Identification of transition based models for gene regulatory networks

Research Proposal
By
DEEPIKA VATSA
Entry number : 2013EEZ8262

Supervisor
Dr. Sumeet Agarwal

Department of Electrical Engineering
Indian Institute of Technology, Delhi

ABSTRACT

Sub-cellular gene/protein networks are highly complex and stochastic. In the post genomic era, one of the challenges is to use high throughput data to predict models for these biological networks. Among several issues concerning this prediction of gene regulatory networks is noise issue which can be extrinsic as well as intrinsic. The model for biological network should be stochastic in nature to deal with noise in data. Other issue is limited data availability for some networks. In these cases, use of background knowledge about the system would aid in accurately determining the network. Background knowledge also helps in cases of missing data for some places. Although, even in cases of good amount of data availability also, background knowledge provides robustness to the model. Objective of this proposal is to generate models addressing these two issues. Thus, proposed research will help understanding the biological system in more detailed and organised way.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

1.1 SUBCELLULAR BIOLOGICAL SYSTEMS

Biological system is a complex network of biological entities working together. At smaller scale, biological systems are cells, organelles, regulatory pathways etc. All these systems work together to perform a common function. Individually, each one is a complex control system in itself. To understand these systems, one must understand the underlying mechanics of how these systems work.
With the advancement in genome sequencing, today we have large volume of data on subcellular biological systems in the form of gene expression data, protein protein interaction data etc. Using mathematical models for these data can help us retrieve meaningful information on dynamics of these systems.

1.2 MOTIVATION

Uncovering subcellular networks helps us understand how cells work out its various functions. Understanding the dynamics of GRNs can also helps vastly in drug and medicine field [9]. Clear picture of disease affected networks and original networks helps in pin pointing the affected area and thus reduce time and efforts in drug development [18].

1.3 BACKGROUND

1.3.1 *Systems Biology*

Systems biology is a branch that deals with describing relationship among elements in a biological system. Systems biology gathers information about these elements by systematically perturbing the biological system and uses this information to generate predictive mathematical model of the system [31]. Technologies like Microarrays, high throughput proteomics helps in analyzing the response to perturbation to assess systems properties. Thus using these models, Systems biology explains complex biological system. Systems biology with other disciplines of science like mathematics, computer science can address different problems in human biology and medicine [10].

1.3.2 *Gene Regulatory Network*

A gene is the basic unit of heredity in a living organism. Gene expression is the process by which the information loaded in the gene is used in the synthesis of proteins. In gene regulatory network, genes interact with one another and other substances to govern the gene expression levels of mRNA and proteins [5]. A gene regulatory network is shown in figure 1. Gene regulatory networks have an important role in every process of life, including cell differentiation, metabolism, the cell cycle and signal transduction [33]. Understanding the dynamics of these networks can reveal easy target in network, breakdown of pathways which lead to a disease, behavior of network if some part break down.

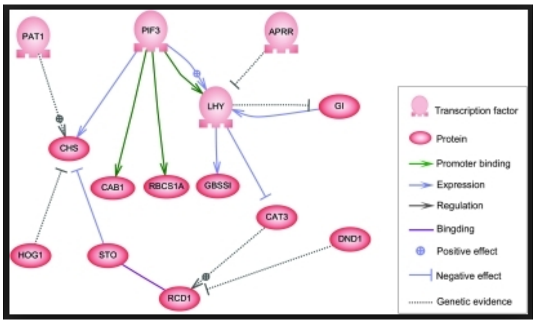


Figure 1: Gene Regulatory Network [5]

1.3.3 *Mathematical modelling techniques*

Mathematical models helps in capturing underlying mechanism of a regulatory network. Several available methods for reconstruction of regulatory networks are: Boolean network, Bayesian network, Petri nets, Relevance networks, ODE, GGM.

Boolean network

Boolean regulatory network was introduced by Kauffman in 1969 [3]. Boolean regulatory network consist of nodes connected to each other. Nodes correspond to genes and connections between genes indicate regulatory relationship between genes. Edge from node A to node B indicates gene A regulates gene B. Node can have either of the two values: 0(OFF) or 1(ON). Value of all nodes at any time point indicates the systems state at that time instant. Value of each node at next time point is determined by values of nodes at previous time point using a Boolean function. Boolean network are suitable for networks of small size since they can be computationally expensive for large networks as the number of states is exponential in number of nodes.

Bayesian network

Bayesian network was introduced by Friedman et al as a tool for identifying regulatory genes from expression data in 2000 [24]. A Bayesian network is a probabilistic graphical model (statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG) [2].

Petri net

Petri nets were invented in 1939 by Carl Adam Petri [6]. Petri nets, also known as Place/transition net are weighted, directed, bipartite graphs consisting of nodes and arcs. Nodes represent transitions and places. Arcs connect place to transition and vice versa. Petri net model represent biochemical reactions. Places contain tokens. When a transition fires, token flow occur from input places to output places. When there are multiple transitions present within places, any one of them can fire, thus firing is non deterministic. Extended Petri nets include two more arcs namely test arc and inhibitory arc. These arcs represent catalytic and inhibitory actions. Extended Petri nets represent biochemical reactions in more clear and concise manner.

Relevance network

Relevance network use information theory to generate graph from expression data. Relevance network computes pair-wise similarity score for all pair of genes in a dataset. Mutual information and Pearson coefficients are appropriate similarity scores for this method. Computationally economical to implement, but less efficient.

Ordinary differential equations (ODEs)

Ordinary differential equations provide a detailed model of regulatory network. Each differential equation describes the change in value of one node of network as a function of values of other nodes. For ODE, an analytical solution can be formulated and the resulting set of algebraic equations then describes the change in node value over time [33]. Popular solutions of ODEs are Hill function and Michaelis Menten functions.

Graphical Gaussian Model(GGM)

Graphical Gaussian models are also known as covariance selection or concentration graphs. Here, partial correlation coefficient is used as a measure of computing independence between genes. GGM are preferred over Relevance networks since partial coefficients provide a strong measure of dependence than independence [4].

1.4 OBJECTIVE AND SCOPE OF WORK

Objective of this work is to develop approaches which helps in understanding the dynamics of subcellular biological systems. In order to achieve this goal, following are the sub-goals I will be looking at:

- Developing predictive models for biological systems that incorporate background knowledge (of the concerned biological system) in addition to data.
- To estimate the effect of noise on prediction of systems and developing probabilistic models.
- Extending models for the inference of large scale systems.

2

LITERATURE REVIEW

2.1 REVIEW

In this chapter, related work is shown which forms the background of the research proposal.

In genomic revolution, with the advancement in high-throughput techniques like Microarrays, ChIP-ChIP technique and ChIP-seq technique, we now have large amount of information on genes in the form of gene expression data, protein-protein interaction data, protein-DNA data etc. However, this information is necessary but not sufficient to understand gene regulation. Thus, mathematical modelling techniques are developed to model gene regulatory networks using this data. Ultimate goal of these approaches is to understand the dynamic behaviour of gene interactions.

In the last decade, many researchers explored this field and come up with different models for GRN inference. Each model requires specific type of experimental data. Experimental data can be steady state (at particular experimental condition) or time series data (over range of time). Steady state data helps in revealing system's structure while time series data helps in capturing systems' structure as well as its dynamic behaviour. Models have been developed for both types of data. Before the actual reconstruction procedure starts, pre-processing and normalization of gene expression data is done. These two form the important steps in the process as more appropriate and informative input data helps in efficient reconstruction of GRN. Pre-processing and normalization helps in removing systemic errors and redundancy in data. LOWESS normalization [48] and quantile normalization [8] are widely used methods for normalizing Microarray generated gene expression data. These steps are essential while dealing with large scale data.

Majority of study of reverse engineering GRN revolves around four modelling techniques:

1. Boolean network

2. Bayesian network

3. Systems of equations

4. Relevance networks

Some reviews highlight their properties and inference methods [50, 13, 32].

State of the art modelling techniques:

1. Boolean network

Boolean networks infer GRN using Boolean logic. This model takes discretized Boolean values for genes i.e. 0 (gene is inactive) or 1 (active gene). Nodes in Boolean network correspond to genes and connections between genes indicate regulatory relationship between genes. Interactions between genes are described by Boolean functions. Boolean functions use logic operators AND ($\wedge$), OR ($\vee$) and NOT ($\neg$). Boolean networks are dynamic models, thus, they use time-series data. Value of each gene at next time point is determined by values of genes at previous time point using a Boolean function. Thus, the aim is to find Boolean functions for each gene using their discretized expression values.

Boolean networks are simplified networks as they deal with discretized values. But whether discretization of gene expression values can extract the meaningful biological information from data without much information loss. This is proved in [53]. Also, binary approach in Boolean network makes it noise resilience and computationally efficient [53]. Boolean networks are suitable for networks of small size since they can be computationally expensive for large networks as the number of states is exponential in number of nodes. Thus, Boolean networks are suitable for coarse-scale qualitative modelling approach and not for fine-scaled quantitative approach.

Some algorithms exist to infer Boolean network like REVEAL, restricted Boolean network model, probabilistic Boolean network model etc. REVEAL is proposed by Liang et al [37]. This algorithm uses mutual information of genes to determine regulatory relationship among them and finally extract minimal Boolean network. It works well if the indegree value of genes is smaller. Also, this method is computationally expensive for large networks as the number of states is exponential in number of nodes. This problem of exponential number of states can be solved using restricted Boolean network model proposed by Higa et al [30]. This method aims to find most promising regulatory relationships (up-regulation and down-regulation) among genes using time series data set and restricted Boolean network model. Restricted Boolean network model is restricted in the sense that it restricts the number of Boolean functions allowed to explain the network. They proposed three rules by analyzing time series data to produce constraint set which explains the regulatory relationships among genes. Constraint set is produced by analyzing perturbations in states in time series data. Regulatory connections with high frequency are selected as highly confident regulatory connections. However, this method is highly sensitive to intrinsic noise as well as extrinsic noise as it is likely to change the constraints in constraint set. To deal with noise, Ouyang et al [47] proposed an algorithm which is an extension to the one proposed by Higa et al [30]. In this method, error is computed for each predictor set of target gene and the predictor set with min error is selected as true regulatory gene set. Also, unlike Higa et al [30] method, it uses complete time series data to infer regulatory relationship and not just consecutive states which makes it more robust to noise.

Probabilistic Boolean network (PBN) model is introduced by Shmulevich et al [52, 51]. Probabilistic Boolean network consists of finite number of Boolean network with perturbation over fixed set of variables [52]. Perturbation probability in all Boolean networks is common. Each Boolean network has its own selection probability. In each Boolean network, probabilities are assigned to potential Boolean functions (predictors) for target genes in accordance with COD (Coefficient Of Determination) [52]. Marshall et al [40] also used time series data and PBN model to infer GRN. This model takes large time series data and divide this data into subsequences. Dividing point is decided by purity function which in turn is calculated using transition counting table. For each sub-sequence, it infers a Boolean network using essential predictors with minimum cost for each gene. It assumes that Boolean networks are switched with certain switching probability. Complexity in accounting these switches demands large data set. Probabilistic approach provides a better way of modelling since the biological data is stochastic in nature and prone to external noise.

2. Bayesian network

   Bayesian network is a graphical network which represent joint probability distribution of random variables (genes). By making use of probability, they model noise and randomness of regulatory relationship and thus represent stochastic nature of genes. Bayesian network is a directed acyclic graph (DAG) where the edges represent conditional dependence between genes. Since the graph structure is of DAG, Bayesian network cannot detect feedback loops in gene network. Both static as well as time series data can be used construct Bayesian network and known as static Bayesian network and dynamic Bayesian network [25, 34] respectively. BANJO [1] is a ready-to-use tool for Bayesian network and DBN inference. Werhli et al [56] gives a method to combine data and prior knowledge to construct Bayesian network for gene inference. Ong et al [46] also used prior knowledge of operon map to restrict the construction of DBN for E.Coli tryptophan metabolism. Missal et al [43] used information theory technique to determine the mutual information between genes and chi-squared test to identify significant mutual information to construct DBN. Beal et al [14] used state space model to infer GRN based on hidden nodes and data using time series data of T cell. It uses variational Bayesian EM algorithm to update parameters of model. Acerbi et al [11] proposed a new approach named continuous time Bayesian network to infer GRN from time series data. It works with discrete valued quantities. Here, as the name suggest, variables can evolve continuously with time. Thus, it takes into account the amount of time a gene stays in particular state before switching to another state. Thus, this method helps in answering queries like, *for how long gene X have to be up-regulated to have an effect on regulation of gene Y?* [11] This method is applied on T helper 17 cell differentiation and found to be effective in inferring the regulatory mechanism.

3. Systems of equations

They provide a detailed model of inferring GRN by taking into account the concentrations of mRNA, proteins etc. Differential equation describes the change in concentration value of one node of network as a function of concentration values of other nodes. Such model are quantitative in nature and of high complexity since they use continuous expression data. Equations can be linear as well as non-linear. Non-linear equations models demand more number of parameters and experiments to fit the data. Methods exist for solving linear equations such as SVD (Singular Value Decomposition), regression etc. Li et al [36] proposed a method to infer GRN using differential equations and prior biological knowledge. SVD is used to solve differential equations. Gebert et al [26] proposed a model based on differential equations using piecewise linear equations. Bonneau [15] proposed a method called Inferelator to infer regulatory relations between genes. This method uses regression and L1 shrinkage methods on gene expression data to predict model for GRN. This method yields promising results on expression data of *Halobacterium NRC-1*.

4. Relevance network

   Relevance networks are static networks as they can infer the structure of network but not the dynamics. Here, correlated genes are identified using some similarity measure like mutual information, Pearson correlation coefficient etc and a defined threshold. Popular examples of Relevance networks are ARACNE, CLR, MRNET etc. Computationally economical method as they require less data but less efficient as it determines the similarity using pair of genes while in actual case a gene may be influenced by multiple genes. Adam Margolin et al [39] proposed an algorithm ARACNE which first uses mutual information to calculate gene-gene interaction from steady state data and then filter those interactions using DPI (data processing inequality) to remove indirect gene interactions. Major advantages of ARACNE include low computational cost, no need of discretization and no requirement of prior knowledge. ARACNE performed well on gene expression data sets for human B cells thus can be used in applications for analysis of mammalian networks. Zoppoli et al [58] used ARACNE algorithm to infer GRN from time-course expression data by measuring dependencies of genes at different time delays. Firstly it identify time point of initial change of gene expression (IcE) for each gene. It helps in finding possible regulator genes for gene g_a will be regulating gene g_b if IcE(g_a) < IcE(g_b). For all pair of genes, mutual information is calculated for different time delays. From all time delays, maximum MI is find for each gene pair. These max MI are then filtered using appropriate threshold to find directed edges if GRN. TD-ARACNE performs better than ARACNE. Faith et al [23] developed context likelihood of relatedness (CLR) algorithm for GRN inference which uses mutual information between regulators and genes and then compute statistical likelihood of each mutual information by comparing mutual information value against background distribution of mutual information values. Meyer et al [42] proposed MRNET algorithm to infer edges among genes from Microarray dataset. This method uses maximum relevance minimum redundancy (MRMR) technique. This method ranks all gene pairs according to a score which signifies maximum mutual information with other gene in pair and least mutual information with all other genes in data set. All those pair are kept whose score is above appropriate threshold. These pair of genes represent edges in the network.

5. Petri net

   Petri net is a directed graph consisting of two types of nodes, places and transitions. Petri net are well suited for modelling complex concurrent systems. They are similar to state transition diagram and also provide a visual aid to model system behaviour. Steggles et al [55] proposed a technique to construct GRN in the form of Petri net using logic minimization and Boolean rules. This method first construct a Boolean network and then translate Boolean terms into Petri net control structures. This method overcomes the problems associated with Boolean networks like lack of analysis tools and thus more clearly represent the dynamics of system's behaviour. They further extended this work in [12] for multivalued networks. Multivalued logic minimization is used to construct Boolean terms and these terms are then used to generate appropriate transition guards in Petri net. Hamed et al [28] used fuzzy logic and Petri net to deal with incomplete and noisy data. Here, the goal is to find activator-repressor-target triplets from the data. Initially, the expression values of input genes are normalized to [0,1] and then classified qualitatively into low, med and high states based on membership values. Rules are constructed based activator-repressor regulatory logic and confidence degree of each rule is decided apriori using expert experience. Then based on the truth degrees (membership value for each state possible for a gene) of input genes and confidence degree of rules, changes in expression of target gene is computed. Thus, this model fits the gene set which exhibit activator and repressor on target gene. One weakness of this model lies in the determination of truth values of input genes and confidence values of rules which are decided on expert advise. Also, this model sticks to the pair of input gene being activator-repressor and thus loose the possibility for activator-activator, repressor-repressor combinations. And, this model does not take into account the possibility of single gene controlling target gene or more than two genes controlling the target gene. Zimmer et al [35] proposed PNFL (Petri Nets with Fuzzy Logic) method for reconstructing GRN. Using fuzzy logic, it defines a rule based mechanism to model a system. Here, effect or regulatory relationship between genes is evaluated using fuzzy rules. At each time, one target gene is modified. Then PNFL simulate data and this simulated data is compared with original data using an objective function. Each move of modifying target gene is accepted or rejected using simulated annealing method. Inferior arcs are accepted with low probability. This method works well with *in silico* size 10 genes. Strength of this model lies in simulating data after each arc addition and comparing it with base data. This step prunes many incorrect arcs identified by system. Durzinsky et al [21] described an algorithm for reconstruction of extended Petri nets from simulated time series data set by finding all minimal networks that are consistent with the data set. It considers catalysis and inhibition of reactions and model them as control functions on transitions. From time series data set, difference vector set is constructed by taking difference of consecutive states. A difference vector may encode more than one transition thus, each difference vector is decomposed into reaction vectors. Minimal set of these reaction vectors is found which explains all difference vectors in the difference vector set. Then, for every transition that can also fire at terminal state, control functions are found. Control functions prevent the transition to fire at any of terminal state. Terminal state is the last state achieved at each experiment. Terminal state implies the end state of system for corresponding experiment. Thus this method reconstructs extended Petri net model for GRN inference. Strength of this method lies in control functions which describe catalysis and inhibitory events in bio-chemical processes in addition to the topology of input and output places.

Major challenge faced by network reconstruction methods are due to system noise (due to stochastic system) or extrinsic noise (noise due to discretization or measurement error). Thus, new methods are developed to deal with noisy gene data and also considers inclusion of prior knowledge. Following are few such methods.

Emad et al [22] proposed a novel algorithm CaSPIAN (Causal Subspace Pursuit for Inference and Analysis of Networks) for inferring directed edges in gene regulatory network. This method is based on compressive sensing and Granger causality techniques. Compressive sensing technique is used to infer sparse causal interactions among genes. Given y as column vector of expression profile of target gene in different experiments and as sensing matrix where each column vector denote expression profile of a gene other than target gene at a time point in different experiments. Thus, this sensing matrix comprises expression profile of all genes other than target gene at all time points. Given y and , List-SP (List-Subspace Pursuit) method finds a column vector x, where non zero entries denotes the genes that causally interact with target gene. To remove false positives from vector x, Granger causality method is used which used F-test to compare the residuals of all genes with residuals for single input gene recovered from x. If this measure is greater than significance level, that input gene is accepted else rejected from x. To deal with noisy data, white Gaussian noise is added to expression profile of genes. Value of significance level results in trade off between precision and sensitivity in presence of noise, thus changing the value of significance level appropriately helps in getting high precision even in presence of noise. Significance level chosen in exp is 0.01 to 0.05. As prior knowledge, this method used scaffold networks. This method outperforms other methods for different values of parameters.

Chang et al [17] also used compressing sensing method to infer GRN. In no noise case, network is exactly reconstructed while with noise, method gives reliable reconstruction. This method defines a refinement process to deal with hidden nodes and measurement noise in data.

Han et al [29] proposed a novel approach using Bayesian approach to infer Boolean network. Bayesian approach is used to account for un-certainities and inclusion of prior knowledge. And MCMC (Markov Chain Monte Carlo) method is used to sample from posterior distributions of network topology and Boolean functions. This gives well fitted parents sets and corresponding

Boolean functions for each node in data. Thus, this method updates network topology iteratively until the chain converge. Initially, network topology is generated randomly. For different sample size, accuracy decreases as noise level increased. This method outperforms BFE and TDBN method. However, some limitations of this method includes setting in-degree of 2 and assumption of the model to be DAG thus not taking into account feedback loops.

Yip et al [57] presented a method to learn noise model from deletion data and differential equation model from time series data and merged the prediction results of both models to learn the network. However, this model could not distinguish between direct and indirect regulation of genes.

Maetschke et al [38] compared the performance of supervised, semi-supervised and unsupervised methods for GRN inference and clearly shown how supervised methods outperformed other methods.

Mordelet et al [44] proposed SIRENE method that used SVM algorithm for GRN inference. This method takes gene expression data and regulation relationship between TF and genes as input and solve binary classification problem for each TF to return final gene network. The capability of this method lies in the strength of training data. Gillani et al [16] used SVM using different kernels for predicting GRN under different biological conditions i.e. Knockdown, knock-out and multi-factorial expression profile of simulated steady-state data set of E.Coli. Different kernels used are linear kernel, polynomial kernel, Gaussian kernel and sigmoid kernel. Experiments are evaluated on different network sizes. Also, unsupervised method CLR is applied to infer GRN. SVM outperformed unsupervised method in all conditions except multifactorial condition. In overall all cases, SVM with Gaussian kernel outperforms all other methods on network size of < 200.

Brouard et al [16] proposed a supervised learning technique for inferring GRN which in addition to experimental data uses description of genes and relationship between genes as training data. Asymmetric bagging technique is used to learn MLN (set of weighted rules) to predict regulatory interactions.

3

PROPOSED RESEARCH AGENDA

A lot of work has been reported for inferring gene regulatory networks from high-throughput data. However, uncovering regulatory networks and understanding the dynamics still remain a challenge. Some problems faced by researchers in this task are noise issue with the data, limited data availability for large number of genes etc. To cope up with noise issue, model should be robust enough to predict accurate interactions with high probability. And for cases of limited data availability, model should be augmented with prior knowledge about the system so that the network can be reconstructed even with less data.

3.1 PROBABILISTIC TRANSITION MODEL FOR NOISY DATA

Biological data can have intrinsic noise as well as extrinsic noise. Intrinsic noise comes from stochasticity in transcription or translation process while extrinsic noise is due to measurement error while recording simulation or discretization data etc. In case of noisy data, it is difficult to come up with accurate solution of network as for this model has to be robust enough to produce correct interaction with high probability.

Using this idea, we have proposed a new probabilistic model for Petri net to deal with noisy data.This model serves the purpose to deal with noisy data. Noise in the data is intrinsic noise. So we try to develop a stochastic data generation system from predefined network. Here, we deal with discretized values (0 or 1) for genes. For each transition in the network, we have state transition probabilities.

Initially, we try to model this proposed system on water example (fig 2) since it contains single transition. Table 1 below shows the state transition probabilities for water example. This table shows state transition probabilities for all possible current states. Here we are assuming the value of output place in current state (i.e. H_2O) to be 0 and only taking all possible H_2, O_2 value combinations in current state i.e. (0,0), (0,1), (1,0) and (1,1). Probable next state defines all possibilities of values of input places and output place. For example, if in current state (H_2, O_2, H_2O) is (0,1,0), so next state can be (0,0,0)(i.e. O_2 is consumed but H_2O not produced), or (0,0,1) (i.e. O_2 is consumed and H_2O produced), or (0,1,1) (i.e. O_2 not consumed but H_2O produced) or (0,1,0)(no change). Thus, in the presence of H_2 and O_2 in the current state (1,1,0), probability of next state (0,0,1) consuming H_2, O_2 and producing H_2O is the highest.

Thus, this table lists all possible state transitions (noisy and un-noisy) with probabilities. Probabilities for state transitions are assumed here. Figure 3

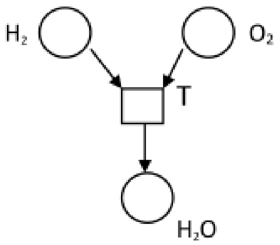
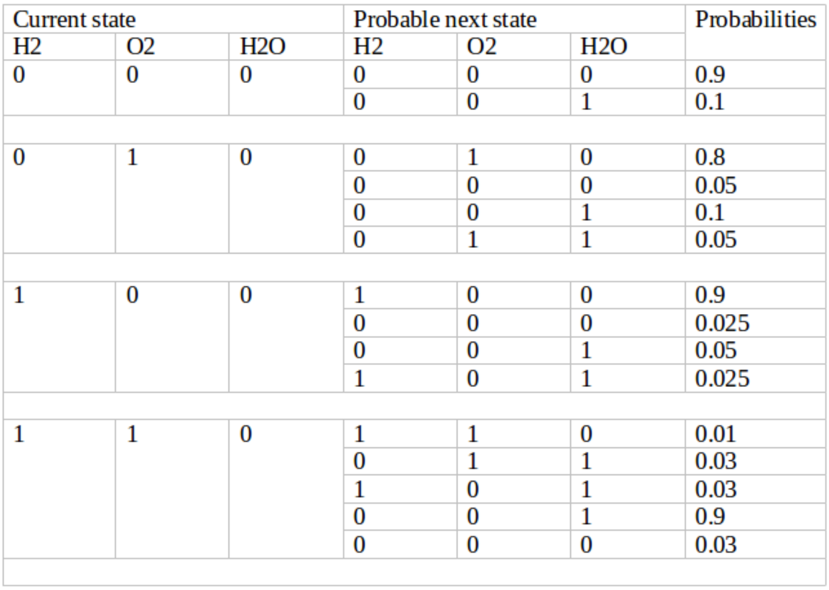


Figure 2: Transition diagram for water network



Current state			Probable next state			Probabilities
H2	O2	H2O	H2	O2	H2O	
0	0	0	0	0	0	0.9
			0	0	1	0.1
0	1	0	0	1	0	0.8
			0	0	0	0.05
			0	0	1	0.1
			0	1	1	0.05
1	0	0	1	0	0	0.9
			0	0	0	0.025
			0	0	1	0.05
			1	0	1	0.025
1	1	0	1	1	0	0.01
			0	1	1	0.03
			1	0	1	0.03
			0	0	1	0.9
			0	0	0	0.03

Table 1: State transition probabilities for water network

shows few probabilistic transitions (values taken from the table) for water example.

Based on these probabilities for state transitions, we will simulate time series data for water example using different initial states. Then these experimentally generated data are used to recover the original network using LGTS model [45] and PRISM [7]. LGTS model with the help of ILP engine helps in learning new transitions (not present in the original network) for the network. PRISM helps in dealing with probabilistic predicates for newly learned transitions and learning the structure of probabilistic transition model.

In this model, we have to change the notion of terminal states as described by Durzinsky in extended Petri net model [21]. There a state if occur more than once consecutively in time series data is taken as terminal state. But in our probabilistic transition model, there is a possibility that a state occur more than once consecutively since it has high probability of not getting change. For example, in the water network described above, we can see in table 1 that when H_2, O_2 and H_2O are 0 in current state, probability of staying in this state is high (0.9). So, next state in data will again be the same state as previous state. In data simulation, if this is the initial state, 90% states in data generated will be this state and 10% states will be the other state (H_2=0, O_2=0 and H_2)=1). In our case, terminal state will be the one which do not have any possibility of

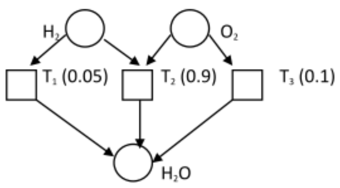


Figure 3: Probabilistic transition diagram for water network

moving further according to our probabilistic state transition table for example state (H_2=0, O_2=0 and H_2)=1). Since this state is not present in the column of 'current state' in table 1, it cannot move further and thus a terminal state.

3.2 LGTS MODEL FOR INCORPORATION OF PRIOR KNOWLEDGE

Inductive Logic Programming (ILP) [45]: It forms a connection between inductive learning and logic programming. It is a learning approach where positive examples, negative examples and background knowledge is used to make hypothesis. Generated hypothesis is such that it can explain all the mentioned positive examples without violating negative examples and background knowledge. Hypothesis are learnt first order logic rules from the facts (examples and background knowledge). Thus, ILP forms a perfect engine to introduce prior knowledge into the model for learning new interactions in inferring gene regulatory networks. Some popular ILP systems are PROGOL, FOIL, GOLEM, ALEPH etc.

Logic Guarded Transition System (LGTS): LGTS has been proposed by A. Srinivasan, M. Bain and K. Sriram [54]. It is a transition based model where the transition are the constraint guards between two states of the system. A transition only fires if it satisfies all the constraints in the constraint box. LGTS model representation is better than Petrinet model in the sense that it makes use of background knowledge which constrain the search space significantly. Also guard function provide a procedure to check control function for each transition.

PRISM (Programming in Statistical Modelling) [7] will be used for implementing above proposed model. PRISM is an extension to Prolog language. PRISM is a programming language for symbolic-statistical modelling. PRISM consists of two parts: learning part and execution part. While learning part takes care of abductive reasoning, execution part deals with probabilistic predicates.

Using above described probabilistic transition model, LGTS model and PRISM, we can have a system with provision for both background knowledge and probabilistic transitions. Each possible transition is assigned probability using probabilistic model. Using these probabilistic transition predicates, if the system learns a new transition which has less chances of occurrence, this transition is not thrown away but kept with low probability. In this way, model can deal with noisy transitions in Petri net.

For the inference of gene regulatory networks, we have used extended Petri net model proposed by Durzinsky [21] and LGTS model proposed by Ashwin Srinivasan [54].

Extended Petri net model: Petri nets or Place/transition net are weighted, directed, bipartite graphs consisting of nodes and arcs. Nodes represent transitions and places. Arcs connect place to transition and vice versa. Petri net is a simple model to represent biochemical reactions. Places contain tokens. When a transition fires, token flow occur from input places to output places. Pure Petrinet do not represent catalytic or inhibitory actions in reactions. To represent these actions, extended Petri nets are used which includes two more arcs namely read arc and inhibitory arc in addition to directed arcs in pure Petri net. Extended Petri nets represent biochemical reactions in a more clear and concise manner. Symbols used in extended Petri net representation are shown in figure 4.

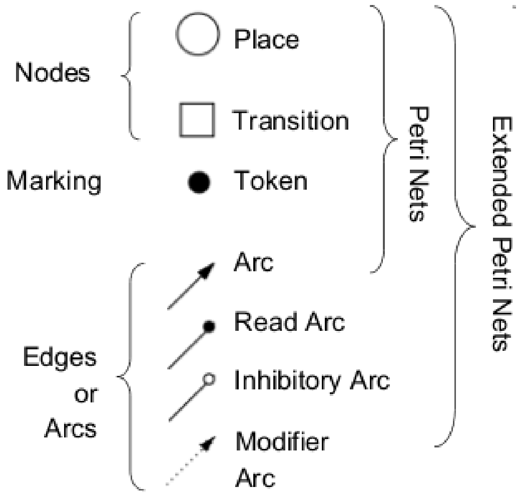


Figure 4: Symbols used in extended Petri net model [41]

LGTS model is a special system which covers pure and extended Petri nets [54]. This model tries to find a FSM (Finite State Machine) consistent with the time series data set. Transition from one state to another is controlled by guard functions which contains constraints for the transition to fire. Control functions in extended Petri net model are equivalent to guard functions in LGTS. However, LGTS model is more expressive as it contains logical as well as linear constraints in guard function making it a more strict check function. Figure 5 shows graphical representation of Extended Petri net model and LGTS model.

These two models are chosen for inference of GRN as each biochemical reaction can be modeled easily. Effect of deletion of gene acting as activator or inhibitor in a reaction can be easily seen. Also, graphical representation helps in better understanding.

Durzinskys extended Petri net model uses combinatorial algorithm [21] to generate extended Petri net model for the network. Each bio-chemical reaction is represented as transition in the model. Catalytic and inhibitory reactions are

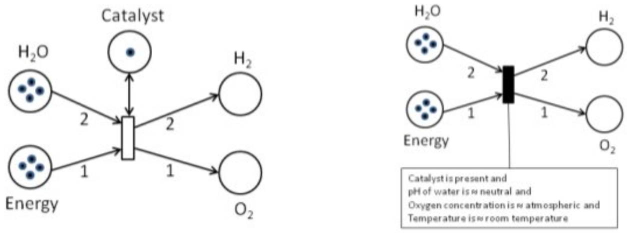

Catalyst
H_2O
H_2
2
2
1
1
Energy
O_2
H_2O
H_2
2
2
1
1
Energy
O_2
Catalyst is present and
pH of water is ≈ neutral and
Oxygen concentration is ≈ atmospheric and
Temperature is ≈ room temperature

Figure 5: Graphical representation of Extended Petri net model and LGTS model [41]

represented using read and inhibitory arcs respectively. Initially, time series data set is feed to the system which is nothing but a state matrix where the last state refers to the terminal state. Then, difference vector set is computed by taking difference of all consecutive state vectors in the state matrix. Each difference vector in the set is a result of firing of one or more transitions. To find all transitions in Petri net, these difference vectors are decomposed into sub-reactions (reaction vectors). These reaction vectors corresponding to each difference vector can occur in any order, thus all possible permutations of these reaction vectors are taken into account. Minimal set of these reaction vectors is found which explains all difference vectors in the difference vector set.

Then second step towards reconstructing extended Petri net model is to find control functions for the transitions which could fire at terminal state. A reaction vector can fire at terminal state if the sum of these two vectors is a valid state. Control functions are minimal Boolean functions of places connected to transitions by control arcs (read or inhibitory arcs). Control functions prevent transitions to fire at terminal state. In this algorithm, Quine Mc Cluskey method is used to find control functions for all reaction vectors.

However, in our implementation of Durzinsky's method, difference vectors are only used to find transitions and control functions. We could find original network from difference vector set itself thus reaction vectors are not considered. Also, all places are searched exhaustively to find control functions for transitions.

In LGTS implementation, system is fed with state matrix (time series data set). System then finds all the difference vectors. Each difference vector is checked if it is a legal transition (i.e. it can not fire at terminal state). If it is a legal transition, it is allowed to fire at that state but if not, appropriate transition type (read or inhibitory) and control places are found by applying guard functions at that difference vector. Guard functions contains conditions for pre-state (state prior to transition fire), post-state (state after transition fires) and some invariant conditions. Applicable guard functions for difference vector are again checked at terminal state. Only those guard functions are selected for difference vectors which do not let difference vector to fire at terminal state. A minimal set of guard functions applicable at transition is found. Finally, system returns guard functions for all transitions in the network.

Data sets used are of phosphate regulatory network and MAPK cascade network.

Experiments are executed under two heads:

1. Incorporation of prior knowledge
2. Estimation of effect of noise

All the implementation is done in Prolog. Prolog is a logic programming language and well-suited for implementation of both the models.

3.3.1 *Networks*

Networks considered for reconstruction procedure are:

1. Phosphate regulatory network in E.Coli

   In enteric bacteria, phosphorus compounds are needed for growth. Amount of phosphorus is controlled by PHO regulon which contains a set of genes. Expression of these genes is controlled by signal transducing proteins which form phosphate regulatory network. In inorganic phosphate limiting conditions, PhoR protein gets phosphorylated which further phosphorylates PhoB protein which in its phosphorylated form binds to promoter region of PhoA gene and activates it. This alkaline phosphate then degrade organic phosphate to inorganic phosphate. The transfer of inorganic phosphate from cytoplasm to periplasm is done by PstSCAB complex. Although, when inorganic phosphate is present in abundance, phosphorylated PhoR protein gets dephosphorylated which prevent further production of inorganic phosohate. Phosphate regulatory network is shown in figure 6. And extended Petri net model of this network is shown in figure 7.

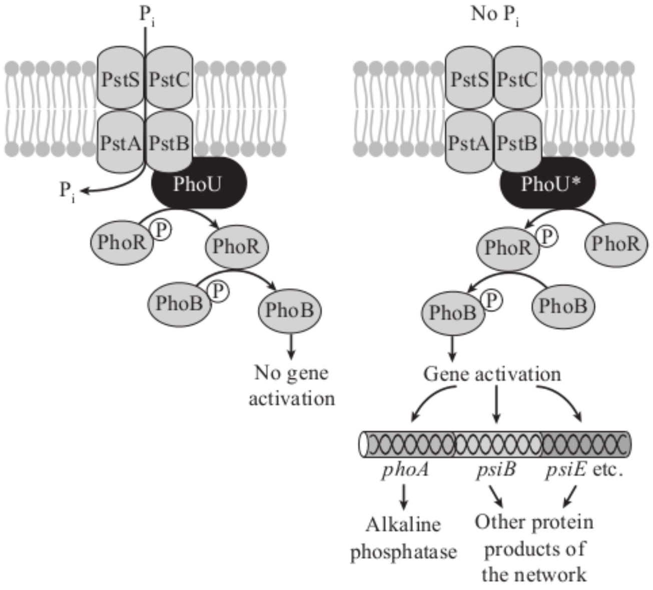
   

   Figure 6: Phosphate regulatory network in E. Coli [41]

   Simulated time series data set for the above network is generated using Snoopy Petri net tool [21].

2. MAPK cascade pathway

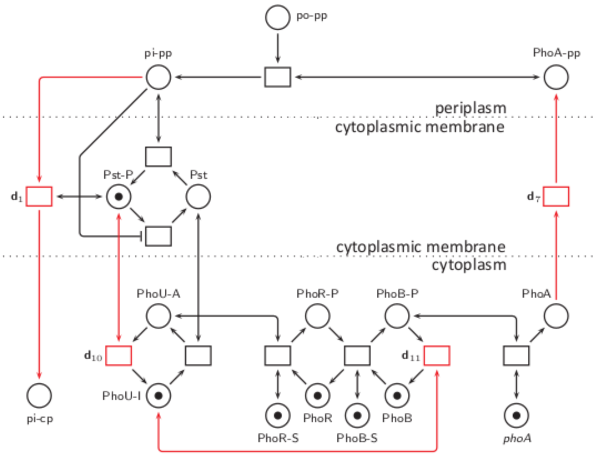


Figure 7: Phosphate regulatory network [21]

MAPK (Mitogen activated protein kinases) is a central signalling pathway that is used in cell tissues to communicate extra cellular events to the nucleus [54]. Initiation of the pathway happens when a protein from receptor protein binds to 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 acts as switch. MAPK cascade pathway is shown in figure 8.

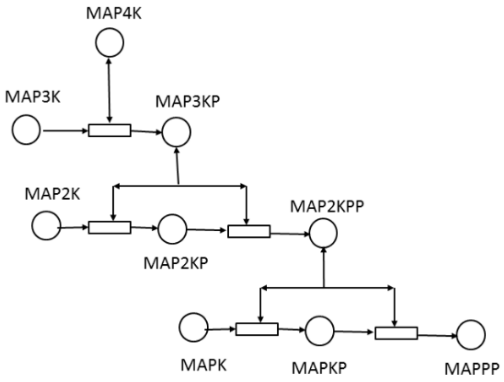


Figure 8: MAPK cascade pathway [54]

3.3.2 *Results*

Results obtained from experiments are discussed in this section.

Experiment 1: Incorporation of background knowledge

The experimental results using the approaches detailed in the above section are reported here. These experiments are performed on Phosphate regulatory network and MAPK cascade pathway. Time series data set used for Phosphate regulatory network is taken from [21] and for MAPK network is taken from [54]. Data set of Phosphate regulatory network is originally constructed using Petri net tool Snoopy.

Case 1: Phosphate regulatory network

Time series data set for phosphate regulatory network consists of 16 places. Total number of experiments done to obtain this data set is 11. All experiments in total have 47 state vectors.Each steady state obtained at the end of experiment is terminal state.

Durzinsky has used background knowledge in his implementation. As background knowledge, places $pi_c p$, $po_p p$, PhoR, PhoRP and PhoA are thrown out from set of potential catalysts and inhibitors [21]. We have used this background knowledge in our LGTS implementation.

Durzinsky's results: Durzinsky obtained 60 different networks consistent with the data. They obtained 2 alternatives of control arc for d1 transition, 2 for d7 transition, 3 for d10 transition and 5 for d11 transition making a total of 60 different possible alternatives for the network. Control arcs obtained by Durzinsky for d1, d7, d10 and d11 transitions are shown in figure 9. In this figure, activator arcs (read arc) are represented by bidirected arcs while inhibitor arcs are represented by flat end arcs.

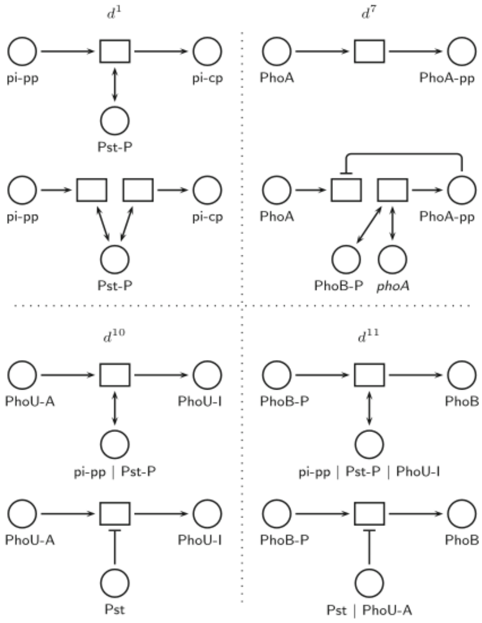


Figure 9: Control arcs obtained for four transitions in Phosphate regulatory network by Durzinsky [20]

Result obtained using extended Petri net model (Durzinsky model): Using this model, we have obtained 768 different networks consistent with the data. Split up is 2 alternative control arc for d2, 3 for d6, 2 for d8, 2 for d9, 4 for d10, 8 for d11 making a total of 768 (2x3x2x2x4x8) networks. Alternative control arcs for these transitions are shown in figure 10. In this figure, activator arc (read arc) is represented by black dot as arc head and inhibitor arc as hollow dot at arc head. Red colored arcs represent incorrect control arcs found.

Result obtained using LGTS model: Here we have obtained 30 different networks conformal with data. 2 alternatives for d2 transition, 3 for d10 transition and 5 for d11 transition. Control arcs obtained using LGTS model for d2, d10 and d11 transitions are shown in figure 11. In this figure, activator arc (read

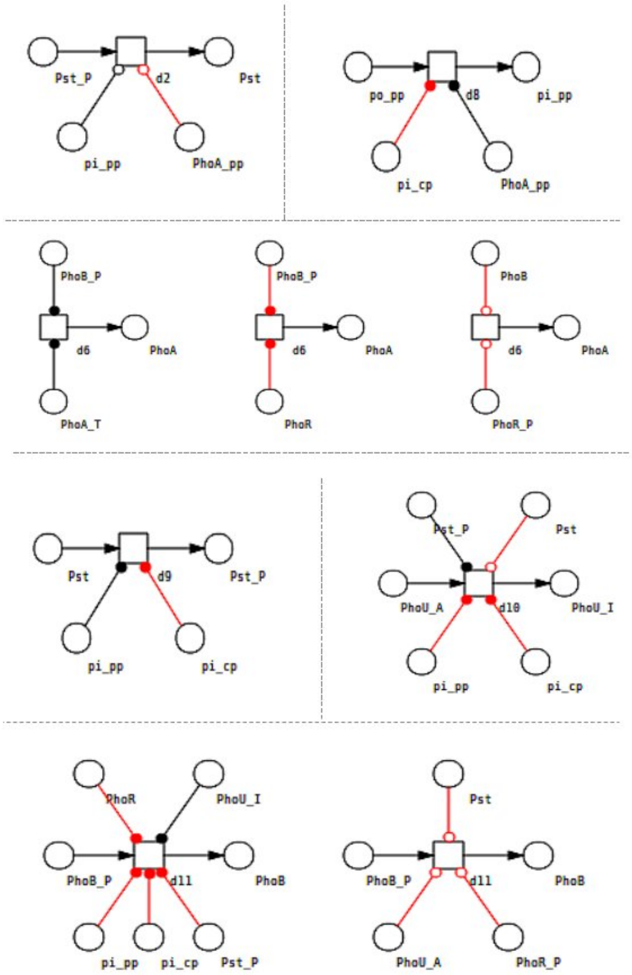


Figure 10: Control arcs obtained for transitions in Phosphate regulatory network using Durzinsky model

arc) is represented by black dot as arc head and inhibitor arc as hollow dot at arc head. Red colored arcs represent incorrect control arcs found.

Although Durzinsky's implementation and our implementation of LGTS model considered same background knowledge, we obtained different number of conformal networks. This difference is due to the fact that Durzinsky considered decomposition of difference vectors into reaction vectors while we considered only difference vectors in our implementation. Due to this, Durzinsky obtained two alternatives for d1 and d7 transition (see figure). For d10 and d11 transition, Durzinsky and our LGTS implementation obtained same alternative control arcs, 3 for d10 and 5 for d11 (see figure). Interestingly, we have obtained two alternatives for d2 transition which is not there in Durzinsky's result. So, Durzinsky has obtained total of 2x2x3x5 = 60 networks while we obtain 2x3x5 = 30 networks.

Difference in the results obtained by our implementation of Durzinsky's model and our implementation of LGTS model is due to the absence of background knowledge in former implementation.

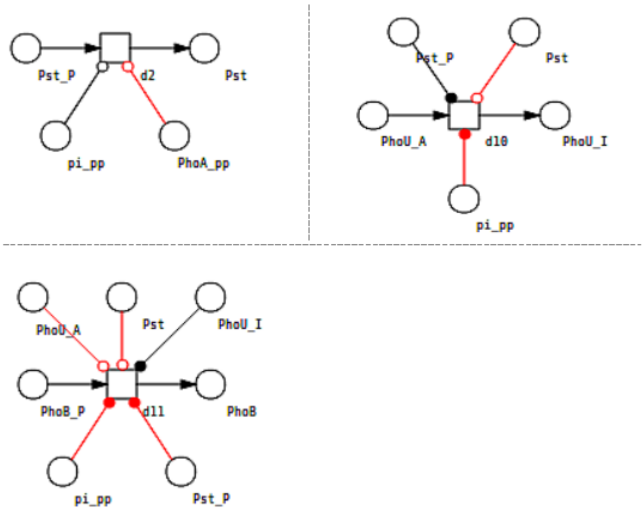
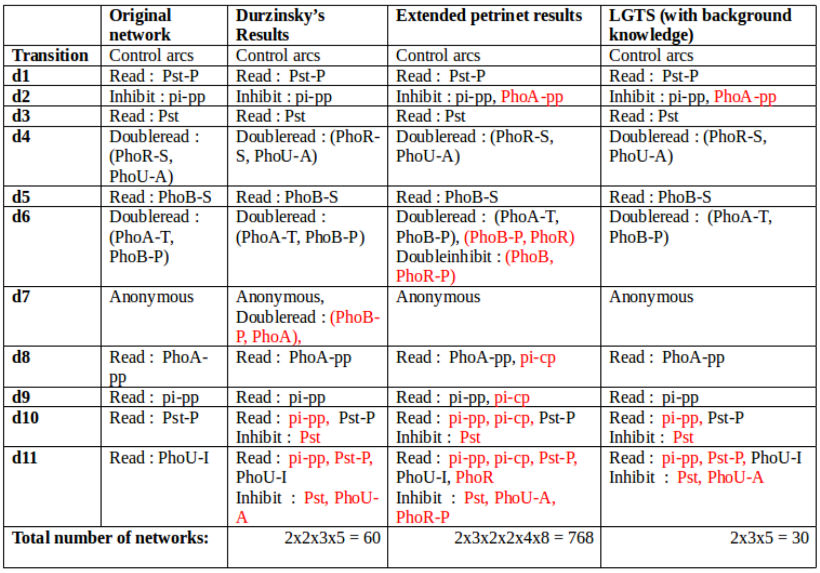


Figure 11: Alternative control arcs obtained for transitions in Phosphate regulatory network using LGTS model

Since, LGTS implementation obtain 30 networks including original network conformal to data, thus, LGTS results are correct results.

So, extended Petri net model in the absence of background knowledge gave poor results while LGTS clearly shows distinction by including background knowledge and substantially reduced the number of conformal networks from 768 to 30. As background knowledge in LGTS, set of catalysts and inhibitors are introduced as predicates in the code. However, this number can be further reduced using more background knowledge of the network. Thus, it implies that using background knowledge can substantially reduce search space and also helps in reconstructing networks which have less data available for them.

All the results obtained are summarized in table 2. Incorrect control arcs obtained are marked with red colour.



	Original network	**Durzinsky's Results**	**Extended petrinet results**	**LGTS (with background knowledge)**
Transition	Control arcs	Control arcs	Control arcs	Control arcs
d1	Read : Pst-P	Read : Pst-P	Read : Pst-P	Read : Pst-P
d2	Inhibit : pi-pp	Inhibit : pi-pp	Inhibit : pi-pp, PhoA-pp	Inhibit : pi-pp, PhoA-pp
d3	Read : Pst	Read : Pst	Read : Pst	Read : Pst
d4	Doubleread : (PhoR-S, PhoU-A)	Doubleread : (PhoR-S, PhoU-A)	Doubleread : (PhoR-S, PhoU-A)	Doubleread : (PhoR-S, PhoU-A)
d5	Read : PhoB-S	Read : PhoB-S	Read : PhoB-S	Read : PhoB-S
d6	Doubleread : (PhoA-T, PhoB-P)	Doubleread : (PhoA-T, PhoB-P)	Doubleread : (PhoA-T, PhoB-P), (PhoB-P, PhoR) Doubleinhibit : (PhoB, PhoR-P)	Doubleread : (PhoA-T, PhoB-P)
d7	Anonymous	Anonymous, Doubleread : (PhoB-P, PhoA),	Anonymous	Anonymous
d8	Read : PhoA-pp	Read : PhoA-pp	Read : PhoA-pp, pi-cp	Read : PhoA-pp
d9	Read : pi-pp	Read : pi-pp	Read : pi-pp, pi-cp	Read : pi-pp
d10	Read : Pst-P	Read : pi-pp, Pst-P Inhibit : Pst	Read : pi-pp, pi-cp, Pst-P Inhibit : Pst	Read : pi-pp, Pst-P Inhibit : Pst
d11	Read : PhoU-I	Read : pi-pp, Pst-P, PhoU-I Inhibit : Pst, PhoU-A	Read : pi-pp, pi-cp, Pst-P, PhoU-I, PhoR Inhibit : Pst, PhoU-A, PhoR-P	Read : pi-pp, Pst-P, PhoU-I Inhibit : Pst, PhoU-A
Total number of networks:		2x2x3x5 = 60	2x3x2x2x4x8 = 768	2x3x5 = 30

Table 2: Performance of different approaches on phosphate regulatory network

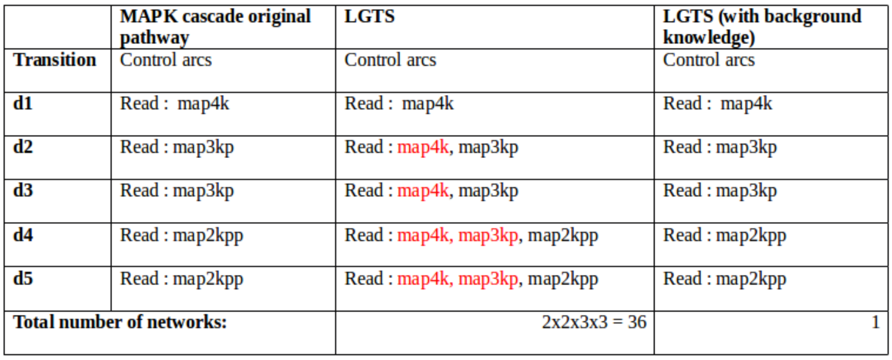
Case 2: MAPK cascade pathway

This experiment is performed using LGTS model with and without background knowledge. This is done to see impact of background knowledge with less data. Time series data set for MAPK cascade pathway consists of 9 places. Total number of experiments done to obtain this data set is 3. All experiments in total have 14 state vectors.

We have used state vectors of all 3 experiments with LGTS model (without background knowledge) i.e. total of 14 state vectors and 1 experiment with LGTS model (with background knowledge) i.e. 6 state vectors.

As background knowledge, we have used protein phosphorylation information i.e. which protein helps in phosphorylating which other protein [54].

In LGTS model (without background knowledge), we have obtained total of 36 networks conformal with data while in LGTS (with background knowledge), we have obtained correct network without any alternatives. This shows the strength of background knowledge even in the presence of less data. Results obtained are summarized in table 3.



	MAPK cascade original pathway	**LGTS**	**LGTS (with background knowledge)**
Transition	Control arcs	Control arcs	Control arcs
d1	Read : map4k	Read : map4k	Read : map4k
d2	Read : map3kp	Read : map4k, map3kp	Read : map3kp
d3	Read : map3kp	Read : map4k, map3kp	Read : map3kp
d4	Read : map2kpp	Read : map4k, map3kp, map2kpp	Read : map2kpp
d5	Read : map2kpp	Read : map4k, map3kp, map2kpp	Read : map2kpp
Total number of networks:		2x2x3x3 = 36	1

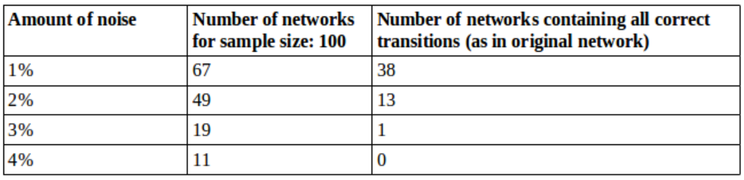
Table 3: Performance of LGTS model on MAPK cascade pathway

Experiment 2: To estimate the impact of measurement noise

In this experiment, noise is introduced randomly in the simulated data set and then noisy data set is used for generating the network. Simulated data set is the discretized data set with all entries as 0 (absence) or 1(presence). To introduce x% noise, we randomly flip x% of total entries in the data set. In this experiment, for different noise levels, we have generated 100 noisy samples. Then these noisy samples are fed to the LGTS system and number of networks produced are stored.

Case 1: MAPK cascade pathway

Time series data set for MAPK cascade pathway consists of 9 places. Total number of experiments done to obtain this data set is 3. All experiments in total have 14 state vectors. So, total number of entries in the data set is 126 (14x9). Introducing 1% noise in the data et randomly flips 1 entry in the data set. We run the code for 100 noisy samples. For 100 noisy samples, LGTS system could produce some network for 67 of them. Out of these 67 networks, 38 networks consists of all correct transitions (as in original network) with some noisy transitions. Introducing 2% noise in the data set randomly flips 2 entries in the data set. For 100 noisy samples, LGTS system could produce some network for 49 noisy samples. Out of these 49 networks, only 13 networks consists of all correct transitions with some noisy transitions. Introducing 3% noise in the data set randomly flips 3 entries in the data set. For 100 noisy samples, LGTS system could produce some network for 19 noisy samples. Out of these 19 networks, only 1 network consists of all correct transitions with some noisy transitions. With 4% noise, 5 entries in the data set got flipped. For 100 noisy samples, LGTS system could produce some network for 11 noisy samples. Out of these 11 networks, not a single network consists of all correct transitions. All these results are summarized in table 4.

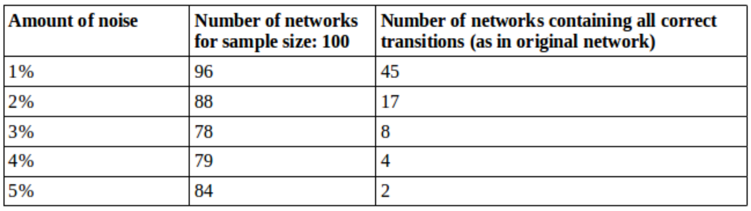


Amount of noise	Number of networks for sample size: 100	Number of networks containing all correct transitions (as in original network)
1%	67	38
2%	49	13
3%	19	1
4%	11	0

Table 4: Performance of LGTS model on MAPK cascade pathway for noise level 1-4%

Case 2: Phosphate regulatory network

Time series dataset for phosphate regulatory network consists of 16 places. Total number of experiments done to obtain this data set is 11. All experiments in total have 47 state vectors. So, total number of entries in the data set is 752 (47x16). Introducing 1% noise in the data set randomly flips 7 entries in the data set. For 100 noisy samples, LGTS system could produce some network for 96 of them. Out of these 96 networks, 45 networks consists of all correct transitions (as in original network) with some noisy transitions. Introducing 2% noise in the data set randomly flips 15 entries in the data set. For 100 noisy samples, LGTS system could produce some network for 88 of them. Out of these 88 networks, 17 networks consists of all correct transitions with some noisy transitions. Introducing 3% noise in the data set randomly flips 22 entries in the data set. For 100 noisy samples, LGTS system could produce some network for 78 of them. Out of these 78 networks, only 8 networks consists of all correct transitions with some noisy transitions. Introducing 4% noise in the data set randomly flips 30 entries in the data set. For 100 noisy samples, LGTS system could produce some network for 79 of them. Out of these 79 networks, only 4 networks consists of all correct transitions with some noisy transitions. Introducing 5% noise in the data set randomly flips 37 entries in the data set. For 100 noisy samples, LGTS system could produce some network for 84 of them. Out of these 84 networks, only 2 networks consists of all correct transitions with some noisy transitions. All these results are summarized in table 5.



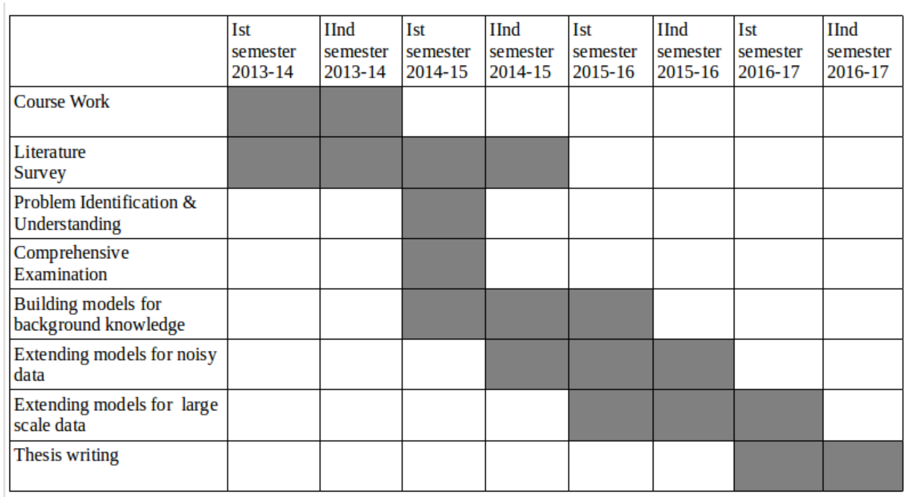
Amount of noise	Number of networks for sample size: 100	Number of networks containing all correct transitions (as in original network)
1%	96	45
2%	88	17
3%	78	8
4%	79	4
5%	84	2

Table 5: Performance of LGTS model on phosphate regulatory network for noise level 1-5%

In this measurement noise experiment for both networks, we cannot get original network exactly even with small amount of noise introduction. Here, we were not decomposing the difference vectors into reaction vectors and only try to reconstruct network using difference vectors. But even with decomposition of difference vectors into reaction vectors (as done in Durzinsky's method), we cannot get original network with small amount of noise. This is because say, for a single noisy difference vector d[-1, 1, 0, 1], if we decompose it in two reaction vectors r1 and r2 such that r1[-1, 1, 0, 0] is the correct difference vector (in noise-less case) and r2 [0, 0, 0, 1] is some other vector, still in the results we will mention both the reaction vectors for that transition. And since r2 is not the correct transition, we will not get original network.

So, this method is highly sensitive to measurement noise as it completely depends on difference vectors of states in state matrix and do not have any procedure to deal with measurement noise.

3.4 TIMELINE



	Ist semester 2013-14	IInd semester 2013-14	Ist semester 2014-15	IInd semester 2014-15	Ist semester 2015-16	IInd semester 2015-16	Ist semester 2016-17	IInd semester 2016-17
Course Work	▨	▨						
Literature Survey	▨	▨	▨	▨				
Problem Identification & Understanding			▨					
Comprehensive Examination			▨					
Building models for background knowledge			▨	▨	▨			
Extending models for noisy data				▨	▨	▨		
Extending models for large scale data					▨	▨	▨	
Thesis writing							▨	▨

ACKNOWLEDGEMENTS

I would like to thank Dr. Sumeet Agarwal and Prof. Ashwin Srinivasan for their constant guidance and support.

REFERENCES

[1] Banjo. https://www.cs.duke.edu/~amink/software/banjo/.

[2] Bayesian network. http://en.wikipedia.org/wiki/Bayesian_network.

[3] Boolean network. http://fias.uni-frankfurt.de/~willadsen/RBN/.

[4] Gaussian graphical model. http://strimmerlab.org/notes/ggm.html.

[5] Gene regulatory network. http://en.wikipedia.org/wiki/Gene_regulatory_network.

[6] Petri net. http://en.wikipedia.org/wiki/Petri_net.

[7] Prism. http://sato-www.cs.titech.ac.jp/prism/.

[8] Quantile normalization. http://en.wikipedia.org/wiki/Quantile_normalization.

[9] System biology and drug discovery. http://www.sbmc06.de/doc/fischer.pdf.

[10] Systems biology. http://medicine.tamhsc.edu/graduate-studies/faculty/systems.html.

[11] Enzo Acerbi, Teresa Zelante, Vipin Narang, and Fabio Stella. Gene network inference using continuous time bayesian networks: a comparative study and application to th17 cell differentiation. *BMC Bioinformatics*, 15(1).

[12] Richard Banks and L.J. Steggles. *A High-level Petri Net Framework for Multi-valued Genetic Regulatory Networks*. Technical report series. University of Newcastle upon Tyne, Computing Science, 2007.

[13] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernardo. How to infer gene networks from expression profiles. *Molecular systems biology*, 3:78, 2007.

[14] Matthew J. Beal, Francesco Falciani, Zoubin Ghahramani, Claudia Rangel, and David L. Wild. A bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21(3):349–356, February 2005.

[15] Richard Bonneau, David Reiss, Paul Shannon, Marc Facciotti, Leroy Hood, Nitin Baliga, and Vesteinn Thorsson. The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, 7(5), 2006.

[16] Cline Brouard, Christel Vrain, Julie Dubois, David Caste, Marie-Anne Debily, and Florence dAlch Buc. Learning a markov logic network for supervised gene regulatory network inference. *BMC Bioinformatics*, 14(1), 2013.

[17] Young Hwan Chang, Joe W. Gray, and Claire J. Tomlin. Exact reconstruction of gene regulatory networks using compressive sensing. *bioRxiv*, 2014.

[18] Bor-Sen Chen and Cheng-Wei Li. Analysing microarray data in drug discovery using systems biology. *Expert Opinion on Drug Discovery*, 2(5):755–768, 2007. PMID: 23488963.

[19] Alberto de la Fuente, Paul Brazhnik, and Pedro Mendes. Linking the genes: inferring quantitative gene networks from microarray data. *TRENDS in Genetics*, 18(8):395–398, 2002.

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

[21] Markus Durzinsky, Wolfgang Marwan, and Annegret Wagler. Reconstruction of extended petri nets from time series data and its application to signal transduction and to gene regulatory networks. *BMC Systems Biology*, 5, 2011.

[22] Amin Emad and Olgica Milenkovic. Caspian: A causal compressive sensing algorithm for discovering directed interactions in gene networks. *PLoS ONE*, 9(3), 03 2014.

[23] Jeremiah J. Faith, Boris Hayete, Joshua T. Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J. Collins, and Timothy S. Gardner. Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. *PLoS Biol*, 5(1), January 2007.

[24] Nir Friedman, Michal Linial, and Iftach Nachman. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.

[25] Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the structure of dynamic probabilistic networks. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 139–147. Morgan Kaufmann Publishers Inc., 1998.

[26] J. Gebert, N. Radde, and G.-W. Weber. Modeling gene regulatory networks with piecewise linear differential equations. *European Journal of Operational Research*, 181(3):1148 – 1165, 2007.

[27] Zeeshan Gillani, Muhammad S. H. Akash, MD. Matiur Rahaman, and Chen Ming. Comparesvm: supervised, support vector machine (svm) inference of gene regularity networks. *BMC Bioinformatics*, 15(1), 2014.

[28] Raed I. Hamed, S. I. Ahson, and R. Parveen. A New Approach for Modelling Gene Regulatory Networks Using Fuzzy Petri Nets. *Journal of Integrative Bioinformatics*, 7(1), 2010.

[29] Shengtong Han, Raymond K. W. Wong, Thomas C. M. Lee, Linghao Shen, Shuo-Yen R. Li, and Xiaodan Fan. A full bayesian approach for boolean genetic network inference. *PLoS ONE*, 9(12), 12 2014.

[30] Carlos HA Higa, Vitor HP Louzada, Tales P Andrade, and Ronaldo F Hashimoto. Constraint-based analysis of gene interactions using restricted boolean networks and time-series data. *BMC Proceedings*, 5(2):555–565, 2011.

[31] Trey Ideker, Timothy Galitski, and Leroy Hood. A new approach to decoding life: systems biology. *Annual review of genomics and human genetics*, 2(1):343–372, 2001.

[32] Hidde De Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9:67–103, 2002.

[33] Guy Karlebach and Ron. Shamir. Modelling and analysis of gene regulatory networks. *Nat Rev Mol Cell Biol*, 9(10):770–780, 2008.

[34] Sun Yong Kim, Seiya Imoto, and Satoru Miyano. Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in Bioinformatics*, 4(3):228–235, 2003.

[35] Robert Kffner, Tobias Petri, Lukas Windhager, and Ralf Zimmer. Petri nets with fuzzy logic (pnfl): Reverse engineering and parametrization. *PLoS ONE*, 5(9), 09 2010.

[36] Jinshan Li and Xiang sun Zhang. An optimization model for gene regulatory network reconstruction with known biological information , 2007.

[37] Shoudan Liang, Stefanie Fuhrman, and Roland Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 18–29, 1998.

[38] SR Maetschke, PB Madhamshettiwar, MJ Davis, and MA Ragan. Supervised, semi-supervised and unsupervised inference of gene regulatory networks. *Brief Bioinform*, 15(2):195–211, 2014.

[39] AdamA Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(1), 2006.

[40] Stephen Marshall, Le Yu, Yufei Xiao, and Edward R. Dougherty. Inference of a probabilistic boolean network from a single observed temporal sequence. *EURASIP J. Bioinformatics Syst. Biol.*, 2007:5–5, January 2007.

[41] Wolfgang Marwan, Christian Rohr, and Monika Heiner. Petri nets in snoopy: A unifying framework for the graphical display, computational

modelling, and simulation of bacterial regulatory networks. In *Bacterial Molecular Networks*, volume 804 of *Methods in Molecular Biology*, pages 409–437. Springer New York, 2012.

[42] Patrick E. Meyer, Kevin Kontos, Frederic Lafitte, and Gianluca Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J. Bioinformatics Syst. Biol.*, 2007:8–8, January 2007.

[43] Kristin Missal, Michael A. Cross, and Dirk Drasdo. Gene network inference from incomplete expression data: Transcriptional control of hematopoietic commitment. *Bioinformatics*, 22(6):731–738, March 2006.

[44] Fantine Mordelet and Jean-Philippe Vert. Sirene: supervised inference of regulatory networks. *Bioinformatics*, 24(16):i76–i82, 2008.

[45] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *JOURNAL OF LOGIC PROGRAMMING*, 19(20):629–679, 1994.

[46] Irene M. Ong, Jeremy D. Glasner, and David Page. Modelling regulatory pathways in e. coli from time series expression profiles. *Bioinformatics*, 18(1):S241–S248, 2002.

[47] Hongjia Ouyang, Jie Fang, Liangzhong Shen, EdwardR Dougherty, and Wenbin Liu. Learning restricted boolean network model by time-series data. *EURASIP Journal on Bioinformatics and Systems Biology*, 2014(1).

[48] John. Quackenbush. Microarray data normalization and transformation. *Nature Genetics*, 2002.

[49] John Jeremy Rice, Yuhai Tu, and Gustavo Stolovitzky. Reconstructing biological networks using conditional correlation analysis. *Bioinformatics*, 21(6):765–773, 2005.

[50] Thomas Schlitt and Alvis Brazma. Current approaches to gene regulatory network modelling. *BMC Bioinformatics*, 8(6):S9, 2007.

[51] Ilya Shmulevich, Edward R. Dougherty, Seungchan Kim, and Wei Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

[52] Ilya Shmulevich, Edward R. Dougherty, and Wei Zhang. From boolean to probabilistic boolean networks as models of genetic regulatory networks. In *Proc. IEEE*, pages 1778–1792, 2002.

[53] Ilya Shmulevich and Wei Zhang. Binary analysis and optimization-based normalization of gene expression data. 18(4):555–565, 2002.

[54] Ashwin Srinivasan, Michael Bain, and K. Sriram. Knowledge-guided identification of transition-based models of biological systems using ilp.

[55] L. J. Steggles, Richard Banks, and Anil Wipat. Modelling and analysing genetic networks: From boolean networks to petri nets. In *Proceedings of the 2006 International Conference on Computational Methods in Systems Biology*, CMSB'06, pages 127–141. Springer-Verlag, 2006.

[56] Adriano V. Werhli and Dirk Husmeier. Reconstructing Gene Regulatory Networks with Bayesian Networks by Combining Expression Data with Multiple Sources of Prior Knowledge. *Statistical Applications in Genetics and Molecular Biology*, 6(1), January 2007.

[57] Kevin Y. Yip, Roger P. Alexander, Koon-Kiu Yan, and Mark Gerstein. Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data. *PLoS ONE*, 5(1):e8121, 01 2010.

[58] Pietro Zoppoli, Sandro Morganella, and Michele Ceccarelli. TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, 11(1), March 2010.