

# Constraint Minimization for Efficient Modeling of Gene Regulatory Network

Ramesh Ram<sup>1</sup>, Madhu Chetty<sup>1</sup>, and Dieter Bulach<sup>2</sup>

<sup>1</sup> Gippsland School of IT, Monash University,  
Churchill, Victoria 3842, Australia

{Ramesh.ram, Madhu.chetty}@infotech.monash.edu.au

<sup>2</sup> CSIRO Livestock Industries, Australian Animal Health Laboratory,  
5 Portarlington Rd, Geelong VIC 3220, Australia

Dieter.Bulach@csiro.au

**Abstract.** Due to various complexities, as well as noise and high dimensionality, reconstructing a gene regulatory network (GRN) from a high-throughput microarray data becomes computationally intensive. In our earlier work on causal model approach for GRN reconstruction, we had shown the superiority of Markov blanket (MB) algorithm compared to the algorithm using the existing Y and V causal models. In this paper, we show the MB algorithm can be enhanced further by application of the proposed constraint logic minimization (CLM) technique. We describe a framework for minimizing the constraint logic involved (condition independent tests) by exploiting the Markov blanket learning methods developed for a Bayesian network (BN). The constraint relationships are represented in the form of logic using K-map and with the aid of CLM increase the algorithm efficiency and the accuracy. We show improved results by investigations on both the synthetic as well as the real life yeast cell cycle data sets.

**Keywords:** Causal model, Markov blanket, Constraint minimization, Gene regulatory network.

## 1 Introduction

Gene regulatory networks (GRNs) represent gene-gene regulatory interactions in a genome to display relationships between various gene activities. Amongst different approaches available, these networks can also be modeled accurately by Markov blanket (MB) graph a powerful versatile method for modeling any dynamic physical system. This technique was first proposed by Sprites et al [1] who stated that MB can adequately represent all connections and interactions in a network. Since then, the work on MB has been rapidly expanding with a focus on the study of causality which plays an important role in modeling, analysis and design of GRNs. Learning any Markov blanket Bayesian network structure and inferring gene networks [2, 3] involves application of constraints. Although these constraints are typically conditional-independence statements, the non-independence based constraints may also be entailed by the structure where latent variables exist [3]. The conditional independence tests

used in practice are statistical tests such as partial correlation, mutual information, and conditional probabilities etc. that indicate a causal influence. In order to use the conditional independence tests to reconstruct the structure, several assumptions have to be made, e.g. causal sufficiency, causal Markov and faithfulness [2]. With these assumptions, we can ascertain the existence of an edge, its direction and whether it is positive or negative. The Sprites-Glymour-Scheines (SGS) algorithm [1], used for obtaining a causal DAG from a dataset, assumes that graphs are acyclical. It is formulated using the concept of d-separation [2] in which all possible combinations are tried before determining the existence of an edge between every pair of variables in the dataset. However, the SGS algorithm fails to always assign directions to each of the edges. This limitation of SGS algorithm is overcome by the inductive causation (IC) algorithm [3], which is capable of assigning directions. Some algorithms do not make use of independence tests but take into account d-separation in order to discover structure from data [2]. For example, Cheng *et al* [4], applied mutual information instead of conditional independence tests. All these algorithms are referred as constraint-based algorithms [1, 4]. Constraint-based algorithms have certain limitations such as poor robustness or computation time which increases exponentially with the number of constraints. These limitations make these approaches impractical for large datasets of tens or even hundreds of variables.

In our recently proposed causal model [5, 6] approach for constructing GRN, the network was inferred by applying the following three sequential steps to identify the sub-structures of a larger network: i) Perform conditional independence (CI) tests for each node's Markov blanket ii) Assign direction to the edges and iii) Assign sign of regulation to the edges. However, due to the huge size of network search space and the limited amount of microarray data, it was impractical to test each and every constraint. Moreover, with the increase in the condition set needed for causal discovery, more and more CI tests had to be performed, resulting eventually in lower accuracy.

By simplifying the complex logic involved with the constraints in the Markov blanket algorithm, the computational efficiency of the MB algorithm [6] can be enhanced thereby resulting in improved accuracy for network reconstruction. In this paper, we propose a technique for minimizing the constraints and hence the condition set needed for testing the structure with respect to data. The statistical tests following the logic is translated into a Boolean function after which a logic gate minimization technique such as K-map [7] is applied and the minimized logic is translated back to the constraints and used on the data. We have achieved this by a novel independence-based algorithm which we refer here as the Markov blanket-Constraint Logic Minimization (MB-CLM) algorithm. The MB-CLM algorithm heuristically uses Markov Blanket neighborhood of a node and makes model evaluation simple. In order to evaluate and validate a Markov Blanket, there is invariably a need for checking a set of conditions. However, from the available set of alternatives, it is possible to have a potentially smaller set of conditions that can establish the desired conclusion for the given network but with a faster computation speed and increased reliability. This is because a conditioning set  $S$  splits the data set into  $2^S$  partitions. With a smaller conditioning set, the data set is split into larger partitions thereby making dependence tests more reliable. This smaller or minimal set will fulfill the necessary and sufficient conditions required for GRN reconstruction.

The rest of the paper is organized as follows. Section 2 provides a background on causal model and also explains the fundamentals of the technique of logic minimization. Section 3 describes the methodology. Section 4 gives the results of the experiments from the synthetic and real datasets. Finally, Section 5 provides conclusion and the future work.

## 2 Background

In this section, we briefly present the causal model approach for GRN reconstruction and also the notion of Markov blanket both of which are important for understanding the MB-CLM algorithm.

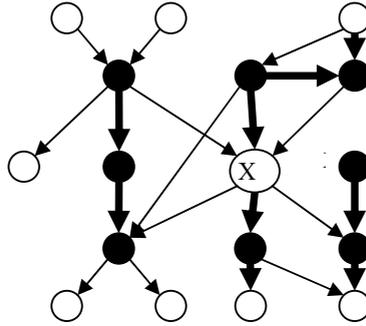
### 2.1 Causal Model Approach

A causal GRN structure is represented by a directed graph whose nodes represent the genes and the directed edges between nodes indicate the causal relationships. Pearl *et al* and Sprites *et al* [1, 2] proposed algorithms to infer a causal structure from experimental data by using partial correlations, if the underlying causal structure is a directed acyclic graph (DAG). Recently, we reported a technique for causal modeling by means of a novel scoring function [5]. In this work, the central step of determining the fitness of the data given a whole network, is decomposed into a task of determining a set of scores of the local models that includes: i) Fitness of structure ii) Direction of causality and iii) Sign (positive/ negative) of regulation. The task of network reconstruction is cast into a search for candidate gene networks with high scores. This highly computationally expensive search is usually carried out stochastically by using, for example a genetic algorithm (GA). The search creates and evolves different networks and eventually obtains a network that best fits the microarray data. Due to the stochastic nature of the GA, the GA is repeated several times and the resulting network structures are combined in a predefined manner to reconstruct the final gene network. While evaluating the fitness, the putative network is actually decomposed into MB and conditional independence tests are applied in order to detect whether or not connections are direct or indirect. The direction and sign of regulation are recovered by estimating the time delay and correlation between expression profiles of pairs of genes. The entire methodology has already been validated by using a synthetic dataset reported in our earlier work [6] and *Saccharomyces cerevisiae* (yeast) [8] microarray dataset. The results of validation are found to be in agreement with the known biological findings.

### 2.2 Markov Blanket

A Markov blanket [2], central to the concept of causal modeling, includes the node  $X$  under consideration and also its parents and children. It is denoted as  $MB(X)$  and is a minimal set of variables such that every other variable is independent of  $X$  given  $MB(X)$ , i.e.

$$\forall Y \in \{X_1, \dots, X_n\} \setminus \{MB(X), X\}, X \perp\!\!\!\perp Y \mid MB(X)$$



**Fig. 1.** Markov Blanket of X

An example of MB is shown in Fig. 1. Several studies [9-11] have sought to identify the MB of a target node (X in Fig. 1) by filtering nodes using statistical decisions from conditional independence tests. In Fig.1, the shaded nodes (black) are Markov blanket neighbours since there is either an edge or a child in common between target node and shaded node. The non-shaded nodes (white) are independent of the target node, X. A MB DAG can be constructed by combining the MB’s of all the nodes in the dataset.

A MB for a node X in a GRN dataset has two important features. First, all the nodes within a MB have a similar set of dependencies and therefore exhibit a similar behavior. In a similar manner, genes in a cell are also organized into small groups and the sets of genes required for a similar biological function or response are co-regulated by the same inputs in order to coordinate their joint activity. In other words, the MB neighbours (shaded nodes in Fig. 1) of a target node (gene) show the gene expression patterns emerging only due to a disruption of that gene. Second, they can also have a causal interpretation: a directed edge from one gene to another,  $X \rightarrow Y$ , represents the claim that X is a direct cause of Y with respect to other genes in a DAG. Keeping other genes fixed, if X is varied by an intervention (e.g., activation or repression), then both X and Y would co-vary [1, 2]. A MB DAG can thus provide both biological and causal insight into relations between a reduced set of predictor nodes (parents, children, spouses) and the target node.

The technique of logic gate minimisation, well known for electronic circuit minimisation, and on which the proposed constraint logic minimisation algorithm is based, is presented next.

### 3 Logic Gate Minimization Technique

The proposed CLM algorithm is uses the K-map technique applied for logic gate minimisation. To illustrate the minimisation technique, let us consider an arbitrarily chosen four input Boolean network. Let the network be, for example, characterized by the following Boolean function to give an output of 1(i.e. true output):

$$f(a, b, c, d) = \sum m (0, 3, 4, 7, 8, 11, 15) \tag{1}$$

		cd			
		0	0	1	1
ab	0	0	1	1	0
	0	1		1	
	1	1		1	
	1			1	
	1	1		1	

Fig. 2. K-Map for the function given by Eqn. (1)

Here,  $f$  is the boolean function.  $a, b, c$  and  $d$  are the four independent inputs. The numbers on the RHS are minterms (i.e. decimal value equivalent of the 4 bit inputs). For example, the value 3 on RHS, means that the four bit input combination of 0011 (i.e.  $a'b'cd$  or the decimal equivalent value of 3) results in 1.  $m$  indicates that all the values within bracket are minterms.

The above equation indicates that any combination of the inputs with values of either 0, 3, 4, 7, 8, 11 or 15 would result in a true output. Since the default value of  $a$  is considered as 1, it implies that  $a'=0$ . Thus, the above function in Eqn. 1 can be expanded as

$$f = a'b'c'd' + a'b'cd + a'bc'd' + a'bcd + abc'd' + ab'cd + abcd \tag{2}$$

The above equation is known as a Sum of Product (SOP) equation and the products are the minterms mentioned above.

The K-map for the above function is shown in Fig. 2. All rows and columns in the K-map above are unique since only one variable changes its value within its square. The relevant K-map elements are given a value of 1 to include all possible constraints with true outputs. The first row, for example has input  $a = 0$  (i.e.  $a'$ ) and input  $b = 0$  (i.e.  $b'$ ). Similarly column 3, for example has both  $c = 1$  and  $d = 1$ . Thus, an element, for example in row 1, column 3 corresponds to input  $a'b'cd = 1$ . It can be noted that this corresponds to the second term on RHS in Eqn. (2) above. It can be further noted that between two adjacent columns (or rows), only one of the variables changes its value. For example, in Fig. 2, the input  $cd$  given as 00, 01, 11, 10 in the columns ensures that there is only one input change.

Now let us consider grouping the common terms and minimisation of the function using the K-map shown in Fig. 2. By grouping:

- i) Four 1 in column 3 (all rows), we get the common term  $cd$
- ii) Two 1 in column 1 (row 1 and row 2), we get common term  $a'c'd'$

Considering the above groupings, we can rearrange the RHS terms from Eqn.(2) appropriately to facilitate logic minimisation. Further, noting that

$$a + a' = 1 \tag{3}$$

We can simplify Eqn. 2 as follows:

$$\begin{aligned}
 f(a,b,c,d) &= (a'b'cd+abcd+a'bcd+ab'cd)+(a'b'c'd'+a'bc'd') +ab'c'd' \\
 &= cd(a'b'+ab+a'b+ab')+ a'c'd' (b+b')+ ab'c'd' \\
 &= cd + a'c'd' + ab'c'd'
 \end{aligned} \tag{4}$$

Since the constraint when applied to the MB scoring for GRN reconstruction will also evaluate to either true or false, the principles of logic gate minimisation presented in this section can easily be extended and applied to GRN modeling. The variables a, b, c, d in Eqn. (4) above will correspond to constraints that can be either CI tests or tests involving delays and directions. This technique is presented next.

## 4 Method

In our GRN reconstruction method reported earlier [6, 12], the network is evaluated at the MB of every node with respect to data resulting in a set of constraints to be satisfied per MB. In general, all these constraints should always be satisfied to validate a true MB with respect to data. Since the dataset under consideration is noisy and high dimensional, it is acceptable if all the constraints are not necessarily satisfied for MB validation. For example, consider a MB having say three constraints. A combination of say two constraints may leave the evaluation of third constraint (*don't care*) unnecessary. However, if the two constraints fail, only then the need to evaluate the third constraint may arise. Since the Markov blanket scoring can be viewed as a logic circuit minimisation, we can get a function similar to Eqn. 1 and the underlying logic constraints can thus be represented using K-map explained in the previous section for optimising the computations. In order to show how this can be achieved, the algorithm for learning MB is presented next.

### 4.1 The Markov Blanket Network Inference Algorithm

A static causal directed acyclic graph (CDAG) model for representing GRN consists of nodes representing genes and arc giving direction and sign of regulation. A matrix element  $E(a,b)$  of the gene expression matrix  $E$  indicates the expression ratio of gene  $a$  at time  $b$ . The overall inference approach (*Learn\_MB* algorithm) is as follows:

- i) *Gene Expression Matrix E*: Obtain  $E$  corresponding to the set of nodes from dataset  $D$  that are affected by node  $X$ . This set involves parents, children and spouse nodes of node  $X$ .
- ii) *Causal relation R*: In the putative MB network  $H(X)$ , the causal relationships are defined as gene  $a$  affects gene  $b$  either directly or indirectly. We thus create  $n$  binary causal relation  $R$  using the causal relationship.
- iii) *Adjacency matrix A*: The adjacency matrix  $A$  is derived directly from the binary relation  $R$ . If there is a relation that gene  $a$  affects gene  $b$ , then the value of element  $(a, b)$  in the adjacency matrix  $A$  is set to 1, i.e.  $A(a, b) = 1$ .
- iv) *Skeleton matrix S*: A skeleton matrix  $S$  includes direct and indirect effects observed in a putative MB. The adjacency matrix  $A$  (of size  $n \times n$  where  $n$  is the number of nodes in the MB) includes direct relationship between genes. The indirect effects are included as follows: The row  $i$  and column  $j$  in adjacency

matrix  $A$  and skeleton matrix  $S$  represent direct and indirect causal relationship between gene  $i$  and gene  $j$  respectively. For example, consider an indirect relationship between gene  $i$  and  $k$  (where  $k$  is any other column corresponding to gene  $k$ ) such that  $A(i, j)$  and  $A(j, k)$  both equal 1. Then  $S(i,k)$  is set to Binary $\{A(i, j) \text{ AND } A(j, k)\}$  (for  $k = 1, \dots, n$ ). In this manner, all indirect effects are captured in the skeleton matrix  $S$  from the adjacency matrix  $A$ .

- v) *Constraints set C*: The direct and indirect effects from adjacency matrix  $A$  and skeleton matrix  $S$  are converted as conditional dependence (CD) and conditional independence (CI) constraints respectively. A conditioning set is needed for each of the constraints which will contain all the nodes in  $H(X)$  minus the variables involved in the constraint. For example, if there is an indirect relation, such that gene  $a$  and  $b$  are conditionally independent then the condition set is given as  $H(X)-\{a,b\}$ . The outcome of the CI and CD constraints is either a 1 (constraint fits the data) or a 0 otherwise. The test is done using statistical methods namely partial correlation. The constraints that are not CI or CD determine the direction and sign of the arcs in the MB and are similar in nature to independence tests. In our case, the direction and sign between gene  $a$  and  $b$  is obtained by the following two equations:

$$f(\text{dir}_H(a,b), \text{dir}_D(a,b)) = 0 \mid 1 \tag{5}$$

$$f(\text{sgn}_H(a,b), \text{sgn}_D(a,b)) = 0 \mid 1 \tag{6}$$

In Eqn. (5), the function  $f$  compares the direction between the putative network  $H$  and the dataset  $D$  while in Eqn. (6),  $f$  compares the sign. Furthermore, there are additional constraints which compare estimated time delay with the actual time delay. All these constraints comprise the constraint set  $C$ .

- vi) *Constraint set reduction*: If a relation exists such that gene  $b$  and  $d$  are conditionally independent (conditioned on gene  $a$ ) and further gene  $c$  and  $d$  are conditionally independent (conditioned on gene  $a$ ), then gene  $b$  and  $c$  are conditionally dependent (conditioned on gene  $a$ ). Such tests are therefore unnecessary to implement and can be eliminated from the constraint set  $C$  resulting in updating the Adjacency matrix  $A$  and the Skeleton matrix  $S$ . Further, the condition set for the CI and CD constraints is also reduced such that the CI/CD test outcome is independent of the removal of a variable from the condition set and is in conformance with d-separation theory [2].
- vii) *Constraints Evaluation*: Next, a table of constraint set  $C$  is created. Here, the combination of constraints that entail the validity of the putative Markov blanket with respect to the dataset or otherwise are computed. A threshold value (explained in Section 4.2 below) is used when constraints are tested with respect to data.
- viii) *Fitness Score*: Comparison based on the value of each element in the skeleton matrix and the evaluation table determines the goodness of fit. This will show if the putative network is consistent with the experiment data  $D$  and the causal relation  $R$ .

## 4.2 MB Threshold Setting

We now discuss the factors involved in determining the threshold value used in step vii) of the MB algorithm presented above. The p-values, commonly used in any statistical analysis, are used for determining the threshold. For large p-values, the Learn\_MB algorithm begins to rapidly increase the number of false positives without any corresponding increase in true positives. An appropriate value for the MB threshold, producing a near optimal result, can be selected a-priori using the Bonferroni-corrected p-value based on the number of potential network interactions. Alternatively, the threshold can be identified by analyzing the distribution of MB scores as a function of the length of the shortest path connecting each gene pair (degree of connectivity). The algorithm depends on the MB being enriched for directly matching interaction among genes and decreases rapidly with its distance from the hub. There is no unique choice for the threshold which can separate directly and indirectly interacting genes, and most methods that attempt to use a single threshold either recover many indirect connections or miss a substantial number of direct ones.

## 4.3 Complexity Analysis and Discussion of the Plain MB Algorithm

The order of complexity for each conditional dependence/independence test taken is  $O(nD)$ , where  $D$  is the dataset of input to the algorithm. The computations are required for constructing the table of constraints and for each combination of the variables (genes) included in the constraint test that exists in the data set. As a worst case scenario, each dependence test uses  $O(D)$  space to store each variable combination of the conditioning constraint set that appears in the data.

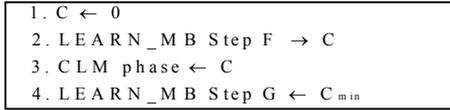
The number of constraints tested is usually reported as a measure of the performance of Bayesian net reconstruction algorithms [1, 4]. To determine the number of tests in this algorithm, we assume that the steps 2 and 3 go through MB variables (parents, children, spouses) in an unspecified but fixed order. Therefore, the order of the entire algorithm is  $O(n)$  in the number of independence tests. The algorithm benefits by further computational optimizations from constraint minimization using the proposed K-map technique.

In the next section, we present the CLM algorithm which when combined with learn\_MB algorithm presented earlier results in an integrated MB-CLM algorithm.

## 4.4 CLM Algorithm

The Constraint minimization approach is given as follows:

1. Obtain the Markov blanket  $H(X)$ . Let the set of constraints be  $C$ .
2. Get the constraint set  $C$  from step v) of Learn\_MB algorithm in section 3.1
3. Assign binary codes for constraints in constraint set  $C$ . Use the constraint evaluation table to generate a truth table and the logic diagram.
4. Perform minimization with the help of K-map.
5. Remove unnecessary constraints before performing constraint evaluation.
6. Execute the minimized logic on the dataset  $D$ .



**Fig. 3.** Learn\_MB and CLM integration step

In Fig. 3,  $C$  represents the set of constraints. Initially, the constraints are obtained from the Learn\_MB algorithm. The above mentioned CLM approach, shown as a CLM phase, takes the constraint set  $C$  as input and returns a minimized set  $C_{min}$  back to the learn\_MB algorithm for evaluation and validation.

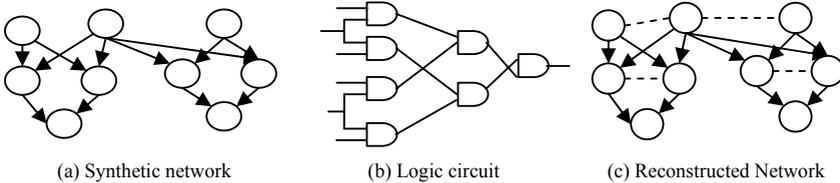
## 5 Experiments and Results

For examining the effect of the minimization technique on the GRN reconstruction, simulations are next conducted using both, the synthetic and the real datasets. The synthetic datasets are realistic and are generated by systematic approach reported earlier [12] for synthetic GRN modeling. The real life data set chosen for investigations is the widely studied yeast cell cycle data set.

### 5.1 Synthetic Datasets

Figure 4 shows an example of reconstruction of an artificially constructed synthetic network using MB-CLM technique. Figure 4a shows the original synthetic network. Amongst various network architectures possible, we chose a network type referred as random network. The generated network is of 3x3 dimensions with an up/down branching factor of 2. The branching factor refers to the number of parents, children and spouses connected to a node. The up branching factor specifies the number of parents of each node directly above it, excluding nodes in the boundary of the network as they are exogenous (without parents). Figure 4b shows the logic circuit corresponding to the constraints involved and Fig. 4c shows the reconstructed network using MB-CLM algorithm.

In our simulations we used plain MB algorithm and MB-CLM algorithm with a MB threshold value of 0.90 in both cases and tested the algorithms using synthetic network 5 x 4 nodes and corresponding synthetic data of upto 100 samples.



**Fig. 4.** Synthetic network and minimized constraint logic

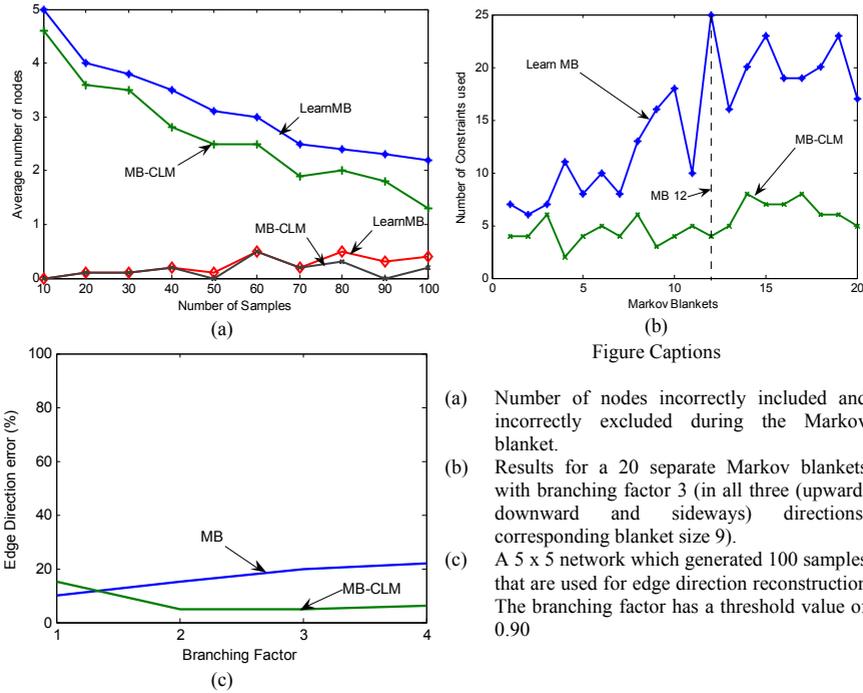


Figure Captions

- (a) Number of nodes incorrectly included and incorrectly excluded during the Markov blanket.
- (b) Results for a 20 separate Markov blankets with branching factor 3 (in all three (upward, downward and sideways) directions, corresponding blanket size 9).
- (c) A 5 x 5 network which generated 100 samples that are used for edge direction reconstruction. The branching factor has a threshold value of 0.90

Fig. 5. Simulation Results

Figure 5a shows a plot of the number of nodes of the MB incorrectly included or excluded for plain Learn\_MB algorithm and MB-CLM algorithm, averaged over all nodes in the domain. It can be observed that due to the constraint minimization, the accuracy of results have increased, as a result the number of nodes incorrectly included is less for the MB-CLM algorithm compared to the Learn\_MB algorithm. Hence, there is better accuracy and reliability with the MB-CLM algorithm. On the other hand, as can be seen from Fig.5a, the use of Learn\_MB algorithm resulted in a slightly higher number of missing nodes. Although the nodes incorrectly included are very low for both Learn\_MB and MB-CLM algorithm, the nodes incorrectly excluded fall more rapidly with increasing sample size in the case of MB-CLM algorithm compared to Learn\_MB algorithm. From Fig. 5b, it can be observed that MB 12 has very high constraints which are minimized by MB-CLM algorithm. The CLM algorithm thus can help with large reduction of constraints in certain circumstances. The effect on percentage Direction Error (DE) by increasing MB (via branching factor increase) is shown in Fig. 5c. DE for the MB and the MB-CLM algorithm remains close for lower branching factors but decreases slightly for MB-CLM algorithm with increase in branching factor. The decrease is due to the large number of parents for each node (i.e. more V structures) which provides greater opportunities to recover the directionality of an edge with increased number of tests.



The *time delay* learning when CLM algorithm was used revealed the activatory influences  $CLN1 \rightarrow CLN2$ ,  $CLB6 \rightarrow CLB5$ ,  $CLN1 \rightarrow CLB6$  and  $CLN3 \rightarrow CLB6$  ( $CLN3$  is also the G1-specific cyclin). These find support in the literature [14-19]. The *time delay* learning was also able to infer the inhibitory influence of  $SIC1$  on the genes  $CLB1$ ,  $CLB2$  and  $SWI5$  [18]. The *plus/minus* learning using the CLM algorithm showed a positive relation between the genes  $SIC1$ ,  $CDC20$  and  $CDC34$ . The gene  $CDC20$  is required for proteolytic degradation of G1 regulators which explains the negative connections discovered by CLM from gene  $CDC20$  to  $SWI6$  and  $MCM1$ , both of which are encoding transcription factors. The gene  $CDC20$  is transcribed in the late  $S/G2$  phase, whereas the genes  $CLN2$  and  $CLB5$  are expressed in G1 phase, supporting the negative connection between  $CDC20$  and these genes.

## 6 Conclusion

In this paper, a Markov blanket (MB) based constraint minimization algorithm (MB-CLM algorithm) for efficiently learning the GRN is presented. The CLM algorithm initially uses the original Learn\_MB algorithm to convert a putative MB structure into a set of constraints which are then tested against the given data. The MB-CLM heuristically minimizes this constraint set using K-map logic minimization technique to improve MB inference resulting in a superior GRN reconstruction. The performance of MB-CLM algorithm is investigated using both the synthetic data and real data (yeast). Experiments with synthetic data show that the number of nodes incorrectly included (or excluded) with only MB algorithm reduces significantly when CLM algorithm is incorporated. Simulations studies with yeast data discovered new regulatory relations when CLM algorithm was used. Using MB-CLM algorithm, both the *time delay learning* algorithm and the *plus/minus learning* algorithm revealed interactions which were not reconstructed with MB algorithm. These newly discovered relations were validated to be correct by the biological knowledge. Thus, in the MB-CLM algorithm improves the overall process of GRN reconstruction.

## References

1. Sprites, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search: Adaptive Computation and Machine Learning, 2nd edn. MIT Press, Cambridge (2000)
2. Pearl, J.: Causality: Models, Reasoning and Inference. Cambridge University Press, Cambridge (2000)
3. Verma, T.S., Pearl, J.: 'A theory of inferred causation', Principles of Knowledge Representation and Reasoning, pp. 441-452 (1991)
4. Cheng, J., Bell, D.A., Liu, W.: An algorithm for Bayesian belief network construction from data. In: Book An algorithm for Bayesian belief network construction from data, pp. 83-90 (1997)
5. Ram, R., Chetty, M., Dix, T.I.: Causal Modeling of Gene Regulatory Network. In: Book Causal Modeling of Gene Regulatory Network, pp. 1-8 (2006)
6. Ram, R., Chetty, M.: Learning Structure of Gene Regulatory Networks. In: Book Learning Structure of Gene Regulatory Networks, pp. 525-531 (2007)
7. Mano., M.M.: Digital Design, 3rd edn. Prentice Hall, Inc., Englewood Cliffs (2002)

8. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell.* 9, 3273–3297 (1998)
9. Aliferis, C.F., Tsamardinos, I.: Algorithms for Large-Scale Local Causal Discovery in the Presence of Small Sample or Large Causal Neighborhoods. In: Book Algorithms for Large-Scale Local Causal Discovery in the Presence of Small Sample or Large Causal Neighborhoods, Vanderbilt University (2002)
10. Aliferis, C.F., Statnikov, I.T.A.: HITON, A Novel Markov Blanket Algorithm for Optimal Variable Selection. In: Book HITON, A Novel Markov Blanket Algorithm for Optimal Variable Selection, Vanderbilt University (2003)
11. Chickering, D.M.: Learning Equivalence Classes of Bayesian-Network Structures. *Journal of Machine Learning Research* 2, 445–498 (2002)
12. Ram, R., Chetty, M.: Framework for path analysis for learning Gene regulatory network. In: Book Framework for path analysis for learning Gene regulatory network, pp. 264–273. Springer, Heidelberg (2007)
13. Chen, K.C., Calzone, L., Csikasz-Nagy, A., Cross, F.R., Novak, B., Tyson, J.J.: Integrative analysis of cell cycle control in budding yeast. *Mol. Biol. Cell.* 15, 3841–3862 (2004)
14. Futcher, B.: Transcriptional Regulatory Networks and the Yeast Cell Cycle. *Current Opinion in Cell Biology* 14, 676–683 (2002)
15. Gardner, T.S., et al.: Inferring genetic networks and indentifying compoud mode of action via expression profiling. *Science* 301, 102–105 (2003)
16. Güldener, U., Münsterkötter, M., Kastenmüller, G., et al.: CYGD: the Comprehensive Yeast Genome Database. *Nucleic Acids Research* 33, D364–D368 (2005)
17. Shinohara, A., Iida, K., Takeda, M., Maruyama, O., Miyano, S., Kuhara, S.: Finding Sparse Gene Networks. *Genome Inf.* 11, 249–250 (2000)
18. Stucki, J.W.: Stability analysis of biochemical systems. A practical guide. *Progress Biophys. Mol. Biol.* 33, 99–187 (1978)