International Journal of Bio-Science and Bio-Technology
Vol.6, No.1 (2014), pp.41-52
http://dx.doi.org/10.14257/ijbsbt.2014.6.1.05

A Dynamic Bayesian Network-based Model for Inferring Gene Regulatory Networks from Gene Expression Data

Lian En Chai[1], Mohd Saberi Mohamad[1*], Safaai Deris[1], Chuii Khim Chong[1], Yee Wen Choon[1] and Sigeru Omatu[2]

[1]Artificial Intelligence and Bioinformatics Research Group, Faculty of Computing, Universiti Teknologi Malaysia, Skudai 81310 Johor, Malaysia
[2]Department of Electronics, Information and Communication Engineering, Osaka Institute of Technology, Osaka 535-8585, Japan

lechai2@live.utm.my, saberi@utm.my, safaai@utm.my, ckchong2@live.utm.my, ywchoon2@live.utm.my, omatu@rsh.oit.ac.jp
[*]Corresponding author

Abstract

Driven by the need to uncover the vast information and understand the dynamic behaviour of biological systems, researchers are now garnering interests in inferring gene regulatory networks (GRNs) from gene expression data which is otherwise unfeasible in the past due to technology constraint. In this regard, the dynamic Bayesian network (DBN) has been broadly utilized for the inference of GRNs, thanks to its ability to handle time-series microarray data and model feedback loops. Unfortunately, the commonly found missing values in gene expression data, and the excessive computation time owing to the large search space whereby all genes are treated as potential regulators for a target gene, often impede the effectiveness of DBN in inferring GRNs. This paper proposes a DBN-based model with missing values imputation and potential regulators selection (ISDBN) which deals with the missing values and reduces the search space by selecting potential regulators based on gene expression changes. The performance of the proposed model is assessed by using S. cerevisiae cell cycle and E. coli SOS response pathway time-series expression data. The experimental results showed reduced computation time and improved accuracy in detecting gene-gene relationships when compared to conventional DBN. The results of this study showed that ISDBN performs better than conventional DBN in terms of accuracy and computation time for GRNs inference. Moreover, we foresee the applicability of the resultant networks from ISDBN as a framework for future gene intervention experiments.

Keywords: *Dynamic Bayesian network, missing values imputation, gene expression data, gene regulatory networks, network inference*

1. Introduction

In the post-genomic era, the exponentially increasing data generated from numerous biological fields such as genomics, metabolomics, proteomics and transcriptomics poses a serious challenge for biological researchers to analyse and utilise. Therefore, due to the increasing dependency on computation to scrutinise the high throughput data, the inclination of molecular biology evolving into a quantitative science is inevitable. Along with the science and technology breakthroughs, researchers are now concentrating on the holistic view of the system rather than just approaching one gene or one protein. The term systems biology is

ISSN: 2233-7849 IJBSBT

thereby coined to describe the evolving nature and holism of bioscience research. It represents a new perspective as a biology-based interdisciplinary study field which focuses on the complicated interactions of biological systems from a holistic approach.

One of the goals of systems biology is to understand the underlying mechanism and nature of the regulation of protein synthesis along with its reactions to external and internal stimuli. The framework of sequence information transfer between sequential information-carrying biopolymers was first described in the central dogma of molecular biology. Despite carrying the same genomic data, the regulation process causes the pattern of gene expression products for different kind of cells in an organism to be significantly different temporally and spatially. Nonetheless, these differences of protein makeup are crucial to the processes of life. This sophisticated way of regulating gene expression can be described as gene regulatory networks (GRNs). In other words, GRNs are a collection of gene segments in a cell and their interactions between each other and other substances in the cell, thus resulting in the governing of gene-product abundance.

In recent years, the advent of DNA microarray technology enabled researchers to facilitate new experimental methods for understanding gene expression and regulation. This technology which probes the expression of hundreds to tens of thousands of gene simultaneously via a nucleic acid hybridization approach, is capable of showing the increment, decrement or inert expression of every gene in the test condition relative to the control condition (also known as gene expression profiling), thus providing a holistic viewpoint of gene expression to the researchers instead of only a few genes as in the classical experiments [1]. Over the years, numerous organisms and mammalian cells have been profiled, including *S. cerevisae* [2], human cancerous tissue [3], and *E. coli* [4]. The profiling data contain the answers to various problems such as the set of behaviours exhibited by the system under different conditions; anomalies of the system if certain parts cease to function; and the robustness of the system under extreme conditions [5]. However, the technology breakthrough in experimental methods for large-scale studies of gene regulation also implies that researchers must deal with the massive amount of expression profiling data generated by the microarray experiments. Nevertheless, motivated by the desire of researchers to understand the complex phenomena of gene regulation, gene expression profiling data have obtained significant importance in the inferring of GRNs to describe the phenotypic behaviours of a specific system. For instance, Segal *et al.* [6] constructed condition specified GRNs by utilising a probabilistic model on a *S. cerevisiae* gene expression dataset with a subset of 2355 genes. The inferred GRNs were proved to be fundamental in predicting the functions of several previously unannotated proteins.

The traditional way of inferring GRNs from gene expression profiling data involves the generation of an initial condition-specific model, that is, a hypothetical model which simulates the system's set of behaviours under experimental conditions. The subsequent steps would be focused on disproving the hypothesis by comparing the prediction of the model based on new conditions against the observed gene expression data to give a glimpse of the hypothetical model's competency. The model must be revised if the predicted system behaviours do not match with the observed data, and this set of routine iterates until a model which accurately describe the system's behaviours is attained [7]. It is obvious that the traditional trial and error method for inferring GRNs is not feasible due to its time-consuming nature of repeating the routine to achieve competency of the model. As a result, researchers have started to rely on computational methods to automate the inference procedure.

From a computational viewpoint, a GRN is a directed graph represented by nodes (genes) and edges (interactions) to describe the causal relationship between gene activities. The simplest interactions include activation (up-regulation), inhibition (down-regulation) or constitutive expression. The absence of an edge between two nodes suggests that there is no relationship between them. The general idea behind inferring GRNs using computational methods is the reverse engineering paradigm, or more widely known as inferring [8]. Recent researches have also showed that the integration of available domain knowledge (such as functional and structural information) to the method is capable of deriving a realistic model by narrowing down the search space, thus shortening the time and effort spent on validation and verification [7]. Over the years, various computational methods have been developed to infer GRNs from gene expression data. In particular, Bayesian network (BN), which models conditional dependencies of a set of variables via probabilistic measure, was widely utilized by researchers in inferring GRNs from gene expression data. BN's effectiveness in inferring GRNs is mainly due to its ability to work on locally interacting components with a relatively small number of variables; able to assimilate other mathematical models to avoid the overfitting of data; and allows the combination of prior knowledge the strengthen the causal relationship. Despite the advantages stated above, BN has two critical limitations in which it does not allow feedback loops and is unable to handle the temporal aspect of time-series microarray data.

In view of the fact that feedback loops represent the importance of homeostasis in living organisms, researchers have developed the dynamic Bayesian network (DBN) as a promising alternative. Since the pioneering work of Murphy and Mian [9], DBN has attracted particular attention from numerous researchers. Perrin *et al.* [10] used an extended expectation-maximisation (EM) algorithm as a penalised likelihood maximisation method to estimate the parameters of the model. Alternatively, Yu *et al*. [11] proposed an influence score metric for DBN to identify the nature and estimate the relative magnitude of the interactions between the genes. Zhang and Moret [12] applied DBN as one of the two base inference methods in part of their refinement algorithm for inferring transcriptional regulatory networks. Nevertheless, conventional DBN typically assumes all genes as potential regulators against target genes, and consequently causes the large search space and the excessive computational cost which inhibit the efficiency of DBN [13]. In addition, the missing values commonly found in expression data may influence up to 90% of the genes [14], thus affecting the inference results. To tackle the two problems, we proposed a model of DBN with missing values imputation and potential regulators selection (ISDBN) which would work out against the missing values problem and reduce the search space by selecting potential regulators based on gene expression changes. The details of our model are discussed in the following section.

2. Methods

In this section, we discuss the particulars of the proposed DBN-based model (ISDBN) for inferring GRNs from gene expression data. ISDBN primarily consists of three main steps: missing values imputation, potential regulators selection and DBN inference. ISDBN differs from conventional DBN whereby the missing values are ignored and all genes are considered as potential regulators against a target gene. Figure 1 illustrates the schematic overview of ISDBN.

2.1. Missing Values Imputation

Missing values in gene expression data occur for numerous reasons. For one, the spots on the slides are miniscule and they are packed very tightly. A deficiency, a smudge, or even a speck of dust will corrupt the signals at a number of spots. After the array are scanned through and digitalised, the questionable spots are manually labelled as missing. Also, it may occur due to various technical reasons, such as bleed-over from neighbouring spots, hybridisation failures or background noise in the scanned image.

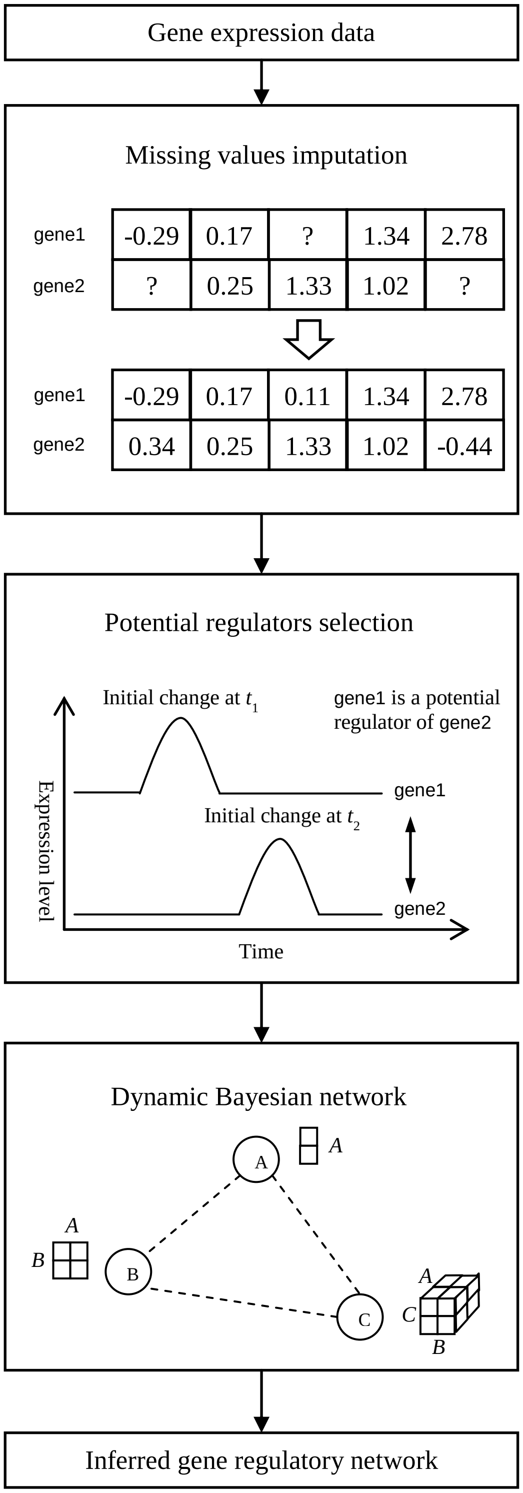
Gene expression data
Missing values imputation
gene1 -0.29 0.17 ? 1.34 2.78
gene2 ? 0.25 1.33 1.02 ?
gene1 -0.29 0.17 0.11 1.34 2.78
gene2 0.34 0.25 1.33 1.02 -0.44
Potential regulators selection
Initial change at t_1
gene1 is a potential regulator of gene2
gene1
Initial change at t_2
gene2
Expression level
Time
Dynamic Bayesian network
A
A
A
B
B
A
C
C
B
Inferred gene regulatory network

Figure 1. Schematic overview of ISDBN

Unfortunately, many downstream gene expression analysis methods lose effectiveness even with a few missing values. Traditional methods of treating missing values include reiterating the microarray experiment which is not economical feasible, or simply substitute the missing values by zero or row average. A better solution is to use imputation algorithms to estimate the missing values by utilising the observed data structure and expression pattern. Widely utilised imputation methods include KNNimpute [15], BPCA [16] and LLSimpute [17]. In particular, LLSimpute exploits local similarity structures by treating target gene with missing values as a linear combination of similar genes based on a similarity measure. The work of Kim *et al.* [17] showed that LLSimpute registered the overall lowest normalized root mean squared error (NRMSE) for all five experimental datasets (Including the *S. cerevisiae* cell cycle dataset used in this study) against other methods. Based on this information, LLSimpute was implemented as the missing values imputation method in our proposed model. In essence, LLSimpute consists of two main steps. The first step is to select k genes by the L_2-norm, where k is defined as a positive integer that determines the number of coherent genes to the target gene. For example, to impute a missing value g located at $\boldsymbol{x}_{11}$ in a $m\times n$ matrix $\boldsymbol{X}$, the k-nearest neighbour gene vectors for $\boldsymbol{x}_1$,

$$\boldsymbol{v}_{s_i}^{\mathrm{T}} \in \boldsymbol{X}^{1\times n} \quad 1 \leq i \leq k \tag{1}$$

are first computed, whereby the gene expression data is summarised as a $m\times n$ matrix $\boldsymbol{X}$ (m is the number of genes, n is the number of observations), and $\boldsymbol{x}_1$ represents the row of the first gene with n observations. s_i is a list of k-nearest neighbour genes vectors with their respective observations, which in turn corresponds to the i-th row of the transpose vector $\boldsymbol{v}^{\mathrm{T}}$. The second step involves regression and estimation of the missing values. A matrix, $\boldsymbol{A} \in \boldsymbol{X}^{k\times(n-1)}$ whereby the k rows of the matrix contains vector $\boldsymbol{v}$, and two vectors, $\boldsymbol{b} \in \boldsymbol{X}^{k\times 1}$ and $\boldsymbol{w} \in \boldsymbol{X}^{(n-1)\times 1}$, are subsequently formed. The vector $\boldsymbol{b}$ contains the first element of k vectors $\boldsymbol{v}^{\mathrm{T}}$, while vector $\boldsymbol{w}$ contains n – 1 elements of vector $\boldsymbol{x}_1$. A k-dimensional coefficient vector $\boldsymbol{y}$ is then computed such that the least square problem

$$|\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{w}|^2 = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{w})^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{w}) \tag{2}$$

is minimised as

$$\min_{\boldsymbol{y}} |\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{w}|^2 \tag{3}$$

Let $\boldsymbol{y}^*$ to denote the vector whereby the square is minimised such that

$$\boldsymbol{w} \simeq \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}^* = y_1^*\boldsymbol{a}_1 + y_2^*\boldsymbol{a}_2 + \cdots + y_k^*\boldsymbol{a}_k \tag{4}$$

where $\boldsymbol{a}_i \in \boldsymbol{A}^{k\times 1}$, and therefore, the missing value g can be estimated as a linear combination of coherent genes such that

$$g = \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{b}^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}})'\boldsymbol{w} \tag{5}$$

where ($\boldsymbol{A}^{\mathrm{T}}$) **'** exists as the pseudoinverse of $\boldsymbol{A}^{\mathrm{T}}$.

2.2. Potential Regulators Selection

Yu *et al.* [18] showed that in most cases, transcriptional factors (TFs) experience changes in expression level prior to or concurrently with their target genes. With this in mind, it is possible to devise an algorithm to reduce the search space by limiting the potential regulators of each target genes. First, we classified the gene expression values into three states: up-, down- and normal regulation. The three states indicate whether the expression value is greater than, lower than or similar to the threshold. This threshold can be either determined experimentally or fixed as the average expression level of the genes across experiments. In this study, we determined the threshold for up-regulation and down-regulation based on the baseline cut-off of the gene expression values. As such, we decided to use ≥1.2 (up-regulation) and ≤0.7 (down-regulation) for the *S. cerevisiae* dataset, and ≥1.4 (up-regulation) and ≤0.7 (down-regulation) for the *E. coli* dataset. Next, the time points of initial up-regulation and down-regulation of each gene were determined, and genes with prior or concurrent expression changes were selected as the potential regulators for those genes with later expression changes. As genes with late expression changes might comprise a large number of potential regulators, we only allowed five time points as the maximum time gap for prior expression changes to avoid selecting potential regulators for a target gene across the whole gene expression dataset. For example, Figure 2 shows the expression profiles of CLN1, CLN2 and GLK1 from the *S. cerevisiae* dataset. We found that the initial up-regulation of CLN1 and GLK1 occurred at the 100th minute and CLN2 at the 110th minute. As CLN1 and GLK1 experienced up-regulation before CLN2, both were included in the subset of potential regulators for CLN2. The same potential regulators selection was applied to other up- and down-regulated genes.

2.3. Dynamic Bayesian Network

The network inference step is done by applying DBN, which is actually an extension of BN to describe the stochastic evolution of a network against time. This is mainly because BN is limited to steady-state data (static data), and DBN readily handles the temporal aspect to identify the causal relationships among variables in time-series data. It also enables the modelling of cyclic structure while inheriting the advantages of BN. Basically, in modelling time-series data, values of a set of variables are observed at different points in time. The general idea that time does not flow backward gives the assumption of an event can cause another event in the future but not vice-versa. As such, the design of DBNs on time-series data is unidirectional whereby the network should flow forward in time. Assuming each time point as single variable $\boldsymbol{Y}_i$, the simplest causal model for a sequence of data $\{\boldsymbol{Y}_1,\ldots,\boldsymbol{Y}_t\}$ would be a *first-order Markov chain*, in which the state of the next variable is dependent on the previous variable only. The *Markov chain* does not represent the dependencies between variables over more than one time step directly. A simple way to extend this model is to assume that the observable variables are dependent on their respective hidden discrete variables known as states. The sequence of hidden states can be classified as a *hidden Markov model* (HMM), which is regarded as one of the simplest form of DBN. DBN consists of two steps: parameter learning followed by structure learning. In the first step, we calculate the joint probability distribution (JPD) by applying the chain rule of probabilities and conditional independencies based on Bayes theorem. Assuming we have a microarray dataset which contains m genes and n observations. The dataset could be summarised as a $m \times n$ matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$ whereby each row, vector $\boldsymbol{x}_m = (\boldsymbol{x}_{m1},\ldots, \boldsymbol{x}_{mn})$ corresponds to a gene expression vector measured at time *t*. First, time dependency is assumed in DBN modelling. The relationship is depicted as a directed acyclic graph (*first-order Markov chain*) whereby only forward edges are allowed. The JPD of the model has the general form of:

$$P(\boldsymbol{x}_{11}, \dots, \boldsymbol{x}_{mn}) = P(\boldsymbol{x}_1)P(\boldsymbol{x}_2|\boldsymbol{x}_1) \dots P(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}) \quad (6)$$

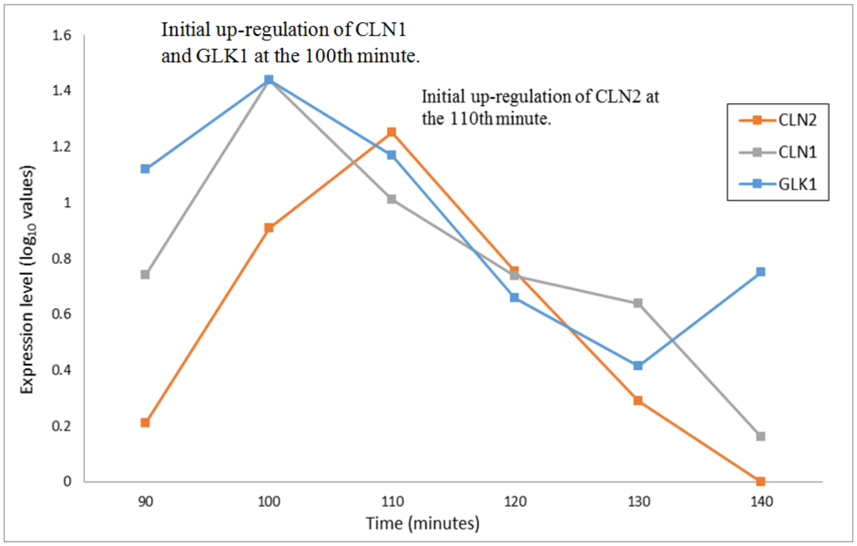

Figure 2. Expression profiles of CLN1, CLN2 and GLK1 corresponding to the *S. cerevisiae* dataset. CLN1 and GLK1 exhibit up-regulated expression at the 100th minute and CLN2 at the 110th minute. Since CLN1 and GLK1 have expression changes prior to CLN2, both were selected as the potential regulators of CLN2

Next, the gene regulations are modelled according to the construction of conditional probability, $P(\boldsymbol{x}_i \mid \boldsymbol{x}_{i-1})$ for $i = 2, \dots, n$. Suppose that the network structure is stable through all time points, the conditional probability could be decomposed into the product of conditional probability of each gene given its parent genes $\boldsymbol{p}$:

$$P(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}) = P(\boldsymbol{x}_{i1}|\boldsymbol{p}_{i-1,1}), \dots, P(\boldsymbol{x}_{in}|\boldsymbol{p}_{i-1,n}) \quad (7)$$

Based on the threshold defined earlier, we discretised the expression values of the results obtained from the previous step into three categories: -1, 0 and 1, which correspond to down-, normal and up-regulation respectively. The conditional probabilities of each subset of potential regulators against their target genes were then computed in a data matrix. The second step of DBN inference is to learn and search for the optimal network structure based on the parameters of the previous step. This is done by using a polynomial time-based search strategy which utilises a scoring function based on the Bayesian Dirichlet equivalence (BDe). The minimal description length (MDL) is another generally used scoring function, and despite that MDL has a faster computation time, BDe are more preferred for its accuracy in statistical interpretation [19]. Hence, for each target gene, the subset of potential regulators that has the minimal network score was selected as the final set of regulators. The final results were then imported into GraphViz (*http://www.graphviz.org*) for network visualization and analysis.

3. Results and Discussion

3.1. Experimental Data and Setup

The experimental data used in this study includes the *S. cerevisiae* cell cycle time-series gene expression data [2] and the *E. coli* SOS response pathway gene expression data [20]. The *S. cerevisiae* cell cycle progression comprises of four phases (G1, S, G2 and M). At G1 phase, the cyclin-dependent kinase, CDC28 associates with cyclin CLN3 and when it accumulates more than a certain threshold, SWI4, SWI6, MBP1 are activated, subsequently promoting the transcription of CLN1 and CLN2. This induces DNA replication and activates CLB1 and CLB2. The association of CLB1 and CLB2 to CDC28 promotes entry into mitosis. This dataset contains a total of 6178 genes observed at two short time series (CLN3, CLB2; both 2 time points) and four medium time series (alpha, CDC15, CDC28 and elu; 18, 24, 17 and 14 time points), and contains 5.912% missing values (28127 out of 475706 observations). We focused on the sub-network around CDC28 which contains around 20 genes. The *E. coli* SOS response pathway is an error-prone repair system which responses to damaged DNA by arresting cell cycle and inducing DNA repair. Under normal circumstances, the repressor protein, lexA negatively regulates the SOS genes by binding to the promoter region of these genes. DNA damage is signified by the blockage of DNA polymerase which would result in the accumulation of single-stranded DNA (ssDNA). The recA protein, which acts as a sensor of DNA damage, is activated by binding to these ssDNA. The activated recA then facilitates the self-cleavage of lexA repressor.

The drop in lexA level in turn causes the SOS genes to be de-repressed. This continues until the damage is repaired, whereby the level of activated recA drops, lexA accumulates and represses the SOS genes again. This dataset contains 8 genes observed at evenly spaced 50 instants with 6 minutes intervals, and contains 11.5% missing values (184 out of 1600 observations).

We applied our model under the framework of BNFinder [19], whereas the missing values imputation and the potential regulators selection were both implemented in MATLAB. To evaluate the performance of ISDBN, we compared the accuracy and computation time of our proposed model against conventional DBN (typified as BNFinder). We first compared the inferred results of both models to the established *S. cerevisiae* cell cycle pathway at KEGG (*http://www.kegg.jp*) and the well-known *E. coli* SOS response pathway [21], and followed by comparing the computation time of both models on a 3.2GHz Intel Core i3 computer with 2GB main memory. The results of Experiment 1 and Experiment 2 are summarised in Table 1 and Table 2 respectively. In both tables, the first row represents the network inferred by ISDBN and the second row represents the network predicted by using BNFinder (Listed as DBN). An edge indicates a relationship between the two connected genes. 'Correctly inferred relationships' denotes the number of relationships found in the established networks and also in the inferred results, 'sensitivity' is the percentage of correctly inferred relationships out of all inferred relationships, and 'specificity' relates to the percentage of correct inference that no relationship exists between two genes.

3.2. Experiment 1

In this experiment, ISDBN managed to correctly infer 30 relationships (Figure 3) out of the established 35 relationships. On the other hand, conventional DBN correctly inferred 27 relationships – it missed out YHP1-MCM1, SWI4-CLN1 and CDC28-WHI5. YHP1 is one of the two transcriptional repressors that bind to MCM1 in the early cell cycle regulation process; SWI4 is a transcriptional activator that regulates the cyclin CLN1 during DNA synthesis and repair; and CDC28 releases the transcriptional repressor WHI5 during early cell cycle phases. A closer look at the original expression profiles of the six genes revealed numerous missing values scattered across each expression profiles. Obviously replacing the missing values as zero or row averages has weakened the statistical relationships between the three pairs of genes, thus causing conventional DBN to erroneously determine that the three relationships were non-existent. However, in ISDBN, the missing values were imputed based on a linear combination of similar genes, and the three relationships were correctly identified. Both models were able to capture the cyclic nature of the cell cycle pathway, for instance, the partial pathway of CDC28-SWI4/6-YOX1-MCM1-CLN3-CDC28 which represents transcription regulation during G1 phase of the cell cycle. As a measure of performance, ISDBN reported 85.71% sensitivity and 94.91% specificity compared to conventional DBN’s 77.14% sensitivity and 93.22% specificity.

Table 1. The Results of Experiment 1

Inference model	Correctly identified relationships	Sensitivity	Specificity	Computation time (HH:MM:SS)
ISDBN	30	85.71%	94.91%	00:25:09
DBN	27	77.14%	93.22%	01:08:23

Although an edge denotes a relationship between two genes, there are 4 possible states for each relationship: correct direction and regulation type, correct direction but incorrect regulation type, incorrect direction but correct regulation type, and incorrect direction and regulation type. By selecting potential regulators to only those which exhibit earlier or concurrent expression changes, ISDBN was able to correctly predict most of the relationships’ direction and regulation type (three wrong regulation assignments and two misdirected edges). In contrast, conventional DBN has seven wrong regulation assignments and nine misdirected edges. Also, since we limited the subset of potential regulators, the search space is relatively small and therefore ISDBN registered a computation time of 25 minutes and 9 seconds against conventional DBN’s 1 hour 8 minutes and 23 seconds. We expect that the difference of computation time between the two models would be more significant on larger dataset.

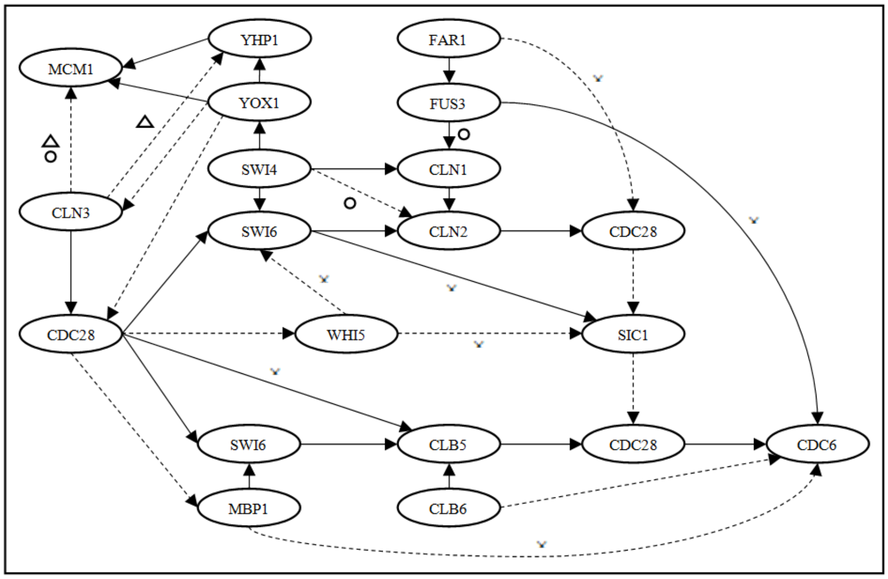

Figure 3. Predicted cell cycle sub-network for *S. cerevisiae* dataset using ISDBN. Dash edges (---) denote down-regulations and straight-lined edges (—) denote up-regulations. A cross represents an incorrect inference; a triangle represents a misdirected relationship; a circle represents an incorrect regulation type; an edge without any attachment is a correct inference

3.3. Experiment 2

In the second experiment, ISDBN correctly identified nine relationships (lexA–recA, lexA–polB, lexA–umuD, lexA–uvrY, lexA–uvrA, lexA-uvrD, lexA–ruvA, lexA–lexA, recA–recA) (Figure 4) out of the established ten relationships, whereby conventional DBN correctly inferred eight relationships. As the dataset is relatively small compared to the previous experiment, both models have similar capabilities to infer relationships from this dataset. Although ISDBN reported 90% of sensitivity against conventional DBN's 80%, the relatively significant difference in percentage is due to the fact that the total of established relationships is only ten. On the other hand, ISDBN has a lower specificity (66.67%) compared to conventional DBN (72.22%). Both models again were able to identify two self-cyclic regulatory relationships: recA, which corresponds to its ability to self-activate when DNA damage is detected, and lexA, which indicates its self-cleavage mechanism when the level of activated recA is raised. ISDBN reported two incorrect regulation assignments and one misdirected edges, while conventional DBN has three wrong regulation assignments and one misdirected edges. Regarding the computation time, ISDBN demonstrated a computation time of 8 minutes and 43 seconds while conventional DBN recorded 15 minutes and 17 seconds. This is more or less due to the fact that the dataset used in this experiment was relatively small, therefore the computation time for both models were also relatively short.

Table 2. The Results of Experiment 2

Inference model	Correctly identified relationships	Sensitivity	Specificity	Computation time (HH:MM:SS)
ISDBN	9	90%	66.67%	00:08:43
DBN	8	80%	72.22%	00:15:17

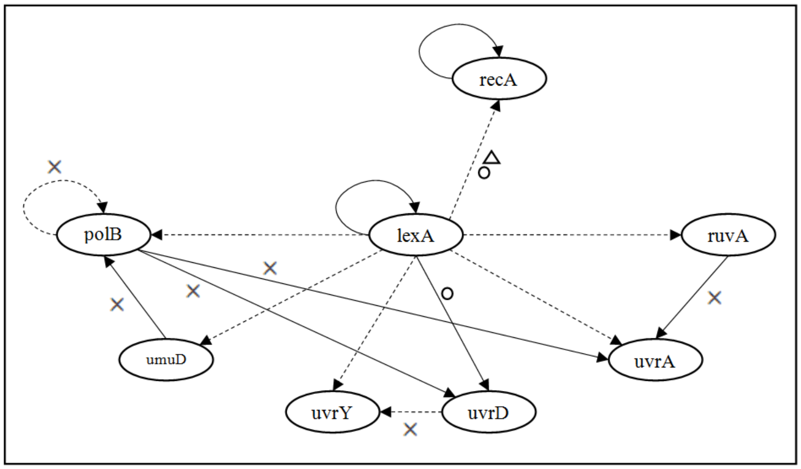

Figure 4. Inferred SOS response pathway for *E. coli* dataset using ISDBN. Dash edges (---) denote down-regulations and straight-lined edges (—) denote up-regulations. A cross represents an incorrect inference; a triangle represents a misdirected relationship; a circle represents an incorrect regulation type; an edge without any attachment is a correct inference

4. Conclusions

In this study, we addressed two problems found in conventional DBN in inferring GRNs from gene expression data: the missing values which would influence the inference results, and the excessive computation time due to the large search space since conventional DBN assumes all genes as potential regulators against a target gene. To this end, we proposed a DBN-based model with missing values imputation and potential regulators selection (ISDBN) to tackle both problems. First, instead of replacing missing values with zeros or row averages, we treated missing values by exploiting local similarity structures as a linear combination of similar genes. In this way, we were able to capture most of the statistical correlation between genes that has missing values spread across their expression profiles. Second, by exploiting the fact that most transcriptional factors generally exhibit prior or concurrent expression changes, we were able to reduce the search space by limiting the number of potential regulators for each target gene, and in turn contributed to the decreased computation time. Based on the datasets of *S. cerevisiae* cell cycle pathway and *E. coli* SOS response pathway, ISDBN showed promising results in terms of computation time and accuracy when compared to conventional DBN.

In addition, we are also interested in taking account of the transcriptional time lag which is commonly ignored in inferring GRNs from gene expression data. The lack of an algorithm to handle transcriptional time lag is one of the main factors that contributed to the relatively low accuracy of inferring GRNs using DBN. Also, it should be noted that presently, our proposed model could only handle inter-time slice edges. To learn DBN with both inter- and intra-time slice edges remains an interesting point of research. It is suggested to learn intra-time slice edges separately before combining with the inter-time slice edges and post-processing as an alternative to describe gene-gene interactions [22]. Lastly, in spite of the broad practice of using DBN to infer GRNs from gene expression data, it is in no way to completely substitute gene intervention experiments. The resultant networks should be treated as a guideline or framework of the studied biological pathways for future hypotheses testing and intervention experiments.

Acknowledgements

We would like to thank Malaysian Ministry of Higher Education for supporting this research by granting an Exploratory Research Grant Scheme (Grant number: R.J130000.7807.4L096) and a Fundamental Research Grant Scheme (Grant number: R.J130000.7807. 4F190). This research is also funded by an e-science research grant (Grant number: 06-01-06-SF1029) from Malaysian Ministry of Science, Technology and Innovation. We would also like to thank Institutional Scholarship MyPhD and Ministry of Higher Education of Malaysia which has provided finance for this study.

References

[1] R. Jornsten, H. Y. Wang, W. J. Welsh and M. Ouyang, Bioinformatics, vol. 21, no. 22, **(2005)**, pp. 4155-61.

[2] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein and B. Futcher, Mol Biol Cell, vol. 9, **(1998)**, pp. 3273-97.

[3] S. Muro, I. Takemasa, S. Oba, R. Matoba, N. Ueno, C. Maruyama, R. Yamashita, M. Sekimoto, H. Yamamoto, S. Nakamori, M. Monden, S. Ishii and K. Kato, Genome Biol., vol. 4, **(2003)**, pp. R21.

[4] H. Salgado, S. Gama-Castro, M. Peralta-Gil, E. Diaz-Peredo, F. Sanchez-Solano, A. Santos-Zavaleta, I. Martinez-Flores, V. Jimenez-Jacinto, C. Bonavides-Martinez, J. Segura-Salazar, A. Martinez-Antonio and J. Collado-Vides, Nucleic Acids Res., vol. 34, **(2005)**, pp. 394–7.

[5] G. Karlebach and R. Shamir, Nat Rev Mol Cell Bio., vol. 9, no. 10, **(2008)**, pp. 770-80.

[6] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller and N. Friedman, Nat Genet., vol. 34, no. 2, **(2003)**, pp. 166-76.

[7] W. P. Lee and W. S. Tzou, Brief Bioinform., vol. 10, no. 4, **(2009)**, pp. 408-23.

[8] M. E. Csete and J. C. Doyle, Science, vol. 295, **(2003)**, pp. 1664-9.

[9] K. Murphy and S. Mian, "Modelling gene expression data using dynamic Bayesian networks", Computer Science Division, University of California: Berkeley, **(1999)**.

[10] B. E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet and F. d'Alche-Buc, Bioinformatics, vol. 19, Suppl. 2, **(2003)**, pp. ii138-48.

[11] J. Yu, V. A. Smith, P. P. Wang, A. J. Hartemink and E. D. Jarvis, Bioinformatics, vol. 20, no. 18, **(2004)**, pp. 3594-603.

[12] X. Zhang and B. Moret, Algorithm Mol Biol., vol. 5, **(2010)**, pp. 1.

[13] Y. Jia and J. Huan, BMC Bioinformatics, vol. 11, Suppl. 6, **(2010)**, pp. S27.

[14] M. Ouyang, W. J. Welsh and P. Geogopoulos, Bioinformatics, vol. 20, no. 6, **(2004)**, pp. 917-23.

[15] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R. B. Altman, Bioinformatics, vol. 17, **(2001)**, pp. 520-5.

[16] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara and S. Ishii, Bioinformatics, vol. 19, **(2003)**, pp. 2088-96.

[17] H. Kim, G. Golub and H. Park, Bioinformatics, vol. 21, no. 2, **(2005)**, pp. 187-98.

[18] H. Yu, N. M. Luscombe, J. Qian and M. Gerstein, Trends Genet., vol. 19, **(2003)**, pp. 422-7.

[19] B. Wilczynski and N. Dojer, Bioinformatics, vol. 25, no. 2, **(2009)**, pp. 286-7.

[20] M. Ronen, R. Rosenberg, B. I. Shraiman and U. Alon, Proc Natl Acad Sci., USA, vol. 99, **(2002)**, pp. 10555-60.

[21] M. Radman, Basic Life Sci., vol. 5A, **(1975)**, pp. 255-367.

[22] N. X. Vinh, M. Chetty, R. Coppel and P. P. Wangikar, Bioinformatics, vol. 27, no. 19, **(2011)**, pp. 2765-6.