Inference of BN and BDN Models of Genetic Networks Using Monotonous Time Transformations

Juha Kesseli, Pauli Rämö, and Olli Yli-Harja
Institute of Signal Processing, Tampere University of Technology,
P.O.Box 553, 33101 Tampere, Finland

Abstract

This paper considers the problem of inferring a Boolean Network (BN) from gene expression data that is available as a frequently sampled time-series. The particular problems that arise are discussed and Monotonous Time Transformations (MTT) are presented as a possible solution. The method is also applied to a simple inference test for Boolean Delay Networks (BDN). The results with data generated by a simulation model show that the method presented can improve the inference performance in the described cases. The measurements currently available are not yet suitable for testing the methods because of the low sampling rates and the amount of noise present.

Keywords: Boolean network model, genetic regulatory network, network inference, monotonous time transformation, Boolean delay network.

Contact: kesseli@cs.tut.fi, phone:+358 3 3115 3866, fax: +358 3 3115 4989

1 Introduction

Boolean Network (BN) models have been used for crude characterization of biological systems for decades (Kauffman 1993). They have been considered interesting because of their apparent simplicity and their properties that have been studied at some length. Cells in different conditions are typically identified with the BN model residing in different attractors. The characterization of attractors and their properties has therefore drawn some attention, although lately it has also been proposed that attractors of BNs in the traditional sense might not be the best way to approach the question of different types of cells. The problem is that even though the model is robust against small changes in the state, the attractors themselves are not robust against changes in the system functions. A biological system should be assumed rather robust, so alternative characterizations are being sought (Aldana et al. 2002).

One of the basic problems concerning the BN modeling of genetic regulatory networks is related to selecting the discrete timestep needed in the model. The common answer to this question is formed along the lines of the selection not being too critical if we are satisfied with modeling the qualitative aspects of the dynamics only. This

is based on the observation that many types of system behavior, e.g. attractors and their properties, are not dependent on individual functions of the model corresponding exactly to their biological equivalents. A more practical answer to neglecting the selection problem is related to the data used for inferring networks. The data is typically measured only at few timepoints constrained by the costs of more frequent sampling.

These answers are not, for obvious reasons, entirely satisfactory. As with any model, we would like to obtain as good a fit with reality as we can. The cost of measuring mRNA and protein levels is decreasing so that sampling rates possible with reasonable costs are increasing. Another way of increasing time resolution might be to take samples from the culture with a very high sampling rate and freeze most of them. After a preliminary analysis for a selected few of the samples the most interesting ones from the freezer could be used for measurements as well. In this way the timepoints with the most interesting information could be sampled quite frequently without excessive costs.

Selecting an arbitrary timestep has its problems. Consider inference of a network with N genes and assume a maximum in-degree of K in order to simplify computation and to tune the dynamics as desired. From the dynamics point of view $K = 2$ is often considered an interesting choice (Kauffman 1993). Denote the set of genes $G = \{g_1, \ldots, g_N\}$ and $\mathbf{B} = \{0, 1\}$ so that the state-space can be denoted as $\mathbf{B}^N$, $|\mathbf{B}^N| = 2^N$. If we do the measurements concentrating on modeling of the cell cycle, we are only looking at the states of the model on the attractor corresponding to this cell cycle. If we took n samples during one cell cycle, we would, in effect, be forcing the number of states on the attractor to be n. It can be seen that with too high a number of samples the exact modeling of the data is impossible without hidden states. This is due to the fact that the system has a total of 2^N states, and it is not reasonable to assume that even a significant part of the state-space would be covered by an attractor.

It is thus not possible to model the dynamics of the system with too short a timestep. Also, in this case we could have in the data lots of consecutive samples with no changes in the states. We could find a best-fit solution, but we are in effect forcing the length of an attractor to be the number of sampling points. With too short a sampling interval the network obtained will certainly not reproduce the attractor like it should when being simulated. With too long a timestep we are clearly losing information. A justified compromise is needed.

Increasing the amount of data is important for networks of non-trivial size at least for reducing the effects of noise. The publicly available time-series data has been of rather low quality for inference purposes because of the amount of noise present and the low sampling rate (Spellman et al. 1998). Theoretically, inference of a Random Boolean Network (RBN) with a maximum in-degree K requires only $\Theta(\log(N))$ state transitions as input data on average. For this result to hold it is assumed that the transitions are obtained uniformly from the state-space (Kauffman 1993). In reality, the available measurements are extracted only from few attractors. Therefore it is likely that even this asymptotical property does not hold in practice and more data from diverse conditions is needed to properly do the inference (Akutsu et al. 1999). It is also apparent that the particular cases of Boolean networks that represent living organisms can not be assumed to be representative of an average RBN realization in any sense.

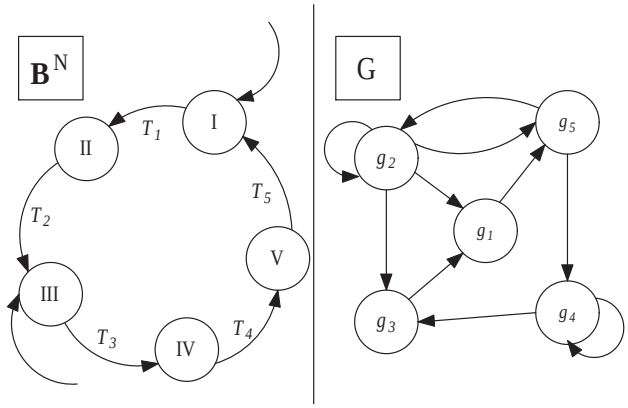

Figure 1: The state space $\mathbf{B}^N$ of a BN has possibly different delays between the states that are determined as a part of the MTT sampling method. These delays are not directly transferrable to the space of genes G (of size N) in which the network to be inferred resides.

2 Monotonous Time Transformations for BN Inference

For purposes of this section we assume that the data used for inferring the network is available as a continuous-time N-dimensional signal

$$x(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T \in \mathbf{R}^N.$$

Such signals can be obtained from simulation models using e.g. differential equations (Chen et al. 2000) or by fitting some suitably selected smooth curves to discrete, sampled real-world data. In order to have sufficient coverage of the state-space several time-series with different initial conditions should be used. The functions to be inferred are descriptive of the changes occurring in the regulatory network. For this reason we now focus on the moments at which the binarized continuous-time gene expression signals change their values.

Denote the set of time instants t_j^i at which the binarized value of gene g_i changes by $T^i = \{t_j^i : j = 1, \ldots, n_i\}$, where n_i is their number. In practice the sets T^i are distinct, $T^i \cap T^j = \emptyset$ for $i \neq j$, so for $T = \bigcup T^i$ it holds that $n = |T| = \sum |T^i| = \sum n_i$. Adding genes to the data increases the number of time instants in T rather quickly. So selecting all the instants between points of T as timepoints in the inference would lead to similar problems as with too short a fixed timestep. In fact, in this case the problems would be even more evident. At each timepoint only one bit in the state would change.

The timepoints in T can now be clustered using any clustering method in order to decrease their number. The changes in gene expression occurring during one resulting cluster of timepoints are then combined to represent a single state change containing changes in several genes. This corresponds to assignment of the set of sampling instants $S = \{s_l | l = 1, \ldots, m\}$ between the clusters. At each of these instants a single data vector $x_l = x(s_l)$ is taken as a sample. In effect, the data-dependent selection of

timepoints is doing a Monotonous Time Transformation (MTT) on the data and then proceeding with the inference as in the usual case with uniform sampling. With different clustering method we obtain different time transformations.

As a result of the inference with e.g. the best-fit method (Shmulevich et al. 2001) we are getting an ordinary BN (Fig. 1). The only difference is that in simulating the model it should not be assumed that time is running at a fixed pace. Instead, the transition from one state of the BN to another happens with a delay that could be different for each state. A suitable estimate for the state lengths can also be found from the data for those states that are present in the time-series. This information can be used in running the network to reproduce the delays. Since the attractor structure of the BN system is assumed to capture the most interesting parts of the dynamics, the variable delays should not be a significant problem as such. Most of the states are probably never seen in a time-series so their lengths cannot be learned. During simulation those states can be modeled with any constant length. The state lengths are mainly considered for visualization purposes, since they don't change the attractor structure of the system.

Unfortunately, we are facing a problem similar to the one of selecting a fixed sampling rate for many clustering methods. The problem is now in selecting a suitable number of cluster centers or the sampling instants. The methods that select the number of clusters automatically are therefore appealing. The monotonous time transformation helps in cases where the variations in the timescales of changes in the original signal cause problems for traditional inference. Data resulting from non-uniform sampling is also at least conceptually simple to use in this method. The interpolation from non-uniform samples that is probably needed is not, however, in practice quite as simple.

3 Clustering Methods

The state of a BN at a given time is the vector $x \in \mathbf{B}^N$ of values representing the current binarized expression for each gene. The proposed clustering methods and the corresponding Monotonous Time Transformations are based on heuristic assumptions about the locations of characteristic states in the binary data. The clustering is done by locating the sampling points S and gathering the new data of the data values on these points. Each gene is sampled using the same sampling points. There does not exist a clustering result that would meet all of the optimal clustering criteria. One possibility to formalize the optimal criteria can be the following:

a) Sample each state only once.

b_1) Locate all the states in the data.

b_2) Locate the states which seem to have the greatest importance.

c) Sample each gene exactly once between two changes in its binarized value.

d) Consistent sampling i.e. all the cell cycles should be sampled similarly.

As can be seen the set of criteria is self-conflicting and not properly defined. However, using these guidelines a satisfactory transformation result can be achieved. Some

method	optimization criterion	user given parameter
(i)	none	m = number of clusters
(ii)	a , b_1	none
(iii)	a , b_2	m = number of clusters
(iv)	a , b_2	m = number of clusters
(v)	a , b_2	none
(vi)	a , c	none
(vii)	a , c	none

Table 1: Table of Monotonous Time Transformations. The first column is the name of the method and the optimization parameters are shown in the second column. The third column shows which parameters have to be given by the user to the algorithm.

algorithms may require a parameter which has to be given by the user but these methods are not preferred. Table 1 summarizes the proposed methods.

The methods can be divided roughly into two categories. The first category methods ((ii)-(v)) are based on locating the most important states in the data i.e. trying to optimize the criterion b. The category two methods (vi) and (vii) are developed to optimize the criterion c. The criterion a is met with all of the proposed clustering methods. No clustering method used here is based on the criterion d, although the overall performance of an algorithm can be seen to be typically reflected also in its success in this respect. The method (i) is the traditional approach and no time transformation is performed. The data is sampled with intervals of equal length.

The method (ii) takes one point to S between each point of T i.e. $S_l = (t_l + t_{l+1})/2$, $l = 1 \ldots n-1$. Therefore all the states are sampled once. This, however, leads to a situation where the cell cycle length is determined by the number of individual changes of genes in the data. The methods (iii) and (iv) require a user given parameter m of the desired number of sampling points. In method (iii) n initial clusters are formed so that each contains a single point from T. After this the closest clusters are combined and the location of the new cluster is set at the midpoint of the corresponding time interval. The procedure is repeated until the desired number of clusters is reached. The sampling points are then put in the middle of the time intervals left outside the clusters.

The method (iv) clustering is done by selecting the m longest intervals between the change points T and putting a sample point in the middle of the interval,

$$S_l = \frac{t_{\hat{k}_l} + t_{\hat{k}_l+1}}{2},$$

where

$$\hat{k} = \underset{\substack{k \in \{1,\ldots,n\}^m \\ i \neq j \Rightarrow k_i \neq k_j}}{\arg\max} \sum_{i=1}^{m} |t_{k_i+1} - t_{k_i}|.$$

The method (v) does not need any user given parameter. The method is the same as (iv) but instead of selecting the m longest intervals all the intervals longer than a certain value l are selected. An estimate for a suitable l can be selected as $l = \ln(N)L/C$.

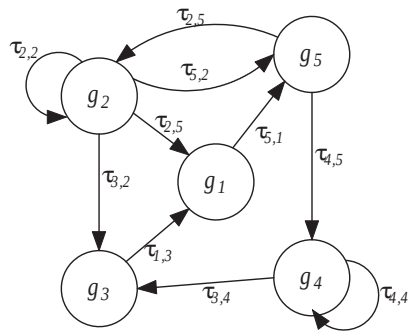

Figure 2: The delays in Boolean Delay Network are between the genes in the network and not between the states as in the BN model shown earlier.

N is the number of genes in the data, L is the total length of the data and C is the total number of state changes. The formula can be derived by creating a large number of uniformly distributed random vectors of length L with C gene change points. L and C were varied. The number of clusters is then observed to follow the equation $n = kC$, where the coefficient k equals $1/e$ and the interval length is the constant $l = L/C$. By reversing the problem it can be noticed that by multiplying the interval size by the constant $\ln(C/n)$ the number of clusters n is obtained. If the number of clusters is desired to be equal to the average change count of each gene, $N = C/n$ can be selected.

The method (vi) starts from the sample point 1 and goes through the whole data point by point. A zero vector v of length N is initialized in the beginning. The vector v is updated by adding one $v_i = v_i + 1$ when a gene change t_j^i for the gene g_i is met. The first gene that has had two changes causes the vector v to reset and a sampling point is put before the change. The exact position is in the middle of the change point and the previous change of any gene. This is repeated until the whole data has been processed. In this method no changes in the states of individual genes are missed in the sampling. The method (vii) is similar to (vi) but the vector v is reset and the sampling point is put when all its values are at least one e.g. all genes have changed their value at least once.

4 Inferring Boolean Delay Networks

Boolean Delay Network (BDN) is a Boolean model in which non-negative real-valued delays have been added to the connections between the genes (Fig. 2). This kind of a network is also called a Continuous-Time Boolean Network (CTBN) (Öktem et al. 2002). The method of Monotonous Time Transformations is here applied to BDNs. For the networks in this section the in-degree bound $K = 2$ has been selected for convenience of notation.

First, a BN is inferred with MTT just like in the previous section. As a result, for

gene g_i we are getting a Boolean function $f_i : \mathbf{B}^2 \rightarrow \mathbf{B}$, by which the update equation for the gene is computed as

$$x_i(n+1) = f_i(x_{c_{i,1}}(n), x_{c_{i,2}}(n)),$$

inputs of which are the binary states of the two genes $g_{c_{i,1}}$ and $g_{c_{i,2}}$.

The assumption made in order to obtain a BDN is that the MTT allows us to discover reasonably correct functional relationships determining the state transitions. This can be rephrased as a restriction on the lengths of delays we assume present in the network. The delays are assumed to be of approximately the same order with no huge scale variations. If we assume that the functions f_i and connections $c_{i,1}$ and $c_{i,2}$ are found using MTT and BN inference, the only parameters left to uncover in the BDN are the delays.

Different types of delay structures can be used. In (Öktem et al. 2002) there are separate delays for each connection. That is, for gene g_i we have delays $\tau_{i,c_{i,1}} \in \mathbf{R}^+$ and $\tau_{i,c_{i,2}} \in \mathbf{R}^+$, where $\tau_{i,j} \in \mathbf{R}^+$ denotes the delay from gene g_j to gene g_i. The state transition equation now becomes

$$x_i(t) = f_i(x_{c_{i,1}}(t - \tau_{i,c_{i,1}}), x_{c_{i,2}}(t - \tau_{i,c_{i,2}})).$$

In order to make a crude model of the different delays for buildup and decay of mRNA this could be generalized so that the delays are dependent on the state of the gene they are propagating. A change of a bit in the state from zero to one would occur with a different delay than one from one to zero. In case of several changes occurring in rapid succession e.g. the last one should be considered dominating to avoid a contradictory definition.

In this paper we are restricting the inference to the case $\tau_i = \tau_{j,i}, j \in \{1, \ldots, N\}$. This means that there are only a total of N delays in the network and the delay from a single gene is the same for all affected genes. The method of estimating the delay τ_i is based on computing the cross-correlation sequence

$$r_i(\tau) = x_i(t) \star f_i(x_{c_{i,1}}(t), x_{c_{i,2}}(t)).$$

The lag τ with the highest value of cross-correlation is selected as the lag τ_i for gene g_i in the model.

5 Test Results

The data used for testing was generated with an implementation of the differential equation model for 28 cell-cycle dependent proteins published in (Chen et al. 2000). A subset of 14 proteins were selected for testing on the basis that the model can be simulated with this smaller number of variables as well. The choice allows simpler visualization of the results and makes the solution a little less non-unique. The expression data consists of 14 real-valued continuous-time expression ratios of proteins instead of mRNAs.

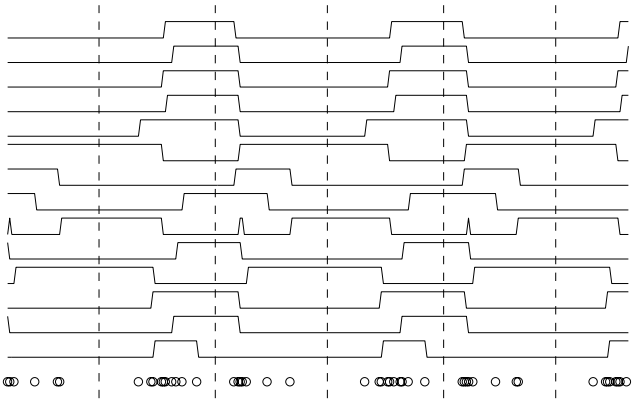

Figure 3: Clustering example with method (v). The binarized original data is drawn for 14 genes. The vertical lines depict the positions of the sampling points. Small circles at the bottom show all the positions of the gene changes.

The inference proceeds similarly for protein data as for gene expression data, since the proteins are another, in some modeling respects equal, part of the regulatory network in the cell. In the future it would be interesting and beneficial to have simultaneous mRNA and protein measurements available in a time-series data. The behavior of the particular protein time series available as a test data here is not particularly interesting and the expression levels seem to follow nearly identical phases. This makes the network inference highly non-unique and the resulting networks are thus not biologically accurate. More interesting results would likely be obtained in the case of more diverse data. Because biological conclusions are not made in this paper the terms protein concentration and gene expression ratio can be used interchangeably for purposes of this section.

The expression ratios of the original data are thresholded to binary values individually by using the mean value as the threshold. This thresholding method is not optimal but in our experiment proved to be sufficient. The thresholding process naturally discards a lot of information of the data. With a BN we are trying to model only the main features, so the loss of details should be acceptable.

The test results are obtained by clustering the data with all of the clustering methods and comparing the original data to the output data created by the network. Figure 3 illustrates the results of the clustering process with method (v).

BN inference is done to the sampled data using the best-fit algorithm (Shmulevich et al. 2001) with the maximum indegree $K = 2$. The inference testing is done by running the inferred Boolean network and the original data is then compared to the output data. If the output data resembles the original data it can be said that the network has learned the single attractor pattern from the original data.

To enable a direct comparison with the original data the output data has to be synchronized to the binarized original data. This is done by estimating the state lengths or in other words the delays between the states of the network. The length of a state

method	m	% correct BN	% correct BDN
(i)	6	20	4
(ii)	32	22	1
(iii)	6	8	8
(iv)	7	37	12
(v)	5	49	20
(vi)	5	7	0
(vii)	11	10	8

Table 2: BN and BDN results are displayed in this table. m is the number of clusters and the result numbers are the percentages of correct states at each sample point. The results should be considered suggestive because a slight misalignment of the time-series can cause a significant reduction in the percentage. Final conclusions can be drawn only by studying the graphical results.

is estimated from the data using the sampling point by moving it at the beginning of the corresponding state in the binarized original data and calculating the distance to the following sampling point similarly moved. The effect of the state delays and the behavior of the Boolean network can be seen in Fig. 5.

The behavior of the network depends on the starting points. The starting points were selected to be the same as in the sampled data. The percentage of correct states at each sample point compared to the binarized original data is the final result. The percentage of correct states at each time instant were from 7 to 49 percent. For those methods that require a user-given parameter the value was selected to give competitive results. These results are shown in Table 2 along with the number of clusters for each method. Visualizing the output data indicates which characteristics of the data the network has learned. An example of the possible problems arising in inference with equal sampling intervals can be seen in Fig. 4. See Fig. 5 as an example of output data with a time transform applied. It should be noted that the BN inference of the sampled binary data is more successful in most cases. Adding the inferred state lengths makes the situation more complicated.

For BDN the test results are more difficult to calculate. Because of errors in the delay parameter inference the output data does not follow exactly the phase of the original data. Also in this case the numerical comparison of different clustering methods proved to be difficult due to the problems mentioned earlier. An example of the output data is illustrated in Fig. 6.

6 Discussion

Inference of Boolean Networks is critically dependent on the sampling of the data. In typical situations until now there has not been too much choice in the sampling, the best choice clearly being as many points of uniform sampling as is feasible, this rate being limited by the increasing costs. Increasing the sampling rate and blindly applying the traditional inference methods leads to problems, however. After a decent

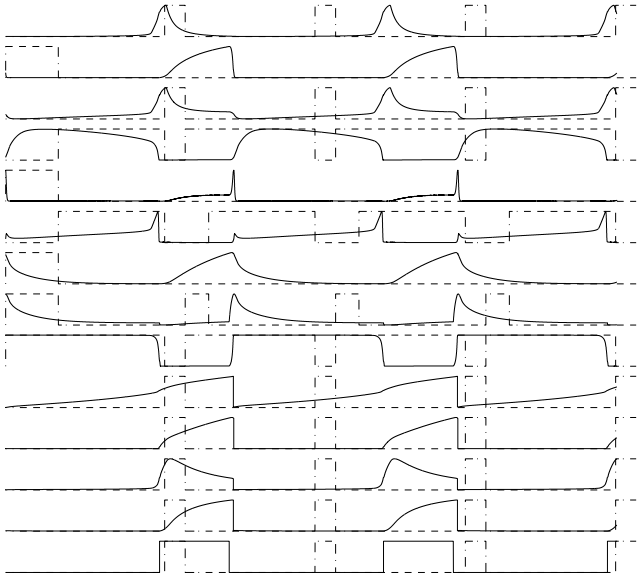

Figure 4: Output data for BN inferred without time transforms, i.e. using method (i). Solid lines are the original protein concentrations and dash-dotted lines are the output data. In this example difficulties that often arise with the method can clearly be seen.

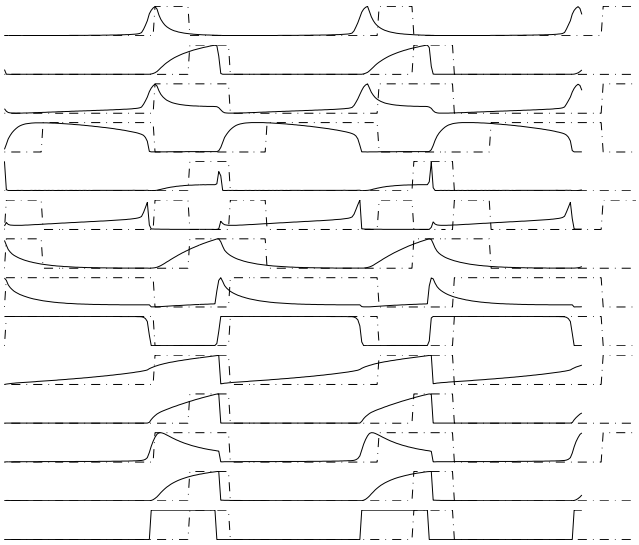

Figure 5: Output data for BN using the clustering method (vi). Solid lines are the original protein concentrations and dash-dotted lines are the output data. It can be noted that the output data follows the original data well. Even some small characteristics of the time-series can be modelled using the system. However, the output data does not follow exactly the phase of the original data.

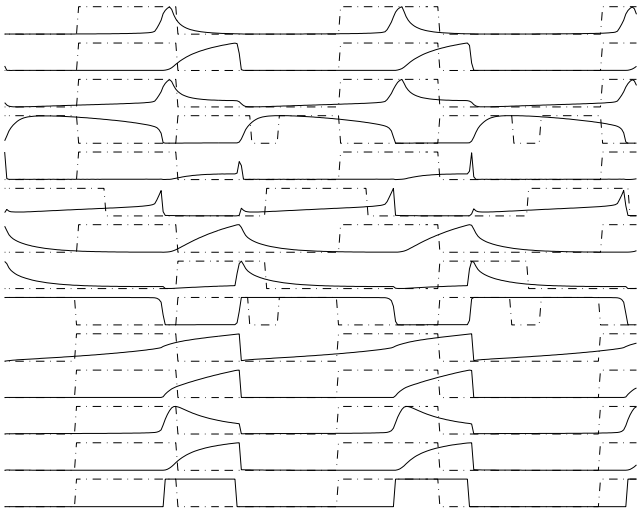

Figure 6: BDN output data using the clustering method (iv). It can be clearly seen that the phase of the output data is different than in the original data. This is the reason why the numerical results are poor. However, the output data follows approximately the shape of the original data.

rate of sampling has been achieved at least two things should be taken into account in the following analysis. In this paper we have been looking for methods to help in this newly-emerging problem area.

First of all, too frequent a uniform sampling creates data that might be out of reach of the modeling capabilities of BNs even though with a proper sampling the modeling would still be possible. The increased sampling rate should be turned into an advantage as a means to cancel out noise in the data, not to increase the amount of data used for BN modeling. The additional data needed for modeling even a close to unique network should consist of e.g. different types of perturbation tests and measurements from cells living in different types of conditions.

Secondly, for the discrete-timestep BN model the states that are interesting might not at all be seen in the data sampled uniformly with a reasonable rate. The timescales of variations in the expression levels are not constant in time and thus it makes sense to make a Monotonous Time Transformation (MTT) for the frequently sampled data as proposed in this paper. In this way, the states of the system that are the most significant can be extracted as states of the resulting BN. The results show that making a suitable transformation improves the results as compared with uniform sampling of any kind.

Boolean Delay Networks were presented as another application of the MTTs. Their applicability to modeling cells would give interesting possibilities since their behavior is not restricted to simple cyclic attractors as with BNs. Their stability properties and inference calls for further attention, however, in order to make further conclusions from even the results presented here.

Currently at least the publicly available data suffers from a high noise level, which

also is to some degree unknown. Until the measurements are made more accurate and also cheaper so that the sampling rates can be increased, the methods as proposed here are not feasible for such real-world data. To use the methods an interpolation to close to continuous time is needed. Noise and low sampling rates make this currently impossible in practice. The simulations, however, give a reassuring background for examining the models used and also for further studies. *Saccharomyces cerevisiae* (budding yeast) is a good target for future experiments with real data since there exists a lot of biological information on its regulatory networks and many results can be verified against this knowledge (Lee et al. 2002). The simulation model used to generate the data is an example of the available level of this knowledge.

References

[1] Akutsu, T., Miyano, S. and Kuhara, S. (1999) Identification of genetic networks from a small number of gene expression patterns under the Boolean network model, in *Pacific Symp. on Biocomputing*, 17–28.

[2] Aldana, M., Coppersmith, S. and Kadanoff, L. P. (2002) Boolean Dynamics with Random Couplings, submitted to *Springer Applied Mathematical Sciences Series*.

[3] Chen, K. et al. (2000) Kinetic Analysis of a Molecular Model of the Budding Yeast Cell Cycle, in *Mol. Biol. Cell*, 11, 369–391.

[4] Kauffman, S. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford Univ. Press, New York.

[5] Lee, T. I. et al. (2002) Transcriptional Regulatory Networks in Saccharomyces cerevisiae, in *Science*, 298, 799–804.

[6] Öktem, H. et al. (2002) A Computational Model for Simulating Continuous Time Boolean Networks, in *GENSIPS 2002*.

[7] Shmulevich, I., Yli-Harja, O. and Astola, J. (2001) Inference of Genetic Regulatory Networks Under the Best-Fit Extension Paradigm, in *IEEE Workshop on Nonlinear Signal and Image Proc.*

[8] Spellman, P. et al. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization, in *Molec. Biol. Cell*, 9, 3273–3297.