 (2004)