Learning transition models of biological regulatory and signaling networks from noisy data

Deepika Vatsa
Department of Electrical Engineering,
Indian Institute of Technology Delhi
deepika@ee.iitd.ac.in

Sumeet Agarwal
Department of Electrical Engineering,
Indian Institute of Technology Delhi
sumeet@iitd.ac.in

Ashwin Srinivasan
Birla Institute of Technology and Science
Goa Campus
ashwin@goa.bits-pilani.ac.in

ABSTRACT

In this paper, we present an extended 2-step probabilistic LGTS (PLGTS) transition system which aims to identify the network structure and stochastic nature of biological processes using time series data. This work is a step towards system identification in a noisy environment using transition systems. Here, the noise implies noise in transitions between states in the observed data. Interestingly, noise in the data helps in assisting system identification. Experimental results on synthetic data show that noise actually helps in understanding the system dynamics as well as constraining the solution space; thus helping to identify the most probable network structure for a given data set.

CCS Concepts

•**Computing methodologies** → **Maximum likelihood modeling;** Discrete-event simulation; •**Applied computing** → **Biological networks;** •**Software and its engineering** → **Petri nets;** •**Theory of computation** → *Constraint and logic programming;*

1. INTRODUCTION

Biological networks describe the interactions among the biological entities eg. DNA, RNA, proteins and small molecules within the cell. Biochemical processes involve biochemical reactions where different reactant molecules react with one another to produce product molecules. Suitable conditions and control elements are required for the reaction to happen. Lack of suitable conditions or any intrinsic noise within the system may hamper the progress of reaction which may either lead to incomplete state of the system or no change in state of the system at all.

Efficient modeling of biological system requires us to look at the nature of biochemical processes it involves. Biochemical processes are inherently noisy, so biochemical systems need to have robust design features to deal with such noise. This suggests that considering observed data to be perfect and designing deterministic methods for network inference is likely to be inadequate and may result in incomplete and inaccurate network inference. It has been seen that network reconstruction is a challenging task [12, 5].

In this paper, we try to model the network inference task using transition systems (see section 2), keeping in mind the intrinsic noise in biochemical processes. In this work, we are dealing with synthetically generated qualitative time series data. To make the data realistic, transition noise (see section 2) is imparted in the specification prior to data generation. The proposed network reconstruction model is tested for different levels of noise and sample sizes of data sets. Along with the minimal networks consistent with the data set, reconstruction model also estimates the firing probabilities of transitions of the network.

The paper is organized as follows: Section 1 gives an introduction to biological networks, Section 2 gives background about transition systems and transition noise, Section 3 gives a brief review of the work done in this field, Section 4 discusses the data sets and network inference method, Section 5 present experimental results followed by Section 6 which include concluding remarks.

2. BACKGROUND

Transition systems Transition systems such as Petri net [1] and Logical Guarded Transition System (LGTS) [9] are very helpful in illustrating the reactions in the biological network and also aids in understanding biological system properties like concurrency, rate constants, activator/inhibitor effects etc. in a graphical manner.

Petri net is a directed bipartite graph which represents the changes in the network through a series of transitions. It consists of two types of nodes, namely places (represented as circles) and transitions (represented as squares) connected by arcs. The connection between control places (activator or inhibitor) of a biochemical network and transition is represented by special arcs (read arc or inhibition arc). Read arcs are represented as a filled circle at the end of the arc while inhibition arc as a hollow circle at the end of the arc. Just like a chemical reaction, a transition is enabled (to fire) when the input places (reactants in a chemical reaction) contains sufficient tokens (molecules in a chemical reaction). Each fired transition transfers the tokens from input places to output places, thus changing the state of the network. Petri nets have been used for network inference modelling [7, 6].

LGTS transition system [9] is basically a generalisation

CODS '16, March 13 - 16, 2016, Pune, India

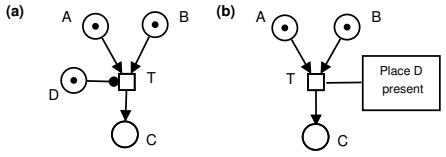
ACM ISBN 978-1-4503-4217-9/16/03…$15.00

DOI: http://dx.doi.org/10.1145/2888451.2888469

of a Petri net transition system in that it constrains the transitions using guard functions. Guard functions define the conditions which need to be fulfilled before a transition can fire. So, for instance, in the chemical reaction of H_2O formation, H_2 and O_2 must react in the presence of high energy, thus high energy becomes the guard parameter for this reaction. LGTS identifier uses logical consequence finding to identify the network structure given the state sequences. As the network structure, it identifies the transition set and the minimal set of applicable guard functions for each transition in the transition set.

Biochemical network represents the interaction among biological entities. Suppose in a biochemical reaction, reactant A reacts with reactant B to produce product C in the presence of entity D. This simple reaction can be graphically represented as Petri net and LGTS as shown in figure 1 (a) and (b).

Transition noise The term *transition noise* is introduced in [11] where it implies that a transition (in effect of noise) can fire to any state which lies within a k-bit Hamming distance of the parent state. However, in this work, the scope of transition noise is limited and the effect of noise is taken to be that transitions (with varying probabilities) can fire from any parent state where the output place is absent. Here in the presence of noise, the transition could fire to produce its output place even when its pre-state is not fully satisfied, i.e., either some input place is missing or some control place is missing. Figure 1 (c) shows complete and incomplete pre-states for a one-transition network. To model this type of transition noise, transition probabilities are assigned to all incomplete pre-states of transitions. Thus, in ideal conditions, only the complete pre-state could transition to the post-state while in stochastic conditions, a whole set of pre-states could transition to the post-state.



(c) Complete Pre state of T : [(A,1),(B,1),(C,0),(D,1)]
Complete Post state of T : [(A,0),(B,0),(C,1),(D,1)]

Incomplete pre states of T :
[(A,0),(B,0),(C,0),(D,0)],
[(A,0),(B,1),(C,0),(D,0)],
[(A,1),(B,0),(C,0),(D,0)],
[(A,0),(B,0),(C,0),(D,1)],
[(A,0),(B,1),(C,0),(D,1)],
[(A,1),(B,0),(C,0),(D,1)]

Figure 1: (a) Petri net representation, (b) LGTS representation, and (c) Complete and incomplete pre-states of one transition network in (a)

3. RELATED WORK

Some studies have attempted regulatory network inference using transition systems from time series data. Durzinsky et al. [4] proposed a Petri net model for network reconstruction. They further extended their work on gene networks containing activator and inhibitor genes. The Extended Petri net model proposed by Durzinsky et al. [3, 2] reconstructs the network as an extended Petri net (include catalytic and inhibitory genes) using time series data.

Srinivasan et al. [9, 10] proposed Logical Guarded Transition Systems (LGTS) for network reconstruction using logic programming. These are generalizations of Petri nets. They assume transitions to be constraint guards between two states of a system. A transition only fires if it satisfies all the constraints in the constraint box. Here, background knowledge can also be used which constrains the search space significantly. Thus, given a data set, they produce the transitions for a network that explains the data set.

Srinivasan et al. [11] show network structure reconstruction using transition systems on noisy data sets. Data simulation assumes the correct transitions of the network to be known and model the transition noise as a Markov process. Thus, using initial states, state space trajectories are traced for different levels of noise. Network structures are obtained using logical programming system LGTS for different noisy data sets. Then, using PRISM, the final network structure is produced using probability estimates of transition sequences. The system finally produces a probabilistic automaton structure. Our work in this paper follows a different approach, though based on similar concepts. Our work allows concurrent transition firing at a single time point in the network. The scope of transition noise forms the basis of the difference in the two approaches. In this work, transition noise effects the network at a local level (i.e. at a particular transition) while in [11], transition noise effects the network globally.

4. MATERIALS AND METHODS

4.1 Data set

For our model implementation, we have experimented with networks which involve single output transitions, multiple output transitions, control places etc.

Networks considered are as follows:

- MAPK cascade pathway

  MAPK (Mitogen activated protein kinases) is a central signaling pathway that is used in cell tissues to communicate extra cellular events to the nucleus [10]. Initiation of the pathway happens when a protein binds to a receptor protein at the cell membrane. This triggers a chain of phosphorylation reaction in a cascade fashion. Each phosphorylated protein acts as a switch for phosphorylation of another protein. In this pathway, three proteins, namely MAP4K, MAP3KP and MAP2KPP act as the switch. The Petri net structure of MAPK cascade pathway is shown in figure 2 (a).

  Time series data set for MAPK cascade pathway is taken from Ref. [10]. Data set is a state matrix representing place value at each state. A total of 3 simulations were done to obtain this data set. All the simulations put together to give a total of 14 state vectors. All transitions in the network are single output transitions.

- Phosphate regulatory network in *E.Coli*

  The phosphate regulatory network in *E.Coli* is a network of phosphate-sensing proteins. The proteins in the network control the phosphate level by regulating

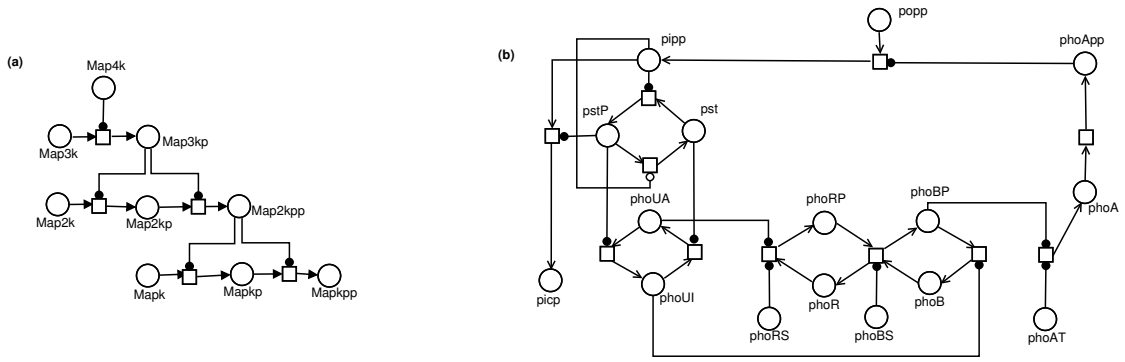


Figure 2: Petrinet model of (a) MAPK cascade pathway [9] and (b) phosphate regulatory network [3]

gene expression. A Petri net model of the phosphate regulatory network [3] is shown in Figure 2 (b).

Time series data set for phosphate regulatory network is taken from Ref. [3]. Data set is obtained from 11 simulations and the state matrix of all simulations contain a total of 46 state vectors. The data set contains single as well as multiple output transitions.

4.2 Network inference model

For network inference, we have proposed a Probabilistic LGTS (PLGTS) system (extension to the LGTS system) that select the most probable network structure given the observed data set. We have also implemented a data simulator for data generation in PRISM [8]. PRISM (Programming in Statistical Modelling) is a programming language for modelling complex systems using rules and probabilities. The data set is then used by PLGTS to reconstruct network and quantify noise. We also compared the performance of LGTS and PLGTS on noiseless and noisy data sets.

Figure 3 shows the structure of a PLGTS transition system. PLGTS system has two main steps: system identification and noise quantification. System identification step takes the data set and returns the most probable set of transitions for the concerned network. Noise quantification step estimates the firing probability of transitions and also helps in reducing the number of networks (in case of networks involving control places). This transition system allows multiple independent transitions firing at a single time point.

Data set generation The data simulator takes as input the network specification to generate a data set. Specification includes initial states of the network with occurrence probabilities and state transition probabilities for all transition sets. Simulator picks an initial state based on its occurrence probability. Then based on the firing probability of the pre-places of transition sets, next state of the network is generated. Simulator stops generating data when it reaches a point where the next state generated is same as the current state. In noiseless data set case, the end state of the data is taken as the terminal state. While in noisy data set case, instead of considering the whole set of end states of sampled data sets as a terminal state set, a procedure is applied to produce the terminal state set. In this procedure, each end state of the sampled data is assigned a terminal state probability *P(Tr)* which is computed as :

$$P(Tr) = \frac{number\ of\ times\ end\ state\ fired\ in\ data}{number\ of\ times\ end\ state\ occurred\ in\ data}$$

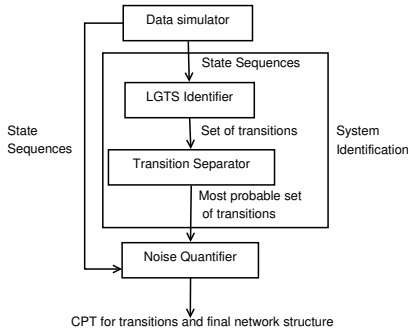


Figure 3: Structure of PLGTS transition system

A terminal state threshold *Thresh(Tr)* is chosen empirically which filter the end states to produce terminal state set. Any end state whose *P(Tr)* is less than *Thresh(Tr)* is selected as the terminal state.

System Identification After the data set and the terminal state set is obtained, LGTS identifier identifies the set of transitions consistent with the data. Since the data can be noisy, this transition set produced by LGTS identifier may contain noisy transitions as well. This set of transitions is thus filtered using a transition separator unit which produces the most probable transition set that explain the observed data set. Transition separator unit separates the correct transitions from noisy transitions by first decomposing multiple output transitions into single output transitions and then ranking the set of transitions for each output place in decreasing order using conditional probability measure. The probability of each transition is computed conditioned on its parent state. Highly ranked transition for each output place is selected and these selected transitions form the transition set for the network.

The procedure for system identification task is given as Algorithm 1:

Noise Quantification Using the final transition set and observed state sequences, the noise quantifier unit generates the Conditional Probability Tables (CPTs) for transitions, thus modelling the stochastic nature of the network. Here, for each transition, first find all (pre-state, post-state) pairs from observed data. For each pre-state Pr_S, generate a set of post-states it fired to $[Po\_S_1, Po\_S_2, .., Po\_S_n]$. Firing probability from pre-state Pr_S to post-state Po_S_k is computed as $(C(Pr\_S \rightarrow Po\_S_k)/C(Pr\_S))$ for $1 \leq k \leq n$ where $C(Pr\_S \rightarrow Po\_S_k)$ denotes number of times Pr_S

Algorithm 1: findtransitionset(S,TS)

Data: Observed set of state sequences S
Result: Most probable transition set TS that explains S
Given S, function $lgts(S,T)$ returns set of transition T where $T = (T_s \cup T_m)$; T_s and T_m denotes single output place transition and multi output place transition
Generate output place set O from T_s
Decompose T_m using T_s and generate decomposed transition set T_D
Update output place set O using T_D
Group transitions in T_D according to common output place s.t. for set $O = \{O_1, O_2, ..., O_n\}$ we have corresponding set $T^{'} = \{T_1^{'}, T_2^{'}, ..., T_n^{'}\}$ where $T_k^{'} \subset T^{'}$ for $1 < k < n$ and $T_k^{'}$ contains output place O_k
Set $TS = \emptyset$
for *each* O_k *in* O **do**
    Rank $T_{kr}^{'}$ according to $P(C(T_{kr}^{'})|C(Par(T_{kr}^{'})))$ where $T_{kr}^{'} \in T_k^{'}$ for $1 < r < size(T_k^{'})$, $C(x)$ denotes count of x in S and $Par(x)$ denotes Parent state of x
    Select $T_{kr}^{'}$ with maximum $P(C(T_{kr}^{'})|C(Par(T_{kr}^{'})))$
    Update $TS = TS \cup T_{kr}^{'}$
Done

fired to Po_S_k and $C(Pr\_S)$ denotes number of times Pr_S occurred in the data. Finally, a CPT is generated for each transition dictating the firing probability of complete and incomplete pre-states. CPTs efficiently capture the entire dynamics of the system, as inferred from the observed data.

5. EXPERIMENTAL RESULTS

In this work, we have investigated the performance of LGTS and PLGTS models on noiseless and noisy data sets. For experiments on noisy data sets, different noise levels from low to high (1% to 20%) are introduced in the noiseless data set. Here, x% noise introduction implies assigning x% probability to incomplete pre-states to fire to the post-state while complete states can fire with a probability of 1-x%. For noisy data set experiments, 10 experimental realizations are done for each sample size, and the most consistent network structure is recorded as the final result.

CASE 1: MAPK cascade pathway

In this case, no background knowledge is used. The work is categorised into four experiments as follows:

- **Experiment 1: LGTS model on noiseless MAPK data set**

  In this experiment, the LGTS model could discover the correct network structure (i.e., correct transition set and control places) along with some incorrect control places for transitions. Thus, in total, 12 networks were found to be consistent with the data set. Table 1 shows the control places found for each transition (also see supplementary figure 1 online). Highlighted control places are the incorrect control places found.

- **Experiment 2: LGTS model on a noisy MAPK data set**

  In this experiment, 1% noisy data set is fed to the LGTS model for network structure identification. It is found that some spurious transitions are also identified along with the correct transitions present in the original network. Thus, a single transition set for the network could not be found in this case. For network identification using LGTS system solely on the basis of the data set generated in the presence of noise, it would be difficult to select the most likely transitions.

MAPK cascade pathway		**LGTS**	**PLGTS**
Transitions	**Control places**	**Control places**	**Control places**
map3k, map3kp	Read : map4k	Read : map4k, map2k, mapk	Read : map4k, map2k, mapk
map2k, map2kp	Read : map3kp	Read : mapk, map3kp	Read : map3kp
map2kp, map2kpp	Read : map3kp	Read : mapk, map3kp	Read : mapk, map3kp
mapk, mapkp	Read : map2kpp	Read : map2kpp	Read : map2kpp
mapkp, mapkpp	Read : map2kpp	Read : map2kpp	Read : map2kpp
Total number of networks:		3x2x2 = 12	3x2 = 6

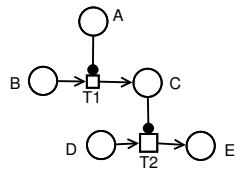
Table 1: Performance of LGTS and PLGTS model on noiseless MAPK data set

Phosphate regulatory network		**LGTS**	**PLGTS**
Transitions	**Control places**	**Control places**	**Control places**
pst, pstP	Read : pipp	Read : pipp	Read : pipp
pstP, pst	Inh : pipp	Inh : pipp, phoApp	Inh : pipp
pipp, picp	Read : pstP	Read : pstP	Read : pstP
phoUA, phoUI	Read : pstP	Read : pipp, pstP, Inh : pst	Read : pstP, Inh : pst
phoUI, phoUA	Read : pst	Read : pst	Read : pst
phoR, phoRP	DoubleRead : (phoUA,PhoRS)	DoubleRead : (phoUA, PhoRS)	DoubleRead : (phoUA, PhoRS)
phoRP, phoR, phoB, phoBP	Read : phoBS	Read : phoBS	Read : phoBS
phoBP, phoB	Read : phoUI	Read : pipp, pstP, phoUI, Inh : pst, phoUA	Read : phoUI, Inh: phoUA
phoA	DoubleRead : (phoBP, phoAT)	DoubleRead : (phoBP, phoAT)	DoubleRead : (phoBP, phoAT)
phoA, phoApp	Anonymous	Anonymous	Anonymous
popp, pipp	Read : phoApp	Read : phoApp	Read : phoApp
Total number of networks:		2x3x5 = 30	2x2 = 4

Table 2: Performance of LGTS and PLGTS model on noiseless Phosphate regulatory network data set

- **Experiment 3: Probabilistic LGTS (PLGTS) model on a noiseless MAPK data set**

  Here, PLGTS is fed with a noiseless MAPK data set. As per expectation, system identification unit in this model also finds the same network structure (i.e., transition set and control places) found by the LGTS model in Experiment 1. However, the CPTs generated for transitions in the noise quantifier unit gives an indication of more likely control places for the transition on the basis of the firing probabilities of all possible combinations of control places (found for a transition) and particular transition. Thus, after ruling out less likely control places, the number of networks in this case are reduced to 6. These 6 networks include the correct MAPK network. Table 1 shows the final control places found for each identified transition. Thus, it can be seen here that the probability estimation step aids



(a) Suppose, control places found for transition T1 are A and D.
Now, let the state sequence of the network is :
S1 : [(A,0),(B,1),(C,0),(D,1),(E,0)]
S2 : [(A,0),(B,1),(C,0),(D,0),(E,1)]
Here, transition T1 does not fire since its control place A is OFF.
Thus, the firing probability of transition T1 in presence of control place D will become low and D will be less probable control place for transition T1.

(b) In unnoisy environment, let the state sequence S for this two-transition network is:
S1 : [(A,1), (B,1), (C,0), (D,1), (E,0)]
S2: [(A,1), (B,0), (C,1), (D,1), (E,0)]
S3 : [(A,1), (B,0), (C,1), (D,0), (E,1)]
State difference D1 : [(A,0),(B,-1),(C,1),(D,0),(E,0)]
State difference D2 : [(A,0),(B,0),(C,0),(D,-1),(D,1)]
Let the transitions found for this data set be:
Tr 1 : [(Read (A), Read(D))] [(A,0), (B,-1), (C,1), (D,0), (E,0)]
Tr 2 : [Read (C)] [(A,0), (B,0), (C,0), (D,-1), (E,1)]

In noisy environment, suppose transition T2 fires prior to T1 and state sequence be like:
S1 : [(A,1), (B,1), (C,0), (D,1), (E,0)]
S2: [(A,1), (B,1), (C,0), (D,0), (E,1)]
S3 : [(A,1), (B,0), (C,1), (D,0), (E,1)]
State difference D1 : [(A,0),(B,0),(C,0),(D,-1),(E,1)]
State difference D2 : [(A,0),(B,-1),(C,1),(D,0),(D,0)]
In this case, transition T2 fired prior to T1 even in the absence of its control place C. Here, unlike in above case, control place for transition T1 which results from difference of states S2 and S3 will be Read(A) only, since value of place D in state S2 is 0, so, it cannot be a read control place for transition T1.

Figure 4: Example explaining possible cases that helps in reducing number of networks

in selecting more likely control places for each transition and finally help in providing the more probable network structures.

- **Experiment 4: Probabilistic LGTS (PLGTS) model on noisy MAPK data set**

  This experiment explores the performance of the Probabilistic LGTS model on noisy data. For each noise level, the model could finally found 6 networks except in the case of sample size 10 under 20% noise where 12 networks are found. The MAPK cascade pathway has a total of 5 transitions. In all cases, the correct transition set is identified. The performance of the PLGTS model on noisy MAPK network data set is shown in table 3. It is evident that larger sample size helps in identifying correct network structure for the given data set.

  Reduction in the number of networks (from 12 to 6) is achieved by ruling out less probable control places for each transition. According to observations, there are possibly two reasons for the low firing probability of transitions in the presence of incorrect control places: firstly, deactivation of incorrect control place due to noise and secondly, disabled transition when the correct control place is OFF while the incorrect control place is ON. Figure 4 shows an example of reduction in the number of networks for both the cases. Thus, it is interesting to note here that noise is actually assisting in ruling out the less likely network structures consistent with the observed data set.

CASE 2: Phosphate regulatory network

Here in all experiments, as background knowledge, information on possible activators and inhibitors is used [1]. Here also, the work is categorised into four experiments as follows:

- **Experiment 1: LGTS model on noiseless Phosphate regulatory network data set**

  In this experiment, it has been seen that LGTS model could recover the correct network structure (i.e., correct transition set and control places) along with some incorrect control places for transitions. A total of 30 networks are found to be consistent with the data set. Table 2 shows the control places found for each transition (also see supplementary figure 2 online). Highlighted control places are the incorrect control places found.

- **Experiment 2: LGTS model on noisy Phosphate regulatory network data set**

  In this experiment, a 1% noisy data set is fed to the LGTS model. It is found that the correct transition set is not found for the network. Also, some spurious transitions are identified. Thus, in the noisy data case, LGTS model performs poorly.

- **Experiment 3: Probabilistic LGTS (PLGTS) model on noiseless Phosphate regulatory network data set**

  In this experiment, the PLGTS model could discover the correct network structure (i.e., the correct transition set and control places) along with some incorrect control places for transitions. A total of 4 networks are found for the given data set. Table 2 shows the control places found for each transition. It can be seen that the number of candidate networks is significantly reduced with the PLGTS model.

MAPK Cascade Pathway												
Noise level	**Sample size**											
	10			**25**			**50**			**100**		
1%	6 / 12	Y	5	6 / 12	Y	5	6 / 12	Y	5	6 / 12	Y	5
5%	6 / 12	Y	5	6 / 12	Y	5	6 / 12	Y	5	6 / 12	Y	5
10%	6 / 12	N	4	6 / 12	Y	5	6 / 12	N	4	6 / 12	Y	5
20%	12 / 12	N	3	6 / 12	N	4	6 / 12	Y	5	6 / 12	Y	5

Phosphate Regulatory Network												
Noise level	**Sample size**											
	25			**50**			**100**			**125**		
1%	4 / 45	N	9	8 / 90	N	10	8 / 90	N	10	4 / 45	N	10
5%	4 / 45	N	10	4 / 45	Y	11	4 / 45	Y	11	4 / 45	Y	11
10%	4 / 90	N	10	4 / 15	Y	11	4 / 15	Y	11	4 / 15	N	9
20%	4 / 16	N	8	4 / 10	N	8	4 / 12	N	9	4 / 8	N	7

Table 3: Performance of PLGTS on MAPK cascade network and Phosphate regulatory network for different noise levels and sample sizes. Each cell in the table has 3 entries. First entry shows the number of networks obtained. As x / y representation here, y denotes number of networks obtained after system identification step while x denotes final number of networks after noise quantification step. Second entry shows whether the correct network is being identified (by *Y*) or not (by *N*). Last entry shows the number of correctly obtained control places.

- **Experiment 4: Probabilistic LGTS (PLGTS) model on noisy Phosphate regulatory network data set**

  In this experiment, PLGTS could discover a total of 4 networks for most of the cases. Table 3 shows the

[1] Places picp, popp, phoR, phoRP, and phoA are not included in possible activators list [3]

results of PLGTS on different noise levels and sample sizes. The phosphate regulatory network has a total of 11 transitions. For all noise levels, correct transition set is identified. However, in case of noise level 20% and sample size 25, one spurious transition is also found along with the correct transition set. Although more number of correct control places are identified for sample size 100. Thus, larger sample sizes help in identifying correct network structure in noisy environment. Here also, the number of candidate networks is significantly reduced (from 45 to 4 in most cases) for the observed data set. Thus, the performance of the PLGTS model seems good in that it could find the correct transition set and most of the correct control places for all noise levels.

6. CONCLUSION

Here we investigate the identification of transition systems on biological networks using noiseless and noisy data sets. We have looked into one aspect of intrinsic noise in the biochemical network where it effect the reaction occurring probability of complete and incomplete pre-states of the network. Extending the LGTS system, we have described a new probabilistic LGTS transition system (PLGTS) whose two-step approach first identifies the most consistent network structures given the observed data set using the logic programming as well as conditional probability. And in the second step, it quantifies the noise in the data by parameter estimation. Experimental results on MAPK and Phosphate regulatory network data sets show that the PLGTS system yields promising results on both noiseless as well as noisy data sets in terms of the number of networks obtained.

In terms of network reconstruction from noisy data sets, this work is a variant of the work presented in Ref. [11]. While the baseline work in Ref. [11] and in this paper relies on the same concept, both follow different approaches to reconstruct the network structure. Ref. [11] follows a logic programming approach and uses PRISM for finding the network transitions and a probabilistic automaton structure featuring the firing probability from one state to the next state in case of noise. While in this work, in addition to finding the most probable network structure given the noisy data set, our model also focuses on finding the probabilities of transition sets (for different pre-states / parent states), thus describing the noise effects in local portions of the network. This work gives the deeper insight of how network dynamics change in the presence of noise and shows that noise actually assist in constraining the solution space by finding more probable network structures.

The approach followed in this work can apparently be represented as a Dynamic Bayesian Network (DBN) where places in the network at the time *t+1* shows a conditional distribution given the places in the network at the time *t*. The connection between the Petri net and DBN is shown in figure 5. Although we will be examining the precise connection between Petri net and other probabilistic graphical models such as DBNs in future work. For some more details on this, see supplementary section 2 online.

Also, it would be interesting to test the performance of this approach on other large and complex biological and signaling systems. We will also try to extend the system mechanism to work on real biological data sets.

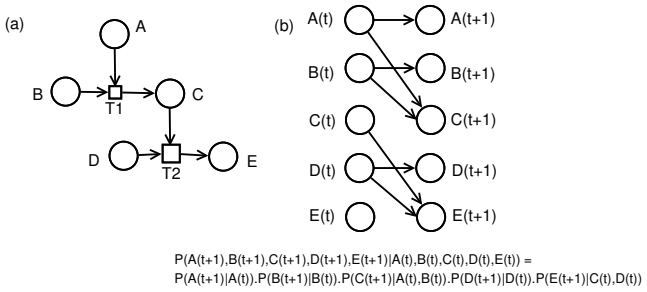


Figure 5: Connection between (a) Petri net and (b) DBN. Input places of transitions in Petri net become parent nodes in DBN. Nodes involved in conditional distribution in DBN are shown as solid edges.

7. SUPPLEMENT

Supplementary material is available at url: http://web.iitd.ac.in/~sumeet/Supplementary_Material_CoDS_2016.pdf.

8. REFERENCES

[1] R. David and H. Alla. *Discrete, Continuous, and Hybrid Petri Nets*. Springer Berlin Heidelberg, 2nd edition, 2010.

[2] M. Durzinsky, W. Marwan, and A. Wagler. Reconstruction of extended petri nets from time-series data by using logical control functions. *Journal of Mathematical Biology*, 66(1):203–223, 2013.

[3] M. Durzinsky, A. Wagler, and W. Marwan. Reconstruction of extended petri nets from time series data and its application to signal transduction and to gene regulatory networks. *BMC Systems Biology*, 5(1), 2011.

[4] M. Durzinsky, A. Wagler, and R. Weismantel. A combinatorial approach to reconstruct petri nets from experimental data. In *Computational Methods in Systems Biology*, volume 5307 of *Lecture Notes in Computer Science*, pages 328–346. Springer Berlin Heidelberg, 2008.

[5] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, 107(14):6286–6291, 2010.

[6] M. Mayo. Learning petri net models of non-linear gene interactions. *Biosystems*, 82(1):74–82, 2005.

[7] J. H. Moore and L. W. Hahn. Petri net modeling of high-order genetic systems using grammatical evolution. *Bio Systems*, 72(1-2):177–186, Nov 2003.

[8] T. Sato and Y. Kameya. Prism: a language for symbolic-statistical modeling. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 1330–1335, 1997.

[9] A. Srinivasan and M. Bain. Knowledge-guided identification of petri net models of large biological systems. In *Inductive Logic Programming*, volume 7207 of *Lecture Notes in Computer Science*, pages 317–331. Springer Berlin Heidelberg, 2012.

[10] A. Srinivasan and M. Bain. Identification of transition-based models of biological systems using logic programming. Technical report, The University of New South Wales, Technical Report, 2014.

[11] A. Srinivasan, M. Bain, D. Vatsa, and S. Agarwal. Identification of transition models of biological systems in the presence of transition noise. In 25^{th} *International Conference on Inductive Logic Programming*, August 2015.

[12] J. Stark, D. Brewer, M. Barenco, D. Tomescu, R. Callard, and M. Hubank. Reconstructing gene networks: what are the limits? *Biochemical Society Transactions*, 31(6):1519–1525, 2003.