SCALE-FREE PRIOR IN GENE REGULATORY NETWORK RECONSTRUCTION

A thesis submitted in partial fulfillment
of the requirements for the degree of

DUAL DEGREE

in

Computer Science & Engineering

by

ABDUL HADI SHAKIR
Entry No. 2010CS50202

Under the guidance of

Prof. SUMEET AGARWAL
(Prof. PARAG SINGLA)

Department of Computer Science and Engineering,
Indian Institute of Technology Delhi.
June 2015.

Certificate

This is to certify that the thesis titled **SCALE-FREE PRIOR IN GENE REGULATORY NETWORK RECONSTRUCTION** being submitted by **ABDUL HADI SHAKIR** for the award of **Dual Degree** in **Computer Science & Engineering** is a record of bona fide work carried out by him under my guidance and supervision at the **Department of Computer Science & Engineering**. The work presented in this thesis has not been submitted elsewhere either in part or full, for the award of any other degree or diploma.

Prof. SUMEET AGARWAL
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Prof. PARAG SINGLA
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

Abstract

Reverse engineering gene regulatory network using expression data is a tough task due to the complex nature of these networks. Bayesian inference have long been put into play for learning these networks. However, they are limited to using only the information derived from data. Here, we attempt to include expert knowledge as a prior information for network reconstruction. One such property of biological networks is *scale-free* structure. We combine scale-free prior on graph structure with genetic algorithm as a sampling algorithm for network inference. We compare scale-free prior with uniform prior and show that the former outperforms the latter under the condition that network being learned has scale-free nature. We also compare it against ARACNE and find that scale-free prior performs relatively low. In addition, we ran our scale-free prior on HCC1954 breast cancer cell data and find that it recalls a significant fraction of known interactions.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Sumeet Agarwal for the continuous supervision and support of my M.Tech project. This would not have been possible without his patience, motivation and immense knowledge.

Besides my advisor, I would like to thank Prof. Parag Singla for coordinating my project in the CSE department. I would also like to thank the rest of my thesis committee: Prof. Mausam, Prof. K.K. Biswas, and Prof. Amitabha Bagchi, for their encouragement, insightful comments, and hard questions.

My sincere thanks also goes to my classmate Ashesh Mishra and Bharat Ratan for involving me in fruitful discussion about data and life, and for pointing out mistakes and reasons in times of my project blackout.

Last but not the least, I would like to thank my family: my parents Umme Kubra and Akhtar Hussain, and my brothers Amir and Nasir for their love and belief in me. It was their support that has helped me sail swiftly over the past five years.

ABDUL HADI SHAKIR

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The recent advancement in the technology for high-thoughput data collection in molecular biology has led to a burst of genomic data. The huge amount of data generated by gene expression arrays have given researchers the opportunity to utilise these information to create a positive impact in the field of biology, medicine and pharmacology. In this thesis we talk about computational methods that have been making this possible and on ways to improve these methods.

1.1 GENE REGULATORY NETWORK

A gene regulatory network (abbreviated as GRN) is defined as a "collection of DNA segments in a cell which interact with each other indirectly (through their RNA and protein expression products) and with other substances in the cell to govern the gene expression levels of mRNA and proteins" [7].

GRNs can be represented by a typical graph or network like structure - the genes being the node and the interaction being represented by edges. More specifically, it is comprised of "nodes", the genes and their regulators, joined together by "edges", which represent physical and/or regulatory interactions [16]. These interactions in turn can be promotory or inhibitory in nature. The Figure 1.1 shows an example of a GRN.

A proper understanding of the GRNs can help us predict causal molecular pathways at cellular level. In addition to predicting physical interactions, GRN edges can also help us identify regulatory relationships and complex cascading pathways, which can be immensely helpful in fields of medicine and pharmacology [16].

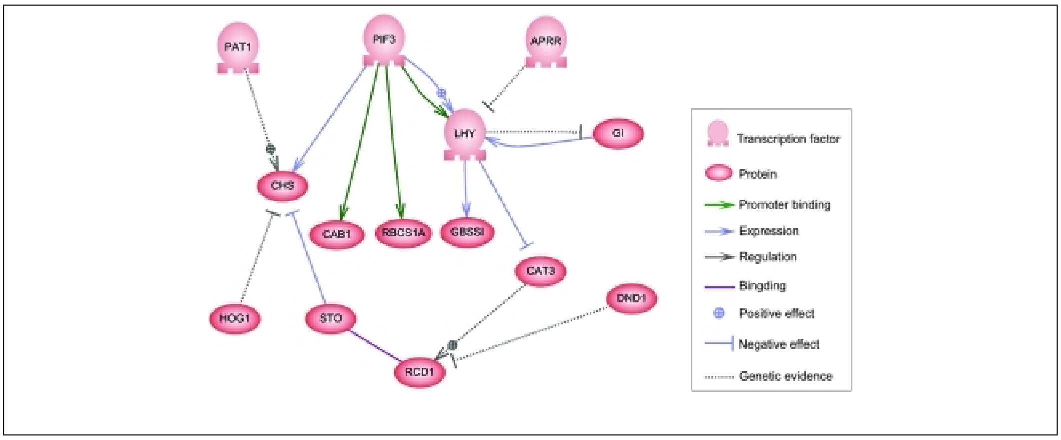


Figure 1.1: Example of a regulatory network

1.2 GENE EXPRESSION DATA

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product. The gene expression mechanism is used by all the life forms. This is the basis of the versatility and adaptibility of any organism.

These expression levels are measured under controlled experimental conditions using DNA microarrays. Multiple microscopic DNA spots are attached to the surface of the microarray. Each spots contains a particular DNA sequence which is called a Probe. The activation level of probes corresponds to the expression level of genes [5]. The array expression data is of 2D-matrix form, where each row corresponds to a gene and the column corresponds to a particular experimental condition. The entries against each gene corresponds to its expression level. Figure 1.2 picturises the gene expression matrix.

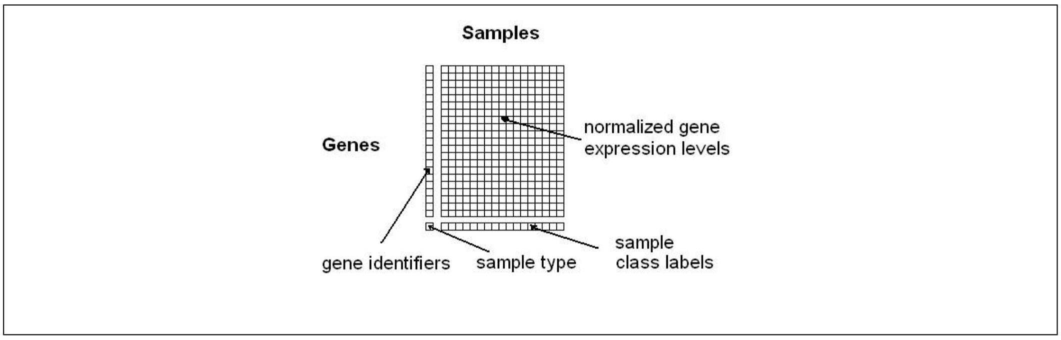


Figure 1.2: Gene expression matrix

1.3 RECONSTRUCTION OF GRNs

Since we do not know the true biological networks apriori, we deploy computational and statistical methods to try learn the GRN. We use machine learning methods to infer information from the gene expression data in order to reconstruct the GRN. Previous efforts for statistical modeling of gene regulatory network has fallen into one of the two categories, Boolean models [1] or systems of differential equations [4]. There is yet another approach adopted by Friedman, *et al* [8], which uses Bayesian networks to analyze gene expression data. This thesis focuses on this approach and suggests some ways to improve its performance.

1.4 MOTIVATION AND PROBLEM DEFINITION

It is very common for the computational methods deployed to reconstruct GRNs to use only the information inferred from the gene expression data. A lot of reserach has been done on the biological networks, that gives us information about structural and organisational properties of these networks. There is a possibility that we can use these prior information in the network reconstruction.

In this thesis we incorporate one such structural property of the biological network, namely the scale-free property as a prior information for network reconstruction. We assess the performance improvement after using scale-free prior, and also compare against network learning methods other than Bayesian networks. The learning algorithm augmented with scale-free prior is also tested on some real data.

Chapter 2

Bayesian Network

Graphical models [15] are a special class of models that are extensively used for formal statistical inference of systems having multiple interacting components. GRNs can be classified as one such system. A graphical model is characterised by two things:

- **Graph**: Describes probablistic relationship between variables, and
- **Parameters**: Specifies conditional distribution specified by the graph.

Bayesian Network (abbreviated as BN) inference is one such generic and a widely used framework for fitting probabilistic models to data.

2.1 BASIC THEORY OF BN

A Bayesian Network [8] is an acyclic directed graph that uniquely specifies a joint probability distribution over a set of random variables. Let us first define some variables before going into the theory. Define:

- $\chi = \{X_1, \cdots, X_n\}$ as set of discrete random variables X_i. These are the various genes in our case.
- $D = \{Y_1, Y_2, \cdots, Y_n\}$ as observed instances of χ. This is the Microarray expression data in our case.
- $< G, \Theta >$ as specifying BN for χ. The variable G corresponds to a directed grpah whose nodes are from χ. Θ is a set of parameters that together quantifies the probability distribution of each variable in the graph. $< G, \Theta >$ will be the reconstructed GRN in our case.

Under the classic *Naive Bayes* assumption, each variable X_i is considered independent of its non-descendants given its parents in G. Thus, the BN uniquely specifies a joint probability distribution over χ given by:

$$P(X_1, \cdots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)) \tag{2.1}$$

The objective is now to find the most probable graph G for explaining the data contained in D. For this we need to introduce a scoring metric that tells us how probable a given graph G explains the data D. This scoring metric is discussed in next section.

2.2 BAYESIAN SCORING METRIC

In order to rank a given graph G as to how probable it explains the data contained in D, we need to introduce a scoring metric called the Bayesian scoring metric [10]. This metric is defined as the log of the probability of G given D.

$$BayesianScore(G) = \log p(G|D) \tag{2.2}$$

$$BayesianScore(G) = \log p(D|G) + \log p(G) - \log p(D) \tag{2.3}$$

Since, $\log p(D)$ is constant for all graphs G, we can remove this term from the scoring metric. Thus, $BayesianScore$ becomes:

$$BayesianScore(G) = \log p(D|G) + \log p(G) \tag{2.4}$$

The term $\log p(D|G)$ is called the log-likelihood, whereas the term $\log p(G)$ is called the log-prior term.

2.3 INCORPORATING PRIORS

In the absence of any specific prior knowledge, it is common to choose as uninformative prior as possible, typically a prior distribution that is uniform over all models. For instance, it is a general practice to maximise only the likelihood. In such case, the scoring metric is defined as:

$$BayesianScore(G) = \log p(D|G) \tag{2.5}$$

There is yet another variant of uniform prior scoring metric that is called the *Bayesian Information Criterion* or *BIC*. *BIC* is actually a penalized maximum likelihood estimate, that penalizes excessive edges in the graph. This is the most extensively used version of the scoring metric and is defined as:

$$BIC = -2log(p(D|G)) + Klog(n) \tag{2.6}$$

where K is the number of edges in the grpah G and n is the number of data points contained in D. *BIC* helps in avoiding over-fitting of data.

In most of the cases taking a uniform prior over the entire range may correspond to a bias towards unrealistic values. There are several properties of biological networks that have already been established. If we make use of these information in addition to data in network inference, we might be able to increase the efficiency of GRN reconstruction. These information will enter the scoring metrics via log-prior term, i.e. $\log p(G)$. One such property is the scale-free property of the biological network. This is discussed in the next chapter.

Chapter 3

Scale-Free prior

3.1 BACKGROUND AND MOTIVATION

A scale-free network is a network whose degree distribution of nodes follows a *power law*, atleast asymptotically. The probability that a node of a scale-free network has k edges, or alternatively k adjacent nodes is governed by the following distribution:

$$p(k) \sim k^{-\gamma} \tag{3.1}$$

This basically suggests that there are very many nodes with only a few link, and in addition few hubs are present with large number of links. Figure 3.1 depicts the scale-free property pictorially.

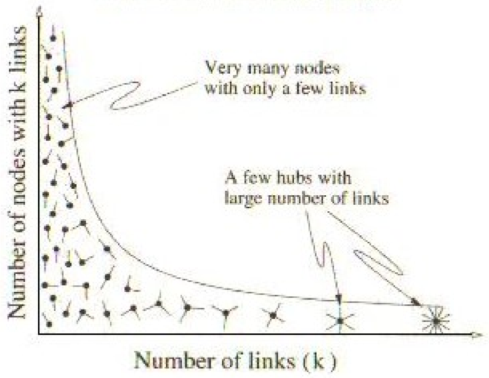

Number of nodes with k links
Very many nodes with only a few links
A few hubs with large number of links
Number of links (k)

Figure 3.1: Scale-free property

There are a broad class of networks that follow this scale-free property, like the Internet or the World Wide Web. In the work done by Newman et al. [22], this property is shown to be the characterisitc of biological networks as well. Therefore, we formulate the scale-free prior in the following section and use that as prior in GRN reconstruction.

3.2 FORMULATING SCALE-FREE PRIOR

The initial formulation of the scale-free prior was done in the work done by Sheridan et al. [26]. A similar work was also done by Bender et al. (2011) [3]. Define: $V = \{v_1, \cdots, v_n\}$ as the fixed set of nodes in a given graph structure G. The scale-free prior probability on the graph structure can be calculated as follows:

- First, assign a probability P_i to each node $i \in 1...N$:

$$P_i = \frac{i^{-\mu}}{\sum_{j=1}^{N} j^{-\mu}} \approx \frac{1-\mu}{N^{1-\mu}} i^{-\mu} \tag{3.2}$$

  The probability P_i decreases with increasing i, and this probability summed up over all i is equal to 1, i.e. $\sum_{i \in 1...N} P_i = 1$. μ is defined as $\mu = \frac{1}{\gamma-1}$, $\gamma \in [2; \infty)$.

- Assuming the nodes are selected independent of each other with probability proportional to P_i, the probability of two nodes not being connected is defined as:

$$\bar{P_{ij}} = (1 - 2P_iP_j) \simeq e^{-2NKP_iP_j} \tag{3.3}$$

  where K is a prameter that controls the mean number of edges.

- The probability of any structure $G_\sigma = (V, E)$ of node set V, edge set E and a permutation $\sigma = \{\sigma_1, ..., \sigma_N\}$ of all nodes in G is given by the product of probability of the edges that are present and the probability of edges not being present. This can be given by:

$$P(G_\sigma) = \prod_{\{v_i,v_j\} \in E} P_{ij} \prod_{\{v_i,v_j\} \notin E} \bar{P_{ij}} \tag{3.4}$$

$$P(G_\sigma) = \prod_{\{v_i,v_j\} \in E} (1 - \bar{P_{ij}}) \prod_{\{v_i,v_j\} \notin E} \bar{P_{ij}} \tag{3.5}$$

$$P(G_\sigma) = \prod_{\{v_i,v_j\} \in E} (1 - e^{-2NKP_iP_j}) \prod_{\{v_i,v_j\} \notin E} (e^{-2NKP_iP_j}) \tag{3.6}$$

- Different permutations of σ are generated, resulting in one graph G_σ for each permutation. If the total number of permutation generated is B, then the final probability of G is averaged over each of them:

$$P(G) = \frac{1}{B}\sum_{\sigma} P(G_\sigma) \tag{3.7}$$

This formulation of scale-free prior can now be used with any of the scoring metric that has scope for incorporating the prior term.

Chapter 4

Genetic Algorithm for GRN reconstruction

4.1 SAMPLING BASED ALGORITHM

In our task of GRN reconstruction using Bayesian inference, we try to search for a graph that scores maximum according to the given scoring metric. The limiting point is that this task is computationally very expensive to be done in real time. The number of possible graphs (or networks) for a given set of variables (genes) V, increases super-exponentially with $|V|$. This motivates us to use sampling based algorithm for optimal network search.

There are two primary sampling techniques that we tried to use:

Genetic Algorithm (GA): In GA we evolve a population of candidate networks. We start with an initial population and successively create next generation population by sharing information among the members of the previous population. This generation is stopped after we reach the convergence criteria or after a fixed number of iterations. The final network contains edges that are present in significant numbers in the population [2].

Markov Chain Monte Carlo (MCMC): This sampling approach is based on the previous work done by Werhli et al. [29]. In this approach we start with a given network G, and try to define a neighbourhood around G by adding, deleting or reversing edges in G. Then one of the network from the neigbourhood is selected using Metropolis-Hastings sampler [9]. A similar process is now applied to this new network to create a chain of networks. The chain is finally terminated when we approach the convergence criteria, and the final network of the chain is our network of interest.

We tested both GA and MCMC on our syntheic data with 10 and 20 nodes. We observed that GA outperformed MCMC in both the instances. The comparing criteria and the performance are discussed in next chapter. Usefulness of GA in network reconstruction approaches have also been established in the work done by Wahde and Hertz [28], and Spieth et al. [27]. Therefore, we decided to work using GA algorithm in the future course of our experiments with Bayesian inference and scale-free prior.

4.2 GA SPECIFICATION

Before we go into the specifications of GA, let us define some variables:

- P: Defined as the initial set of populations. $P = \{G_j : j \in 1, ..., p\}$ of p networks.
- q: It is a parameter of GA that defines the *crossover* rate as q and the *selection* rate as $(1 - q)$. $q \in [0; 1]$
- m: It is a parameter of GA that defines the *mutation* rate. $m \in [0; 1]$
- $Fitness$ of a network is defined as the network score given by a scoring metric (discussed in Chapter 2).

From the current population P, we create the next generation population P' using the following three steps:

Selection: A fraction $(1 - q)p$ of the individuals in P are selected with probability proportional to their fitness. If fitness of the selected network is greater than the median fitness of P, the network is added to next generation population P'.

Crossing over: A fraction $\frac{qp}{2}$ random pairs of individuals are choosen with probability proportional to their fitness. Crossing over is performed for each of these pairs. In order to achieve this, the columns of the adjacency matrix of each network is attached to one another to represent it as a vector. Then a two point cross over is performed for these vectors. If the individual obtained after performing cross-over has fitness greater than the median fitness of P, it is added to P'. If the population size remains less than p after performing cross-over, we add individuals from P to P' randomly unless there are p members in P'.

Mutation: A fraction of mp networks are choosen from P'. For each of the selected network, a random edge is drawn and its type is changed to one of the remaining type. If performing this step increases the fitness of the network, the old network is replaced by the mutated network.

Successive iterations of population evolution is performed unless we reach the convergence criteria or a maximum number of iterations. The GA is said to converge if the median score of the population does not change over 10 iterations in a row.

When the GA terminates we have a population of networks. In the final network we include edges that are present in some significant indivduals of the population. If the significance threshold is $\tau \in [0;1]$, then an edge is included in the final network if it occurs in more than $\tau \times p$ individuals in the final population.

4.3 SCALE-FREE PRIOR IN GA

The scale-free prior can be easily incorporated in GA by virtue of the $fitness$ measure of the individual network in the population. Since fitness is nothing but one of the scoring metric discussed in Chapter 2, the prior information can be incorporated in the log-prior term of the Bayesian score metric. In addition, the initial population to start the GA is also drawn from the prior type, i.e. scale-free prior in our case.

Chapter 5

Results and Discussion

5.1 SYNTHETIC DATA

To test the performace of our learning algorithm we first test it on synthetic data. If it performs well on the synthetic data, it is then tested on real data.

5.1.1 Normal Sampling

The normal sampling of data is similar to the one used in C.Bender et al. [3]. A scale-free network was generated using γ as the control parameter. γ basically controls the 'scale-free'ness of the network. Some randomly selected edges were changed to inhibitory edges, such that the network has atleast 20% of the edges that are inhibitory in nature. A node is said to be **activated** if all the edges incident on it is activating and there are no inhibiting edges incident on it. In other cases it is said to be **inhibited**.

The expression level of activated node was sampled from the normal distribution $\mathcal{N}(2000, 400)$, that of the inhibited node was sampled from $\mathcal{N}(1200, 400)$ and that of stable node was sampled from $\mathcal{N}(1600, 400)$. The parameters for the normal distribution (mean and standard deviation) were choosen similar to the values observed in real data.

5.1.2 Mendes model

Another set of synthetic data was generated using Mendes model [20] that follows multiplicative Hill Kintetics [11] to approximate the transcriptional interactions. In this model there is one differential equation for each gene (given by equation 5.1). The RHS of equation has two terms - one posotive term that represents transcription and one negative term that represents mRNA breakdown.

$$\frac{dG_i}{dt} = s(G_1, \cdots, G_n) - b(G_i) \tag{5.1}$$

where G_i = abdundance of mRNA of gene i, $s(G_1, \cdots, G_n)$ = rate law representing mRNA synthesis and $b(G_i)$ = mRNA breakdown.

The expression level of each gene is sampled at different time points. Although this model is a simplification of real biological network, it gives a reasonably complex interaction network that very well approximates the transcriptional interaction. Any good learning algorithm should be expected to perform well on this set of data. All the simulations for Mendes model was performed using the *sysgensim* software [24].

5.2 PERFORMANCE METRIC

Before we discuss about the performance metrics used for assessing the performance of our algorithm, let us define some variables:

- N_{TP} as number of true positive arcs in reconstructed network
- N_{FP} as number of false positive arcs in reconstructed network
- N_{TN} as number of true negative arcs in reconstructed network
- N_{FN} as number of false negative arcs in reconstructed network

We define two quantities - sensitivity(SN) that counts true occurences and specificity(SP) that counts false occurences as follows:

$$SN = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{5.2}$$

$$SP = \frac{N_{TN}}{N_{FP} + N_{TN}} \tag{5.3}$$

At different inclusion threshold ($\tau \in [0;1]$) in GA, we obtain different values of SN and SP. τ is varied in the range 0 to 1, and a set of SN and SP values are obtained. The SN values when plotted against $(1 - SP)$

gives us the receiver operator characteristic (ROC) curves. The area under the ROC curve (also known as AUC) gives us the measure of algorithm's performance. Greater the AUC, better is the performance. AUC for a good learning algorithm is expected to be atleast 0.5.

All the results and numbers that have been mentioned in the sub-sections to follow have been averaged over five different synthetic data generations.

5.3 POPULATION SIZE IN GA

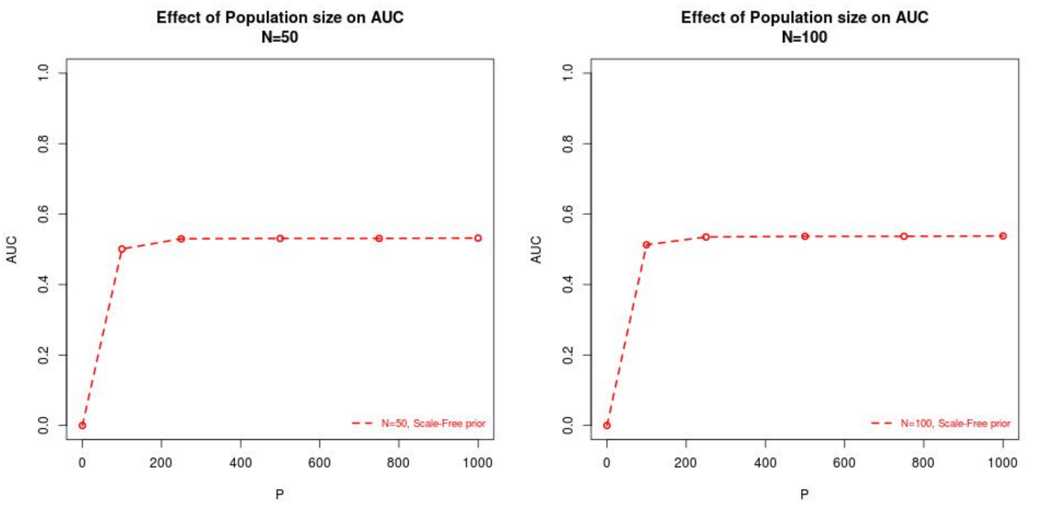
The size of the population in the GA is a parameter that has to be choosen in advance. There are many possible population size that can be used for network reconstruction. Trying different population sizes for the same experiment setting can be computationally expensive and redundant at the same time. Therefore, it will be better if we find an optimal population size at which the performance of the learning algorithm saturates.

In order to achieve this, we ran GA with scale-free prior on network with 50 and 100 nodes; and data generated from Mendes model. The γ for scale-free prior was set to 2.3. The population size was varied from the set $\{100, 250, 500, 750, 1000\}$. The AUC obtained for each of them is plotted in the following figure:

We observe in Figure 5.1 that the AUC for both 50 and 100 nodes saturates at a population size of 500. Increasing the population size after 500 leads to no or very insignificant change in the AUC. We therefore, fix the population size to 500 for all our future course of experiments.

5.4 PERMUTATION COUNT FOR SCALE-FREE PRIOR

As discussed in section 3.2, we use the method of averaging over several node permutations to obtain the final prior value. In order to obtain an optimal choice of the permutation count, we performed an experiment with



(a) 50 nodes (b) 100 nodes

Figure 5.1: AUC with varying population size

different permutation counts to see at what count the prior value saturates. The permutation count was varied from the set $\{100, 250, 500, 750, 1000\}$, the data was generated from Mendes model for a network size of 50 and 100 nodes. The results are tabulated below:

Permutation Count	Prior Value, N=50	Prior Value, N=100
100	-42.84	-72.25
250	-41.84	-70.01
500	-41.55	-69.53
750	-41.34	-69.21
1000	-41.32	-69.19

Table 5.1: Prior values for varying permutation count

We observe from the Table 5.1 that variance among different permutation counts for a given graph size is very less and it almost converges after a permutation count of 250 for both N=50 and N=100. Thus a value of 500 as the permutation count seems a reasonably good choice for future experiments.

The method of averaging over permutations is not an standard one but gives us a close to actual prior value of the graph. The greater the number of permutations, we achieve a more precise value of prior. There is yet another approximate method for calculating $P(G)$. Under this method the nodes in G having greater number of nodes are assigned a lower index i. This method tries to maximise $P(G|\sigma)$ instead of averaging $P(G|\sigma)$.

5.5 GA vs MCMC

In section 4.1 we talked about the need for sampling algorithm. A choice was to be made between GA and MCMC as our sampling-based algorithm for our future experiments. In order to achieve this, we ran both GA and MCMC for N=10 and N=20 nodes, believing that the one which performs well in smaller networks will also perform well in bigger networks. The data was generated using Mendes model. The γ for scale-free prior was set to 2.3. The network reconstruction was done with both uniform as well as scale-free prior. The results are tabulated below:

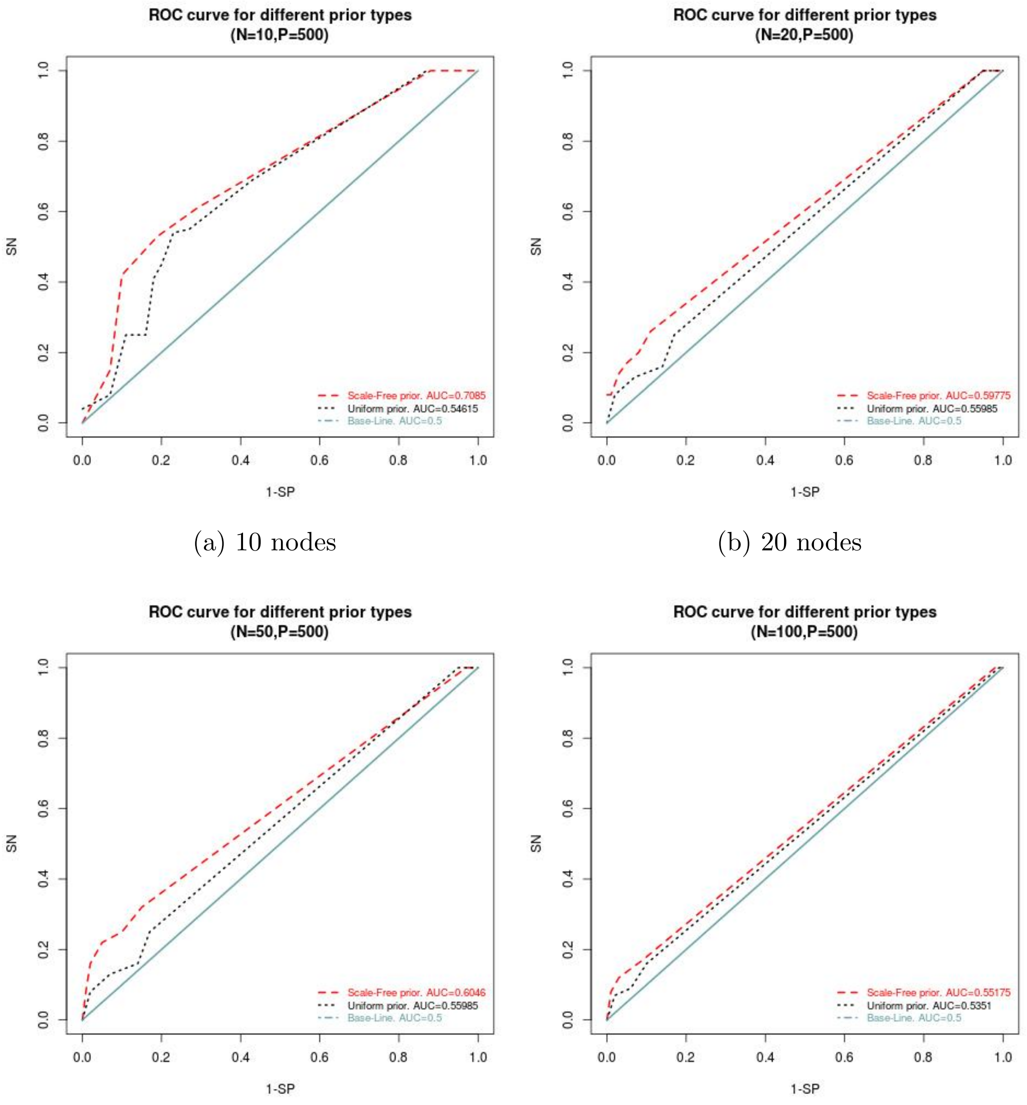
Experiment Setting	AUC for MCMC	AUC for GA
N=10, Uniform prior	0.518	0.633
N=10, Scale-free prior	0.569	0.675
N=20, Uniform prior	0.504	0.540
N=20, Scale-free prior	0.531	0.578

Table 5.2: AUC comparison for GA and MCMC

We observe in Table 5.2, that GA outperforms MCMC in both N=10 and N=20 network sizes and with either of the prior type. We therefore selected GA as our sampling-based algorithm. Thus, for all our future course of experiments we will be using GA with a population size of 500.

5.6 UNIFORM vs SCALE-FREE PRIOR

In order to compare uniform prior and scale-free prior, we ran GA with both types of prior on the synthetic data. The population size was set to 500. The data was generated for $\{10, 20, 50, 100\}$ nodes using both Normal sampling as well as Mendes model. The network from which data was sampled, was generated using scale-free structure with $\gamma = 2.3$. The γ parameter of the scale-free prior in GA for reconstructing the network was also set to 2.3. ROC analysis was done on each of them. The ROC curves are plotted below:



(a) 10 nodes

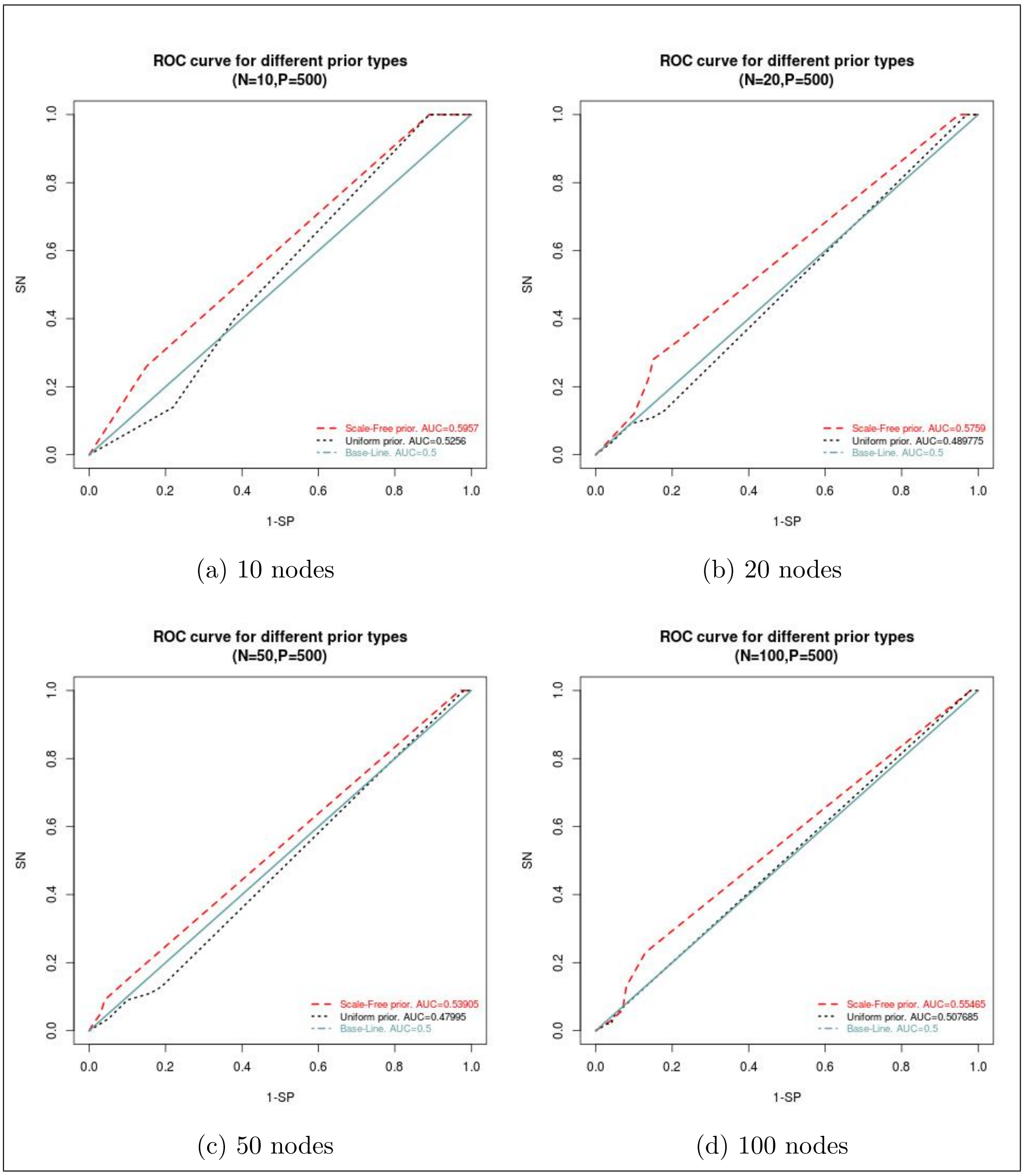
(b) 20 nodes

(c) 50 nodes

(d) 100 nodes

Figure 5.2: ROC curves for different prior types (Normally sampled data)

We observe in Figure 5.2 and Figure 5.3 that scale-free prior outperforms uniform prior under all the given experiment setting. This suggests that when the original network has a scale-free structure, combining the information from data with scale-free prior can help improve performance than using the



(a) 10 nodes

(b) 20 nodes

(c) 50 nodes

(d) 100 nodes

Figure 5.3: ROC curves for different prior types (Mendes model data)

information from data only (using uniform prior).

5.7 FINE GRAINED SAMPLING OF γ

The use of scale-free prior in GA for network reconstruction depends on the choice of scale-free parameter γ. In most of the real scenarios, this γ is not known in advance. In this section we try to study the sensitivity of scale-free prior performance to γ. The data is generated similar to previous section for 100 nodes using both Normal sampling and Mendes data. The scale-free parameter (γ) of data generating network is set to 2.3. The γ of scale-free prior in GA is varied from the set $\{1.7, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 3.0\}$. The ROC analysis is done for each of them and the AUCs obtained are tabulated below:

Experiment Setting	N=100, Normal data	N=100, Mendes model
Uniform prior	0.522	0.512
$\gamma = 1.7$	0.521	0.512
$\gamma = 2.0$	0.517	0.517
$\gamma = 2.1$	0.539	0.527
$\gamma = 2.2$	0.550	0.530
$\gamma = 2.3$	0.573	0.537
$\gamma = 2.4$	0.543	0.525
$\gamma = 2.5$	0.538	0.513
$\gamma = 2.6$	0.500	0.513
$\gamma = 3.0$	0.526	0.510

Table 5.3: AUC comparison for varying scale-free parameter(γ)

We observe that scale-free prior's performance is quite sensitive to the choice of γ. Indeed, within a delta of 0.2 around the original scale-free parameter ($\gamma = 2.3$), the algorithms performs reasonably well relative to the uniform prior. Thus, we get a range under which we can expect the scale-free prior to work well, i.e. around 0.2 delta of original γ.

5.8 COMPARISON WITH OTHER METHODS

We compared our GA augmented with scale-free prior with the state of the art network learning algorithm *ARACNE*, as well as another version of Bayesian inference called *bnlearn*. They are briefly discussed below:

ARACNE: This network inference algorithm was proposed by Margolin and Basso et al. [19]. This learning method uses information theoretic approach to reconstruct GRN. It works under the assumption that all gene dependency can be inferred from pairwise statistical information and that no higher order ananlysis needs to be done.

ARACNE follows Data Processing Inequality(DPI) to reconstruct GRN. It first finds the pairwise mutual information between all the genes. Then it creates a graph in which gene g_i is connected to gene g_j, if $MI(g_i, g_j) > I_0$, where I_0 is some threshold value. Then it considers all the gene triplets in this thresholded graph and removes the edge with the minimum value.

bnlearn: This is a newly developed and extensively used implementation of Bayesian inference that was originally developed as an *R* package [25]. It provides options for using analytical expression for likelihood optimisation from data. Thus it is nothing but a more comprehensive way of doing likelihood optimisation for network inference.

Synthetic data for generated for 10,20,50 and 100 nodes using scale-free networks with scale-free parameter of 2.3. Network reconstruction was performed using *bnlearn*, *ARACNE*, GA with uniform prior and GA with scale-free prior ($\gamma = 2.3$). The results avaraged over five different runs are tabulated below:

Experiement Setting	bnlearn	ARACNE	GA(uniform)	GA(scale-free)
N=10, Normal data	0.625	0.710	0.633	0.675
N=20, Normal data	0.554	0.641	0.540	0.578
N=50, Normal data	0.568	0.636	0.545	0.571
N=100, Normal data	0.529	0.613	0.522	0.573
N=10, Mendes data	0.522	0.685	0.526	0.621
N=20, Mendes data	0.520	0.622	0.513	0.558
N=50, Mendes data	0.514	0.596	0.490	0.535
N=100, Mendes data	0.515	0.589	0.512	0.537

Table 5.4: AUC comparison for different learning methods

We observe in Table 5.4 that ARACNE outperforms all the learning algorithms. However, we observe that GA with uniform prior performs marginally suboptimal compared to bnlearn, and that GA with scale-free prior performs better than bnlearn.

5.9 PRECISION-RECALL CURVES

As yet another performance metric, we evaluated the scale-free prior against other algorithms using precision and recall, to plot Precison-Recall curves(PRCs). Using the defintion of N_{TP}, N_{FP}, N_{TN} and N_{FN} from section 5.2, we define *Recall* and *Precision* as:

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{5.4}$$

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{5.5}$$

Recall is basically the fraction of true interactions correctly inferred by the algorithm, whereas *Precision* is the fraction of true interactions among all the inferred ones.

Different precision and recall values were obtained for the Bayesian inference algorithm by varying the probability threshold, identical to that done for SN and SP curve in previous sections. PRC for ARACNE was obtained using different Mutual Information (MI) threshold.

This analysis was done on data generated using mendes model for 100 nodes. The scale-free parameter of the generating network was set to 2.3. For GA, the learning was done using a population size of 500, and the γ for scale-free prior set to 2.3. The PRCs are shown below:

We observe in Figure 5.4 that ARACNE outperforms all other algorithms. We also observe that GA with scale-free prior is somewhat better than bnlearn and GA with uniform prior. However, we see that the recall for ARACNE never reaches 1. This is because the *data processing inequality (DPI)* used in ARACNE eliminates some interaction even at very low MI threshold. A similar pattern was studied by Margolin and Basso et al. [19].

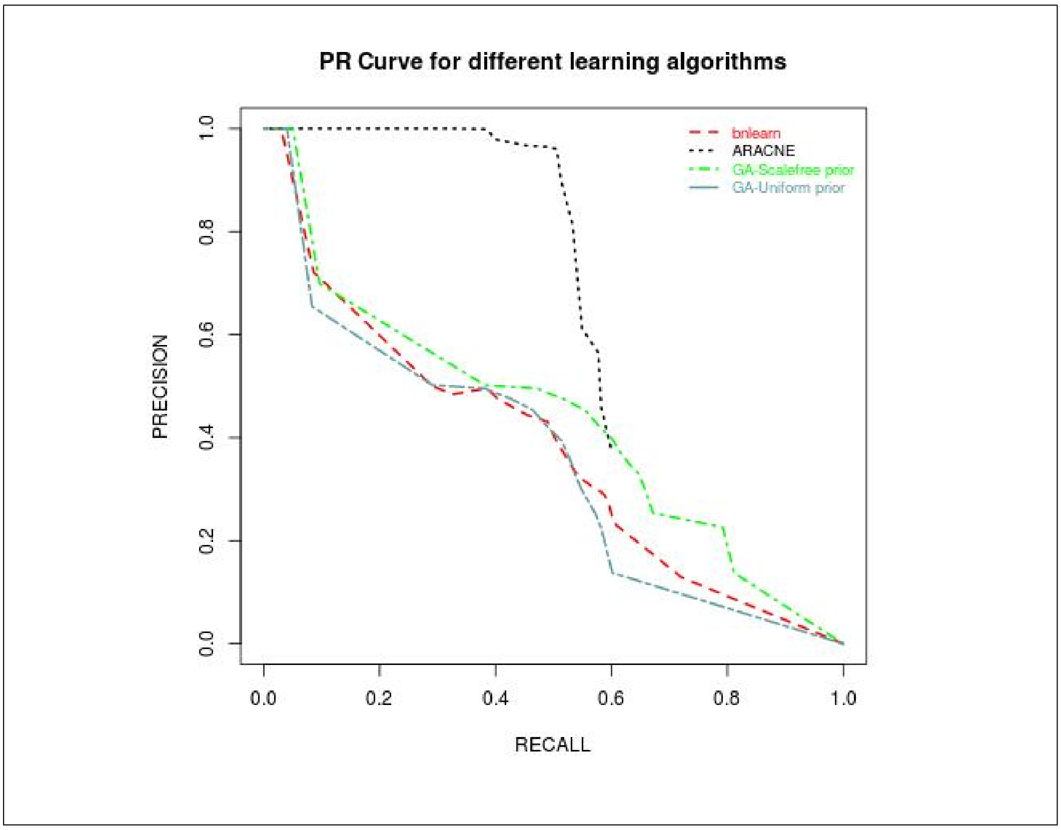


Figure 5.4: Precision Recall Curve

5.10 REAL DATA

After testing our scale-free prior on synthetic data, we evaluated it using real data. The HCC1954 data was used for this purpose. The details are discussed in following subsections.

5.10.1 HCC1954 data

HCC1954 is human breast cancer cell lining data. The dataset contains expression data for 31 different genes. The GRN for HCC1954 is not know in advance. However, there are some very well established interaction for this breast cancer cell. They are enumerated below (→ indicates activation and ⊣ indicates inhibition):

1. HRG→ERBB1 [23]

2. EGF/HRF→ERBB2/3 [12]

3. HRG→PKC_α [17]

4. EGF→p38 [13]

5. MAPK signalling cascade (EGF→AKT⊣ GSK3$_\alpha$) [14]

6. ERBB3→SRC [18]

7. ERBB3→PDK1 [6]

8. MEK1/2→ERK1/2 [14]

5.10.2 Results on HCC1954

GA with scale-free(SF) prior was run on HCC1954 dataset. The γ for scale-free prior was varied from the set $\{1.7, 2.0, 2.3, 2.5, 3.0\}$. GA with uniform(UNI) prior and ARACNE was also run on this dataset.

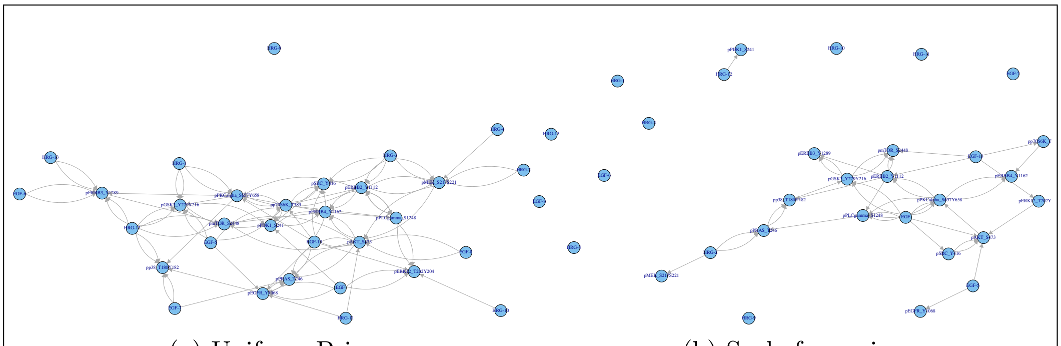
In order to compare the performance we report the number of interactions inferred among the enumerated 8 interactions in previous subsection. We also report the total number of interaction inferred, which will tell us the precision of algorithm.

Algorithm	# True interactions inferred	# Total interactions inferred
UNIFORM	3	92
SF, $\gamma = 1.7$	6	45
SF, $\gamma = 2.0$	4	41
SF, $\gamma = 2.3$	5	37
SF, $\gamma = 2.5$	5	37
SF, $\gamma = 3.0$	5	34
ARACNE	4	29

Table 5.5: Results on HCC1954 data

We observe in Table 5.5 that scale-free prior with $\gamma = 1.7$ infers the maximum number of true interactions. However, this is achieved at the cost of a higher number of interactions being predicted, suggesting that it has a lower precision. A similar pattern is observed in other γs of scale-free prior. Uniform prior performed worst with least number of true interactions and maximum number of total interactions being predicted. ARACNE has least number of interactions being predicted but also has a lower number of true interactions being inferred. Thus, we can conclude that scale-free prior within certain margin of error performs reasonably well (certainly better than uniform prior).

Further, to pictorially visualise the effect of scale-free prior we plot the inferred GRN for HCC1954 using both uniform and scale-free prior. The networks are shown below.



(a) Uniform Prior

(b) Scale-free prior

Figure 5.5: Inferred GRNs on HCC1954

We observe in Figure 5.5 that GRN learned with uniform prior is very dense with lots of edges. On the other hand, GRN learned with scale-free prior has lesser number of edges and there are distinctly visible interaction hubs present. Thus, deploying scale-free prior in Bayesian inference helps construct networks that are rich in scale-free property.

Chapter 6

Conclusion and Future work

In this thesis we explored into gene regulatory network reconstruction using Bayesian inference. The conventional approach has been to use only the information gained from the expression data, giving uniform importance to all the candidate networks. This may correspond to bias towards unrealistic networks. Therefore, we try to include prior information in network learning. There are lots of expert knowledge available, particularly when it comes to the field of gene regulatory networks. We try to incorporate one such property called the scale-free property for network reconstruction.

We discuss the theoretical background for Bayesian inference in chapter 2. We then motivated for the incorporation of scale-free prior in Bayesian inference for GRN reconstruction in chapter 3. The scale-free prior is formulated for networks with a given structure. In chapter 4 we talk about the need for sampling algorithm and the choice of genetic algorithm for network reconstruction. We also talk about the inclusion of scale-free prior in genetic algorithm that favours learning of networks that are rich in scale-free property.

In chapter 5 we perform various experiments with synthetic and real data to establish the importance of scale-free prior. We show that under the condition when networks have scale-free nature, learning with scale-free prior performs better than learning with uniform prior. However, there is still a large performance gap between ARACNE and Bayesian learning with scale-free prior. We also ran our scale-free prior on HCC1954 breast cancer cell data and find that it retrieves a significant fraction of the known interactions.

In this thesis we have shown that the use of scale-free prior in Bayesian inference can help improve network learning performance under the assumption that the network being learned has scale-free property. This assumption is backed by several studies discussed in chapter 3. We have shown the advantage of using sampling based algorithm like GA over MCMC. The scale-free prior is clubbed with GA, and a quantitative study is done on network learning algorithms which was not done previously. We rank several algorithms and find that there is substantial performance gap between ARACNE and Bayesian inference algorithms. This motivates us for the future work of this project and it is discussed in paragraphs to follow.

This work on priors can be extended in a number of ways. Node ordering independent scale-free prior could be a good addition to look into in future. We have explored one feature of the network called scale-free property. There are several other high-level network features in biological networks like *network motifs* that has been discussed by Milo et al. [21]. We could think of formulating priors for such features. Also, we can think of improving ARACNE's performance by allowing it to construct networks that favor scale-free structure. We will need to introduce scale-free scoring parameter in ARACNE's information theoretic approach, which will enable it to perform even better on scale-free networks.

Bibliography

[1] Réka Albert. Boolean Modeling of Genetic Regulatory Networks. pages 459–481. 2004.

[2] Christian Bender, Frauke Henjes, Holger Fröhlich, Stefan Wiemann, Ulrike Korf, and Tim Beißbarth. Dynamic deterministic effects propagation networks: learning signalling pathways from longitudinal protein array data. *Bioinformatics*, 26(18):i596–i602, September 2010.

[3] Christian Bender, Silvia, Frauke Henjes, Stefan Wiemann, Ulrike Korf, and Tim Beissbarth. Inferring signalling networks from longitudinal data using sampling based approaches in the R-package 'ddepn'. *BMC Bioinformatics*, 12(1):291+, 2011.

[4] Jiguo Cao and Hongyu Zhao. Estimating dynamic models for gene regulation networks. *Bioinformatics*, pages btn246+, May 2008.

[5] Gary A. Churchill. Fundamentals of experimental design for cDNA microarrays. *Nat Genet*, 32 Suppl:490–495, December 2002.

[6] P. Cohen, D. R. Alessi, and D. A. Cross. PDK1, one of the missing links in insulin signal transduction? *FEBS Lett.*, 410(1):3–10, Jun 1997.

[7] Eric H. Davidson and Douglas H. Erwin. Gene regulatory networks and the evolution of animal body plans. *Science*, 311(5762):796–800, February 2006.

[8] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of computational biology : a journal of computational molecular cell biology*, 7(3-4):601–620, August 2000.

[9] Paolo Giudici and Robert Castelo. Improving Markov Chain Monte Carlo Model Search for Data Mining. *Machine Learning*, 50(1):127–158, January 2003.

[10] David Heckerman. A Tutorial on Learning with Bayesian Networks. In Dawn Holmes and Lakhmi Jain, editors, *Innovations in Bayesian Networks*, volume 156 of *Studies in Computational Intelligence*, chapter 3, pages 33–82. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2008.

[11] Jan-Hendrik S. Hofmeyr and Hofmeyr Cornish-Bowden. The reversible Hill equation: how to incorporate cooperative enzymes into metabolic models. *Comput. Appl. Biosci.*, 13(4):377–385, August 1997.

[12] J. T. Jones, R. W. Akita, and M. X. Sliwkowski. Binding specificities and affinities of egf domains for ErbB receptors. *FEBS Lett.*, 447(2-3):227–231, Mar 1999.

[13] I. Y. Kim, H. Y. Yong, K. W. Kang, and A. Moon. Overexpression of ErbB2 induces invasion of MCF10A human breast epithelial cells via MMP-9. *Cancer Lett.*, 275(2):227–233, Mar 2009.

[14] J. W. Kim, S. S. Sim, U. H. Kim, S. Nishibe, M. I. Wahl, G. Carpenter, and S. G. Rhee. Tyrosine residues in bovine phospholipase C-gamma phosphorylated by the epidermal growth factor receptor in vitro. *J. Biol. Chem.*, 265(7):3940–3943, Mar 1990.

[15] Steffen L. Lauritzen. *Graphical Models.* Oxford Statistical Science Series. Oxford University Press, New York, USA, July 1996.

[16] Lesley T. Macneil and Albertha J. Walhout. Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. *Genome research*, March 2011.

[17] Campaner S Campiglio M Pilotti S Mnard S Tagliabue E Magnifico A1, Albano L. Protein kinase Calpha determines HER2 fate in breast carcinoma cells with HER2 protein overexpression without gene amplification. *Cancer Res.*, 447(2-3):227–231, June 2007.

[18] W. Mao, R. Irby, D. Coppola, L. Fu, M. Wloch, J. Turner, H. Yu, R. Garcia, R. Jove, and T. J. Yeatman. Activation of c-Src by receptor tyrosine kinases in human colon cancer cells with high metastatic potential. *Oncogene*, 15(25):3083–3090, Dec 1997.

[19] Adam A. Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo D. Favera, and Andrea Califano. ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics*, 7(Suppl 1):S7+, March 2006.

[20] Pedro Mendes, Wei Sha, and Keying Ye. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19(suppl 2):ii122–ii129, September 2003.

[21] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002.

[22] M. Newman. The structure and function of complex networks, 2003.

[23] M. A. Olayioye, I. Beuvink, K. Horsch, J. M. Daly, and N. E. Hynes. ErbB receptor-induced activation of stat transcription factors is mediated by Src tyrosine kinases. *J. Biol. Chem.*, 274(24):17209–17218, Jun 1999.

[24] Andrea Pinna, Nicola Soranzo, Ina Hoeschele, and Alberto de la Fuente. Simulating systems genetics data with SysGenSIM. *Bioinformatics (Oxford, England)*, 27(17):2459–2462, September 2011.

[25] Marco Scutari. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3):1–22, July 2010.

[26] Paul Sheridan, Takeshi Kamimura, and Hidetoshi Shimodaira. A Scale-Free Structure Prior for Graphical Models with Applications in Functional Genomics. *PLoS ONE*, 5(11):e13580+, November 2010.

[27] Christian Spieth, Rene Worzischek, and Felix Streichert. Comparing evolutionary algorithms on the problem of network inference. In Mike Cattolico, editor, *GECCO*, pages 305–306. ACM, 2006.

[28] M. Wahde and J. Hertz. Coarse-grained reverse engineering of genetic regulatory networks. *Bio Systems*, 55(1-3):129–136, February 2000.

[29] 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.