# Building Gene Networks with Time-Delayed Regulations

Iti Chaturvedi, Jagath C. Rajapakse

**Abstract**

We propose a method to build gene regulatory networks (GRN) capable of representing time-delayed regulations. The gene expression data is represented in a linear model using a dynamic Bayesian network (DBN) and a skip-chain model using a hidden Markov model. The method therefore finds both short- and long-term regulatory interactions. The algorithm was tested on time-series data of the yeast cell-cycle. We compare the accuracy of GRN built by the present method with those built by using a higher-order DBN. The proposed method better fits the expression data and found core genes that are crucial in cell-cycle regulation.



*Key words:* Dynamic Bayesian networks, Gene regulatory networks, Viterbi algorithm, Skip-chain model, Genetic algorithms

## 1. Introduction

Gene expressions if collected over enough time points can be used to derive gene regulatory networks(GRN)(1). The GRNs represent causal activities of genes and gene products in biological systems and provide a basis for signal transduction in biological pathways. Since the signal transduction is transient, the study of the dynamics of the transduction is essential. The existing methods of deriving GRN from gene expression time-series can be

broadly classified into three categories: networks built by using 1. boolean rules 2; differential equations; and 3. stochastic modeling (2). Boolean networks are known to have lower sensitivity than Bayesian networks (3). They are not causal and use mutual information and minimum regulation linkage. Ordinary differential equations (ODE) have very high complexity as they describe processes at a very refined level (4).

Bayesian networks (BN) have been introduced for building gene regulatory networks in the stochastic framework. Pathways have a natural representation in BN where genes are present at the nodes of the network and the edges represent causal interactions among them. The causal dependencies are in terms of conditional probabilities which infer 'cause and effect' relationships among the genes in the network. However, BN are acyclic, and cannot track time-delayed, feedback, and self-regulatory events. Building gene regulatory networks has been extended by using the dynamic Bayesian networks (DBN) where the parents are selected from the previous time instant and the time-series are assumed to be first-order stationary (5). The first-order assumptions allow feedbacks but still deprive DBN of representing variable time-delayed interactions. The DBN formulation of GRN has been extended to higher-order which are capable of extracting higher-order regulatory interactions. Mutual information has been used to determine the best time-delay (6). These generative models become intractable in very high-orders. Therefore, we resort to a conditional skip-chain model which take into account delayed regulatory events of different orders.

We use a skip-chain model where two types of features model the time-delayed regulations: 1. linear features modeling the lower-order delays; and

2. skip features modeling the long-distant delays. In our model, the skip-features are modeled by using a hidden Markov model (HMM) where the log likelihood of the network is decomposed into a sum of consecutive pairs of genes, so the maximum likelihood regulation can be found by using the Viterbi algorithm. The Viterbi skip-features automatically determine the best time delay in the higher-order Markov chain.

Our approach consists of two stages: 1) identification of time-delayed interaction features and computation of Viterbi scores; and 2) prediction of the optimal GRN by using a genetic algorithm (GA). The fitness function of the genetic algorithm includes the Viterbi scores of time-delayed interactions. We demonstrate our method with an application to a long time-series of yeast cell-cycle data. Our method finds core genes that have regulatory effects with different time-delays on the cell cycle.

Earlier fusion of PPIN and GRN has been attempted using a similar model (7). The work was extended in (8) to implement a GRN without prior PPIN. The paper also considered over-fitting problems for small datasets. In this paper a derivation of the skip-chain model is given. We consider larger sets of genes and higher number of time points. Discretization into 3 levels and effect of multiple runs of GA is explored. Lastly, validation has been done with Biogrid. A validation with Biogrid PPI (9) for higher-order interactions shows that the method is more effective than simple HDBN.



## 2. Methods

Consider a set of $n$ genes $G = \{g_i : i = 1, 2, \ldots n\}$ and time-series of gene expressions gathered over $T$ time points for all the genes. Let the

gene expression data be $x = \{x_{i,t}\}_{n\times T}$ in which row vector $x_i = (x_{i,t} : t = 1, 2, \ldots T)$ corresponds to the gene expression time-series of gene $g_i$. The $x_{i,t}$ denotes the expression level of gene $g_i$ at time $t$. Suppose that gene expressions are discretized into a set $\Gamma$ of $d$ levels: $\Gamma = \{1, 2, ..., d\}$. A level of gene expression indicates a state of the gene. Let the set of parent genes regulating the gene $g_i$ be denoted as $a_i$ and the number of states that a node in $a_i$ take to be $q_i$.

### *2.1. Bayesian Networks (BN)*

BN are causal networks that can represent regulations among the genes at the nodes, as edges in the network. The BN decomposes the likelihood of gene expressions into a product of conditional probabilities by assuming independence of non-descendant genes, given their parents :

$$p(x) = \prod_{i=1}^{n} p(x_i|a_i, \theta_i) \tag{1}$$



where $x = (x_1, x_2, \ldots, x_n)$, $p(x_i|a_i, \theta_i)$ is the conditional probability of gene expression $x_i$ given its parents $a_i$, and $\theta_i$ denotes the parameters of the conditional probabilities.

Given the set of conditional distributions with parameters $\theta = \{\theta_i : i = 1, 2, \ldots n\}$, the likelihood can be written as

$$p(x) = \int p(x|S, \theta)p(\theta|S)d\theta \tag{2}$$

Let $\theta_{ijk} = P(x_{i,t} = k|a_i = j)$ and $N_{ijk}$ be the number of instances of $\theta_{ijk}$ that occur in the training data. Using the property of decomposability (5),

$$P(x) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{d}\theta_{ijk}^{N_{ijk}} \tag{3}$$

The model parameters $\theta_{ijk}$ are estimated by using maximum likelihood (ML):

$$\theta_{ijk} = \frac{N_{ijk}}{\sum_{k=1}^{d} N_{ijk}} \tag{4}$$

Then the log-likelihood of the data is given by

$$\log P(x) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{d} N_{ijk} \log \frac{N_{ijk}}{\sum_{k=1}^{d} N_{ijk}} \tag{5}$$

The likelihood approximation is known to be good when a large amount of data points are available (10) . Therefore, we use the maximum likelihood estimate of parameters to obtain the optimal structure of the Bayesian network.

### *2.2. Dynamic Bayesian Networks (DBN)*

The acyclic condition of BN does not allow self- and feedback-regulations of genes, which are essential characteristics of GRN. The dynamic Bayesian networks (DBN) overcome this by modeling the regulatory network from one time point to the next and the temporal expression pattern by unrolling network over time. A first-order DBN is defined by a transition network of interactions between a pair of structures $(S_t, S_{t+1})$ corresponding to time instances $t$ and $t+1$. In time instance $t+1$, the parents of genes are those specified in the time instant $t$. The gene regulations are obtained by unrolling the transition network over time and assuming first-order stationary behaviour over time. The likelihood of the data is given by Eq. (3):

$$P(x) = \prod_{t=1}^{T} \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{d} \theta_{ijk}^{N_{ijk}^{(t,t+1)}} \tag{6}$$

where $N_{ijk}^{(t,t+1)}$ correspond to the number of instances where $x_{i,t+1} = k$ while $a_{i,t} = j$. The first-order DBN has two layers and therefore $2n$ nodes.

### *2.3. Hidden Markov Models (HMM)*

The classical DBN is unable to capture complex time-dependencies and is extended to an $o$-order ($o \geq 2$) Markov chain. It predicts the expression levels of a set of genes based on the expressions of up to $o$ previous time points using frequency statistics. Higher-order dynamic Bayesian networks (HDBN) have been proposed to study time-delayed interactions. However, as $o$ increases it is not possible to compute the statistics using the few time points. Over-fitting occurs as can be seen by a corresponding decrease in likelihood in higher orders.

Therefore, we resort to a first-order hidden Markov model (HMM) to determine delayed interactions. It determines the probability of expression of a gene $g_j$ at time point $t$, given that $g_i$ was observed at $s_t$ where $s_t < t-1$ within the section of the time-series of length $t - s_t$. Let a sequence of hidden states from time point $s_t$ to $t$ be denoted by $y_{s_t:t} = (y_{s_t}, y_{s_t+1}, ...y_t)$ where $y_{t'}$ denotes the gene expressed at time point $t^{'}$ in the path. Correspondingly, we have the observed data $x_{s_t:t} = (x_{i',s_t}, x_{i',s_t+1}, \ldots, x_{i',t})$ where $x_{i',t'} = k$ is the discretized gene expression state. Since only the upregulation, downregulation or no regulation is considered, we only consider $x_{i',t'} \in -1, 0, 1$.



Given the microarray data, the maximum likelihood estimation can be used to estimate the state transition and emission probabilities, which are defined as follows (11):

$$a_{l,m} = \frac{M_{l,m}}{\sum_{m'=1}^{n} M_{l,m'}}, \quad \forall y_{t'} = g_l, y_{t'+1} = g_m \in G \tag{7}$$

$$b_l(k) = \frac{M_l^k}{\sum_{k'=1}^{d} M_l^{k'}}, \quad \forall k \in \Gamma, t^{'} \in \{s_t, s_t+1, \ldots t\}, g_l \in G. \tag{8}$$

where $M_{l,m}$ denotes the number of occurrences where $x_{l,t'} = x_{m,t'+1} = 1$

$\forall t^{'} \in \{s_t, s_t+1, \ldots, t\}$, $\forall k \in \Gamma$ and $M_l^k$ denotes the number of occurrences where gene $g_l$ has been at discrete state level $k$, $\forall t \in \{s_t, s_t+1, \ldots t\}$.

### *2.4. Viterbi Algorithm*

When the expression time-series are modeled with an HMM, the maximum a posteriori (MAP) estimate could be used to find the time-delayed interactions of a pair of genes. The path begins and ends at the known states of genes : say, $y_{s_t} = g_i$ and $y_t = g_j$. We assume that $t - s_t$ is not very large and conditional independence between feature vectors. For a sequence of a set of genes, the most probable path is given by the MAP estimate:

$$\arg\max_{y_{s_t:t}} p(y_{s_t:t}|x) = \arg\max_{y_{s_t:t}} p(x|y_{s_t:t})p(y_{s_t:t}) \tag{9}$$



The Viterbi algorithm(VA) can be used to find the best path by finding the MAP estimate, between two genes at distant time points (12). The state transition and emission probabilities are estimated during training. VA is a dynamic programming procedure and determines the best path in an incremental manner. Let $\delta_m(t^{'})$ be the probability of the most probable path ending at gene $g_m$ with the observation $x_{m,t'}$ at time $t^{'}$. Then, the best path at the next iteration is found as

$$\delta_m(t^{'}+1) = b_m(k) \max_l \{\delta_l(t^{'}) a_{l,m}\} \tag{10}$$

We can divide the path probability by length of path to get a first-order probability as a goodness of fit of the path. We define skip-edge score as the normalized MAP interaction :

$$h(x_i, a_i, s_t, t) = \log \frac{1}{(t - s_t)} \max_{y_{s_t:t}} p(y_{s_t:t}|x) \tag{11}$$

where the parent set $a_i$ has only one gene at time point $s_t$.

Finally, for any pair of genes, we can choose the best time-delayed interaction having the highest probability :

$$\hat{h}(x_i, a_i, s_t, t) = \max_{s_t,t} h(x_i, a_i, s_t, t) \forall\ t - s_t > o \tag{12}$$

where $g_j \in a_i$ and $o$ is predefined linear order. Next we combine this with the HDBN using a skip-chain model.

### *2.5. Linear and Skip features*

In order to handle both short- and long-range interactions, we model gene regulations by using both linear- and skip-chain features. Linear and skip features in microarray data are illustrated in Figure 1. The up-regulated genes are indicated with one. The genes $g_2$ and $g_3$ have a linear second-order interaction as both genes are upregulated at a delay of two time points. Similarly, the order of interaction between $g_1$ and $g_2$ is four and is hence represented by a skip feature. There can be numerous skip-features between a pair of genes. These features could be of different time delays or have the same time delay with different start and end time points. Here, we use our method of choosing an optimal time-delayed skip-feature between two genes as described in the previous section.



We can interpolate the two types of features by expressing the likelihood of a gene expression $x_i$ as a weighted sum of linear and skip-edge scores:

$$\log p(x_i|a_i, \theta_i) \propto \lambda f(x_i, a_{i(t-o:t)}, t) + (1 - \lambda) h(x_i, a_i, s_t, t) \tag{13}$$

where from (Eq. 5) $f(x_i, a_{i(t-o:t)}, t) = \sum_{j=1}^{q_i} \sum_{k=1}^{d} N_{ijk} \log \frac{N_{ijk}}{\sum_{k=1}^{d} N_{ijk}}$

where $f(x_i, a_{i(t-o:t)}, t)$ and $h(x_i, a_i, s_t, t)$ represent the linear- and skip-feature functions and $\lambda$ is a weight determined heuristically.

Linear-chain feature functions $f(x_i, a_{i(t-o:t)}, t)$ represent local dependencies that are consistent with an $o$-order Markov assumption of gene expressions (13). The skip-chain features represent long range dependencies in a GRN (14). The skip-chain feature functions $h(x_i, a_i, s_t, t)$ exploit the dependencies between genes that are arbitrarily distant at time instances $s_t$ and $t$ respectively. Such a skip-feature models a variable length Markov chain up to $T-1$ order where $T$ is number of time points. We use an HDBN to implement a linear-chain model and first-order HMM to implement the skip-chain model.

A classical HDBN uses a GA to find the optimal delays. For a DBN each gene given its parents needs $d^{|a_i|} \times d$ parameters, where $d$ is number of discrete levels and $|a_i|$ is the cardinality of parent set. For $o$-order HDBN, we can further have $o^{|a_i|}$ structural possibilities for each DBN. Hence, the search space and corresponding complexity is very high to find delays. On the other hand, skip models use VA to find the optimal delay and associated probability. Complexity of Viterbi is known to be quadratic on length of delay which is much smaller than the complexity of GA.



## 3. Implementation using a Genetic Algorithm

A genetic algorithm (GA) is used to find the optimal network structure. The individual solutions in the GA is defined by the connectivity matrix $\{c_{i,j}\}_{n\times n}$ where $g_j$ is a parent of $g_i$ in $c_{i,j}$. Each connection $c_{i,j}$ is initialized from the values in $M = \{M(i,j,l)\}_{n\times n\times T}$, where $M(i,j,l)$ is mutual

information between the expressions of genes $g_i$ and $g_j$ at a time lag $l$.

If each gene is allowed to have a maximum of $N_p$ number of parents, then the connections are randomly initialized as follows:

$$c_{i,j} = \begin{cases} \arg\max_l M(i,j,l) & \text{if } M(i,j,l) > \alpha \text{ and } \forall l \leq o \\ 0 & \text{if } M(i,j,l) < \alpha \text{ or } |a_i| > N_p \end{cases} \quad (14)$$

where $|a_i|$ is number of parents of gene $i$, $\alpha$ is the threshold for mutual information, and $o$ is the order of the model. We randomize the order of genes during initialization for each individual. The Bayesian score of this graph of linear edges of low orders can be calculated using Eq. (5).

To account for longer delays, for any two genes $g_i$ and $g_j$ where $c_{i,j} > 1$, we choose the highest Viterbi score among all the possible interaction features $\hat{h}(x_i, a_i, s_t, t)$. Here we only consider skip-edges longer than the linear edges modeled using an HDBN. If there are no such skip-edges, we set the feature probability to 0. Lastly we interpolate the probabilities of the linear and skip-graph using the weight $\lambda$ as in Eq. (14).



A random interpolation weight $\lambda < 1$ can be appended to the individual. The GA then finds the best structure with the highest posterior probability for different combinations of linear score, skip score and $\lambda$. GA does optimization of search for the structure, it parallel processes populations. Mutation and crossover introduce changes in the structure. Here we run the GA for $Q$ generations or if the change in score is less than $\alpha_q$ for 20 consecutive generations. As the low lying structures can easily dominate the others leading to premature search convergence, a minimum similarity threshold of $p_s > 0.7$ is maintained in each generation. Crossover involves swapping several rows and the weights between two parents resulting in possibly lower

energy structures. Lastly the mutation operator in each generation selects a random individual and inverts a random interaction.

The gene expressions were normalized by linear transform, and each profile was divided by mean for the gene. A GA was used to determine $T$ scaling factors. Spearman correlation is calculated between scaled expression and median profile for all genes. Fitness function of the GA is the sum of mean and variance of the correlation. Lastly, the scaled data is discretized into three levels based on relative increase between two consecutive time points.

## 4. Experiments and Results

We evaluated our method on time-series gene expressions of yeast cell-cycle data obtained from Chou et al. (15) (17 time points) and Spellman et al (16) (24 time points, cdc-15 cell cycle arrest). The yeast cell-division cycle consists of four main phases: genome duplication (S phase), and nuclear division (M phase), separated by two gap phases (G1 and G2). The S-G1-M-G2-S form a cycle for cell duplication. The expression values were normalized and discretized into 1 for upregulation, 0 for no regulation, and -1 to denote downregulation by using an approach described earlier (17). We use Chou dataset on nine genes which appear to control the sequential activation of cyclins and other cell cycle regulators (6). Similarly the Spellman *et al.*, dataset was used on subset of genes in different phases.

VA was used to compute skip-feature probabilities of all pairs of genes and GA was used to find the optimal structure. Simulation was done upto order four of HDBN and skip-chain, with a maximum skip-edge length of 10 time points. We plotted the histogram of MI for each pair of genes for

different time delays. The peak of the histogram was taken as the threshold as most interactions below that had negative or low mutual information. The parameters of the GA were determined empirically. A MI threshold of 0.27 was found optimal to recover edges in Spellman dataset and 0.1 for the Chou dataset. The skip-edge weight is determined heuristically by GA along with structure. The GA chooses the network with the best combination of the skip and linear edges Eq. (14). Simulation was done at different numbers of individuals (N) and generations (Q) (N=200/300/400 and Q=300/400/500) for both HDBN and skip-chain model. The GA similarity threshold was set to 0.7, and in the case of Chou dataset it was increased to 0.9 to avoid early convergence. The GA stops when the maximum number of generations is reached or if the score difference is 1 for 20 consecutive generations. We consider edges with confidence over 0.7 over 20 runs of the GA in the final network. The mean and standard deviations were reported. It is observed that optimal $\lambda$ found by GA is larger for small networks where probability of skip-edge is low and is smaller for large networks where probability of skip interactions becomes higher due to longer cascades of genes.



As seen from Table 1, HDBN of order four and skip-chain of order 1 have the highest likelihood in all datasets confirming that the network fits expression data well . As seen, a skip-chain yielded higher number of edges at 0.7 confidence since the networks are more stable. The HDBN shows a peak of the interactions at delay 1 and 4. This indicates that most interactions are first-order or instantaneous, and the fourth-order is insufficient to capture all higher-order interactions. Since, we are modeling variable order delay, a previously proposed validation can be used (7). Here we look for a cascade

of genes in the GRN corresponding to an interaction in PPIN. On a subset of 19 S phase genes for which interaction are available in Biogrid we can clearly see that our model gives higher number of true positives than DBN or HDBN. Figure 2 shows predicted network for 9 genes, it can be seen that delays are consistent with phases of cell cycle. For eg. Ndd1 regulates Swi6 at a delay of 70 mins. Figure 3 shows prediction for subset of 19 genes. The dashes edges correspond to cascade of genes forming a TP in PPI. For eg. HTB2 interaction with MET6, takes the form: HTB2 interacts with HTB1, and HTB1 interacts with MET6.

We also see that the method is robust to the increase in the number of genes. It can be seen that majority of the predicted interactions have time delays. Bigger networks like S(36) of 52 interactions had several 8 time points delay. This is very useful as building higher-order DBN is very time-consuming and has to deal with an exponential number of parameters. Hubs in a network are nodes with high degree of connectivity, which usually represent important nodes in causal networks. Table 2 gives a list of top 10 hubs of networks derived by different methods for 19 genes in S phase. The corresponding hubs in the Biogrid target network are also given. The top core genes produced by all methods seem the same and the core genes produced by the DBN and our method were quite similar.



Further comparison of top 10 hubs predicted by a DBN, an HDBN and a skip-chain using Saccharomyces Genome Database (SGD) showed that while a DBN had hubs involved in instantaneous events such as initialization, silencing, etc., the time-delayed hubs in HDBN were mostly regulatory or feedback associated. Our skip-chain model showed a combination of both

types of regulation, hence representing the robust signaling networks well. As seen in the S phase, some hubs such as HHF1, HTB1, HTA2, and HHT1 are conserved in all the models. These are all histones required to initiate duplication by chromatin assembly and chromosome function. HDBN model picked up a hub ADA2 also seen in Biogrid. ADA2 encodes a chromatin modifying complex. It also plays a role in transcriptional silencing at telomeres which occurs at end of duplication. KIP1 was a FP, it is a kinesin-related motor protein required for mitotic spindle assembly and chromosome segregation. It encodes the inhibitor of several cyclin CDK complexes which control the progression of the cell cycle from G1 to S phase. These events are slower and hence emerge in skip-chain model.

## 5. Discussion and Conclusion



Pathways are often triggered by transcription factors which in turn express genes and produce proteins. Therefore, the regulatory interactions in molecular pathways can be given by GRN. Gene regulations generally includes dynamic feedback loops, cascaded interactions, intermediary factors, etc., which provides for underlying biological mechanisms of regulation. This results in different time delays in regulatory interactions. The delays in regulations are an integral part of biology. In this work we focused on modeling delays in GRN.

We have considered higher-order DBN (HDBN) for representing delays in regulations. When larger delays are involved, implementation of HDBN becomes intractable. Therefore, we proposed a skip-chain HDBN. This involved two components: linear model to represent short delays and skip model to

represent long delays. These two components may represent actions of activator and inhibitor involved in regulatory interactions. Our method was evaluated against earlier approaches, which shows our method better fits the gene expression data when GRN was built. In order to provide a more biological meaningful validation, we performed comparison with the protein-protein interaction data. That validation also showed superiority of our technique over other methods but because of the incompleteness of protein-protein interaction data sources, such comparisons results in large false positives.

Skip-chain models address the difficulties of a HDBN by easily incorporating long time-delayed regulations. The skip-chain model is a first-order HMM and captures long-distance dependencies of input time-course gene expressions. This inference technique leads to lower total training time without loss in accuracy compared to HDBN. The forward Viterbi path through the trellis determines the best long-distant time delay and therefore automatically finds the best higher-order interactions between two genes. The method can be applied to different long- time series by suitably tuning the GA similarity measures.



## References

[1] Wagner J, Stolovitzky G: **Stability and Time-Delay Modeling of Negative Feedback Loops**. *Proceedings of the IEEE* 2008, **96**(8):1398–1410.

[2] Gebert J, Motameny S, Faigle U, Forst CV, Schrader R: **Identifying Genes of Gene Regulatory Networks Using Formal Concept Analysis**. *Journal of Computational Biology* 2008, **15**(2):185–194.

[3] Li P, Zhang C, Perkins EJ, Gong P, Deng Y: **Comparison of probabilistic Boolean network and dynamicBayesian network approaches for inferring gene regulatorynetworks**. *BMC Bioinformatics* 2007, **8**:S13–S20.

[4] Liu B, Thiagarajan P, Hsu D: **Probabilistic Approximations of Signaling Pathway Dynamics**. In *Computational Methods in Systems Biology* 2009:251–265.

[5] Friedman N, Murphy K, Russell S: **Learning the Structure of Dynamic Probabilistic Networks**. *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)* 1998, :139–14.

[6] Zhengzheng X, Dan W: **Modeling Multiple Time Units Delayed Gene Regulatory Network Using Dynamic Bayesian Network**. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on* 2006:190–195.

[7] Chaturvedi I, Rajapakse J: **Fusion of Gene Regulatory and Protein Interaction Networks Using Skip-Chain Models**. In *Pattern Recognition in Bioinformatics*, *Volume 5265 of* Lecture Notes in Computer Science 2008:214–224.

[8] Chaturvedi I, Rajapakse J: **Detecting robust time-delayed regulation in Mycobacterium tuberculosis**. *BMC Genomics* 2009, **10**(Suppl 3):S28.

[9] Breitkreutz BJ, Stark C, Reguly T, Boucher L, Breitkreutz A, Livstone M, Oughtred R, Lackner DH, Bahler J, Wood V, Dolinski K, Tyers M:

**The BioGRID Interaction Database: 2008 update**. *Nucl. Acids Res.* 2008, **36**(suppl1):D637–640.

[10] Friedman N, Murphy K, Russell S: **Learning the Structure of Dynamic Probabilistic Networks**. *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)* 1998, :139–14.

[11] Cappae O, Moulines E, Rydaen T: *Inference in hidden Markov models* 2005.

[12] Hao T, Huang TS: **Improved Graphical Model for Audiovisual Object Tracking**. In *Multimedia and Expo, 2006 IEEE International Conference on* 2006:997–1000.

[13] Galley M: **A Skip-Chain Conditional Random Field for Ranking Meeting Utterances by Importance**. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)* 2006:364–372.



[14] Sutton C, McCallum A: **Collective Segmentation and Labeling of Distant Entities in Information Extraction**. In *Presented at ICML 2004 Workshop on Statistical Relational Learning* 2004.

[15] Chou R, Campbell M, Winzeler E, Steinmetz L, Conway A, Wodicka L, Wolfsberg Ta: **A genome-wide transcriptional analysis of the mitotic cell cycle**. *Molecular Cell* 1998, **2**:65–73.

[16] Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B: **Comprehensive identification of**

**cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization**. *Mol Biol Cell* 1998, **9**(12):3273–97.

[17] Shmulevich I, Zhang W: **Binary analysis and optimization-based normalization of gene expression data**. *Bioinformatics* 2002, **18**(4):555–65.

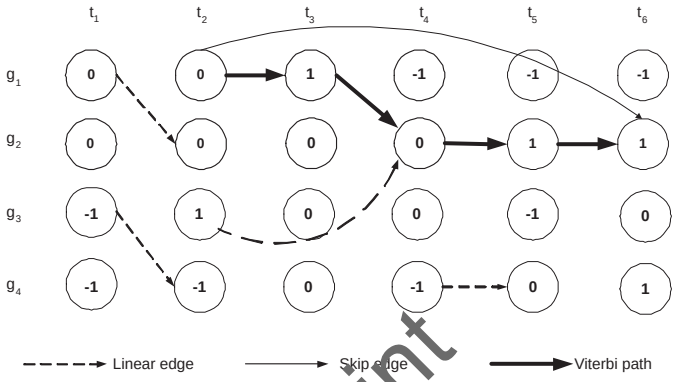




Figure 1: State transition diagram for six time points and four genes in a DBN. The dashed edges are linear $o = 1, 2$ order edges formed by linear features. The solid directed edge is an example of skip-edge over four time points which models a long-distant dependency. The bold directed line shows a skip path computed by Viterbi algorithm.

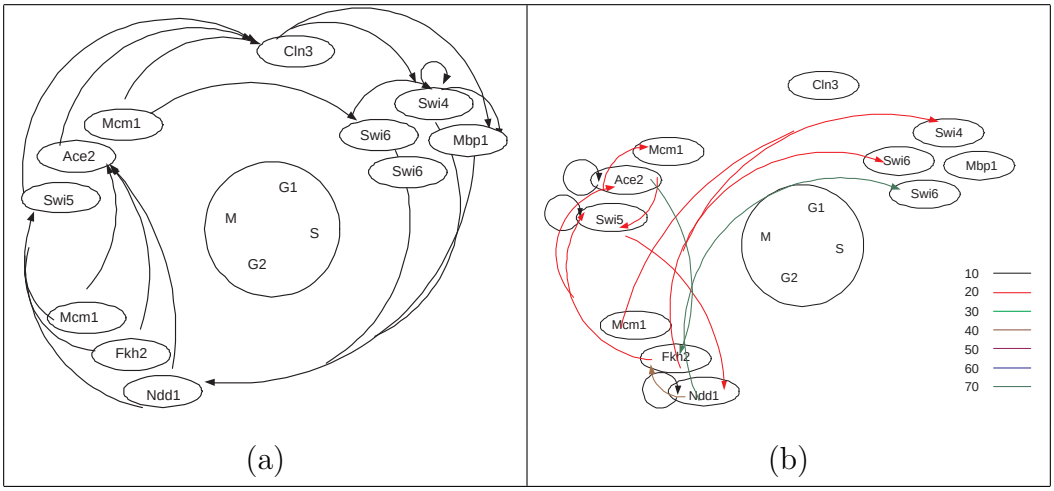


Figure 2: Time-delayed interactions in predicted network of 9 genes (a) Target network, (b) Predicted network



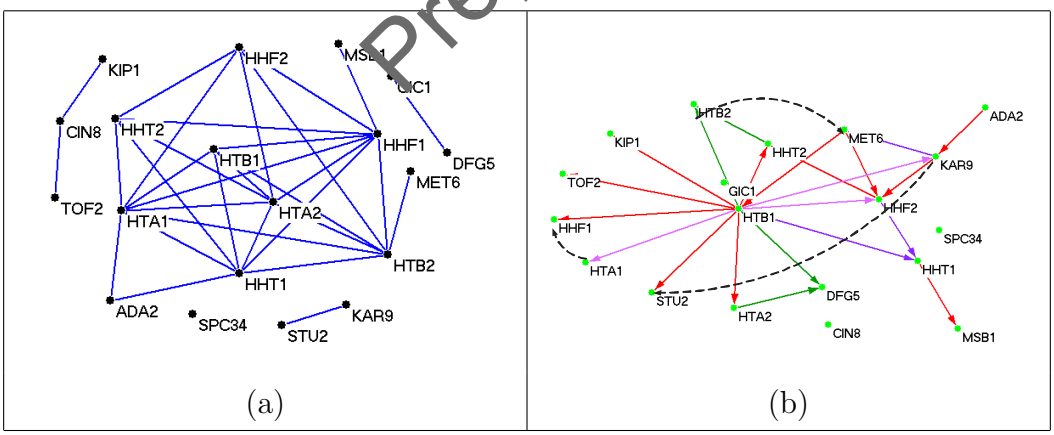


Figure 3: Time-delayed interactions in predicted network of 19 genes (a) Target network, (b) Predicted network. Dashed edges correspond to $k$-skip validation.

Table 1: First- and higher-order regulations in predicted by DBN(d), HDBN(h), and skip-chain(s) models, *(n)* denotes overlapping skip-edges and $o$ is order of the model.

| | | | Order of Regulation | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| #genes | $o$ | ML | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Tot | $k$-TP |
| All(9) | d:1 | -52.66 ± 3.13 | 15 | | | | | | | | 15 | 11 |
| | h:4 | -34.95 ± 0.53 | 3 | 7 | 2 | 6 | | | | | 18 | 7 |
| | s:1 | -29.95 ± 0.77 | 15 | (9) | | (1) | | (2) | | | 15 | 11 |
| S(19) | d:1 | -194.57 ± 13.34 | 20 | | | | | | | | 20 | 27 |
| | h:4 | -120.25 ± 7.76 | 6 | 0 | 1 | 7 | | | | | 14 | 21 |
| | s:1 | -53.36 ± 0.80 | 25 | (12) | | | (3) | (4) | (3) | | 25 | 36 |
| S(36) | d:1 | -426.75 ± 2.76 | 52 | | | | | | | | 52 | 36 |
| | h:4 | -245.91 ± 12.65 | 27 | 6 | 3 | | | | | | 28 | 34 |
| | s:1 | -131.27 ± 2.22 | 52 | (32) | (2) | (2) | (2) | (7) | | (3) | 52 | 37 |
| G2(33) | d:1 | -320.45 ± 3.46 | 34 | | | | | | | | 34 | 10 |
| | h:4 | -300.22 ± 11.71 | 12 | 10 | 10 | 12 | | | | | 44 | 9 |
| | s:1 | -182.10 ± 0.71 | 50 | (31) | (2) | (1) | (2) | (6) | | (4) | 50 | 13 |
| M(60) | d:1 | -702.49 ± 23.60 | 49 | | | | | | | | 62 | 19 |
| | h:4 | -409.13 ± 19.80 | 15 | 12 | 15 | 25 | | | | | 67 | 22 |
| | s:2 | -324.51 ± 8.00 | 33 | 27 | (5) | (43) | | (10) | | | 64 | 18 |



Table 2: Top 10 hubs in different phases of yeast cell-cycle. The number below the gene name is the connectivity in the network.

| | | Rank of genes based on connectivity | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $o$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| S(19) | bg | HHF1 | HTA1 | HHT1 | HTB2 | HTA2 | HTB1 | HHT2 | HHF2 | ADA2 | CIN8 |
| | | 8 | 8 | 8 | 7 | 7 | 7 | 5 | 5 | 5 | 3 |
| | d:1 | HTB1 | HHF2 | HTA1 | HHT2 | HHT1 | MET6 | HTB2 | KIP1 | MSB1 | TOF2 |
| | | 10 | 5 | 4 | 3 | 2 | 2 | 2 | 2 | 2 | 2 |
| | h:1 | TOF2 | HTB1 | HHT2 | HHF2 | HHF1 | HTA1 | ADA2 | - | - | - |
| | | 8 | 6 | 3 | 2 | 2 | 2 | 2 | - | - | - |
| | s:4 | HTB1 | HHF2 | KAR9 | HHT1 | MET6 | HHT2 | HTA2 | DFG5 | HTB2 | TOF2 |
| | | 14 | 5 | 4 | 3 | 3 | 3 | 3 | 2 | 2 | 2 |