AUGMENTING GENE NETWORK INFERENCE USING META-ANALYSIS AND STRUCTURAL PRIORS

TARUN MAHAJAN

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY DELHI

DECEMBER 2017

Augmenting Gene Network Inference using Meta-Analysis and Structural Priors

by

Tarun Mahajan

Department of Electrical Engineering

Submitted

in fulfillment of the requirements of the degree of

Master of Science (Research)

to the

Indian Institute of Technology Delhi

December 2017

Certificate

This is to certify that the thesis titled, "**Augmenting Gene Network Inference using Meta-Analysis and Structural Priors**" is a bona fide record of work done at Indian Institute of Technology, Delhi by **Mr.Tarun Mahajan**, for the award of the degree of **Master of Science (Research)** in Electrical Engineering. The work in this thesis was conducted under our supervision and guidance. The results contained in the thesis have not been submitted elsewhere, wholly or in part for the award of any other degree.

Dr. Sumeet Agarwal
Assistant Professor
Department of Electrical Engineering,
Indian Institute of Technology Delhi,
Hauz Khas, New Delhi - 110016, India

Dr. Jayadeva
Professor
Department of Electrical Engineering,
Indian Institute of Technology Delhi,
Hauz Khas, New Delhi - 110016, India

Acknowledgements

First and foremost, I would like to express gratitude towards my supervisors Professors Sumeet Agarwal and Jayadeva, for their consistent support and guidance. They have been a beacon of light through every step of my masters, from the invaluable insights to the support with my health issues. The flexibility they have afforded me in progressing in accordance with my predisposition has been instrumental to this work. This thesis would not have been possible without them.

I would be remiss without explicit mention of Mr. Divyanshu Mahajan and Mr. Kshitij Rai. I am indebted to them for the love and friendship they have showered on me. Kshitij has also been kind to indulge my inquiries regarding various principles of molecular biology with utmost patience. Divyanshu has been a vitalizing presence throughout my masters. Contrary to his role as the younger brother, he has been gracious and sweet to adjust his busy work schedule to take care of me in moments of sickness. I express my gratitude to them with great fondness and love.

To Ms. Aishwarya Singh Kashyap, I feel a deep sense of appreciation and love. In addition to being an inexhaustible source of emotional strength and warmth, she has most patiently dealt with the fits of stress that have bared their fangs at times. For the different roles she has played, from keeping me honest to self-imposed deadlines and to proof-reading this draft of the thesis, I am forever in debt of her. I wholeheartedly thank her for being a part of my life.

I would like to dedicate this work to my parents, Mrs. Parveen Kumari and Mr. Mohinder Kumar. I owe them everything. The great struggles and pains they

have endured to offer my brother and I the best they could, are eternally etched in my mind. The unconditional love that ceaselessly exists has been my source of inspiration and strength. Mom and Dad, thank you for your love and support.

Finally, I would like to thank all those who could not be mentioned explicitly due to shortage of space, but have assisted and eased my stay at IIT Delhi.

Tarun Mahajan

Abstract

High-throughput omics data is pouring out in copious amounts. It offers measurements for cell components at different stages of the central dogmatic view of information flow. This data can be used to construct a network level understanding of biological systems and functions such as inferring a gene regulatory network. Many computational techniques are readily available to infer gene regulatory networks from high-throughput experiments. One suggested way of improving the performance of these techniques is meta-analysis—combining the predictions from different methods. In this regard, we propose a meta-strategy for combining methods and show that this strategy might help with the inference task both globally and locally. Another proposed way for improving the performance is the use of domain-specific prior information such as the distribution of the number of regulatory links in biological networks. However, there hasn't been a systematic analysis of either the methods that incorporate such structural priors or the priors themselves in the network inference field. We perform a systematic study of three different ways of incorporating such prior information in the network inference task. One of these has been developed as part of this work, leveraging an existing framework. The analysis shows that the prior information helps both globally and locally in network inference. We further explore the possibility of incorporating structural motif related properties into the scale free prior; and suggest possible frameworks to accomplish this. Finally, we propose a framework for including both indegree and outdegree distributions as prior to augment the network inference task.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Sequencing the genome for many organisms at the dawn of 21st century was a remarkable feat. It has ushered the era of "Omics" technologies where high-throughput data is readily available at all levels of the "Central Dogma". At the genome scale there is Gene Sequencing data. At the transcriptome scale, Microarray and RNAseq technologies assist in measuring the expression of thousands of transcription units simultaneously without loss in quality. Proteomics data can be collected using hybrid systems such as the yeast two-hybrid system, Chip-based techniques and Mass Spectrometry-based methods. Some or all of these data sources can be leveraged to answer a multitude of biological questions such as regulation of various cellular mechanisms, interaction between cellular components, understanding signalling networks, prediction of cellular phenotypes, etc. This postgenomic world has seen the birth of Systems Biology. Consequently, biology has invariably moved away from reductionism; the focus has shifted from individual molecules and components to the systematic study of organization of these components into complex networks, which lie at the heart of cellular functions [1]. Depending on the nature of the interacting components, we can have a variety of networks: protein-protein interaction networks, signalling networks,

metabolic networks, transcription regulatory networks etc [2]. Amongst these, a transcription regulatory network or gene regulatory network (as it is most commonly referred to) represents a group of genomic sequences (genes or partial genes); which are engaged in interactions, through intermediary components (proteins and molecules), to create a spatial and temporal pattern of expression of the involved genes [3].

The computational vantage point associates the phrase "gene regulatory network" (GRN) to the web of connections between genes inferred from sequencing data, such as microarray data [4]. Networks identified by such an approach have applications ranging from identifying important modules in cellular functions [5, 6, 7] to study of diseases [7, 8, 9] and drug design [4]. Interest in network inference task has been increasing exponentially over the years [10]. Thus, a systematic study of different aspects of the inference task has value in furthering a better understanding of the advantages and shortcomings of different methods. There have been extensive studies with regards to the network inference task in general [10, 11, 12, 13, 14].

However, a principled analysis of approaches that leverage relevant biological information in addition to the expression data has been missing. Structural properties are one source of additional biological information. Gene regulatory networks are known to have rich topological properties such as modularity [2], scale free degree distribution [2, 15], hierarchy [2, 16], robustness [2] and existence of motifs [17, 18], to name some. The network inference task is a high dimensional problem with a prohibitively large solution space [19]. The available data has low information content, thus solving the problem without regard to these structural constraints is bound to give solutions which might not capture one or more crucial properties of the underlying regulatory networks. Thus, it is of paramount importance that known structural properties are used to trim the search space and arrive at solutions that are biologically plausible.

The present work is motivated by the need for incorporating structural information in the network inference task. We specifically look at one such property, the degree distribution of gene regulatory networks. Gene regulatory networks are known to have scale free degree distribution [2, 15]. In the present work, we offer a brief review and analysis of some of the methods that try to utilize the degree distribution as a form of constraint or prior information to guide the search for a gene regulatory network.

1.2 Organization of the Thesis

The present work has been divided into five chapters in addition to the current one. Chapter 2 introduces the benchmarking process in general. Specifically, the networks and datasets used in this work have been described. Finally, different global and local measures of assessing the performance of network inference tasks are introduced. Chapter 3 begins by formalizing the task of gene network inference, followed by a brief overview of the network inference task in general. A characterization based on the approximation approach used by the network inference tasks is introduced. The chapter ends with the discussion of the experiments that we have conducted to study the properties of some of the widely used network inference methods. Chapter 4 introduces two meta-analytic strategies for combining predictions made by different network inference tasks. The effect of these strategies is studied using the measures of assessment introduced in Chapter 2. The chapter closes by talking about possible directions for future research. Chapter 5 gives a brief overview of the state of network inference task with respect to the use of scale free degree distribution as a source of prior information. Some of the methods that leverage the scale free prior are discussed. We also introduce a simulated annealing based method for leveraging an existing framework for inferring scale free networks. Comparative analysis is conducted against two other scale free prior based methods. A meta strategy for combining the predictions of introduced simulated annealing method with any other given method is proposed.

The introduced meta strategy is analyzed using the networks and performance measures introduced in Chapter 2. The chapter ends with a discussion of possible future directions for research. The last chapter talks about the central tenets of the current thesis and grounds them within the context of the results presented in the previous chapters. The work of the thesis is briefly summarized. The chapter closes by ruminating over the place of this thesis within the space of the network inference research.

1.3 Conclusion

The present thesis is guided towards analyzing the question, "Do structural priors aid the task of gene network inference?" Consequently, we are also concerned with studying the specific effects of capitilizing structural priors, especially degree distribution, on the network inference task. Given the absence of systematic inquiries into these queries, we offer a brief discussion and analysis of some of the methods which try to incorporate a scale free structure prior in the network inference pipeline. To this end, we conduct multiple experiments to elicit global and local behaviours of the degree distribution prior, which have been described in detail in Chapter 5. As a complementary study, we also do a brief analysis of some of the currently available network inference methods. Based on the experiments conducted on these methods, we suggest strategies based on meta analysis to aid the network inference task. These and other experiments are discussed in the following chapters.

Chapter 2

Gene Network Inference Benchmarking

2.1 Introduction

Research in gene regulatory network inference techniques has been doubling every year [10]. With the wide variety of available methods, there has been growing interest in benchmarking the performance [10, 11, 12, 13, 14]. Systematic analyses of global and local properties of some of the methods have been conducted. The DREAM challenges [10, 11, 12] are at the forefront, utilizing both synthetically and experimentally generated datasets. Narendra *et al.* [13] conducted another comprehensive assessment of the global properties of various methods on four different assessment metrics. These attempts at characterizing the performance of different inference methodologies on different data types has generated some useful insights into the global and local behaviours. For instance, one consistent learning from the DREAM challenges is that an ensemble aggregate of many methods, including individually poor performers, is better than any individual method [12]. Previously, the DREAM3 challenge [11] had examined the performance on different indegree and outdegree edges. Further, motif-level local properties have also been studied [11, 12]. However, inferences drawn from these studies are contingent on

the datasets and the evaluation metrics being employed. Thus, expression data and evaluation metrics are two crucial components of the benchmarking process.

Employing data sets that faithfully capture the key properties of biologically existing networks is of paramount importance. Another central player in the assessment process is the set of evaluation metrics leveraged to assign a quantitative and/or qualitative measure of performance to extricate the important properties of the inference methods. In the following sections, we offer brief discussions pertaining to both components of the benchmarking process within the context of this work.

2.2 Data

The aspired aim of the network inference research is to identify important regulatory relationships for an organism, given the expression data under a set of conditions; in turn, detailed experimental interventions can be designed to prod these predicted regulatory links. If successful, network inference methods would usher momentous progress in systems biology-based solutions to current problems such as identifying important modules in cellular functions [5, 6, 7], study of diseases such as cancer [7, 8, 9] and drug design [4]. To be able to faithfully tackle these tasks, we need to have a clear understanding of the strengths and weaknesses of different network inference methods; and for this purpose we need networks for which the true underlying topology of regulatory interactions are known.

The data for the benchmarking process has two components-a matrix of expression values and the topology of interactions between the genes that generates the mRNA expression data. Three different sources have been leveraged to generate benchmark networks [20]-real biological networks constructed from in-vivo studies and maintained in curated databases [21], synthetic genetic networks constructed in-vivo in the lab [22] and in-silico artificial networks [10, 11, 12, 13, 14]. Each

of the sources has its strengths and weaknesses. Though networks constructed from in-vivo experimental studies are representative of the naturally occurring networks, interactions contained therein are only a subset of all the interactions. Thus, such networks would be ill-suited for the benchmarking process, since there would be many false negatives in such a gold standard. In such a scenario, false positives identified by a given network inference method wouldn't necessarily imply an error on part of the inference technique, rather these could be potential novel regulatory interactions [12].

Synthetic genetic networks are usually constructed in a bottom-up approach and thus have a specific design topology. Therefore, the true underlying interactions are known. The flip side is that all of the synthetically constructed pathways and networks have very few genes and interactions compared to naturally occurring networks. These networks would not capture topological properties, such as the degree distribution, modularity, distribution of motifs etc., inherent in naturally occurring networks. Finally, artificial networks based on different network models are a good alternative; the true topology is known and thus inferences about structural performances of a given method would be accurate. Further, these networks could be easily sampled from experimentally constructed networks so as to be representative of biological networks [20].

However, in terms of the expression data, the first two sources offer data that is of practical interest, since it is generated by the biological machinery of transcription and translation and other associated processes. Whereas in-silico expression data is generated using simplistic models of transcription and translation. Also, such models are only an approximation of the actual process of protein expression and inherently lack many complex processes found in the natural world such as post-transcriptional and post-translation changes, biological noise, etc. The in-vivo measured data is also riddled with issues. Generally, only a limited number of samples are available from a single experiment conducted under specific experimental conditions; thus data can be augmented by stacking together measurements from different experiments performed in the same and/or different labs

[12]. Though such augmentation increases the number of available sample points, however the noise in the data might increase.

Despite the aforementioned shortcomings, both experimentally collected and artifically simulated networks and expression data have been used for the purpose of benchmarking [10, 11, 12, 13, 14] to leverage the complementarity of these sources. Therefore, we also employ both sources of data in this work as well. Details pertaining to the network topologies and the corresponding expression data are explained in the following sections.

2.2.1 Network Topologies

Biological networks are known to be sparse random graphs with heavy-tailed degree distributions [2]. Contrary to simple random networks, gene regulatory networks have a scale-free degree distribution for the regulatory edges emanating from the genes [15]. This can be represented as shown in Eq 2.1.

$$P_{out}(k_{out}) \propto k_{out}^{-\gamma_{out}} \tag{2.1}$$

where, P is the probability of having nodes with outdegree k_{out} and γ_{out} is the exponent which usually assumes a value between 2 and 3. However, the indegree distribution can be represented as a limited exponential distribution [15] which might have the form given in Eq 2.2.

$$P_{in}(k_{in}) \propto exp(-\lambda_{in}.kin) \tag{2.2}$$

where, P is the probability of having nodes with indegree k_{in} and λ_{in} is the rate parameter of the distribution. The asymmetry between the indegree and outdegree distributions could be explained by the fact that genes are more likely to be regulating large number of genes than being regulated by many genes. Biologically, this amounts to the physical limitation on the available promoter sites on a gene.

Given these constraints, we have experimented with three different types of network categories, which together contain 25 medium size networks and 3 large-scale networks. These network categories are: Power law Indegree and Power law Outdegree (PIPO); Networks from DREAM Challenges; Exponential Indegree and Power law Outdegree (EIPO) with and without Modularity. The topological properties of these networks are described as follows.

1. **Power Law Indegree Power Law Outdegree (PIPO)** - This category consists of five networks of 100 genes each. The networks were created using the algorithm proposed in [23]. Both indegree and outdegree distributions are independently sampled from a scale-free distribution. The degree distributions are adjusted to ensure that they represent a graph; the sum of indegrees and outdegrees over all of the nodes are ensured to be equal. Finally, the edges are assigned to the nodes using a directed configuration model. The degree exponent for both indegree and outdegree distributions is set to a value of 2.5. Self-regulatory interactions were removed from the network. While generating expression data, it was assumed that 40 % of the edges in the network have a negative regulatory effect, while the rest had a positive regulatory effect. Fig. 2.1 shows the plot for the indegree and outdegree distributions for some of the networks used in this work.

2. **DREAM Synthetic** - Synthetic networks generated using random graph models such as Erdos Renyi, Strogatz-Watts or Barabasi-Albert don't necessarily capture all the structural properties of gene regulatory networks [20]. Thus, the networks for the DREAM challenges have been generated by extracting modules from known regulatory networks of *E. coli* and *S.cerevisiae* [20]. DREAM3 had five synthetic networks of size 100 in the network inference challenge [11]. Two networks are from *E. coli* and 3 from *S.cerevisiae*. DREAM4 had two different sub-challenges in the size 100 category-one subchallenge had data from knockout, knockdown and timeseries experiments, while the other provided multifactorial perturbation steady state data. In this work, we have used the latter for evaluating the performance of the

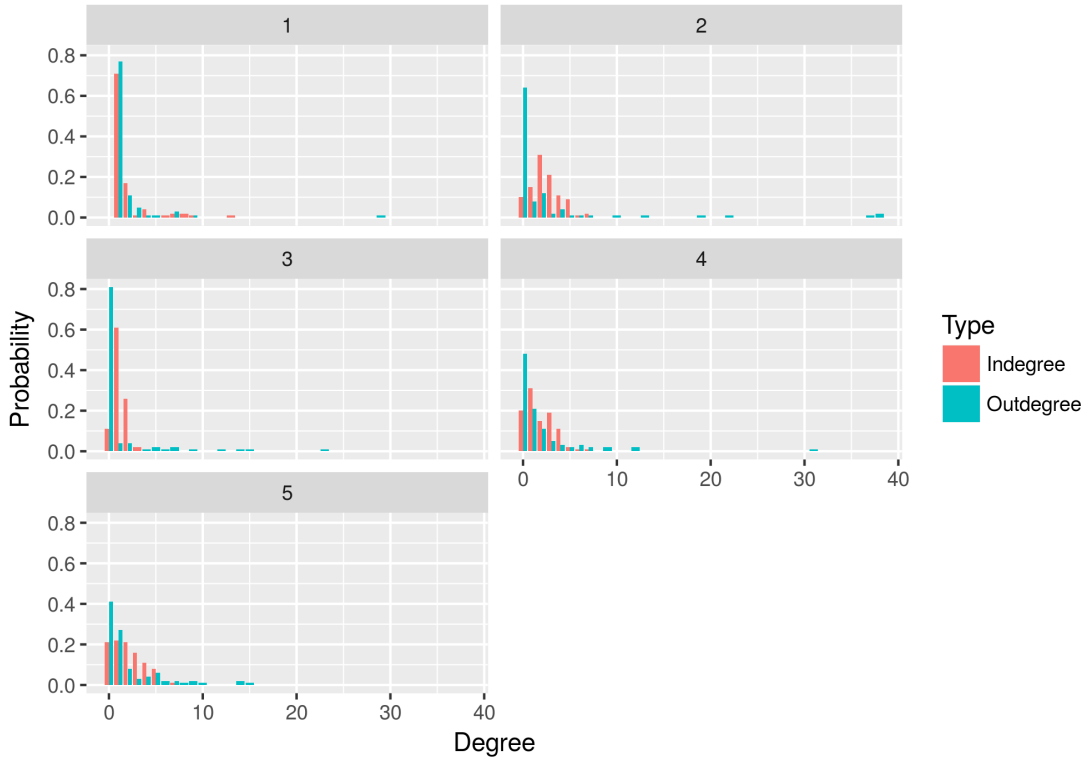


Fig. 2.1. Degree distriution for the size 100 networks.

The indegree and outdegree distributions have been presented for one network from each of the following topologies. 1:PIPO, 2:DREAM4, 3:DREAM3, 4:EIPO, 5:EIPO Modular. PIPO has scale free in and out degrees, thus both are symmetric. However, for all the other topologies indegree is exponential or roughly exponential and the outdegree is scale free. The asymmetry in degree distributions is visible for these topologies.

network inference approaches. Since multifactorial perturbation data was not provided for DREAM3, we generated steady-state multifactorial perturbation expression data using the tool GeneNetWeaver [20]. This tool has been provided by the organizers of the DREAM challenges and was used to generate data for the actual DREAM challenges.

We have also utilized two of the four DREAM5 networks. One of the networks is an artificial in-silico network extracted from the known topology of *E.coli* transcriptional regulatory network from RegulonDB [21] with an addition of 10 % random edges; expression data was simulated using GeneNetWeaver. The other network is a real biological network from *E.coli*. The gold standard for *E.coli* was created using the experimentally validated interactions available from the curated Database RegulonDB.

3. **Exponential Indegree Power Law Outdegree (EIPO)** - This category contains five networks of 100 nodes each. The networks were generated using the MATLAB based software tool SysGenSim [24]. Each network is generated to have exponential indegree, with rate parameter 0.5, and power law outdegree distributions following the scale-free property with an exponent of 2.1. Another set of five networks was generated with the same degree distributions and an additional property of modularity, with 5 modules, enforced on the created networks.

2.2.2 Expression Data

Expression data for all the in-silico networks has been generated using the tool GeneNetWeaver [20]. The network topologies are laden with a dynamical model of gene regulation, which accounts for both transcription and translation. The model incorporates the dynamics of both mRNA and protein concentrations. For a given network, the nodes constitute the genes. Thus, the mRNA level for a gene is controlled by the proteins expressed by its regulator genes. The entire regulatory web of interactions is encoded into a system of differential equations. Further details for the model can be found in [11]. To account for the inherently stochastic nature of molecular reactions, the noise in mRNA and protein concentrations is estimated using chemical Langevin equations. Measurement noise is also added to the generated data.

GeneNetWeaver is capable of generating different types of gene expression datasets incluing steady state and timeseries, which may include expression values for wild-type, knockout, knockdown, dual knockouts and multifactorial perturbation experiments. The details of these experiments are available in [11, 20]. For the 25 medium size networks we focus on steady state multifactorial perturbation data. In real biological systems, where the number of genes and gene products are quite large, multifactorial perturbation data is economical and more readily

available [25]. Single gene perturbation experiments such as single gene knockouts or knockdowns cannot provide faithful information regarding the combinatorial web of interactions on the level of a gene network [25]. Single-gene perturbations are carried out within the context of a fixed background. However, perturbing the system against one background does not offer information regrading the other backgrounds against which a single gene could be altered.

In light of this, multifactorial perturbation data, where the genes are present in a combination of different contexts, such as different gene mutants and/or varying environmental or experimental conditions is quite useful. The DREAM4 challenge consisted of a size 100 multifactorial perturbation sub-challenge; we have used this data for the DREAM4 networks. Expression data for all the other networks, including DREAM3, has been generated by means of GeneNetWeaver by setting the simulation conditions similar to those used for the DREAM4 challenge. For each network, hundred multifactorial simulations are done. Thus, for each network, the expression data is a matrix $X \in \mathbb{R}^{n \times p}$ with $n = 100$ and $p = 100$, where n and p are the number of experiments and the number of genes respectively.

The in-silico network in DREAM5 has also been simulated using GeneNetWeaver [12]. Expression data for *E.coli* was extracted from the gene expression data from the Gene Expression Omnibus (GEO) database [26]. In contrast to DREAM3 and DREAM4 challenges, DREAM5 provided a putative list of transcription factors and details about the experimental conditions for the microarray chips used for compiling the expression data (for instance, the perturbation introduced, target of a gene knockout, or the time point for a time-series experiment). The experimental conditions for the in-silico network was simulated so as to mimic the chip details for *E.coli*.

2.3 Evaluation Metrics

The second important component of the benchmarking process is assessment. We need metrics that can accurately capture the performance of the inference methods. In this work, we know the true edges that exist in the network and predictions from different methods are compared against this known network. Thus, many common metrics from binary classification problems have been leveraged for characterizing the performance of network inference methods such as area under the Precision Recall curve, area under the ROC curve, precision, recall, F-score, Positive Predictive Value (PPV), Negative Predictive Value (NPV), etc and various combinations of these metrics, [10, 11, 12, 13, 14]. These metrics identify any given method's capacity to separate edges or interactions between genes from the non-edges or non-interactions using the gene expression data. In addition to such global metrics, tailor-made metrics have been used to prod the local properties of the network inference methods; for instance, [11] constructs a metric to assess the behaviour of inference techniques on edges incident on genes with different indegrees and outdegress. This metric is based on normalized confidence values assigned by a method to the interactions between the genes in a given network. In the same study, confidence value-based metric is also utilized to characterize the performance of inference methods in identifying different three node motifs in the network. Both global and local metrics taken together afford a clear picture of the strengths and weaknesses of any given network inference technique.

In the present work, we adopt several global and local metrics to characterize the different properties of the network inference methods. We briefly discuss these metrics in the following sections.

2.3.1 Global Assessment

To gauge an unbiased estimate of the performance, we use metrics that assess performance across all recall values. For this explicit purpose, the following metrics

have been used.

- **Precision Recall (PR) Curve**
- **Receiver Operator Characteristics (ROC) Curve**
- **Area under the PR Curve (AUPR)**
- **Area under the ROC Curve (AUROC)**
- **Score** - This metric has been consistently used in the DREAM challenges [10, 11, 12]. It compares the performance of a given method against a null model obtained by assigning random confidence values (uniformly sampled between 0 and 1) to all possible interactions. The aggregate score is based on p-value calculation using a given number of random network predictions for each network in every dataset [10, 11, 12]. For DREAM4 networks, we use the null distribution provided by the organizers. For the other 20 networks, we use 30,000 random network predictions for each network to obtain the null distribution for AUPR and AUROC values. Under this null hypothesis p-values are computed for a given network inference method, which quantify the probability of having the same or better performance, in terms of AUPR and AUROC values, than the inference method under the null hypothesis. It is assumed that both AUPR and AUROC values are independent, thus only marginal distributions are used. The p-values can be converted into a score as follows;

$$Score_{AUPR_i} = -log_{10}(p_{AUPR_i}) \tag{2.3}$$

$$Score_{AUROC_i} = -log_{10}(p_{AUROC_i}) \tag{2.4}$$

where subscript i is for the ith network. If there are N networks, composite scores can be computed by taking the geometric mean of the corresponding

p-values and then computing the scores.

$$\hat{p}_{AUPR} = \left(\prod_{i=1}^{N} {p^i}_{AUPR} \right)^{1/N} \tag{2.5}$$

$$\hat{p}_{AUROC} = \left(\prod_{i=1}^{N} {p^i}_{AUROC} \right)^{1/N} \tag{2.6}$$

Similarly, overall Scores can be computed for individual networks or a group of networks by taking the geometric means of individual p-values (${p^i}_{AUPR}$ and ${p^i}_{AUROC}$, $i \in \{1, 2, \ldots, N\}$) or the geometric means of composite p-values ($\hat{p}_{AUPR}$ and $\hat{p}_{AUROC}$) and then computing the Scores.

- **Causal Inference** - Even with steady state perturbation data many methods infer asymmetric adjacency matrices. To assess the extent of causal information contained in an asymmetric prediction we use the plot of accuracy versus threshold as used in [27]. The adjacency matrix is thresholded by removing all the entries below a given threshold. For the remaining non-zero entries, if an asymmetric prediction is present (asymmetric predictions are those for which entry in the ith row and the jth column is non-zero while that in the jth row and the ith column is not), we measure how accurately the direction of causation is captured by the thresholded matrix. If T represents the total number of asymmetric interactions in the predicted adjacency matrix at a given threshold where gene i regulates gene j and not vice-versa, and of these T interactions, $Pred$ is the number of interactions which are also present in the gold network, accuracy is given by $\frac{Pred}{T}$. Varying the threshold over the entire possible range gives us a curve, and the area under the curve quantifies the extent of causal information captured by a given method.

2.3.2 Local Assessment

The most common local features of a graph are its degree distribution and motifs. GRNs are known to have heavy-tailed degree distributions, such as power-law, which suggests that only a few genes play the role of major regulators. Further, the power-law nature might suggest that these networks are robust to external perturbations. Another important property of GRNs is the existence of highly over-represented motif structures [17, 18], which have important sensory and developmental functions in a cell [28]. Thus, it is important for a good network inference methods to faithfully identify these properties of a GRN. We use different local measures of performance for assessing the impact of different methods on recovering these properties of a given gold standard network.

- **PR and ROC Curves for Indegree (Outdegree)** - A given network is broken down into mutually exclusive and exhaustive sets of nodes and corresponding incoming (outgoing) edges where each set contains nodes of a specified indegree (outdegree). Precision Recall curves can then be obtained for each of these sets independently. For instance, the edge set E can be divided in subsets E^u, $u \in \{1, 2, \ldots, d\}$ where set E^u contains all the edges incident (emanating) on (from) genes with indegree (outdegree) u and d is the maximum indegree (outdegree) in the GRN. For each subset, we can characterize the ability of a given method to identify the edges of indegree (outegree) u from the non-edges using Precision-Recall and ROC curves.
- **AUPR Indegree (Outdegree)**
- **AUROC Indegree (Outdegree)**
- **Degree Score (dScore)** - Similar to the Score defined for the AUPR and AUROC values for the entire edge set, a score value can also be calculated for the AUPR and AUROC values for a subset E^u with edges incident on (emanating from) indegree (outdegree) u nodes. For a given indegree (outdegree) u, we have calculated the null distribution by generating 10,000 random predictions. Corresponding p-values and the scores can be calculated for each

network and entire ensemble as discussed above. The dScore is calculated independently for indegree and outdegree edges. We observe in Chapter 3 that the performance for all the methods decreases exponential with indegree and linearly with outdegree.

Thus, to appropriately characterize these trends, we fit a straight line to the log transformed dScore for indegree against the indegree, and fit a straight line to the dScore values for outdegree against the outdegree. Corresponding to these straight line fits, we figure out the intercept and slope values; these characterize the behaviour for the degree distributions. Higher the intercept and lower the slope the better an inference method is able to capture the degree distribution of the underlying network. Intercept quantifies the performance over indegree or outdegree 1 edges, while the slope characterizes the decline in performance with indegree or outdegree.

For the purpose of comparing any two inference methods, we can calculate the degree at which the straight line fits for both the methods would cross each other. Assume that M_1 and M_2 are two methods with slopes s_1 and s_2 and intercepts I_1 and I_2 respectively. Further assume that d_{cross} is the degree at which they cross. If $d_{cross} < 0$ and $I_1 > I_2$, it implies that M_1 lies above (dominates) M_2 in the dScore vs degree plot. If $d_{cross} > 0$ and $I_1 > I_2$, then d_{cross} quantifies the positive degree up to which M_1 dominates M_2. If $d_{cross} < 0$ and $I_1 < I_2$, then M_2 always dominates M_1. And finally, if $d_{cross} > 0$ and $I_1 < I_2$, then M_1 dominates M_2 after degree d_{cross}.

- **Degree Separation (dSeparation)** - The dScore only quantifies the ability of a network inference method to identify the edges of different indegree or outdegree. To assess the competence of a method to capture the combinatorial nature of regulation, we need to look at the difference in the prediction confidences assigned to the edges incident on or emanating from a given gene. Consider again the subsets E^u, $u \in \{1, 2, \ldots, d\}$, where set E^u contains all the edges incident on genes with indegree (outdegree) u. For

indegree (outdegree) u, we identify all the genes in the subset E^u and calculate the dSeparation as given in Eqs 2.7 and 2.8 for indegree and outdegree edges respectively.

$$dSeparation = \frac{\sum_{i<i' \in E^u_{*j}} w_{ij} - w_{i'j}}{(u-1)p^u} \tag{2.7}$$

$$dSeparation = \frac{\sum_{j<j' \in E^u_{i*}} w_{ij} - w_{ij'}}{(u-1)p^u} \tag{2.8}$$

where E^u_{*j} is the set of genes regulating gene j, E^u_{i*} the set of genes being regulated by gene i, w_{ij} is the prediction confidence assigned to the edge from gene i to gene j by a given method and p^u is the number of genes in E^u. For each gene in E^u, all the edges incident on (emanating from) it are arranged in decreasing order by the prediction confidence and then dSeparation is calculated as given in Eqs 2.7 and 2.8. The obtained dSeparation values can be converted to a score similar to the score for AUPR and AUROC values by comparing against a null model. The null model distribution is obtained by randomly assigning confidence values to edges in the subset E^u and the process is repeated 10,000 times. A p-value is calculated as the probability of observing a dSeparation smaller than or equal to the observed dSeparation for a given method. The log-transformed p-value is finally used as a dSeparation score. Larger the dSeparation score the closer the edges incident upon or emanating from a gene with a given indegree or outdegree are. Aggregate scores can be computed as discussed before for AUPR and AUROC.

- **Motif Bias** - Evaluation of performance on three-node network motifs has been conducted in the DREAM challenges before [11, 12]. Here, we assess the performance on the same three-node motifs as used in the DREAM challenge. These motifs are Cascade, Fan-in, Fan-out and Feed Forward Loop (FFL). The motif topologies are shown in Fig. 2.2. The motif edges are assessed in terms of bias in AUROC against the global performance. For

each motif, the motif edges are compared against the non-edges to obtain an AUROC value and this AUROC value is compared with the AUROC value for the whole network. This quantifies to what extent a given method is able to identify a particular network motif. The motif errors for Cascade, Fan-in and Fan-out motifs, i.e., the edges not present in the motifs but predicted by a given method, are quantified by comparing the errors against all the non-edges. The motif error edges are assumed as the positive case and the set of all the non-edges as the negative case and an ROC curve can be obtained. This curve can be compared against that for a random prediction.

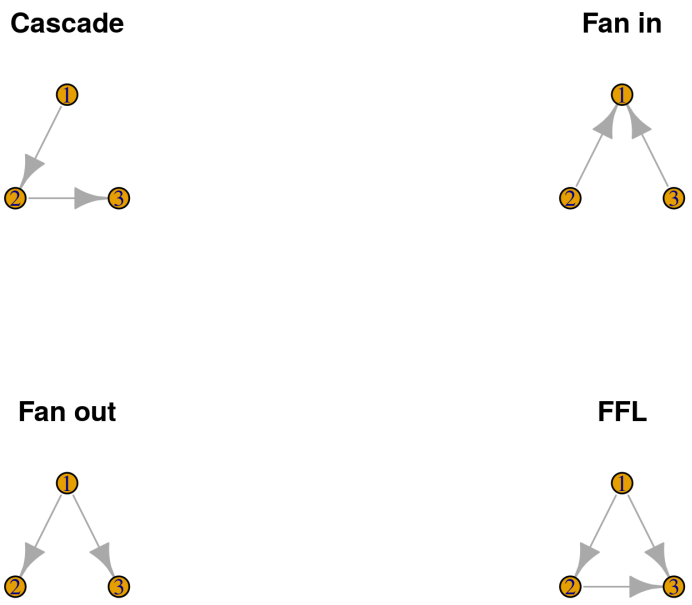


Fig. 2.2. Motif Topologies.

The topologies for four, three node motifs have been presented here. We have used the following motifs: Cascade, Fan in, Fan out and Feed Forward Loop (FFL).

2.4 Conclusion

We have introduced a total of 11 metrics for assessing the performance of network inference methods. Five of these metrics aim at eliciting the global behaviour of

network inference approaches. While six local metrics analyze the behaviour over local properties such as specific indegree or outdegree edges and performance on motif structures. The metrics introduced for studying the behaviour of any given method on indegree and outdegree edges is novel. We have extended the score metric used regularly in the DREAM challenges [10, 11, 12] for edges of a given indegree and outdegree.

Additionally, we have constructed the metric dSeparation to assess the degree of closeness between the edges of a given indegree or outdegree, that are incident upon or emanating from a given gene respectively. The introduced metrics are not specific to this study, and can be adapted for any exploration of network inference methods. We use these metrics in the following chapters to analyze the properties of different network inference methods.

Chapter 3

Gene Network Inference Methods: Introduction and Analysis

3.1 Introduction

3.1.1 Problem Formulation

As discussed in Chapter 2, GRN inference can be defined as a problem to infer the interaction pattern for a group of genes from a given expression data. The source of the data is high-throughput experiments, such as microarray, RNA-sequencing. Though the inference techniques can be applied to data from different sources, here we assume that the source of the data is steady state microarray experiments. Consequently, data is in the form of a matrix, $X \in \mathbb{R}^{n \times p}$, where columns represent genes and the rows represent different conditions(experiments) and/or time points under which the mRNA (gene expression) levels were measured for all the genes. Thus, given matrix X, the task is to find an interaction pattern for the group of genes: for each gene, we need to find a list of genes that are potential regulators

(activators/repressors), which together determine the expression levels for that gene[1].

In light of the aforementioned discussion, a GRN can formulated as a directed graph $G = (V, E)$. The vertex set $V = \{1, 2, ..., p\}$ contains p nodes, which correspond to the genes. Consequently, the edge set E consists of ordered pairs of nodes $\{i, j\}$ which represent the interaction pattern for the GRN. For instance, an edge directed from node i to j accounts for the regulatory effect of gene i on gene j. The nature of the regulatory relationship, whether activating or repressing, is inconsequential within the context of this work, thus has been disregarded.

Generally, the retrieved interaction pattern is a weighted adjacency matrix $W \in \mathbb{R}^{p \times p}$, where the elements of the matrix correspond to the predicted confidence or strength in the corresponding edge in the network. The interaction pattern $I \in \mathbb{R}^{p \times p}$ can then be obtained from W after appropriately dropping all the elements below a specified threshold; the present work is not concerned with the selection of this threshold. To get from X to W an association metric is desired to ascribe prediction confidences to the edges in the network.

Without apriori assumptions, the regulation of any gene in the network would be a function of the other genes, which could be represented in the form of a coupled system of equations as given in Eq 2.1 [2].

$$x_j = g_j(x_1, x_2, ..., x_{j-1}, x_{j+1}, ..., x_p) \tag{3.1}$$

where, x_j, $j \in \{1, 2, \ldots, p\}$ is the jth column of the expression matrix X. The function g: $\mathbb{R}^{n \times p\text{-}1} \rightarrow \mathbb{R}^n$ characterizes the regulatory mechanism for the network. Given the form for g, the inference task consists of estimating the parameters of g and obtaining the weight matrix W. Assume that W_{ij} represents the predicted confidence or strength in the regulatory relationship from gene i to gene j, then

[1]The implicit assumption is that the group of genes also contains the genes that express the transcription factors.

[2]Here we assume that auto-regulatory effects are not present.

W_{ij} can be estimated from Eq 2.2.

$$W_{ij} = f_{ij}(\mathbf{g}, x_i, x_j), i \in \{1, 2, \ldots, p\}, j \in \{1, 2, \ldots, p\}, i \neq j \tag{3.2}$$

where $\mathbf{g}$ represents a vector of functions which characterizes the gene behaviour. Varying degrees of assumptions can be made about the form for $\mathbf{g}$. The nature and the severity of the assumptions defines a trade-off between faithfully capturing the complex structure and the feasibility of a solution. The network inference task is inherently high dimensional; large number of networks are possible and only limited data points are available. This makes the problem under-determined and appropriate assumptions need to be made to find a solution. Depending on the assumptions adopted, the network inference process could be divided into three categories: Pairwise (PW), GeneWise Decoupled (GWD) and Full Conditional Distribution (FCD). Under PW, methods only consider pairwise dependencies between all gene pairs in the network while neglecting the influence of other genes. GWD methods, take each gene independently, and consider the joint effect of the rest of the genes on the selected gene. Finally, FCD methods look at the full conditional distribution structure for all the genes together in a multivariate sense.

There are other ways to categorize network inference methods, such as based on the underlying methodology applied by a method [12] or structural priors being employed. Naturally, there would be overlaps between the classes of different categorization. For instance, the class of degree distributions under structural priors would fall in GWD and FCD categories. Pairwise methods consider association between two genes at a time and thus cannot capture the degree distribution property as prior. However, pairwise methods can be augmented with post-processing methods for incorporating structural properties[29, 30]. In this chapter, we discuss network inference methods under the ambit of PW-GWD-FCD categorization.

3.2 Gene Network Inference Methods: Characterization and Literature Survey

As previously dicussed, the methods for network inference are diverse and different categorization techniques could be used to characterize them. Here we adopt the PW-GWD-FCD characterization introduced in the previous section.

- **Pairwise (PW)** - PW methods assume that pairwise gene interactions are independent of the other genes in the network. Consequently, Eq 2.1 is decomposed into *p(p-1)* independent equations as shown in Eq 3.3.

$$x_j = g_{ij}(x_i), i \in \{1, 2, \ldots, p\}, j \in \{1, 2, \ldots, p\}, i \neq j \tag{3.3}$$

  Mutual information (MI) and correlation ($Corr$) are the most popular pairwise metrics [10, 11, 12, 13, 14, 31, 32, 33, 34, 35, 36], which work within the framework of a relevance network approach [32, 33]. These methods forego the estimation of Eq 3.3 and directly calculate the extent of dependence between two genes, and give the elements of matrix W. The most lucrative properties of these two methods is low computational demand and easy scalability. Correlation between each pair of genes can be easily calculated using a variety of measures of correlation [37]. Mutual information requires discretiation of the data to estimate the joint and marginal distributions. Thus, calculation of mutual information is a little trickier than correlation, however mutual information has proven to be better at the network inference task as shown in [12] and Section 3.3.

  Despite these advantages, such pairwise methods are of limited applicability. The inferred network is essentially a co-expression network rather than a transcriptional regulatory network [19], since edge confidences are assigned by the "guilt by association" principle. Thus, genes with similar expression patterns would be assigned a high probability of being connected. The inferred network is sparse with large number of false positives. Correlation and

mutual information inferred networks are strongly affected by the cascade error [12]. Thus, unaided correlation and mutual information metrics are of limited application.

Many post-processing techniques specific to *MI* and *Corr* have been devised to deal with the menace of false positives. ARACNe [35], *CLR* [34] and *MRNET* [36] are some of the most widely implemented procedures to tackle false positives. Each of the methods uses a different strategy to impose a similar structural constraint, sparsity. ARACNe looks at each triplet of genes in the network, and removes the weakest link using a threshold called the data processing inequality (DPI). This is tantamount to reducing the cascade errors by removing the weak feed forward loop (FFL) edges. The strategy has biological appeal, since it is known that biological networks have more cascade type motifs than FFLs. *CLR* proceeds by constructing a background null distribution for each edge in the network. Two background distributions are constructed, one for the regulator and other for the target gene. Z-scores obtained from each distribution are thresholded at zero and then combined to give a modified z-score to be used as the strength of the edge. *MRNET* uses the maximum relvance minimum redundancy (mrmr) feature selection principle to select potential regulators for each gene; thus, attempting to remove all the redundant edges in the network. In this work, we found that *ARACNe*, *CLR* and *MRNET* all reduce the cascade error to varying degrees in different synthetically generated datasets as shown in Section 3.3.

Network deconvolution [30] is another method that tries to remove indirect edges from the correlation matrix using matrix inversion. Network deconvolution is generally applicable to a large number of methods.

In addition to sparsity, methods have tried to impose higher order structural properties. *WGCNA* is one such method [29]. It tries to use a thresholding principle to impose scale-free structure on the inferred network. The

threhsold is selected such that the inferred network is approximately scale-free. This method has been discussed in detail in Section 4.2.

Netter [38] is another post-processing method aimed at incoporating desired structural properties in any method. Given a prediction list, *Netter* re-ranks the edges in the list using a simulated-annealing approach so as to maximize an objective function. The objective function can be customized to include desired structural properties such as graphlet based penalty or limiting the number of regulator genes.

Pinna *et al.* [39] introduced yet another way of post processing a network inference prediction using structural properties, in this case downranking spurious FFL indirect edges. The method however has been demonstrated for the case of knockout data. For each edge ij from gene i to j in the network, a z-score is computed by comparing the expression of gene j in the knockout strain of gene i against the background distribution of all other strains. A thresholded network is constructed and the indirect edges belonging to FFLs in the condensation graph of the thresholded network are downranked in the final prediction. In this study, we have made a crude attempt at leveraging a similar post-processing ideology, Section 5.5.

Hybrid methods based on combining two different pairwise metrics have also been constructed. *RegCorr* [40] is one such method, which combines pairwise regression with correlation coefficient to achieve performance comparable to the state of the art method *Genie3* [41] on the DREAM4 multifactorial perturbation challenge. A regression based score for edge ij from gene i to gene j is calculated by first regressing gene j on gene i and then using a function of the sum squared residuals for the score. The final prediction matrix is obtained by entrywise product of the correlation matrix and the regression score matrix. The hybrid methodology of this method has inspired the strategy 1 discussed in Section 4.2.

The simple pairwise methods such as Corr, MI and most associated methods have been suggested to be used as a filter for newly proposed mutlivariate methods [13]. These methods cannnot capture combinatorial regulation and other higher-order structural properties inherent in true gene regulatory networks [19]. However, some pairwise methods have performed exceptionally well on large scale biological datasets [42]. Küffner et al. [42] used η^2, which is a non-linear measure of correlation and computed using ANOVA. Two-way anova is used to assess local dependencies between transcription factor and target gene pairs. The method was the best performer on the *E.coli* dataset in the DREAM5 network inference challenge. One crucial factor for the success of this method is the availability of supplemental information regarding the expermental conditions under which the data was collected.

- **GeneWise Decoupling** - Methods under this category, decompose the network inference task into p independent tasks. Thus, Eq 2.1 is considered independently for each gene and not as a coupled system. The final network is constructed from the models for each individual genes. l_1 regularized regression has been extensively used within this category. *TIGRESS* [43] is one of the state of the art methods in this category, which combines l_1 regularization with stability selection. It has been observed that l_1 base methods tend to perform better with some kind of resampling technique [12]. This also resonates with the observation that the performance of any given inference method depends strongly on the specific implementation rather than the general methodology. TIGRESS splits the data into roughly equal sized two samples and infers binary network for each half by using l_1 regularization on each gene independently. This process is repeated a large number of times to generate a frequency matrix that can be normalized to obtain a weighted adjacency matrix.

  Inferelator is another method that leverages l_1 based regularization, [44]. *Inferelator* uses a decoupled dynamical model as an approximation to a coupled system of ordinary differential equations to model the system dynamics.

The derivatives are estimated by finite differences and an l_1 regularized linear regression formulation is used to estimate the network topology. One interesting aspect of [44] is the introduction of a second order interaction term for transcription factors which helps capture combinatorial regulation. This structural constraint has been discussed in Section 4.3, where it has been leveraged to capture interaction between transcription factors for combinatorial regulation in the context of a simulated annealing based method that we introduce.

Apart from l_1 regularized regression models, some novel methodologies have been developed such as *Genie3* [41] which uses random forests. *Genie3* uses an ensemble of tree based regression to compute the strength of regulation by potential transcription factors. Such models are created for each gene independently. *Genie3* is one of the state of the art methods which was the best performer on the DREAM4 multifactorial steady state data and the DREAM5 in-silico network.

- **Full Conditional Distribution (FCD)** - Methods which model the joint distribution of all the genes in a gene regulatory network fall in this category. Graphical methods such as bayesian networks, gaussian graphical models (GGM) and dyanmic bayesian networks belong to this class of methods. Bayesian networks have been used extensively for the gene network inference task [45, 46, 47, 48, 49]; six out of the 29 participants in the DREAM5 challenge used bayesian networks. However, bayesian networks have not performed well compared to some of the other computationally less demanding methods [12]. Exponential search space and limited data drive the optimization towards sub-optimal solutions. Model reduction techniques have been proposed [47] to reduce the size of the search space. Recently, methods are beginning to leverage topological properties of gene regulatory networks to reduce the size of the search space and infer biological relevant networks such as [50], which imposes a scale-free structural prior for the inference task.

Many GGM based methods have been proposed to infer gene regulatory network as well. l_1 based regularization can be leveraged for covariance estimation to infer gene regulatory networks [51, 52, 53]. Many GGM based methods have rather employed biologically inspired priors such as the degree distribution of gene regulatory networks [54, 55, 56] as the constraint in covariance estimation. Methods such as [57] adopt the scale free degree distribution prior and conducts model selection by sampling from the posterior using markov chain monte carlo (MCMC) sampling methods. Some of these methods will be discussed in detail in Chapter 5 where we talk about the task of incorporating knowledge about the degree distribution of gene regulatory netwokrs in the inference task. Specifically, we will elaborate and use two such methods, *Sheridan* [57] and *GLASSO sfprior* [54]. The former will be called *Sheridan sfprior* henceforth when used with a scale free prior, and *Sheridan rndprior* when used with a binomial prior on the degree distribution.

The inherent capability of graphical methods to incorporate structural information as prior information is quite lucrative. Finding efficient ways to incoporate structural properties of gene regulatory networks such as the scale free degree distribution of gene regulatory networks might offer consequential gains for the network inference task.

3.3 Results

We conducted experiments with 17 widely used network inference methods belonging to the six methodological categories defined in [12]. To extricate the strengths and shortcomings of these methods, we evaluated the performances on local and global measures introduced in Chapter 2. All 17 methods were used to infer networks for the 25, size 100 networks described in Chapter 2 and performance was characterized using the metrics introduced in Chapter 2. The results of various experiments that were conducted are described next.

- **Global assessment** - Fig. 3.1 shows the aggregate performance of the methods. An interesting observation is that *RegCorr* which looks at pairwise associations is comparable or better than two of the state of the art methods *Genie3* and *TIGRESS*. With *MI* in place of correlaion in *RegCorr* leads to an improvement in performance; the modified method will be referred to a *RegMI*. Though the performances of both *Genie3* and *TIGRESS* are variable with respect to the parameters, *RegCorr* and *RegMI* are comparable to these methods. In contrast to the pairwise *RegCorr* and *RegMI*, these methods employ more sophisticated approaches.

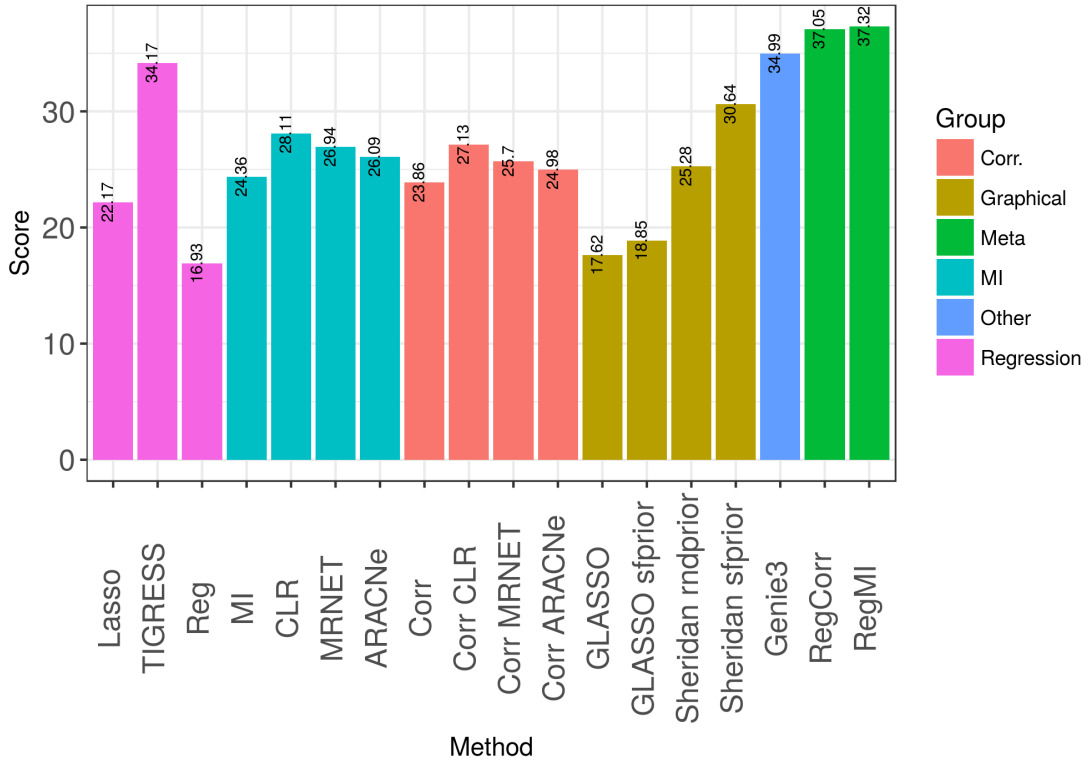


Fig. 3.1. Average overall score for method comparison.

The overall score averaged over all the netwrks. There are a total of 17 methods divided into six groups depending upon the methodology used by these methods for network inference. An additional method called *Reg* has been included for comparison purposes. The method is an implementation of the regression part in *RegCorr*.

In accordance with intuition, methods which aim to idenitfy directed networks generally have better performances compared to those that don't. *RegMI*, *RegCorr*, *Genie3* and *TIGRESS* all infer a directed network. Consequently, all of these methods rank at the top of the list. The extent of causal

information is shown in Fig. 3.2; these methods carry statistically significant causal information.

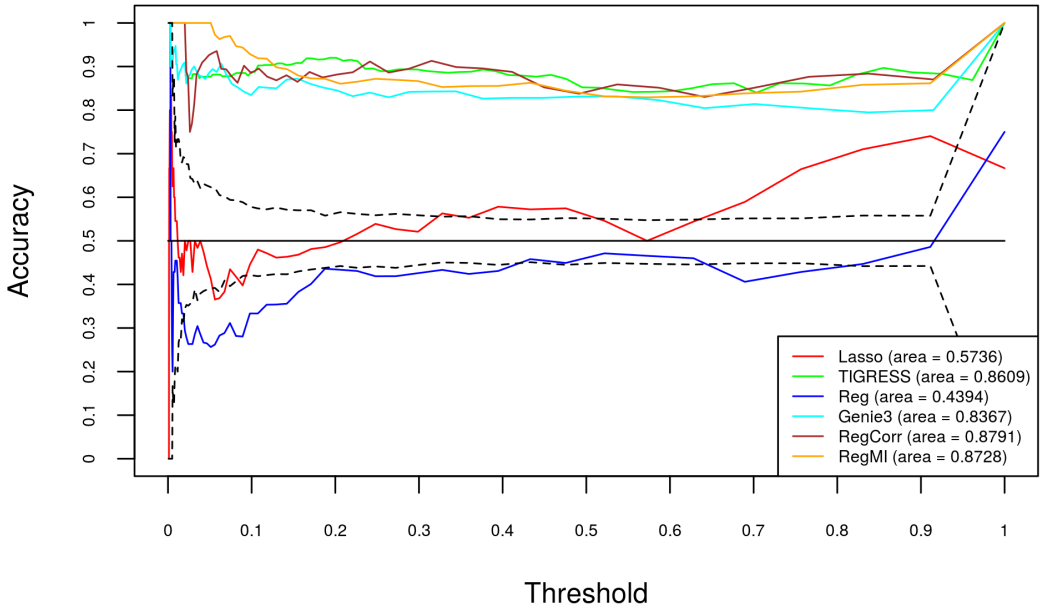


Fig. 3.2. Causal Accuracy vs Threshold for the class of directed methods.

Causal accuracy has been plotted against the threhsold. The plots were generated by thresholding normalized prediction matrices at different values of the threshold. For each threhsold, the accuracy of predicting the direction of causality for asymmetric edges is measured for the obatined thresholded network. The solid horizontal line at accuracy 0.5 represents random guessing. The region between the two dotted curves above and below the solid line represents 95% confidence bound; curves in this region would have causal information statistically indistinguishable from the case of random guessing.

However, *Lasso*, application of simple l_1 regularization for network inference, performs poorer compared to most of the undirected methods. This contradiction could be explained by the fact that the performance of inference methods is strongly affected by the particular implementation of a general methodology [12]. For instance, both *Lasso* and *TIGRESS* perform l_1 regularized linear regression, yet the latter is a state of the art method while Lasso performs quite poor. The performance of such regularization methods is quite dependent on the type of resampling technique being employed; *TIGRESS* uses stability selection based resampling while *Lasso* uses none.

Moreover, it is evident from Fig. 3.2 that *Lasso* is indistinguishable from the case of random guessing when it comes to inferring directionality.

Variation among methods is also evident for graphical methods, where *GLASSO sfprior* and *Sheridan sfprior* both use the GGM framework for network inference, yet *Sheridan sfprior* has almost two times the score compared to *GLASSO sfprior* in Fig. 3.1. This could be due to the fact that *Sheridan sfprior* uses a well defined prior on the degree distribution of models compared to *GLASSO sfprior*. This point has been discussed in Chapter 6. This further supports the claim that performance is highly dependent on specific implementations.

Notably, *Sheridan sfprior* performs the best among all the undirected methods. The incorporation of a well defined prior on the degree distribution gives this method edge. We shall see in Chapters 4 and 5 that when combined in a meta-analytic framework, *Sheridan sfprior* is one of the top performing methods. It is worthwhile to note that *Sheridan sfprior* performs better than *Sheridan rndprior*, which is an implementation of the method described in [57] with a binomial prior on the degree distribution in contrast to a scale free distribution. The difference between these two implementations is further explored in Chapter 6.

One of the key learnings from the DREAM challenges has been that combining multiple methods might lead to a synergistic effect in performance [12]. To explicate such an effect of meta-analysis for *RegCorr* and *RegMI* we have included in Fig. 3.1 another method called *Reg*. This is a method obtained by using the regression part of *RegCorr* in isolation. We see that *Reg* is the worst performer within the setting of our study. However, combination of *Reg* with other low performing methods, *Corr* and *MI*, leads to methods which are the top performers. We leverage this insight in Chapter 4 to design a meta-analytic strategy for augmenting the network inference task.

We had mentioned in Section 3.2 that *MI* performs better than *Corr*, and this is now evident in Fig. 3.1. *MI* based methods also perform better than corresponding *Corr* based methods. *CLR* performs better than *MRNET*, which performs better than *ARACNe*.

Score, though a good indicator of the overall performance, does not allow inference about the properties of different methods across the entire range of recall. Thus, we refer to Fig. 3.3 for the average precision recall curve over DREAM4 networks. We see that *RegMI* lies above all the others. Followed by *RegCorr* and *TIGRESS*. Interestingly, *Genie3* lies below *TIGRESS* for mid and high recall values. However, for small recall values, *Genie3* has the best performance, evident by its dominating curve in the top left corner. This implies *Genie3* offers biologically most relevant preditions compared to the other methods. We also observe that the undirected methods do not begin at high precision values. This is due to the undirected nature of these methods. Specifically, for *Sheridan rndprior* and *Sheridan sfprior* the curves are low and flat at very low recall values. The flatness is due to the fact that a large number of the top edges in their predictions have the same confidence. This issue has been raised again and discused in more detail in Chapters 4 and 5. The synergistic effect inherent in *RegCorr* and *RegMI* is quite visible in Fig. 3.2. While *Reg* is at the bottom, *RegCorr* and *RegMI* are at the top among all the curves. *Corr* and *MI* lie in between.

- **Comparative analysis on indegree/outdegree edges** - To assess the performance of different methods on edges of varying indegrees or outdegrees, we look at the plots for the dScore with different indegrees and outdegrees edges in Fig. 3.4. We find that the performance on indegree edges decreases exponentially. This is in contrast to the study conducted after DREAM3 challenge [11], where the performance was found to decrease linearly. However, a different performance metric was employed there to quantify the performance on indegree edges. The particular metric used there, is a biased estimator of the performance, since it assumes that the median of the

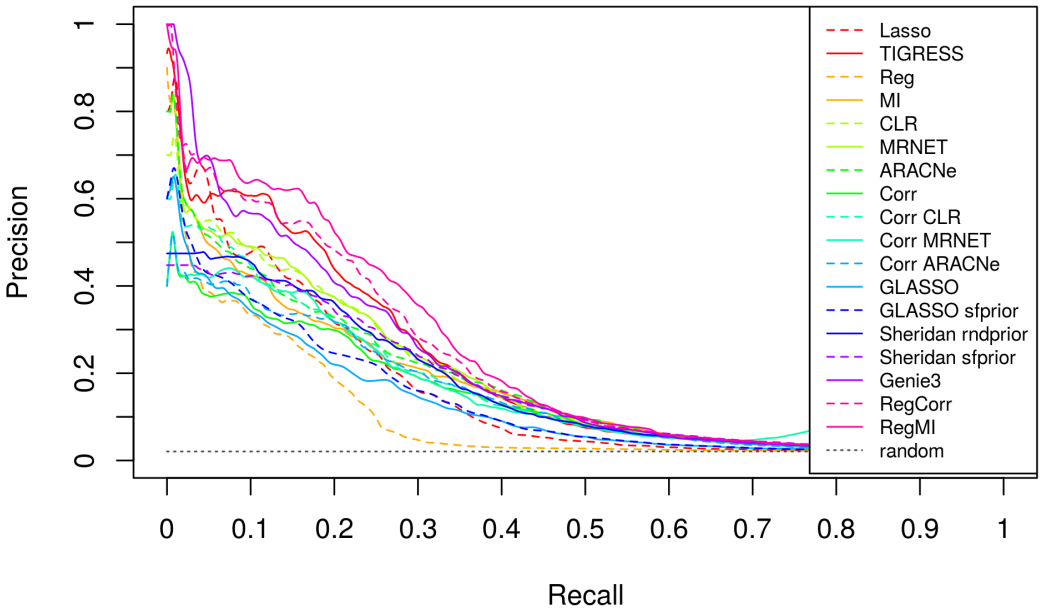
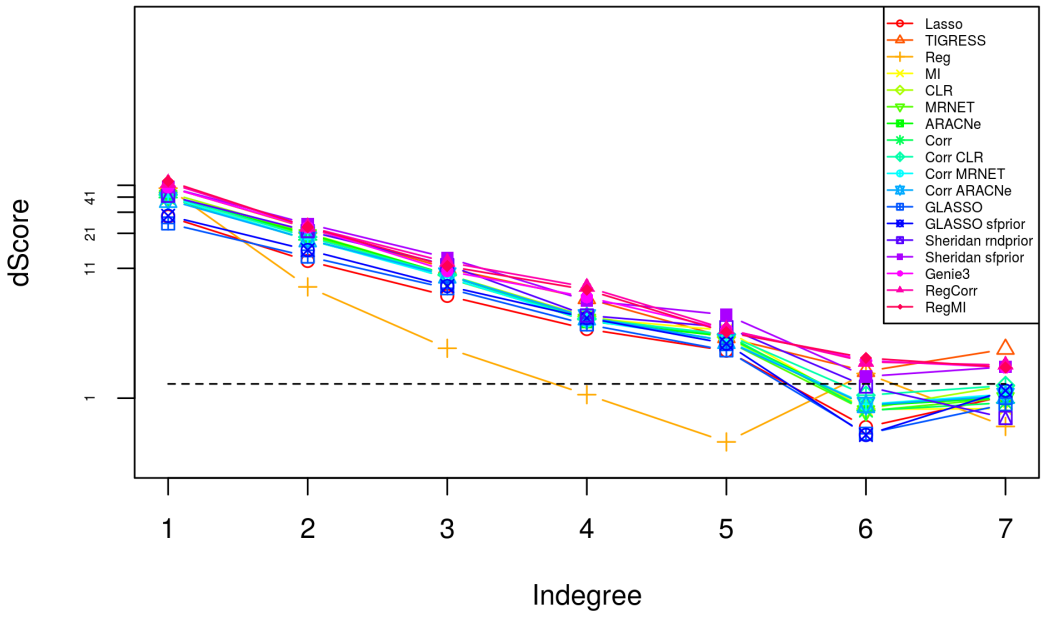


Fig. 3.3. Precision Recall Curve for method comparison.

Precision recall curves for all the methods averaged over the five networks in the DREAM4 challenge.

normalized edge weights for a given indegree is a good approximation of the performance. A metric defined in such a way essentially looks at the threshold at which 50% recall is acheived, which is not a suitable measure of overall performance. We believe that dScore introduced in Chapter 2 captures the performance over the entire range of the threshold and is statistically sound.

To make a comparative assessment of the performance across different methods, we look at the plots for intercept, slope and degree crossing for dScore. Fig 3.5a shows the intercept for indegree dScore averaged across the 25 networks. Except for *Sheridan rndprior*, the intercepts for the other methods seem to follow a trend similar to the average overall score. The large intercept for *Sheridan rndprior* does not imply a better performance at extracting the indegree distribution, as observed by looking at the slope values in Fig. 3.5b. *Sheridan rndprior* has the largest slope, thus the sharpest decline in performance with increasing indegree.

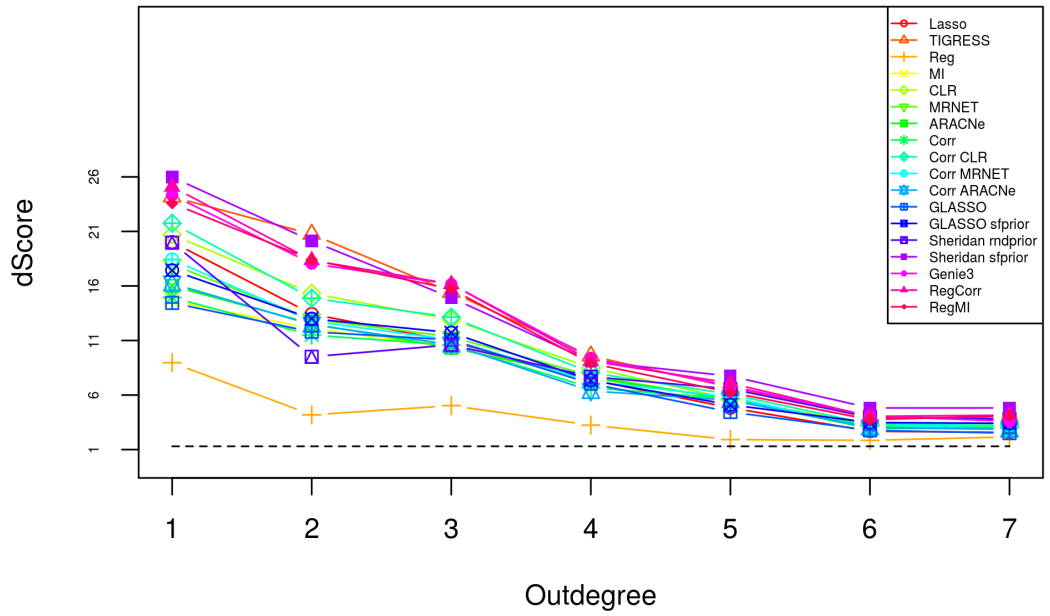


(a)

Fig. 3.4. Average dScore vs Degree for method comparison. ***(cont.)***

Given the intercept and slope values in this format, it is not possible to make any reasonable inference about the comparative performance of the different methods. Thus, we show the pairwise degree crossing map in Fig 3.6. The dark red region along the rows for *RegMI*, *RegCorr*, *Genie3* and *TIGRESS* suggests that these methods are better in general than the undirected methods for inferring the various indegree edges. *RegCorr* dominates *RegMI* completely. Both *RegCorr* and *RegMI* dominate *Genei3* and *TIGRESS* to a limited extent, i.e., only up to a certain indegree. Comparing the rows for *Sheridan rndprior* and *Sheridan sfprior* offers an interesting observation; the elements in the row of *Sherdian sfprior* have a darker red shade. This implies that *Sheridan sfprior* more strongly dominates other methods than *Sheridan rndprior* does. Thus, suggesting that the sfprior works better at inferring the indegree distribution.

Also, we can see from the faint blue shade in the cell for *Sheridan rndprior*

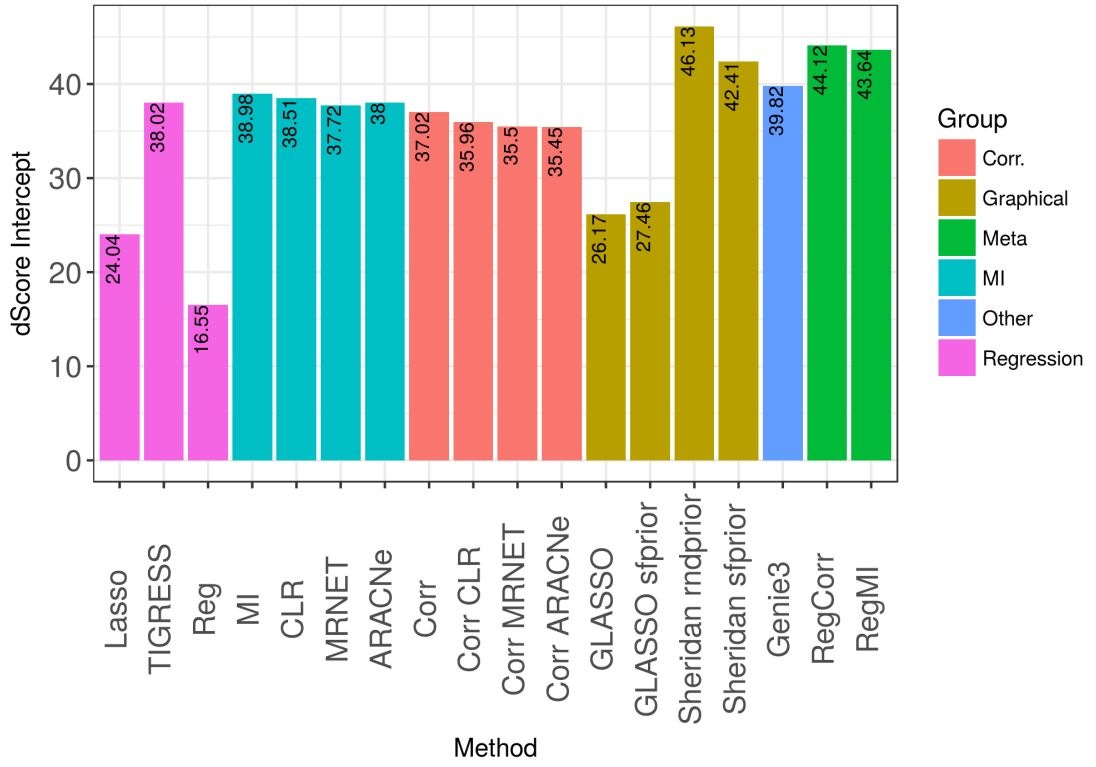


(b)

Fig. 3.4. Average dScore vs Degree for method comparison.

dScore averaged over all the 25 networks plotted against Indegree and Outdegree. (a): dScore vs Indegree plot. The dScore axis is on a logarithmic scale, while the indegree axis is on a linear scale. The exponential decrease in performance is clearly visible for all the methods. (b): dScore vs Outdegree plot for outdegree edges. The dScore and outdegree axes are on linear scales. dScore decreases linearly with the outdegree.

in the row of *Sheridan sfprior*, that after very small indegrees, *Sheridan sfprior* completely dominates *Sheridan rndprior* in terms of inferrring indegree edges. Except for *Reg*, *Lasso* and *Sheridan rndprior*, *GLASSO* and *GLASSO sfprior* are completely dominated by rest of the methods. For extracting the edges of different indegrees, *GLASSO* is one of the poor performing method. For *MI* based variants, *CLR* completely dominates *ARACNe* and *MRNET*. *ARACNe* performs better than *MRNET* at extracting the indegree edges. *CLR*, *ARACNe* and *MRNET* all dominate *MI* after small indegrees. For *MI* and *Corr*, the former dominates over lower indegrees while the later does at higher indegrees. For regression based methods, we see that [43] completely dominates [58], demonstrating the effet of using resamplng tehniques for the same class of methods. Other comparative observations can be easily made

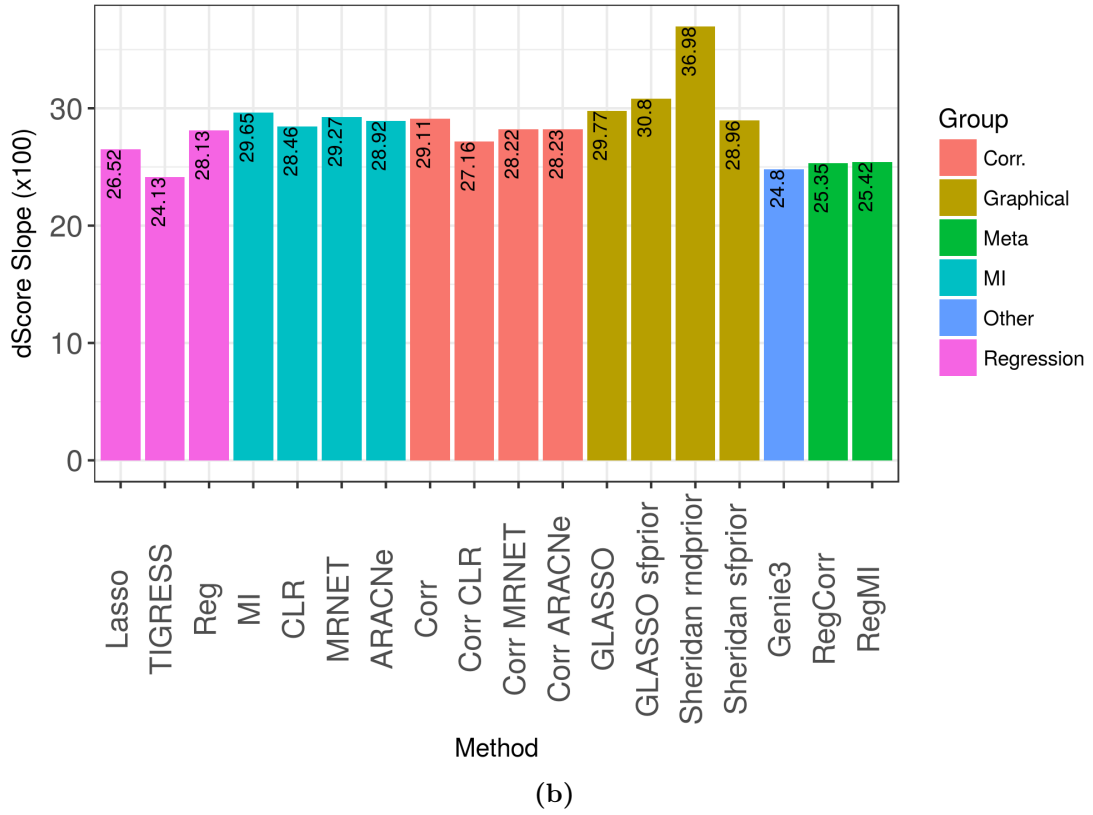


(a)

Fig. 3.5. dScore Intercept and Slope for Indegree for method comparison. ***(cont.)***

from the degree crossing map. We believe that a representation such as the degree crossing map for the comparative analysis of methods might be quite useful. Although, the analysis is contingent upon the linear regresssion fit, as defined in Chapter 2 for the indegree and outdegree dScores against the degree, being a valid model.

We again use the degree crossing map to perform a comparative analysis of the performances for outdegree as well. The degree crossing map is shown in Fig 3.7. Compared to the indegree degree crossing map, the outdegree shows a more restricted behaviour. The shade for strong red regions has diminished in Fig 3.6; complete dominance over all indegrees has been replaced by limited dominance over a range of outdegrees. For instance, the rows for *RegCorr*, *RegMI* and *Genie3* now show red regions with thinner shades of red. However, performance against *TIGRESS* for these methods has changed. For indegree edges, these methods dominated *TIGRESS* for a



(b)

Fig. 3.5. dScore Intercept and Slope for Indegree for method comparison.

Barplots for dScore intercept and slope for indegree averaged across all the networks. (a) dScore intercept; (b) dScore slope.

small range of indegree, after which *TIGRESS* was dominating. However, its vice-versa for outdegree. The general trend that we are observing in moving from indegree to outdegree has two more component in addition to the thinning of red regions; a reduced level of blue shades and swapping of some red and blue regions. However, reduction in shades for blue and red regions is complementary. If method M_1 dominates method M_2 for some range of outdegree, the cell for M_2 in the row of M_1 will hava a red shade, while there will be blue shade in the cell for M_1 in the row of M_2. So, if the red shade reduces, so does the blue shade. Finally, swapping in shade suggests a switch in behaviour; the region of outdegree where M_1 was dominating, now M_2 is dominating in that region and vice-versa. Thus, the overall picture is that the comparative differences for indegree edges are reduced when they trickle down to the outdegree edges. Different inference methods differ in the model they use for identifying regulation on the genes in a network. Essentially, the

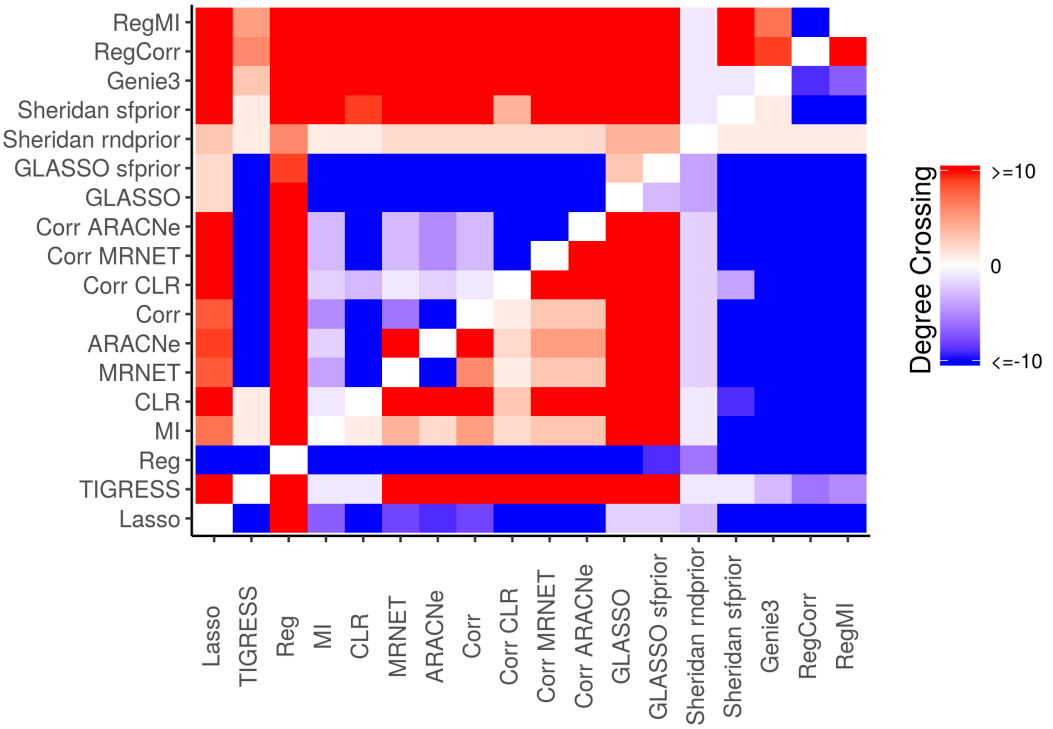


Fig. 3.6. Degree crossing map for Indegree for pairwise comparison of methods.

The degree crossing map represents the pairwise degree crossing for all the methods. The ijth element in the map represents the degree crossing while comparing the ith method against the jth. A positive value means that i completely dominates j up to a degree equal to the degree crossing value and after that j dominates i. When i completely dominates j, we have assigned a positive values of $p-1$ to degree crossing in such a case, to represent the fact that i always lies above j; and when j dominates i, degress crossing is set equal to $-(p-1)$. A negative value implies that j dominates method i up to a degree equal to the absolute value of the degree crossing and after that i dominates.

different methods try to capture the incoming edges on all the genes. Thus, depending upon the assumptions and the adopted models, methods would vary in their strength for identifying incoming links for the genes. Perhaps, this might be the reason for the observed reduction in the degree crossing maps from indegree to outdegree. Methods are trying to identify incoming edges; and the difference in performance between two methods trickles down from the indegree to the outdegree through the relationship given in Eq 3.4.

$$\sum_{i=1}^{p} d^i_{in} = \sum_{i=1}^{p} d^i_{out} \tag{3.4}$$

where, d^i_{in} and d^i_{out} are the in and out degrees for gene i respectively.

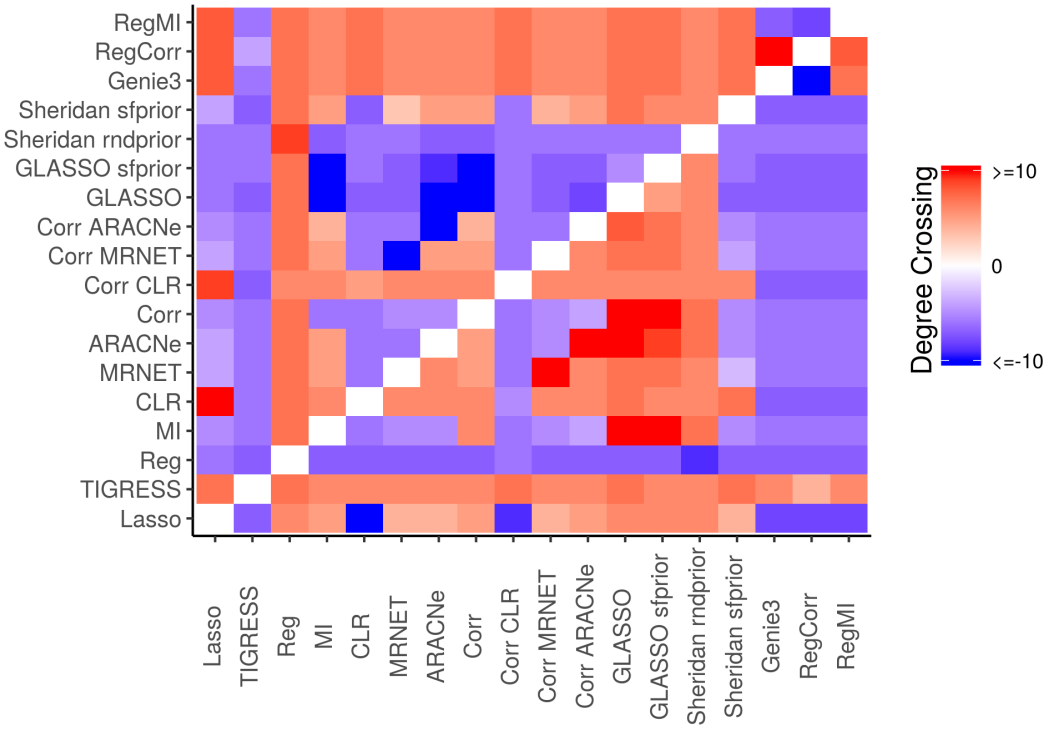
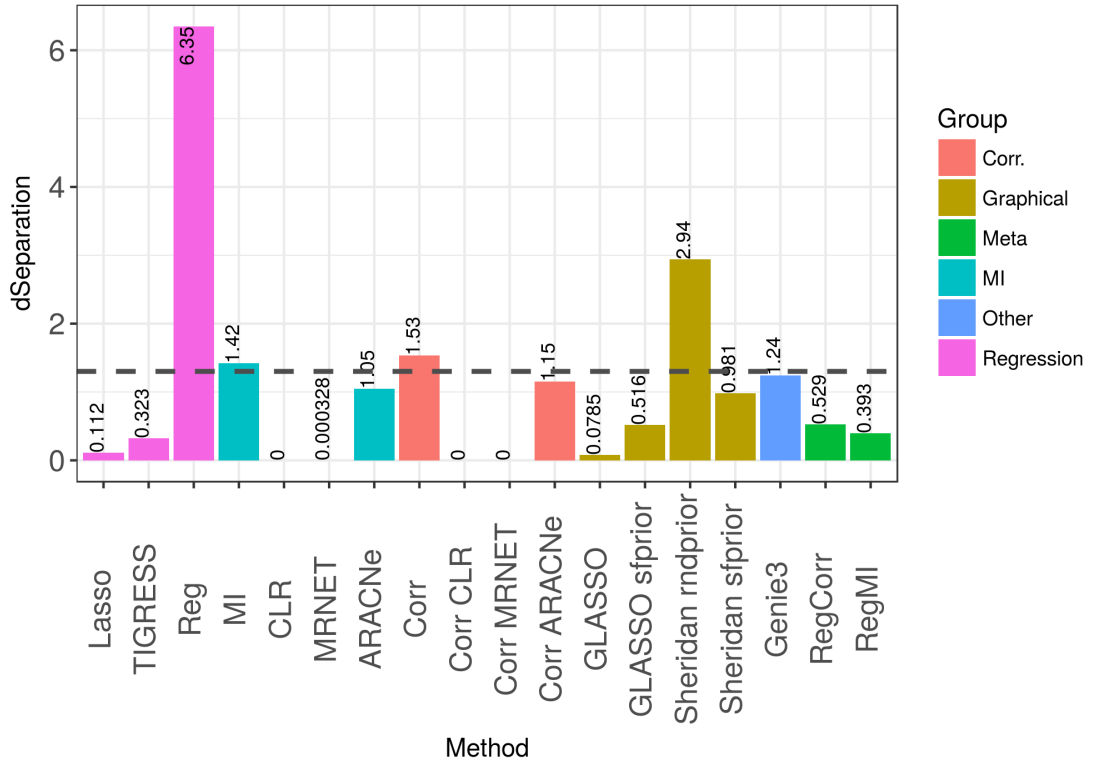


Fig. 3.7. Degree crossing map for Outdegree for pairwise comparison of methods.

The degree crossing map represents the pairwise degree crossing for all the methods. The inference strategy is the same as described in Section 2.3 and the caption for Fig. 3.6.

We also examined dSeparation for both indegree and outdegree 2 edges; the plots for dSeparation for all the methods are given in Fig. 3.8. We find that most of the methods have a statistically insignificant dSeparation. Only *Reg*, *Sheridan rndprior*, *MI* and *Corr* have statistically significant dSeparation. Thus, suggesting that for most methods, all the edges incident on a given gene are randomly distributed among the set of edges with indegree 2. For outdegree 2 edges, except for *RegMI* and *RegCorr* all the methods have statistically insignificant dSeparation. Thus, even for outdegree 2, edges emanating from a given gene are randomly distributed within the set of outdegree 2 edges.

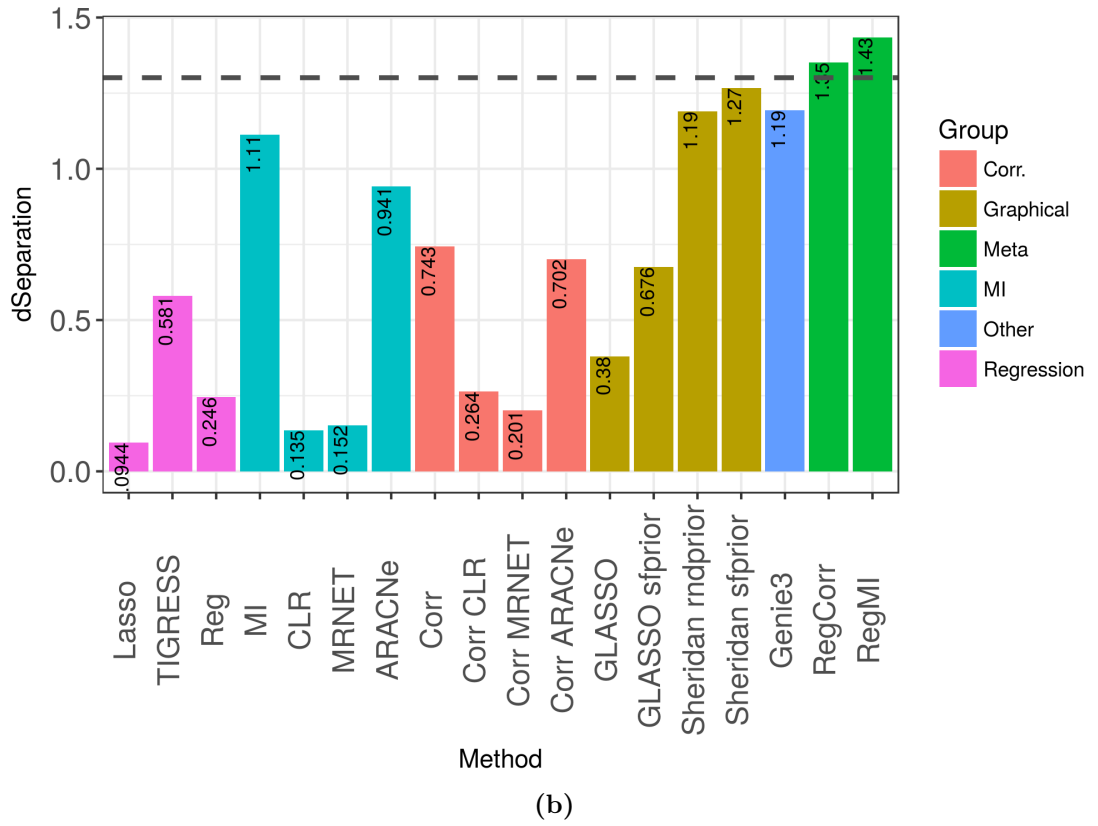
- **Comparative analysis on different three node motifs** - The behaviour of the 17 methods on four different three node motifs is shown in Fig. 3.9.



(a)

Fig. 3.8. dSeparation for method comparison. ***(cont.)***

All the methods are unequivocally affected by the four motif errors. This has been previously observed with respect to the DREAM3 challenge [11]. The negative bias on Fan-in and FFL errors is somewhat similar across all the methods. However, the difference between different methods is visible for cascade errors. Regression based methods are less affected by cascade error compared to *MI* and *Corr* based methods. Among *MI* and *Corr*, *ARACNe* has the least cascade error followed by *MRNET* and then *CLR*, *MI* and *Corr* have the highest bias. This outlines the benefit of the data processing inequality for *ARACNe*. Interestingly, *RegMI* and *RegCorr* are less affected by cascade error compared to *MI* and *Corr*. This could be attributed to the influence of *Reg*, which has one of the lowest biases for cacade error. *Genie3* also has a low bias for cascade errors. On Fan-out motifs also, Regression based methods are less affected by the error compared to *MI* and *Corr* based methods. *ARACNe* for *Corr* has a slghtly reduced bias compared to other *Corr* and *MI* methods, all of which have high bias for Fan-out errror.



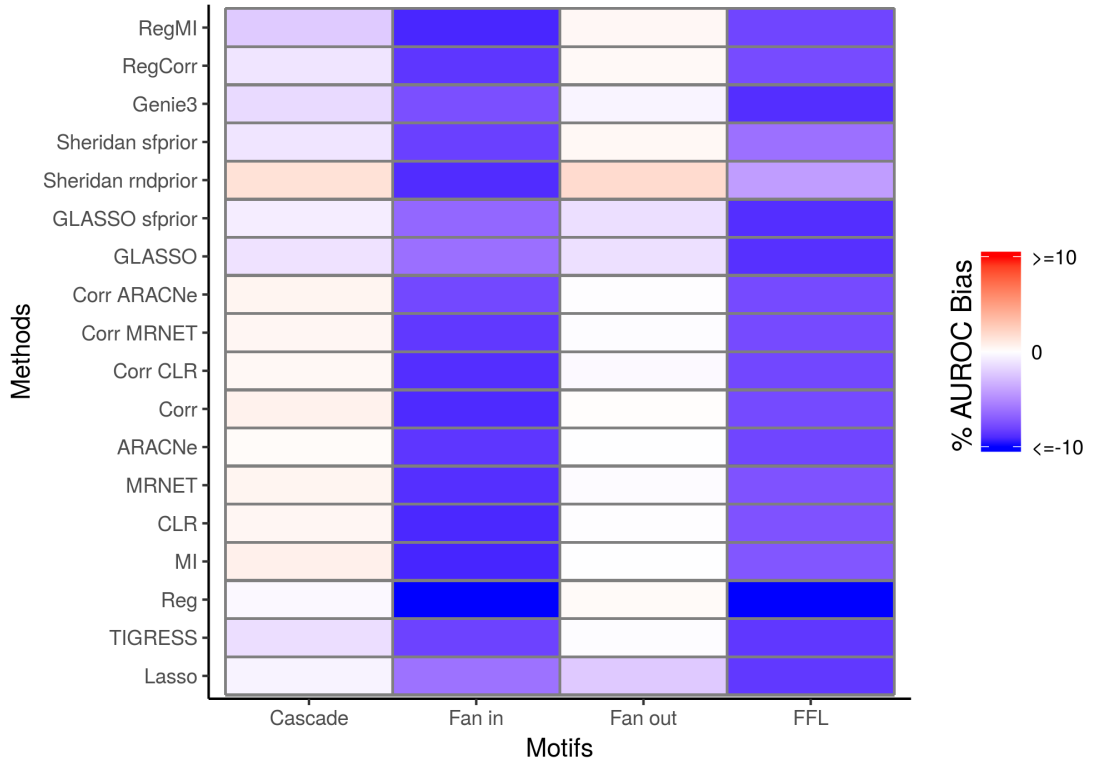
(b)

Fig. 3.8. dSeparation for method comparison.

The dotted horizontal line represents the dSeparation value which corresponds to a p-value of 0.05 for the p-value calcuation introduced in Section 2.3. (a) dSeparation for indegree; (b) dSeparation for outdegree.

GLASSO, *GLASSO sfprior* and *Sheridan rndprior* also exhibit a slightly lower bias compared to the high bias methods such as *Corr* and *MI*.

Given these observations, it is evident that all methods are in need of techniques that lead to a reduction in the various motif errors. This is neceassary if these methods are required to be able to faithfully reconstruct motif structures for the underlying network. For erros such as Fan-in and FFL, majority of the methods are equally poor. While for cascade error, diferent methods might need varying amounts of correction. We will see in Chapter 4 that strategy 1 introduced therein helps to reduce Fan-in error among other things. In Chapter 5 it has been shown that introducing scale free prior on the degree distribution leads to a decrease in cascade error for the prior incorporating methods. Fan-out error is also reduced for some methods.



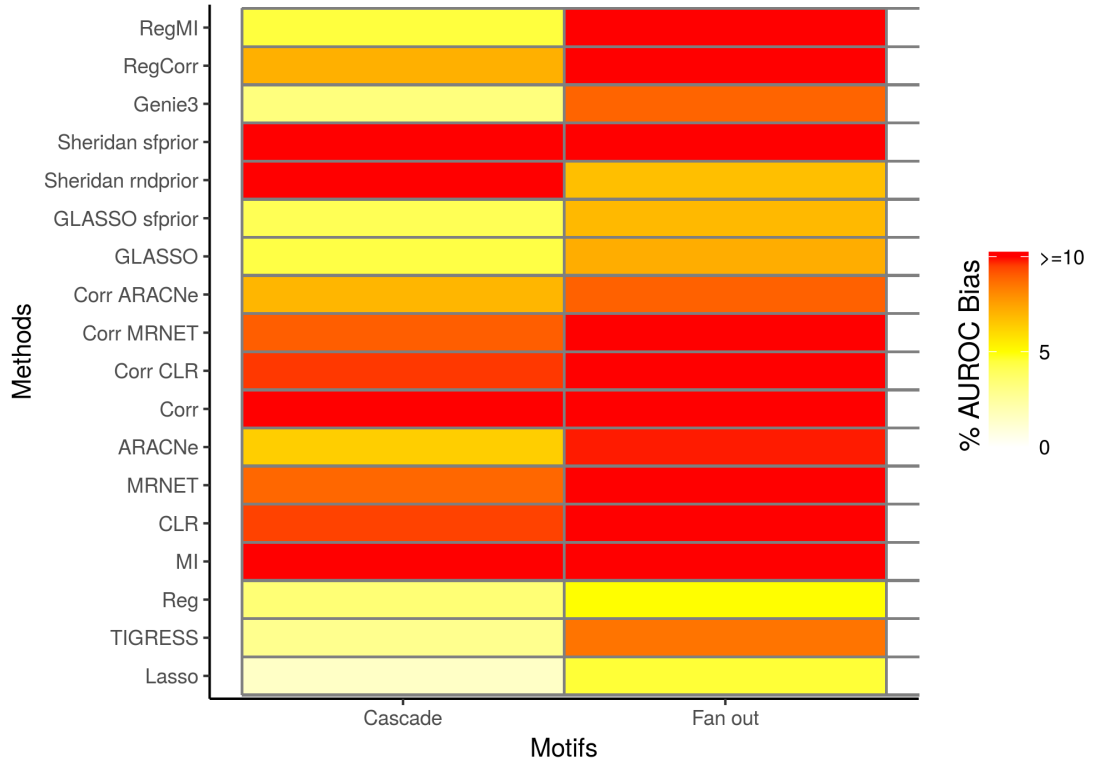
(a)

Fig. 3.9. Percentage Motif Bias for method comparison. *(contd.)*

3.4 Conclusion

In this chapter, we have discussed the network inference task in a more general setting. We have talked about some of the regularly used network inference methods, and tackled them within the context of the strategies used to handle the underdetermined nature of the network inference problem. Further, we have tried to characterize the strengths and shortcomings of some of the widely used network inference methods. Thus, we hope that this chapter might offer an understanding of a subset of the existing work in the network inference domain.

One consistent issue and concern among network inference methods is the incoporation of biologically relevant information. Biological information might be available in different forms, either as known regulatory interactions from prior lab experiments or structural properties. We are concerned with the latter in this work. Across many benchmarking and review studies into the gene network inference task, there hasn't been a systematic study of the effect of incorporating



(b)

Fig. 3.9. Percentage Motif Bias for method comparison.

Percentage Motif Bias for different methods. To assess the capacity of a given method for identifying different motifs, we look at the performance on both the motif edges and the motif errors. (a) Percentage AUROC bias for motif edges; (b) Percentage bias for motif errors.

knowledge about crucial structural properties of gene regulatory networks into the network inference task. Neither has there been a systematic analysis of available methods that try to incoporate such information. However, given the extensive literature [2, 15, 16, 17, 18, 28] reiterating the importance of structural properties such as scale-free degree distribution, modularity, motifs, etc., it seems obvious that leveraging these properties will help in inferring biologically relevant networks. Within this context, the current work is an attempt at studying the properties of some of the network methods which try to capture structural properties. Where the current chapter gives a brief overview of the network inference task in general, Chapter 5 acts complementarily. There, we specifically discuss the task of incorporating structural priors in the network inference task.

Chapter 4

Combination of Methods

4.1 Introduction

One consistent theme across the DREAM challenges has been that combining together the inferences from multiple methods leads to a prediction with better performance [10, 11, 12]. The aggregate score is even robust to the inclusion of the poorly performing methods on a given dataset. There's a two fold rationale that motivates this strategy. First, different methods might have complementary performances in that they might be good at identifying two different properties of the underlying network. Secondly, since the benchmarking studies till date have only focused on limited datasets, it is likely that methods' performances might vary across other previously unseen datasets. For the former situation, aggregation would offer the benefit of leveraging the strengths of different methods to arrive at a consensus prediction better than any individual prediction. While, if the latter is the case, combining predictions from different methods offers robustness against variation in performance. The DREAM organizers have used one possible way of aggregating the predictions by averaging the predictions across all the methods. The community prediction thus obtained has been shown to be comparable, if not better, to the best performer. In view of this observation, we have experimented

with two different ways of aggregating methods. One is a product based aggregation, where we combine a pairwise regression technique with another method. This aggregation strategy is tested on 15 different methods. The second strategy is implemented by applying the post-processing technique introduced in [34] to 11 different methods. We find that the first strategy has a synergistic effect on performance for methods which are inherently undirected; essentially, offering a strategy to direct the edges in symmetric network predictions. While the performance for the directed methods does not degrade much. The second strategy also has a similar directionality effect on most of the undirected methods, however the performance over the directed methods reduces by large amounts. We analyze both the strategies in the following sections. Section 4.2 formulates these strategies, Section 4.3 presents the results of the experiments that have been conducted and Section 4.4 offers a discussion about future prospects.

4.2 Formulation

4.2.1 Strategy 1: Regression based aggregation

This strategy has been inspired by the method introduced in [40]. A meta-method is discussed that works by combining a pairwise regression method with a correlation-based relevance network approach. Thus, this approach assumes marginal independence between pairs of genes while assigning a prediction confidence to the edge between the genes, and thus falls under "Pairwise" category introduced in the previous chapter. In contrast to the relevance network approach however, this method tries to infer a causal prediction, essentially assigning asymmetric weights to the edges between genes i and j and genes j and i.

Given the expression matrix $X \in \mathbb{R}^{n \times p}$, where n is the number of samples and p is the number of genes, the inference problem is decomposed into $p(p-1)$ independent problems corresponding to all the possible edges in the network. For each edge ij

from gene i to j, the confidence in that edge is calculated as given in Eq 4.1

$$W_{ij} =\mid r_{ij} \mid exp(-SSE_{ij}) \tag{4.1}$$

where W_{ij} is the confidence assigned by the method to the edge from gene i to j, r_{ij} is the pearson correlation coefficient between genes i and j and SSE_{ij} is the sum squared error obtained from regressing gene j on i. For computing SSE_{ij}, it is assumed that gene j can be modelled as a polynomial function of gene i, Eq 4.2.

$$x_j = a_{ij}^0 + \sum_{k=1}^{m} a_{ij}^k x_i^k + \epsilon_{ij} \tag{4.2}$$

where a_{ij}^m are the coefficients, m is the order of the polynomial and x_i is the ith column of X and ϵ_{ij} is the error term. It is assumed that Eq 4.2 is applied element wise to the components of column vectors x_i and x_j. Now, SSE_{ij} can be computed as given in Eq 4.3

$$SSE_{ij} = \sum_{u=1}^{n} ({}^u x_j - {}^u \hat{x_j})^2 \tag{4.3}$$

where ${}^u x_j$ is the uth element of the jth column of X and $\hat{x_j}$ is the least squares estimate of x_i after solving Eq 4.2.

The intuitive argument behind using SSE_{ij} is that it captures the capability of gene i to regulate the expression of gene j. Thus, Eq 4.1 assigns a strength of regulation to the regulatory interaction from gene i to gene j. W_{ij} in Eq 4.1, acts as an *AND* combination of the pearson correlation coefficient and the regression term; W_{ij} has a large value only when $\mid r_{ij} \mid$ is high and SSE_{ij} is low. The prediction matrix $W \in \mathbb{R}^{p \times p}$ is computed after calcuating W_{ij} for each edge in the network. The final prediction matrix is obtained by normalizing each column of matrix W by the q-norm [1] of that column, so as to ensure that all edges incident on a gene are on a common scale.

[1] q is assigned a value of 3.5 after experimenting with different values [40]

To generalize this approach for any given method, we modify Eq 4.3.

$$W_{ij} =\mid W_{ij}^{M} \mid exp(-SSE_{ij}) \tag{4.4}$$

where W_{ij}^{M} is the strength of interaction from gene i to j assigned by method M. Thus, W can now be given as in Eq 4.5.

$$W_{ij} = \frac{\mid W_{ij}^{M} \mid exp(-SSE_{ij})}{(\sum_{i=1}^{p}(\mid W_{ij}^{M} \mid exp(-SSE_{ij}))^{q})^{1/q}} \tag{4.5}$$

With the formulation in Eq 4.5, now the prediction confidence metric can be applied to any method that infers a weighted adjacency matrix. From the above formulation, it is evident that if even method M infers an undirected network, matrix W returns a directed network. The word *Reg* will be appended at the beginning of the name of each method after application of strategy 1.

4.2.2 Strategy 2: CLR based post-processing

CLR was introduced in Chapter 3, where it was discussed that *CLR* introduces a sparsity inducing post-processing procedure that works by comparing the entries of the inferred relevance network adjacency matrix against its background and removes the "weak" entries. Till now, this post-processing technique has only been applied to MI and correlation based approaches [10, 11, 12, 13, 14, 34]. Here, we apply *CLR* based post-processing prediction to other methods. For an adjacency matrix W^{M} from method M, the significance of interaction between gene i and j can be computed by comparing W_{ij}^{M} against an empirically estimated distribution of background values. The new z-score values would now be computed as given in Eqs 4.6 and 4.7.

$$z_1(i,j) = max(0, \frac{W_{ij}^{M} - \frac{\sum_{i' \neq j} W_{i'j}}{n}}{\sigma_j}) \tag{4.6}$$

where σ_j is the standard deviation of the values in the jth column of W^M, excluding the diagonal term.

$$z_2(i,j) = max(0, \frac{W_{ij}^M - \frac{\sum_{j' \neq i} W_{ij'}}{n}}{\sigma_i}) \tag{4.7}$$

where σ_i is the standard deviation of the values in the ith row of W^M, excluding the diagonal term. The ijth element of the final prediction matrix W can now be calculated as given in Eq 4.8

$$W_{ij} = \sqrt{z_1^2 + z_2^2} \tag{4.8}$$

As discussed in Chapter 3, W would be sparse with many entries being zero. The word *CLR* will be appended at end of the name of each method after application of strategy 2.

4.3 Results

We conducted experiments to test strategies 1 and 2 on all the size 100 networks. Strategy 1 and 2 were applied to 15 and 11 network inference methods respectively. We analyzed the effectiveness of the strategies in inferring the overall topology, degree distribution and motif structures of the gold standard networks using the global and local metrics introduced in 2. The following discussion explores in detail the results of these experiments.

- **Strategy 1 improves area under the curve (AUC)** - Fig. 4.1 shows the score averaged across all the 25 networks. Apart from *TIGRESS* and *Genie3*, the score improves drastically for all the other methods. We observe similar trends across all medium sized topologies. Furthermore, we see that after application of strategy 1, six erstwhile undirected methods, now have better performances than the top performer in the DREAM4 in-silico multifactorial challenge, i.e., *Genie3*. In fact, most of these methods perform better than

Genie3 across all the size 100 networks. The effect of strategy 1 is synergistic in nature, this can observed by referring back to Chapter 3, where we find that the regression part of strategy 1 performs poorly in isolation.

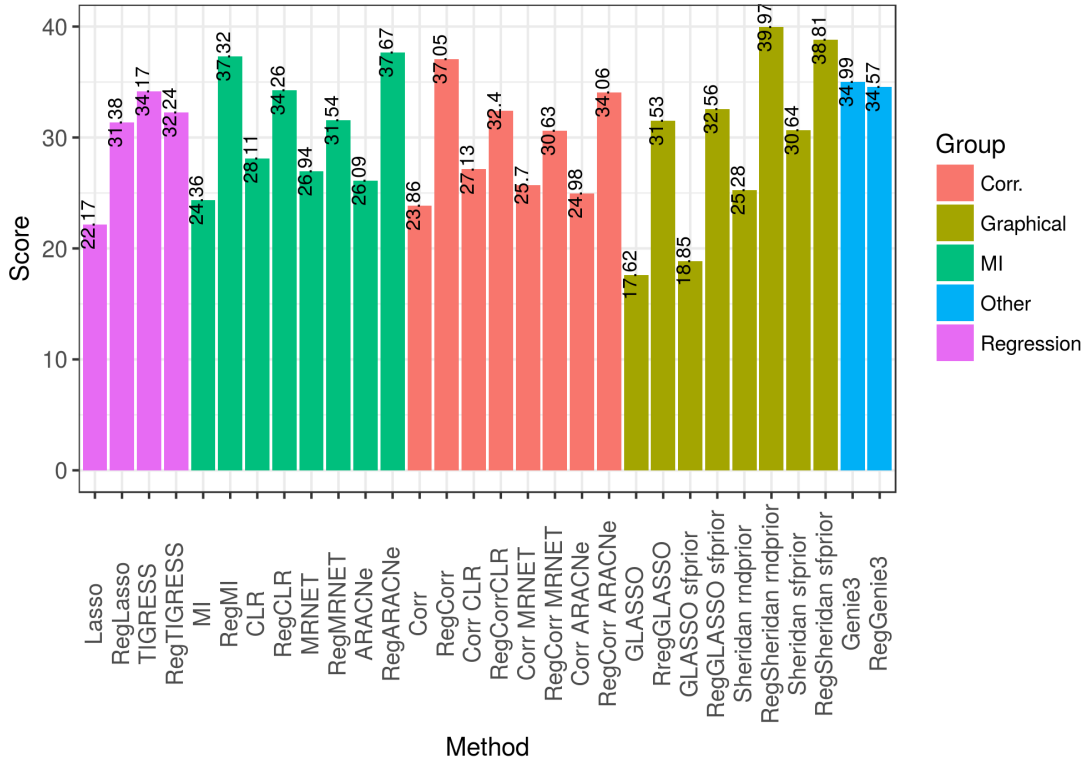


Fig. 4.1. Average Overall Score for studying strategy 1.

The overall score averaged over all the networks for studying the effect of strategy 1.

Interestingly, while *Sheridan sfprior* performs better than *Sheridan rndprior*, *RegSheridan rndprior* performs better than *RegSheridan sfprior*. One possible explanation for this might be the asymmetric nature of indegree and outdegree distributions for all the networks except PIPO. We find that on PIPO networks *RegSheridan sfprior* is better than *RegSheridan rndprior* For the PIPO networks, both indegree and outdegree distributions are scale-free with $\gamma = 2.5$. The effect of asymmetric degree distribution can also be examined by looking at the scatter plot shown in Fig. 4.2, where we see that there is a statistically significant positive correlation between the maximum indegree in a network and the difference in AUPR or AUROC values for *RegSheridan sfprior* and *RegSherdian rndprior*. It is evident that four out

of the five PIPO networks have relatively higher indegrees among all the 25 networks. *Sheridan sfprior* adds a scale free prior on the total degree distribution of the inferred network. However, except for PIPO, other networks tend to have an exponential indegree distribution. As we have seen, strategy one directs the undirected prediction from *Sheridan sfprior* and *Sheridan rndprior*. Perhaps, with the directional nature of Strategy 1, *Sheridan sfprior*'s heavy-tailed distribution fairs poorer than the dense distribution of *Sheridan rndprior* on networks other than PIPO due to the exponential nature of the indegree of the underlying networks.

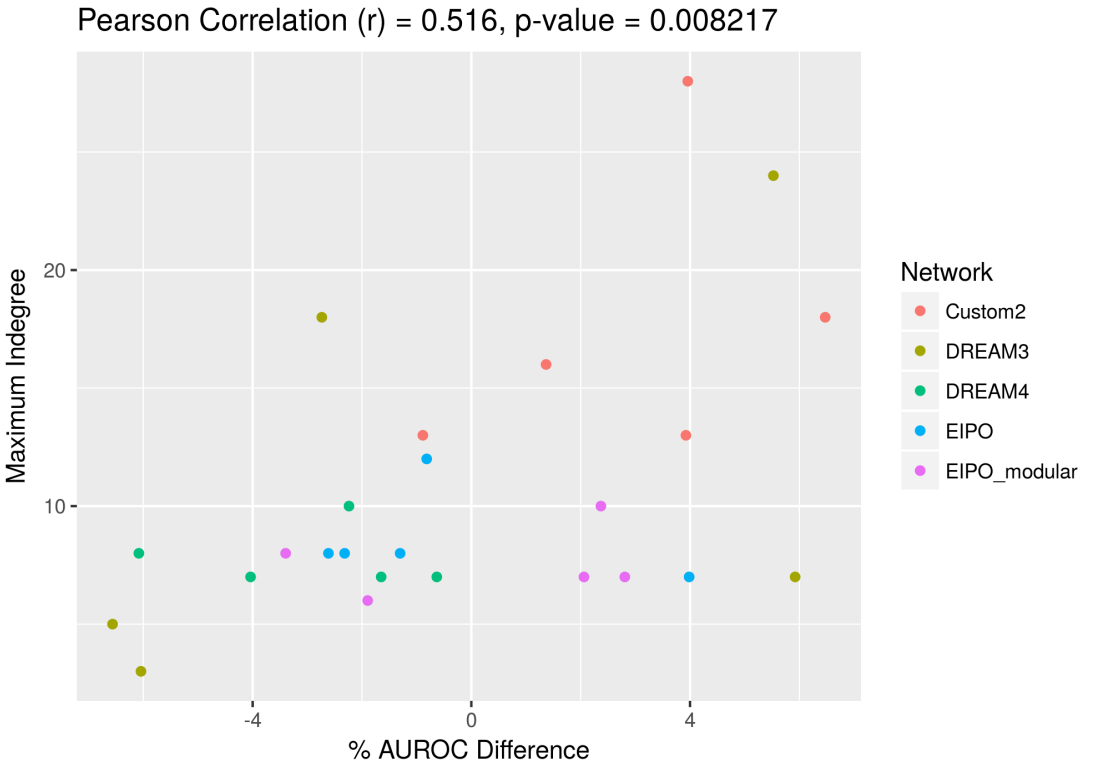


Fig. 4.2. Maximum Indegree vs %AUROC Difference for comparing *RegSherdian sfprior* and *RegSherdian rndprior*.

Scatter plot for studying the source of difference between *RegSheridan sfprior* and *RegSheridan rndprior* versus *Sheridan sfprior* and *Sheridan rndprior*. Difference in performance between *RegSheridan sfprior* and *RegSheridan rndprior* is positive for four out of the five PIPO networks. Difference in AUROC between *RegSheridan sfprior* and *RegSheridan rndprior* appears to be positively correlate with maximum indegree in the network, and thus with the nature of the indegree distribution.

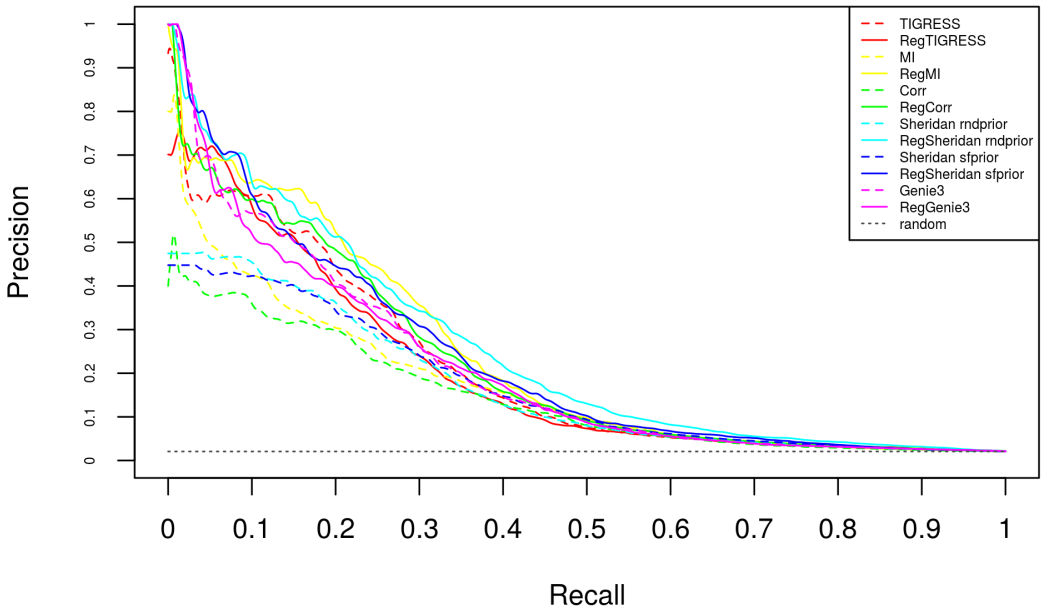
The score metric gives a neat overall picture of the network inference process for comparison among methods, however it does not inform regarding the

precision of a given method. Biologically, the left upper half of a precision recall curve is the most relevant for network discovery, since this region corresponds to predictions with high precision. To elucidate the effect of strategy 1 over the entire region of the ROC and PR curves, we look at Figs. 4.3a and 4.3b. The drastic effect of strategy 1 is quite evident here as well. Specifically, for *Corr*, *Sheridan rndprior* and *Sheridan sfprior*, the curves now begin at the top left half of the precision recall curve.

One of the issues with MCMC-based *Sheridan* method is that a large number of the top edges in the prediction list have the same confidence score. Thus, these edges are indistinguishable, which explains the the low and flat part of the original *Sheridan* method in the left half of the precision recall curve as seen in Fig 4.3a. Strategy 1 changes this behaviour, and the curves now begin with high precision. The erstwhile equivalently scored edges are now being distinguished. Furthermore, the curves with strategy 1 also have a lower slope; *RegSheridan sfprior* stays above all the other methods for all values of recall.

Interestingly, on closer inspection we see that the curves for *Genie3* and *TIGRESS* also show improvement with strategy 1 in early recall region between recall values 0.1 and 0.3 in the precision-recall space. Additionally, *RegTIGRESS* now begins at the top left corner of the precision recall whereas *TIGRESS* did not. The performance improvement for *RegSheridan rndprior* and *RegSheridan sfprior* is also visible in the ROC curve of 4.3b, where it is evident that both of these methods dominate over high recall regions as well. Except for *Genie3* and *TIGRESS*, these arguments also hold for all the other networks as well.

- **Strategy 1 is not limited to imparting directionality alone** - *Corr*, *MI* and derived methods, *GLASSO* and *Sheridan*, all of these methods are inherently undirected in nature. Thus, Fig. 4.1 suggests that methods which try to infer directionality tend to have better performance with strategy 1. Strategy 1 essentially performed an element-wise product of the prediction

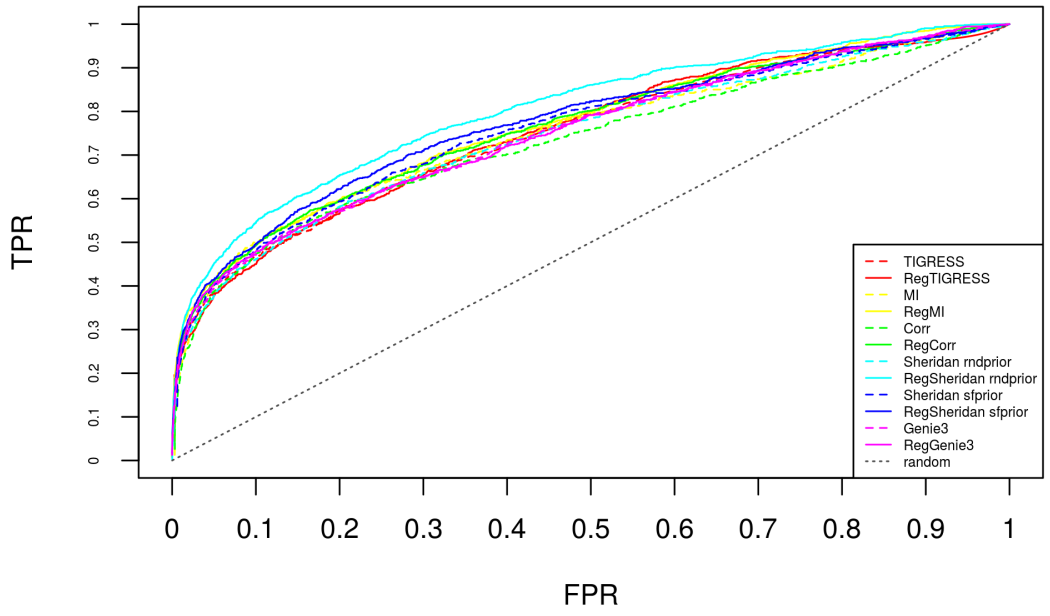


(a)

Fig. 4.3. Average Precision-Recall and ROC curves. ***(cont.)***

matrix inferred by a given inference method with a regression based directed method. Thus, we intuitively expect strategy 1 to impart a directional nature to any undirected method given as input to it. Indeed, that is the case as we can see in Fig. 4.4. All the erstwhile undirected methods, now have high causal content; many of these methods now have higher causal information than *Genie3* as well. This behaviour stays consistent across all the networks. The behaviour of *GENIE3* and *TIGRESS* however, is not consistent across all the networks; sometimes the causal content reduces while at other times it increases slightly.

To assess whether the effect of strategy 1 is only ascribing directionality to an undirected prediction, we convert all the network predictions to undirected predictions by averaging the values in th ijth and jith entries of the prediction matrices. The results after this transformation are shown in Fig. 4.5. Methods like *Lasso*, *GLASSO* and *Sheridan rndprior* consistently



(b)

Fig. 4.3. Average Precision-Recall and ROC curves.

Precision recall and ROC curves for different methods with and without strategy 1. The PR and ROC curves have been averaged over the five DREAM4 networks.

perform better with strategy 1 even after we remove directionality. *RegSheridan sfprior* shows slightly better performance than *Sheridan sfprior* on some networks only. *RegCorr* and *RegMI* show an increase in the overall score averaged across all the networks. However, on DREAM4 networks, *RegMI* based methods have huge dips in score with strategy 1. The behaviour is not consistent across all the methods. If directionality was the only information added by strategy 1, we would expect the score to come down to pre-strategy 1 levels after the symmetry inducing transformation. Thus, this observed effect suggests that something more complex is going on; and the causal information added by strategy 1 is coupled with other information about the underlying networks. This information is being destroyed by the symmetry transformation for some of the methods, while some methods still possess the extra information.

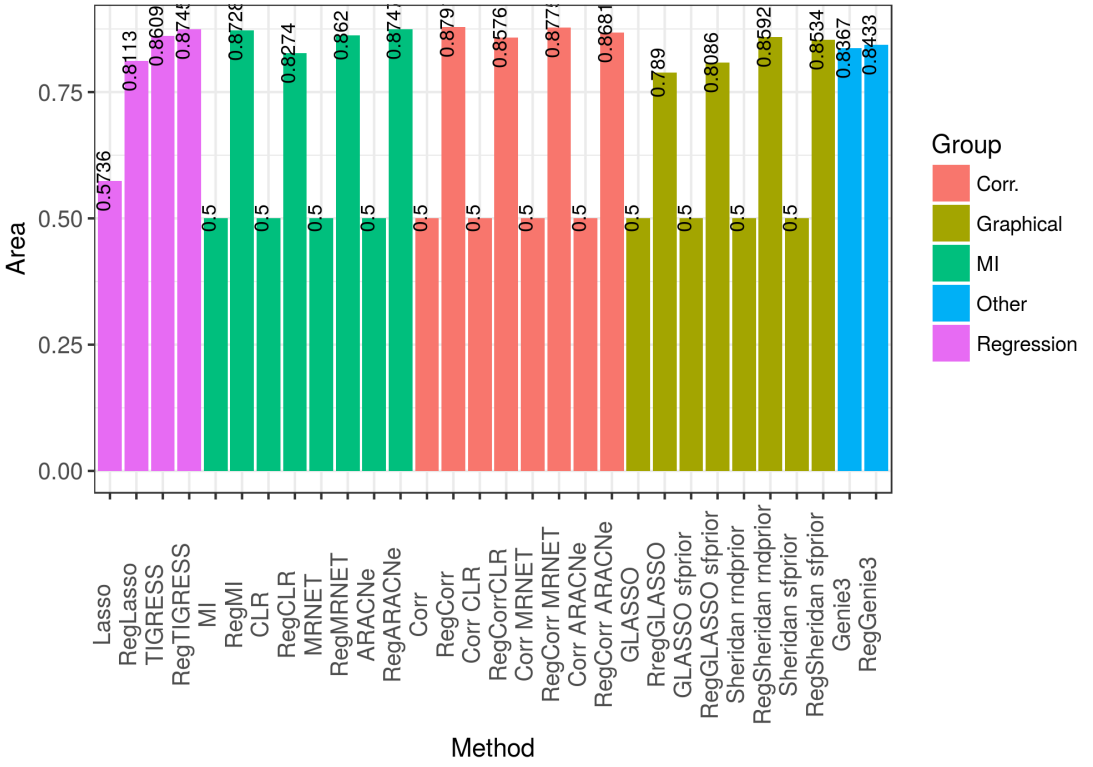


Fig. 4.4. Causal Area for studying strategy 1.

Area under the Causal Accuracy vs. Threshold plot for studying strategy 1. The plot has been averaged over the five DREAM4 networks for all the methods.

- **Strategy 1 aides in extracting the degree distribution of the underlying network** - Experiments similar to the ones conducted in Chapter 3 were used to calculate dScore for all the methods with and without strategy 1 to assess the effect of strategy 1 on degree distribution estimation. The obtained dScore values for different indegrees are log-transformed and then regressed against the indegree values to obtain the intercept and slope values and consequently the degree crossings as seen in 4.6a.

  We see that except for *Genie3* and *TIGRESS* all the other methods are dominated by the strategy 1 versions. The degree crossing for *RegGenie3* against *Genie3* is 3. Thus, we can conclude that overall strategy 1 versions of all the methods except *Genie3* and *TIGRESS* can estimate the indegree distribution better. Interestingly, *RegTIGRESS* climbs above *TIGRESS* at indegree 2, thus performs better than *TIGRESS* at higher indegrees. This also explains the decrease in performance with strategy 1. It was shown in

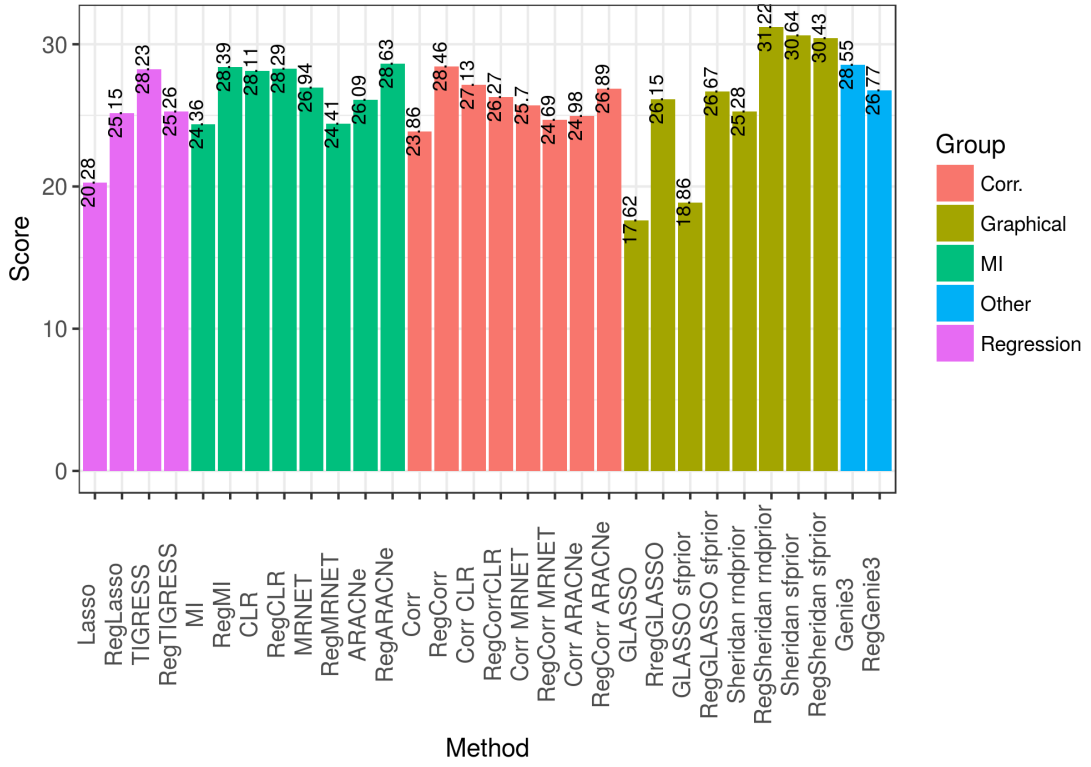
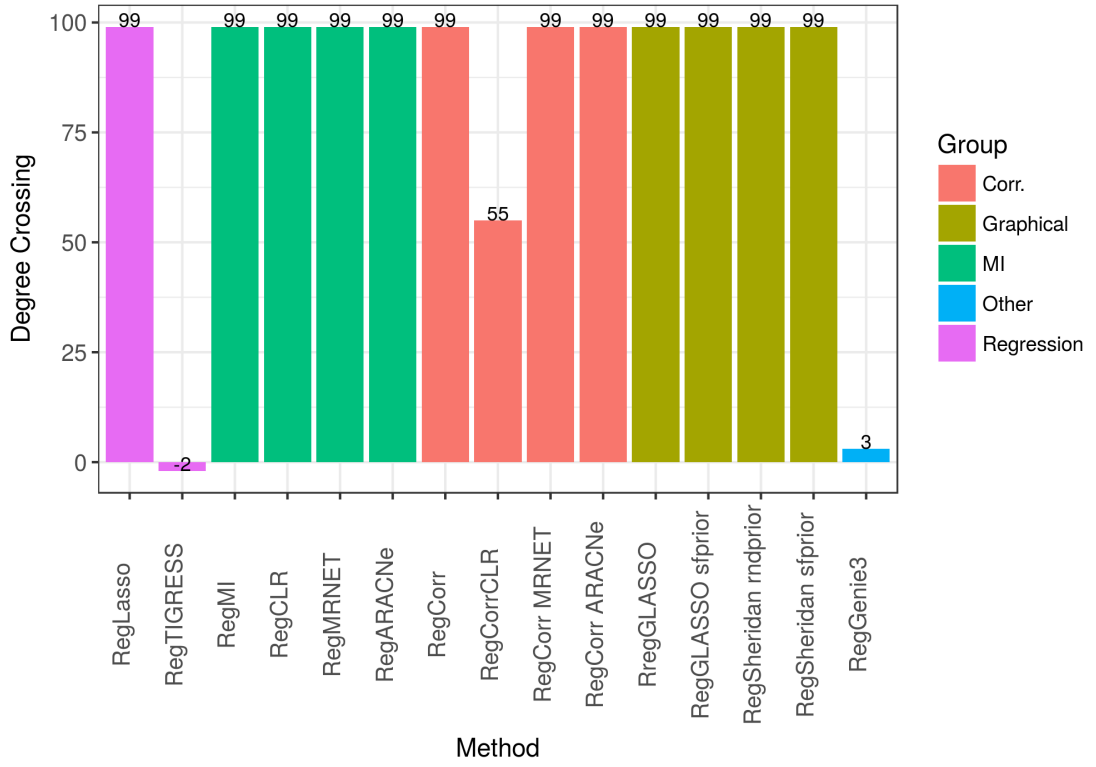


Fig. 4.5. Average Overall Score for studying the directional effect of strategy 1.

The overall score after converting all the directed methods to symmetric predictions.

Fig 3.4 that the best inferred edges for any method are the lowest indegree edges; the performance deteriorates exponentially thereafter. Thus, only *Genie3* doesn't show an improvement in its capability to identify indegree distribution better with strategy 1.

Similar to the above strategy we analyze the performance of strategy 1 on outdegree estimation. The plot for outdegree crossings is given in Fig. 4.7. We see that except for *Genie3*, all the other methods perform better with strategy 1 up to medium outdegrees only. This effect is similar to the trend we observed in Section 3.3 for the difference between indegree and outdegree. Here again, we see that the effect has diluted while trickling down from indegree to outdegree edges.



(a)

Fig. 4.6. Degree Crossing for Indegree for studying strategy 1.

Within the context of the discussion in Section 2.3.2 with regards to degree crossing, a given method's version with strategy 1 is compared against the base method. Thus, with strategy 1 the method would be M_1 and the base method would then be M_2. So, the conclusions regarding degree crossing from 2.3.2 would hold accordingly. A value of $p-1$ for the degree crossing implies that M_1 dominates M_2. Any other positive value means that M_1 dominates M_2 up to a degree equal to the degree crossing value and after that M_2 dominates M_1. A negative value implies that M_2 dominates method M_1 up to a degree equal to the absolute value of the degree crossing and after that M_1 dominates. A value of $-(p-1)$ means that M_2 completely dominates M_1.

- **Strategy 1 exhibits increased confidence in identifying Fan-out and Fan-in motif edges**

  As we have seen, strategy 1 aides in extracting the indegree distribution. While for outdegree distribution it aides for outdegrees up to 6 or 7. Fig. 3.4 shows that the performance on outdegree 7 drops down to the level of random predictions. Contingent on the fact that this trend continues for higher outdegrees as well, we could assume that strategy 1 might help in extracting Fan-in and Fan-out motifs better than base methods alone.

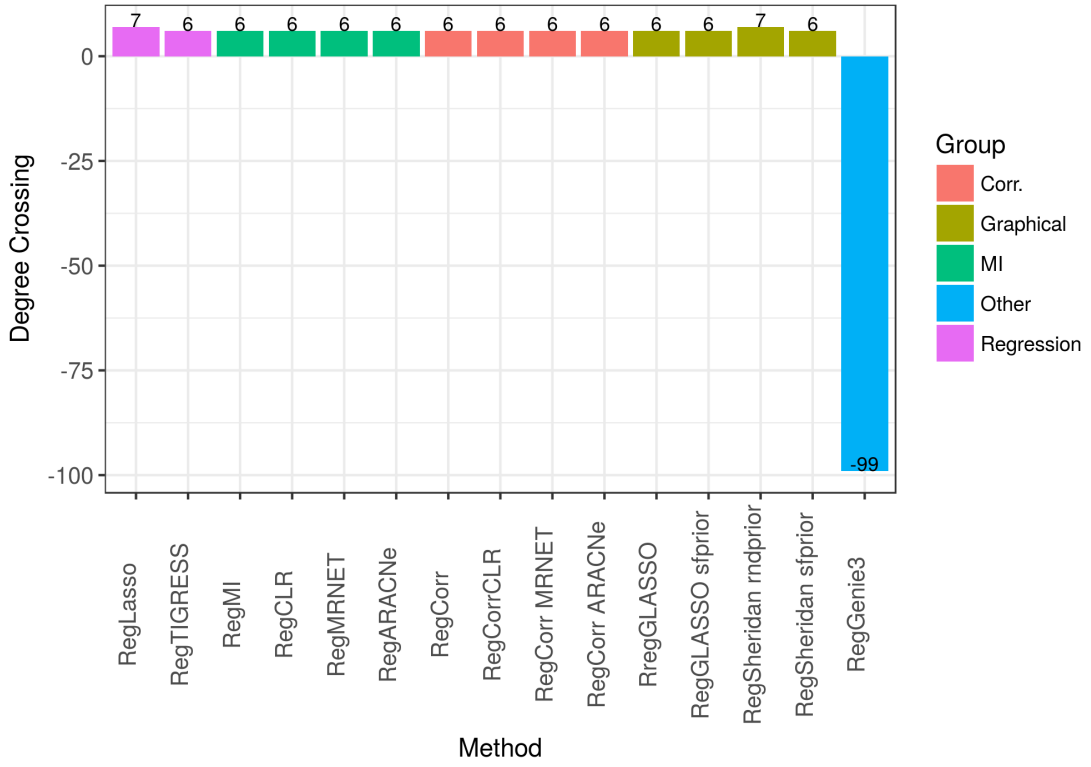
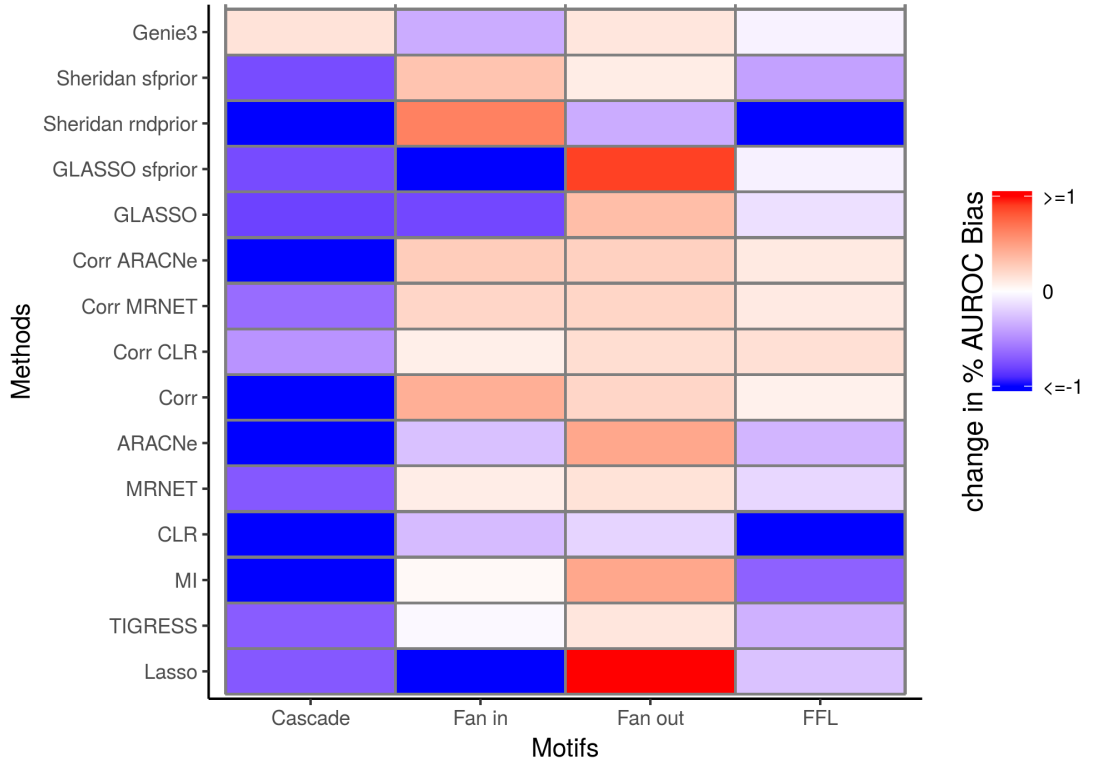


Fig. 4.7. Degree crossing for outdegree for studying strategy 1.

Degree crossing averaged across the networks. The inference strategy is the same as described in Section 2.3 and the caption for Fig. 4.6a.

To confirm this hypothesis, we refer to Fig. 4.8, where we have shown the percentage change in AUROC bias for edges of various three node motifs for the DREAM4 networks. It is clear from Fig. 4.8a that strategy 1 indeed leads to an increase in bias for the prediction of Fan-in and Fan-out edges for most of the methods. The performance on other Network topologies is also similar. However, increase in bias for Fan-out edges is accompanied by an increase in bias for Fan-out error as shown in Fig. 4.8b. This is not unexpected though, as it has been shown before that most methods tend to identify the false edge between nodes 2 and 3 in Fig. 2.2 for Fan-out motifs [11]. Though the indirect edges have been pronounced for Fan-out motifs with strategy 1, this is not the case for other indirect interactions as well. It has been seen in previous studies that regression based inference methods are less prone to the indirect cascade error. Since strategy 1 uses a pairwise regression based method, we expect it to have a reduced bias for the cascade error; and this is evident in Fig. 4.8b.

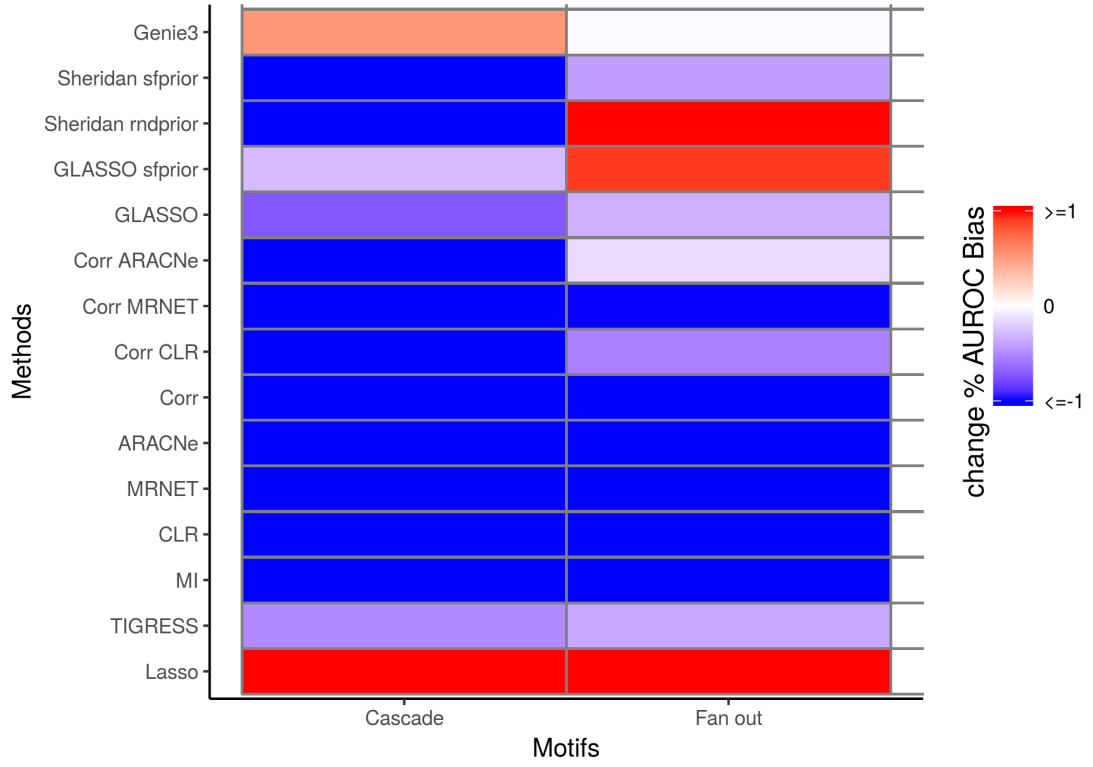


(a)

Fig. 4.8. Percentage Motif Bias for studying strategy 1. ***(cont.)***

- **Strategy 1: Performance on DREAM5 networks**

  To assess the performance of strategy 1 on large scale networks, we leveraged the networks provided as part of the DREAM5 Network Inference Challenge [12]. We used two of the three networks from the challenge- Networks 1 and 3. The former has been extracted from the known topology of *E.coli* and some random edges have been added to the final network; this network has been termed the in-silico network, since the expression data was generated using the tool GeneNetWeaver [20]. While Network 3 is created using available *E.coli* data [12]. For most of the methods, we used the partial prediction list provided by the DREAM organizers [12]. The DREAM5 challenge required the participants to provide a list of the top 100,000 edges predicted by their methods along with predictions confidences. We have used these truncated lists with 100,000 or less predictions. However for the correlation based methods, MI methods and the ANOVA based method [42] (called Other 2 in the DREAM5 challenge) we have generated the full prediction lists and



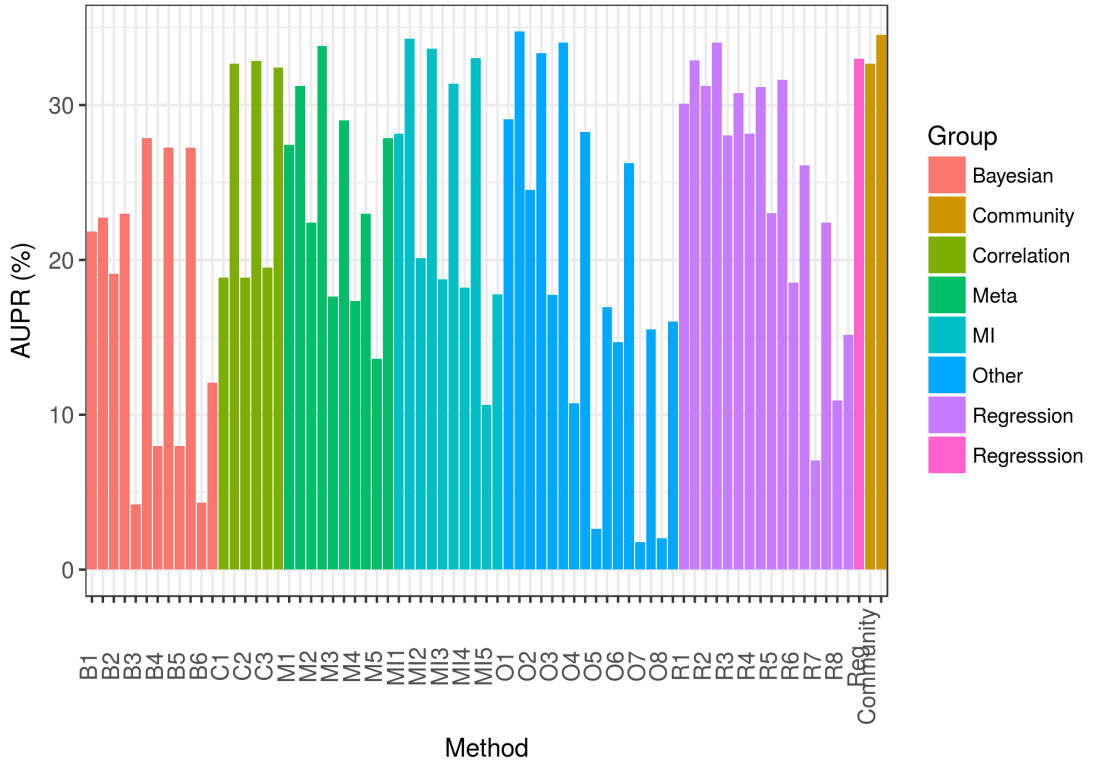
(b)

Fig. 4.8. Percentage Motif Bias for studying strategy 1.

(a) Percentage AUROC bias for motif edges; (b) Percentage bias for motif errors.

then applied strategy 1. Strategy 1 has been applied in a naive manner, neglecting the supplemental information about the different experimental conditions associated with the expression data.

The results for both the networks are shown in Fig. 4.9. We see in Fig 4.9a that strategy 1 improves the performance for all the methods including the community prediction. For the purpose of comparison, we have also included the score for the prediction generated by using the Reg part in isolation. The synergistic effect of strategy 1 is again visible here. If we look at Genie3, called Other1 in the DREAM5 challenge, which was the top performer in the in-silico category, we see that the performance after applying strategy 1 is better than both Reg and Genie3. Similar observations are true for many of the methods on this Network.

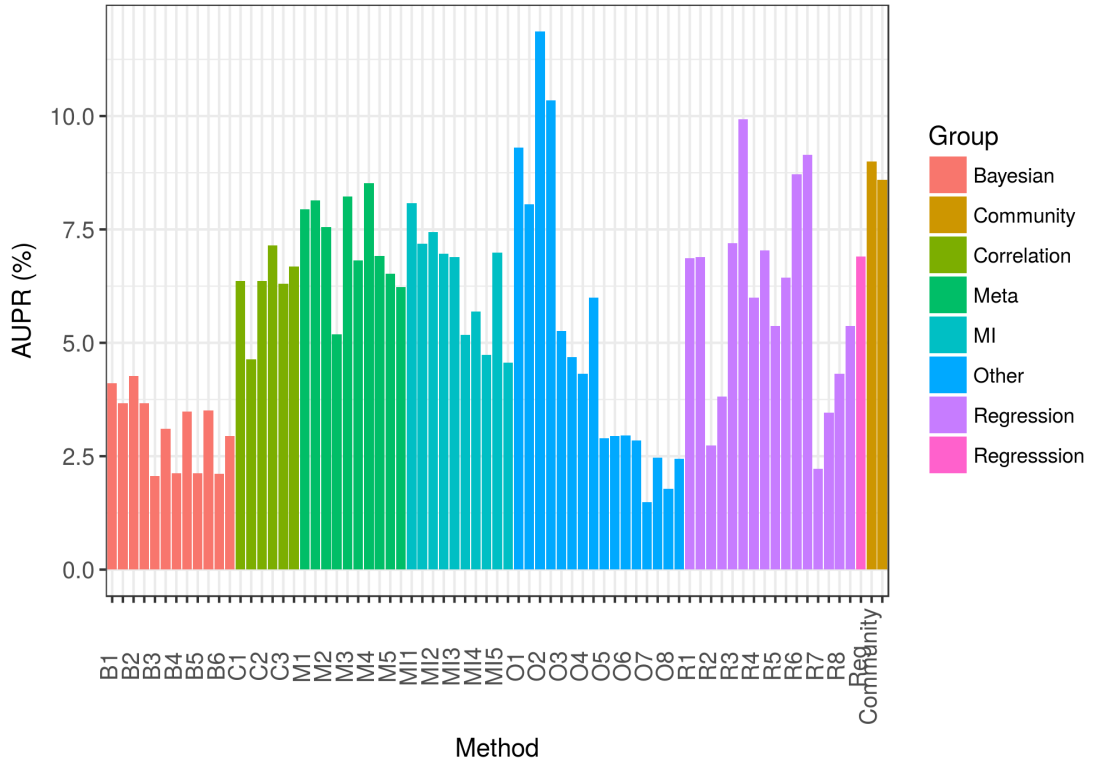


(a)

Fig. 4.9. %AUPR for methods on DREAM5 networks for studying strategy 1. ***(cont.)***

For the *E.coli* dataset, the performances of seven out of the eight regression based methods, from Regression 2 to Regression 8, improves after applying strategy 1. Interestingly, Regression 3 performs better than the second best method Genie3, in terms of AUPR values, after applying strategy 1. Apart from regression based methods, Correlation (Correlation 2 and Correlation 3), Meta1 and Bayesian (Bayesian 3 Bayesian 6) show improvement in performance with strategy 1.

However, the performances deteriorate for the rest of the methods on this network. There are a couple of reasons why this might be the case. Biological networks are more complex than in-silico systems like Network 1 [42]. Use of polynomial regression in strategy 1 might not be appropriate for capturing the complexity in biological networks. Secondly, most methods which have performed well on the *E.coli* network have leveraged the supplemental information provided about the experimental conditions, [12, 42]. In light



(b)

Fig. 4.9. %AUPR for methods on DREAM5 networks for studying strategy 1.

(a) Network 1 - In silico; (b) Network 3 - *E.coli* The abbreviations used for different class of methods- B:Bayesian, C:Correlation, M:Meta, MI:Mutual Information, O:Other, R:Regression. A number after any of these represents a method which was one of the participants in the DREAM5 challenge. For instance, *B1* refers to the method *Bayesian 1* in [12]. *Reg* is appended to all the methods after applying strategy 1. Starting from the left till the method *Reg*, only the odd number bars in the plot have been labelled with an abbreviation for a method. All the even numbered bars till *Reg* corresponds to the application of strategy 1 to the method to its immediate left. For instance, the bar to the immediate right of *B1* refers to the method *RegBayesian 1*. The bar to the right of the *Community* method belongs to *RegCommunity*.

of this information, naive application of strategy 1 without regard for this information would end up destroying the advantage leveraged by methods that exploit this knowledge about experimental conditions.

Lastly, Küffner *et al.* [42] have shown that local measures of dependency are more suitable for biological networks such as the *E.coli* network used in DREAM5 challenge to accurately identify the underlying network. This can be attributed to the fact that most experimental expression data has been collected by different labs and in a variety of different experiments.

Global measures of dependencies tend to neglect interactions which might be supported by a subset of the experiments, and would be picked up by local measures. Thus, to apply strategy 1, we need to modify the regression part to leverage the information provided about different experimental conditions; and perhaps also apply strategy 1 in a more local fashion to different subsets of the data and then integrate the findings.

- **Strategy 2 improves performance for some methods and degrades it for others** - The effect of Strategy 2 has been characterized by using 11 network inference methods. Five methods generate directed prediction networks, the rest are undirected methods. The methods cover entire range of method categories, in terms of underlying methodological approach, discussed in Chapter 3. Both global and local analyses have been conducted on the 25 medium size networks.

  Fig. 4.10 shows the score averaged across all the 25 networks. Six methods exhibit improvement in performance after applying strategy 2, while five experience a decrease. A similar trend is observed for the score averaged across individual network topologies. Except for *TIGRESS*, performance has improved for all the other methods with sparsity assumptions, i.e., *Lasso*, *GLASSO* and *GLASSO sfprior*. Though for *Lasso*, the increment in performacne is quite small. It is interesting to observe that the performance increases for *Sheridan rndprior* with strategy 2 while decreases for *Sheridan sfprior*. The former assumes a dense graph generating prior distribution while the latter uses the scale-free distribution as a prior on the space of graphs. Contrary to such a bayesian way of introducing biologically-inspired specific sparsity constraint, strategy 2 uses a crude way of introducing sparsity in the network. Thus, we see that performance increases for some sparsity based methods while degrades for others.

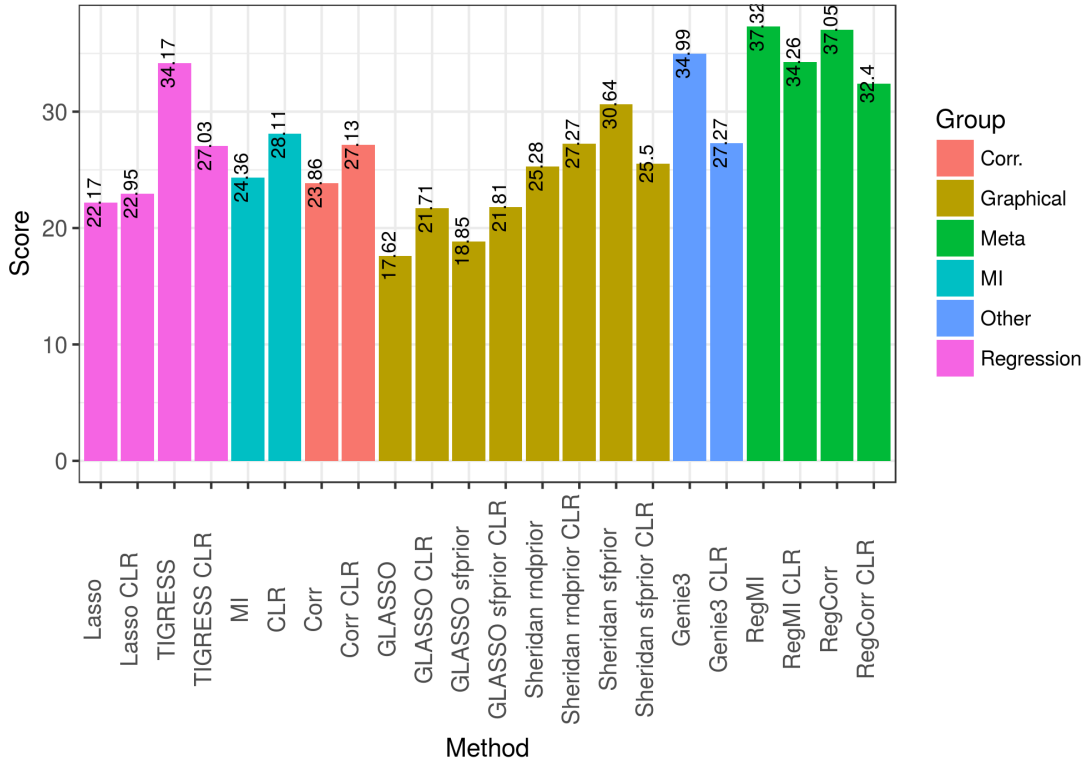
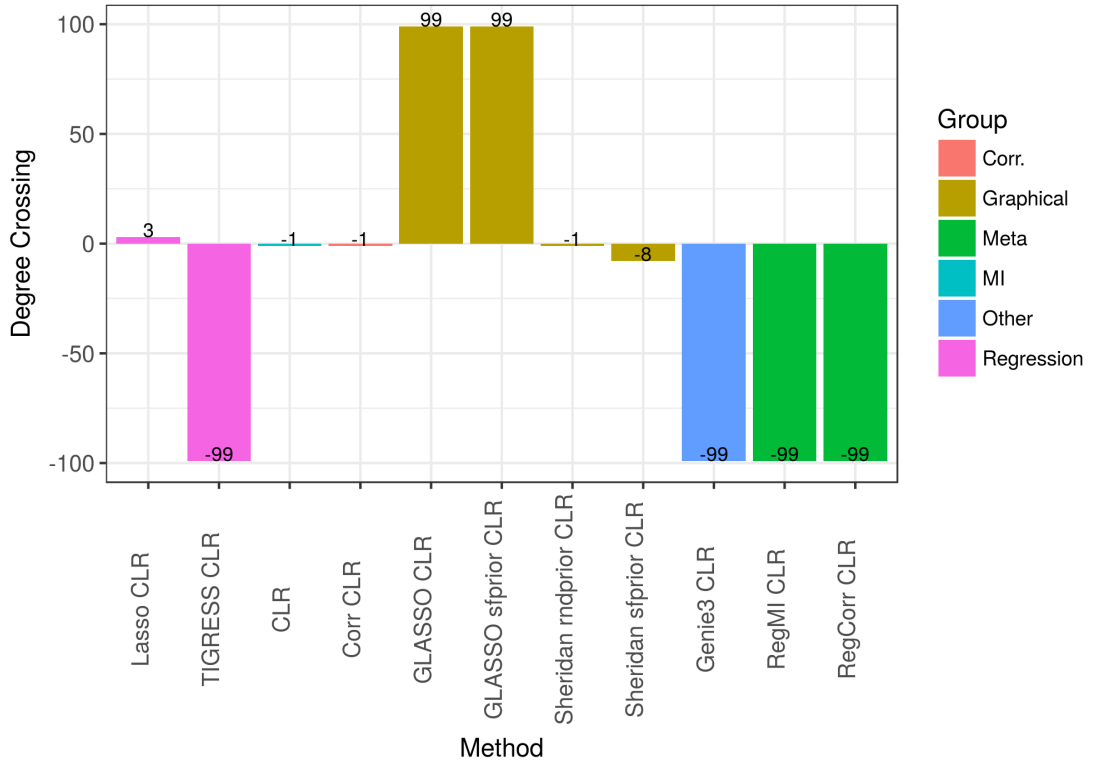


Fig. 4.10. Average Overall Score for studying strategy 2.

The overall score averaged across the five network topologies, PIPO, DREAM4, DREAM3, EIPO and EIPO Modular for studying strategy 2.

- **Strategy 2 aides in extracting the Indegree distribution of the underlying network for *Corr*, *MI*, *GLASSO* and *GLASSO sfprior*** - The degree crossing plot is shown in Fig. 4.11a. From this figure, we see that for *TIGRESS*, *Sheridan sfprior*, *Genie3*, *RegMI* and *RegCorr* strategy 2 leads to a decline in the method's ability to extract the indegree distribution. Whereas *Corr* and *MI* are better able to extract indegrees greater than 1 after application of strategy 2, justifying the use of strategy 2 on these two methods on a regular basis. Interestingly, the behaviour on *Sheridan rndprior* is the same. Moreover, *GLASSO* and its scale-free variant, completely dominate with strategy 2. However, *Lasso* and *Sheridan rndprior* show a very limited ability to aide in indegree estimation, but in different regions to each other. *Lasso* performs well only for low indegrees, while *Sheridan sfprior* does for huge indegrees.

Degree crossing plot for outdegree is given in Fig. 4.12. Disparate to indegree,



(a)

Fig. 4.11. Degree Crossing for Indegree for studying strategy 2.

The degree crossing for Indegree averaged across all the methods for studying strategy 2. Within the context of the discussion in Section 2.3.2, with regards to degree crossing, a given method's version with strategy 2 is compared against the base method. Thus, with strategy 2 the method would be M_1 and the base method would then be M_2. So, the conclusions regarding degree crossing from 2.3.2 would hold accordingly. A value of zero for the degree crossing implies that M_1 dominates M_2. A positive value means that M_1 dominates M_2 up to a degree equal to the degree crossing value and after that M_2 dominates M_1. A negative value implies that M_2 dominates method M_1 up to a degree equal to the absolute value of the degree crossing and after that M_1 dominates.

strategy 2 augmented *GLASSO*, *GLASSO sfprior* and *Sherian rndprior* have limited capabilities on outdegree with degree crossings equal to 7 and 6 respectively. For outdegree distribution, even *Corr* and *MI* have limited estimation capability; the degree crossings are 6 for both. Again, illustrating the dilution effect from indegree to outdegree. However, some methods have reversed their behaviour. For instance, *Sheridan sfprior* now dominates only at low indegrees. *Lasso* and *RegMI* have also switched region of dominance with strategy 2.

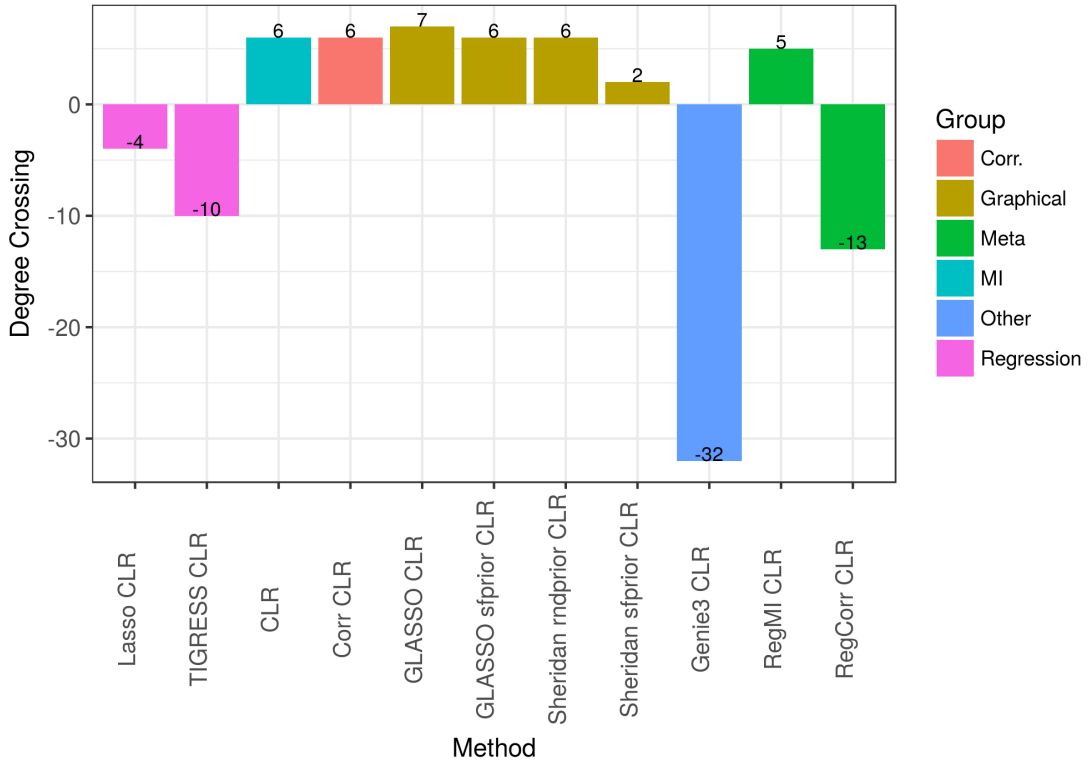


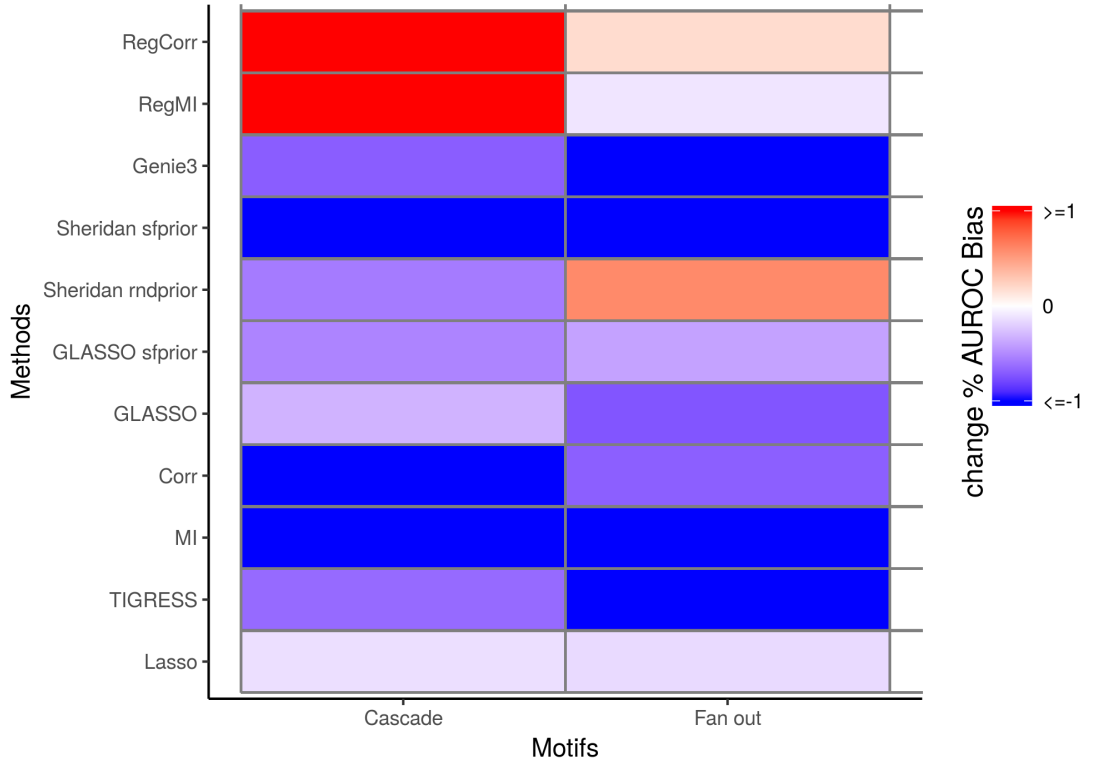
Fig. 4.12. Degree crossing for outdegree for studying strategy 2.

The degree crossing for outdegree averaged across all the networks for studying strategy 2. The inference strategy is the same as described in Section 2.3 and the caption for Fig. 4.11a.

- **Strategy 2 reduces bias for Cascade and Fan-out error** - The aim of strategy 2 is to induce sparsity [34]. Thus, if successful, it would lead to reduction in false positives. In this study, we see that strategy 2 leads to a reduced bias in Cascade and Fan-out error Fig. 4.13. *RegMI* and *RegCorr* are the anomalies here. This could be due to the fact strategy 2 was not directly applied to *RegCorr* and *RegMI*, rather strategy 2 was applied to Corr and MI and then strategy 1 was applied.

4.4 Conclusion

In this chapter we have reaffirmed the conclusion of the DREAM5 network inference challenge that meta methods, which aggregate the predictions from different



(a)

Fig. 4.13. Percentage Motif Bias for studying strategy 2.

methods, might perform better than any of the individual approaches. A rank average based aggregation method was used in the DREAM5 study. However, here we have used a product based aggregation strategy. We have shown that strategy 1 introduced in this chapter consistently improved the performance for the class of undirected methods across 25 synthetically generated, topologically varied networks. The performance enhancement had far-reaching consequences in terms of better estimating the degree distributions of the underlying networks. Further, it was also demonstrated that this strategy even when applied naively, without consideration for the type of experimental data and the underlying conditions, might lead to an increment in performance. This was seen for all the participating methods in DREAM5 challenge over the in-silico network and for some of the methods on the *E.coli* network. Another post-processing based strategy was also introduced, which also improved the performance for a class of undirected method.

However, the results in this chapter also warn against inappropriately used aggregation of different methods. For instance, with the *E.coli* network, performance of the top methods on this network decreased with strategy 1. This highlights the shortcomings of a naively applied meta analysis with regards to strategy. Since this data, like most real biological datasets, had different types of experimental conditions, the application of any meta analysis has to be cognisant of this fact. For instance, the strategy introduced in the method labelled Regression 3 in the DREAM5 challenge, could be used to leverage the knowledge pertaining to the different experimental conditions. It is worthwhile to note that for this method, strategy 1 gave a significant improvement on the *E.coli* network, leading to a performance better than the second best method. Regression 3 uses self organizing maps to assigns an ordering to the various experiments in the expression data. Further, an ordering is assigned to the experimental conditions within an experiment as well. Strategy 1 could potentially be used with such an ordering of the experiments to infer either a global network or multiple networks over the different experiments which can then be aggregated. Another potential way to appropriately apply strategy 1 could be to use local regression techniques such as using gaussian basis function regression on the ordered set of experiments to capture the local trends in the data. It has been shown in [42], that local analysis might lead to significantly better performance than global analysis. Furthermore, simple guidelines such as using appropriately lagged values for the time series data should be regularly used.

Besides the aforementioned implementational insights, the results in this chapter provide another set of evidence in favor of using a meta-analytic approach towards network inference. It has been shown previously on occasions [12, 59] that multiple methods, when combined to leverage the strengths of each method, may give signficantly better performance than any individual method.

Chapter 5

Structural Prior on Degree Distribution

5.1 Introduction

Generally, the data available from microarray experiments, matrix X, is high-dimensional; the number of rows-experimental conditions and/or timepoints-is significantly smaller than the number of columns, the number of genes for which measuements are made. Put succintly, this means that $n \ll p$. A direct consequence is that the inference task is underdetermined: in the unconstrained form a unique solution does not exist [19]. Given p genes, there are $p(p-1)$ possible interactions [1], which grows in an exponential manner with the number of genes. If genes in the network have multiple regulators (transcription factor genes), then we have a case of combinatorial regulation; and this drastically increases the size of the solution space. Due to the large solution space coupled with limited and uninformative data [12, 19] it becomes infeasible to find a unique solution; a large number of solutions with equal conformity to the data are possible. Inference techniques need some form of approximation to reduce the size of the solution space to infer a high fidelity network in a time-judicious manner. Approximations

[1] This assumes that the transcription factors or regulators are among these p genes.

to handle the combinatorial nature of gene regulation were discussed in Chapter 3. Another way to shrink the solution space is to use complementary sources of information. Inference techniques could incorporate domain knowledge in the form of prior information. Curated databases such as RegulonDB maintain lists of verified gene-gene interactions for different organisms; these identified links can be fed into the inference algorithms as prior information to enhance the performance [60, 61, 62].

A rich source of prior information is the known structural properties of GRNs. Lots of effort has been expended in studying these structural properties [2, 15, 16, 17, 18, 28]; GRNs are known to have exponential indegree and scale-free outdegree distributions, have modular structure, are composed of small overexpressed subgraphs that have rich dynamic properties, etc. Other than edge-based priors, these more generic sources of information could be leveraged to augment the network inference task. Modular structural priors have been used fairly often. Knowledge pertaining to degree distribution of the regulatory links in the network has been explored to a limited extent. Motifs have been even less systematically explored for restricting the size of the solution space [39, 63]. In this chapter we focus on degree distributions as a means of aiding the network inference task. Section 1 briefly discusses some of the efforts made at incorporating the degree distribution information in the network inference task, Section 5.2 introduces a simulated-annealing based method for imposing a scale-free distribution on the indegree, Section 5.3 presents the results of experiments conducted on different methods that include the degree distribution prior and Section 5.4 offers a discussion of the results from Section 5.3 and offers potential directions for future research.

5.2 Incorporation of degree distribution as a form of structural prior for network inference

Many network inference methods impose the constraint of sparsity to reduce the solution space of networks [34, 35, 36, 43, 44, 51, 58]. However, GRNs belong to a special subset of sparse networks; GRNs are known to have scale-free degree distribution [2, 15]. Thus, sparsity constraint alone might not be sufficient to appropriately reduce the size of the solution space. A more principled approach is to constrain the solution space to networks which have a scale free degree distribution. Bayesian framework offers the most natural way to impose a prior distribution on the space of networks. This can formally be expressed as given in Eq 5.1

$$P(G, X; E, V) = P(X|G)P(G) \tag{5.1}$$

where G is a graph-directed or undirected and with edge set E and vertex set V; G represents the underlying gene regulatory network, $P(G, X)$ is the joint distribution for data X and the graph G, $P(X|G)$ is the likelihood for the data and $P(G)$ is the prior distribution over the class of possible networks. An unbiased estimate for $P(G)$ would be a uniform distribution. However, biologically such a prior would not make sense and given the limited nature of the available data would not be feasible as well. Methods which impose a simple sparsity constraint do not explicitly offer the prior distribution being appealed to; most of these methods do not use the bayesian approach. Thus, in the absence of any guarantee about the nature of the search space, the structural properties of the inferred network are unknown.

Given the fact that biological networks are scale-free, the obvoius choice for $P(G)$ is a distribution that favors scale-free distributed networks. Different methods have tried to exploit this knowledge [29, 40, 54, 55, 56, 57, 64, 65] for reducing the size of the solution space. With the exception of [40], all the other methods belong to the class of undirected techniques, most of which use the framework of

gaussian graphical models (GGM) to incoporate the scale-free degree distribution as prior information. GGM based methods use a bayesian framework for incorporating the prior while the other methods apply deterministic techniques to restrict the solution space. We bin these methods into three categories-Adjacency Thesholding, Graphical and Binary Programming. The details for these categories are presented next.

- **Adjacency Thresholding**

  WGCNA [29] introduces a simple thresholding based method for selecting a scale-free network from a given adjacency matrix. Two thresholding schemes are possible, "soft" and "hard". The thresholding process is a function of usually a single parameter, and varying the value of this parameter leads to networks with differing amounts of sparsity. The selection of the parameter is done using a "scale free topology criterion"; the parameter is selected such that the thresholded network has scale-free topology.

  Given a positive, symmetric and weighted adjacency matrix $W \in \mathbb{R}^{p \times p}$, where p is the number of genes in the network, and an associated parameter β the method proceeds as follows.

  1. Vary β in the range $(\beta_{min}, \beta_{max})$.
  2. For each value of β threshold the weighted adjacency matrix, and for the obtained network perform a least-squares linear fit between the degree distribution and the degree values on a log-log scale as given in Eq 5.2.

  $$y_\beta(k) = w_0 + w_\beta x_\beta(k) + \epsilon_\beta, k \in \{1, 2, \ldots, d\} \tag{5.2}$$

  where k is the degree, d the maximum degree in the thresholded network, $y_\beta(k) = log(p(k))$, $x_\beta(k) = log(k)$ and ϵ_β is the error term.

  3. Select the value of β for which R^2 value for the least squares fit for Eq 5.2 is greater than 0.80 and w_β is ¡ 0. Further, ensure that the mean connectivity of the thresholded network is high.

When hard thresholding is used, β corresponds to a value in the range $(0, 1)$ [2] such that entries below β are set to 0, while those above β are set to 1. Under soft thresholding, β takes on integer values and each entry of W is modified as given in Eq 5.3.

$$W_{ij} = W_{ij}^{\beta} \tag{5.3}$$

where W_{ij} is the ijth element of W. After applying Eq 5.3, some entries in W become zero.

This way of extracting scale-free networks has issues. The R^2 value for the least squares fit in Eq 5.2 is not a statistically good measure to assess the goodness of fit to a power law curve [66]. The relationship between R^2 and β is noisy [29]. Finally, this method acts as a post-processing procedure that does not utilize the scale-free prior in a systematic way for the task of network inference.

- **Graphical**

  The methods under the ambit of this category, utilize the framework of gaussian graphical models (GGM) to infer a gene regulatory network. The expression data is assumed to belong to a multivariate gaussian distribution, and network inference reduces to the task of estimating the non-zero entries of the precision matrix. To induce sparsity in the inferred precision matrix, l_1 regularization on the elements of the precision matrix has been used regularly. To estimate the l_1 constrained structure of the precision matrix many methods have been devised such as [51, 52, 53] and others. One class of methods within this category either modify the l_1 constraint or introduce new constraints that restrict the solution space to precision matrices which have scale free or approximately scale free degree distribution.

  If $X \in \mathbb{R}^{n \times p}$ is the expression matrix, under the GGM framework, X is assumed to have a p-variate gaussian distribution $\mathcal{N}(0, \Sigma)$, where Σ is the

[2] It is assumed here that the entries of the weighted adjacency matrix are constrained to lie between 0 and 1.

covariance matrix. In order to estimate the underlying GRN, we need to estimate the precision matrix $\Omega = \Sigma^{-1}$. Generally, a constrained optimization problem is solved, which is defined in Eq 5.4.

$$\underset{X}{minimize} S(X, \Omega) = L(X, \Omega) + \lambda C(\Omega) \tag{5.4}$$

where $L(X, \Omega)$, the cost function, takes on different forms depending upon the specific methodology. $C(X)$ is the constraint term added to induce scale-free distributed precision matrix Ω, λ is a parameter and $S(X, \Omega)$ is the total cost function to be minimized. The function $C(.)$ penalizes the precision matrix so as to infer a scale-free or approximately scale-free degree distribution. Depending upon the formulation, the form of $C(.)$ would be different. Usually, different methods proceed by defining an approximate probability term for a graph in terms of the degree distributions of the genes, which is given in Eq 5.5.

$$p(G = (V, E); \epsilon, \gamma) \sim \prod_{i \in V} (deg(i) + \epsilon)^{-\gamma} \tag{5.5}$$

where G is an undirected graph corresponding to the GRN, V is the vertex set, E is the edge set, $deg(i)$ is the degree of gene i, ϵ and γ are the parameters of the distributions. The negative log-likelihood of Eq 5.5 is then used as the function $C(.)$ in Eq 5.4.

[54] estimates $deg(i)$ using the term given in Eq 5.6.

$$deg(i) = \left|\left| \sum_{j \neq i} \Omega_{ij} + \epsilon_i \right|\right|_1 \tag{5.6}$$

where $deg(i)$ is the degree of the ith gene and ϵ is a small term added to avoid the degree from becoming zero. $C(.)$ is given as defined in Eq 5.7

$$C(\Omega) = \sum_{i=1}^{p} log(deg(i)) + \beta \sum_{i=1}^{p} \left|\Omega_{ii}\right| \tag{5.7}$$

Finally, the optimization problem in Eq 5.4 is solved using a minorize-maximize algorithm. It is shown that Eq 5.4 can essentially be solved by repeated application of any of the methods for solving l_1 regularized formulation for GGMs. The constrained term in Eq 5.7 has been shown to be equivalent to imposing and approximate log-normal prior on the underlying graph; and log-normal distribution can be used as an approximation to scale free distribution. Combination of [54] with *GLASSO* will be referred to as *GLASSO sfprior* henceforth.

[55] uses submodular relaxation to approximate $deg(i)$ and arrives at a $C(.)$ given by Eq 5.8

$$C(\Omega) = \sum_{i=1}^{p} \sum_{j=1}^{p-1} (log(j+1) - log(j)) |\Omega_{i,(j)}| \tag{5.8}$$

where $\Omega_{i,(j)}$ is a permutation of the elements of the ith row of Ω such that $|\Omega_{i,(1)}| \geq |\Omega_{i,(2)}| \geq ... \geq |\Omega_{i,(p-1)}|$. The C described in Eq 5.8 mimics a l_1 norm except for the additonal weight, which characterizes how each edge ranks with respect to the other edges of its neighbouring edges. Alternating direction method of multipliers (ADMM) is used to optimize modified Eq 5.4. Similary, different assumptions regarding $deg(i)$ will lead to different forms for the score function $L(.)$, which can be optimized accordingly.

Another class of methods infer the gene regulatory network from a posterior distribution on the structure of the gene regulatory network by imposing a more rigorous prior and using a Markov Chain Monte Carlo (MCMC) method [57] for model selection. [57] introduces a scale-free prior on the space of possible networks using the static model presented in [67]. To a given graph G with vertex v and edge set E, a prior probability is assigned by introducing probability of existence or non-existence of edges between pairs of genes. Under assumption of independence, the probability for the

entire graph can then be given as Eq 5.9

$$P(G) = \prod_{ij \in E} \left(1 - (1 - 2s_i s_j)^{pK}\right) \prod_{ij \notin E} \left(1 - 2s_i s_j\right)^{pK} \tag{5.9}$$

where s_i and s_j are weights assigned to genes i and j, K is a parameter of the model. To assign the weights, a permutation σ of the gene labels is used. The term $(1 - 2s_i s_j)pK$ represents the probability of no edge being present. Under σ the weights are assigned as given in Eq 5.10

$$s_i = \frac{\sigma_i^{-\mu}}{\sum_{k=1}^{p} \sigma_k^{-\mu}} \tag{5.10}$$

where μ is the Zipf exponent and lies between 0 and 1. A metropolis hastings sampler is used to sample from the posterior $P(G \mid X) \propto P(X \mid G)P(G)$. The most crucial parts of the update process in the metropolis hastings sampler are the gene permutation *sigma* and network decomposability. *sigma* is updated by randomly selecting any two genes and swapping their labels. Network decomposability is of paramount importance to the entire algorithm, since likelihood for GGMs can efficiently be updated over decomposable graphs. Thus, an edge addition or deletion, while updating the network, is only accepted if the resulting graph is decomposable. A weighted adjacency matrix can be created by estimating the frequency of each edge along the MCMC chain. An interesting aspect of this method is that the exponent of the scale-free distribution can also be estimated.

The difference in the two classes of graphical methods that incorporate a scale free prior on the structure of networks is the form of the prior. Methods like [54] and [55] use a crude approximation for the prior, while [57] applies a more rigorous one. Moreover, the network model used for the prior in [57] has been shown to faithfully generate scale free networks [67]. Thus, for [57] we have high confidence in the structural properties of the inferred network. [57] even estimates the exponent for the scale free distribution of the underlying gene regulatory network, which is absent from [54, 55] and

similar methods. However, the use of MCMC based sampling makes [57] computationally more demanding. The method in [57] has been referre to as *Sheridan* in this work.

- **Binary Programming**

  RegCorr, [40], introduced a systematic way for inferring scale free networks under a binary programming formulation. The central idea is to incorporate scale-free prior on the indegree distribution of edges by estimating the expected distribution of incoming links to the genes and then solving an integer programming problem to assign the number of regulators for each gene in the network. The indegree distribution is estimated using a truncated scale free distribution as shown in Eq 5.11

$$P_{sf}(d) = \begin{cases} ck_{min}^{-\gamma_{in}}, & \text{if } 1 \leq d \leq k_{min} \\ cd^{-\gamma_{in}}, & \text{if } k_{min} < d \leq N \end{cases} \tag{5.11}$$

  where $P_{sf}(d)$ is the probability of finding genes with indegree d, k_{min} is a positive integer, N is the maximum indegree for the network, γ_{in} is the scale free exponent for the distribution and $c = (k_{min}^{1-\gamma_{in}} + \sum_{d=k_{min}+1}^{N} d^{-\gamma_{in}})^{-1}$. Under such a probability distribution, number of genes with indegree d can be estimated as $Genes(d) = \lfloor pP_{sf}(d) \rfloor$.

  To assign indegrees to each gene in the network, a cost is associated with assignment of a particular indegree to a gene. To caluclate the cost for assiging indegree d to gene j, multivariate regression is performed for gene j assuming that it is regulated by d regulators. The d regulators are selected based on the ordered values in the jth column of matrix SSE introduced in Chapter 4. If σ_j is the list of potential regulators for gene j, then $SSE_{\sigma_j(1)j} \leq SSE_{\sigma_j(2)j} ... \leq SSE_{\sigma_j(p-1)j}$. σ ranks all the genes, except for gene j, in decreasing order of their capability to regulate gene j. Given σ, a cost can be assigned for every assignment of indegree to all the genes in the network. The binary programming formulation can now be defined as follows.

$$
\begin{aligned}
\underset{b}{\text{minimize}} \quad & \sum_{d=1}^{N}\sum_{j=1}^{p} C(d,j)b_{dj} \\
\text{subject to} \quad & \sum_{j=1}^{p} b_{dj} = Genes(d), \quad d = 1,\ldots,N, \\
& \sum_{d=1}^{N} b_{dj} = 1, \qquad\qquad j = 1,\ldots,p, \\
& b_{dj} \in \{0,1\}; \quad d = 1,\ldots,N; \quad j = 1,\ldots,p
\end{aligned}
\tag{5.12}
$$

where $C(d,j)$ is the cost associated with assigning indegree d to gene j, b_{dj} is a binary variable characterizing whether indegree d will be assigned to gene j or not. Formulation 5.12 solves for the binary variables b_{dj}, thus effectively assigning indegrees to all the genes. The first constraint in 5.12 ensures that the assignment of indegrees to the genes in the network follows the scale free distribution introduced in Eq 5.11. The second constraint accounts for the fact that each gene will have a unique indegree; for gene j only one of the variables in $b_{dj}, d \in \{1,2,\ldots,N\}$ would be non-zero.

Cost $C(d,j)$ is computed using multivariate regression. For indegree d and gene j the following regression problem is solved.

$$
{}^{u}x_j = a_{0j} + \sum_{k=1}^{d} a_{kj}\,{}^{u}x_{\sigma_j(k)} + \epsilon_{dj}, \quad u \in \{1,2,\ldots,n\} \tag{5.13}
$$

where ${}^{u}x_j$ is the uth sample in the jth column of expression matrix X, a_{0k} and a_{kj} are the coefficients and ϵ_{dj} is the noise term. Eq 5.13 can be easily solved using least squares regression. Further, sum of squared residuals and sum of deviation squares can also be calculated for the model in Eq 5.13 as follows.

$$
R_{dj} = \sum_{u=1}^{n} ({}^{u}\hat{x}_j - {}^{u}x_j)^2 \tag{5.14}
$$

$$D_{dj} = \sum_{u=1}^{n} ({}^{u}\hat{x}_j - \bar{x_j})^2 \tag{5.15}$$

where R_{dj} is the sum of squared residuals, D_{dj} is the sum of deviation squares, ${}^{u}\hat{x}_j$ is the least squares estimate of ${}^{u}x_j$ from Eq 5.13 and $\bar{x_j}$ is the mean of the expression values for gene j. The cost function can now be defined as given in Eq 5.16.

$$C(d, j) = R_{dj} exp\Bigg(- \frac{D_{dj}/d}{R_{dj}/(n-d-1)}\Bigg) \tag{5.16}$$

The cost function captures how well the expression of gene j is explained by d regulators, thus quantifying combinatorial nature of regulation to some extent. The larger the cost function, the less likely it is that gene i would be regulated by d regulators. The term $(-\frac{D_{dj}/d}{R_{dj}/(n-d-1)})$ is equivalent to the F-statistic for multiple linear regression that characterizes goodness of fit. Thus, effectively, the cost function would be low when R_{dj} is low and $(-\frac{D_{dj}/d}{R_{dj}/(n-d-1)})$ is high. This would suggest that not only the d regulators capture the variation in the expression of gene j well, also the model is statistically relevant.

The binary problem 5.12 can be easily solved using any mixed integer programming solver (MISP). The solution thus obtained gives the required gene regulatory network. [40] does not use this inferred network directly, rather employs it in a post-processing step to change the ordering of the edges in the prediction list generated by the main technique introduced. Edges in the inferred scale free network are upranked in the final predicted adjacency matrix such that they are at the top of the list. If W is the matrix inferred by the main method introduced in [40] and W_{sf} is the inferred scale-free network, then the final prediction matrix is given as follows.

$$\hat{W} = W + \delta * W_{sf} \tag{5.17}$$

where $\hat{W}$ is the final predicted adjacency matrix, $\delta = \underset{d+1\leq i\leq p}{max}\ \underset{1\leq j\leq p}{max} W_{ij}$ is the amount by which the edges in W_{sf} are upranked in W.

5.3 Simulated annealing based inference of scale free networks

The binary programming based inference of scale free networks introduced in [40] and discussed in the previous section only infers a single binary network. Moreover, the inferred network is only used as a post-processing approach. Even the performance benefit offered by the inferred scale free network is highly sensitive to the selection of the parameters k_{min}, γ_{in} and N. No procedure has been offered to make a selection of these parameters. Thus, the application of the prior is extremely limited. In this section we devolop a simulated annealing based procedure to circumvent the need to select the parameters. Further, we show that by recording the frequency of occurence of each edge along the simulated annealing chain, the prior now can be used as an independent network inference method. We also adapt Eq 5.11 to include different priors. Specifically, we consider uniform, binomial and exponential priors.

5.3.1 Formulation

The total cost function $TC(X, P_{sf}) = \sum_{d=1}^{N}\sum_{j=1}^{p} C(d,j)b_{dj}$ in Eq 5.12 represents an estimate of the error inherent in the estimated network given data X and the assumed degree distribution P_{sf}. Thus, given two degree distributions P_{sf}^1 and P_{sf}^2 parametrized by different parameters, we assume that the model with a lower total cost function would be a better fit to the data. With this assumption we can now propose a simulated annealing based approach to optimize the total cost function $TC(.)$. The proposed algorithm, which we call *SAprior* is given as follows.

1. Initialize the parameters.

- Indegree distribution parameters - $k_{min}^{cur} = k_{min}^{0}$, $\gamma_{in}^{cur} = \gamma_{in}^{0}$, $N^{cur} = N^{0}$, $k_{min}^{prop} = k_{min}^{0}$, $\gamma_{in}^{prop} = \gamma_{in}^{0}$, $N^{prop} = N^{0}$.
- Binary programming parameters - $TC^{cur} = 0$, $TC^{prop} = 0$
- Simulated annealing parameters - $T_0 = 1$, $T = T_0$, $T_{step} = 0.1$, $nburnIn = 30$, $nIterations = 150$, $nIter_T = 10$, $counter = 0$

2. Repeat steps 3 to 9 $nIterations$ number of times.

3. Update simulated annealing parameters.

   - $T = (T_0 - T_{step})^{counter}$
   - $counter = counter + 1$

4. Repeat steps 4 to 9 $nIter_T$ number of times.

5. Generate a resampled version of expression data $X_{resampled}$.

6. Generate new estimates for the indegree distribution parameters using the proposal scheme - $k_{min}^{prop} = Proposal_k(k_{min}^{cur}, T)$, $\gamma_{in}^{prop} = Proposal_\gamma(\gamma_{in}^{cur}, T)$, $N^{prop} = Proposal_N(N^{cur}, T)$.

7. If $(k_{min}^{prop} < k_{low})\text{OR}(k_{min}^{prop} > k_{high})\text{OR}(\gamma_{in}^{prop} < \gamma_{low})\text{OR}(\gamma_{in}^{prop} > \gamma_{high})\text{OR}(N^{prop} < N_{low})\text{OR}(N^{prop} > N_{high})$ return to step 6. Otherwise, proceed to step 8.

8. Solve the optimization problem in 5.12 for both the current and proposed estimates for the indegree distrbutions using the resampled data $X_{resampled}$ and update the corresponding cost functions TC^{cur} and TC^{prop} respectively.

9. if $exp(-\frac{(TC^{prop} - TC^{cur})}{T}) > 1$ accept the proposed degree distribution and update the degree distribution parameters. Otherwise, generate a uniform random number $u \in (0, 1)$. If $exp(-\frac{(TC^{prop} - TC^{cur})}{T}) > u$ accept the proposed distribution and update the parameters otherwise reject it. If the proposed distribution is accepted the parameters are updated as follows.

   - Indegree distribution parameters - $k_{min}^{cur} = k_{min}^{prop}$, $\gamma_{in}^{cur} = \gamma_{in}^{prop}$, $N^{cur} = N^{prop}$

If the proposed degree distribution is accepted and $counter > nburnIn$, save the gene regulatory network associated with the accepted degree distribution.

With respect to the parameters in the algorithm, superscript *cur* identifies parameters for the current while *prop* for the proposed indegree distributions. Subcsripts *low* and *high* represent lower and upper bounds on parameters respectively. Description for the other parameters are as follows.

- T_0 = Initial temperature for simulated annealing; T = Current temperature; T_{step} = Step by which the temperature decreases at every iteration.
- $nburnIn = 30$ is the number of burn-in steps after which the graphs in the simulated annealing chain are recorded; $nIterations = 150$ is the total number of outer iterations in the *SAprior*, $nIter_T = 10$ is the number of times the inner loop in *SAprior* is executed, $counter = 0$ is a running counter to keep track of the number of outer iterations.

In addition to solving the binary programming problem, there are three other crucial steps in *SAprior* - resampling the expression matrix, generation of a new estimate for the indegree distribution and acceptance or rejection of the proposed distribution. Next, we discuss these three aspects.

- **Data resampling**

  Data resampling has been previously shown to improve the performance of various network inference techniques [12, 43, 59]. Resampling aides in making statistically stable estimates. Further, resampling adds variation to the estimates made in different iterations. We found that the adjacency matrix accumulated by looking at the frequency of occurence of each edge along the simulated annealing chain was sparser without resampling. The performance was also lower wihtout data resampling. Thus, we have included the resampling step. In our experiments on the size 100 networks we found that taking twice as many resamples, with replacemnt, as there are samples

in the original expression matrix, gives a better performance. Thus, in all our experiments we adopt this resampling strategy

- **Proposing new degree distributions**

  In the binary formulation 5.12, the indegree distribution is contingent upon three parameters k_{min}, γ_{in} and N. Thus, in *SAprior* to generate a new estimate for the indegree ditribution, we need to generate estimates for k_{min}, γ_{in} and N. For this purpose, we need proposal functions for each parameter. For a given current value k_{min}^{cur}, we generate k_{min}^{prop} by uniformly sampling an integer from the interval $(k_{min}^{cur}-1, k_{min}^{cur}, k_{min}^{cur}+1)$. A similar startegy is used for generating N_{prop}. For γ_{in} we use a gaussian proposal distribution centred at (γ_{in}^{cur} and having a variance equal to the current temperature.

- **Acceptance of proposed indegree distribution**

  The central tenet of *SAprior* is that the total cost function in formulation 5.12 can be used as an objective function in a simulated annealing process. Consequently, if the cost function for the proposed ditribution is lower than that for the current distribution, the proposed distribution is accepted. Thus, all proposed indegree distributions which have a better fit to the multivariate regression model compared to the current distribution are always accepted. While proposed distributions with a poorer fit are visited less often. Thus, the networks visited in the simulated annealing chain can be aggregated to give an estimate of the underlying gene regulatory network.

5.3.2 Network inference with SAprior

To gauge an estimate of the underlying gene regulatory network with *SAprior*, we adopt two different strategies. The first is based on aggregation of all the network along the simulated annealing chain. The second strategy uses *SAprior* to augment an adjacency matrix inferred by any other network inference technique. These strategies are discussed as follows.

- **Edge frequencies**

  In this work, the collection of all the points visited by *SAprior* is referred to as the simulated annealing chain. If l represents the lth point in the simulated annealing chain and W_l is the running adjacency or prediction matrix, then we get the following update equation for W_l.

$$W_l = W_{l-1} + W_l^{SA} \tag{5.18}$$

  where W_l^{SA} is the network from the accepted proposed distribution at point l in the chain. Interestingly, W_l^{SA} is binary and sparse, whereas W_l is neither. Consequently, if W_l is normalized, it assigns to each edge in the network a weight or probability of exitence. Thus, W_l can be used for performance analysis within the present framework. It is worthwhile to note that a similar strategy for aggregating multiple binary networks to generate weighted predictions has been used by many methods before [43, 57, 58].

- **Upranking**

  As previously discussed in Section 4.2, [40] uses an upranking strategy to leverage the network inferred using formulation 5.12. We observe that this strategy is not unique to the method introduced in [40]. Thus, using this upranking strategy, we introduce a way of using *SAprior* to augment the predictions made by any other inference technique. Given an adjacency matrix W^M predicted by method M and as before W_l is the running adjacency or prediction matrix along the simulated annealing chain. Then the update equation for W_l can be stated as follows.

$$W_l = W_{l-1} + (W^M + \delta * W_l^{SA}) \tag{5.19}$$

  where, $\delta = \underset{d+1 \le i \le p-1}{max} \underset{1 \le j \le p}{max} W_{ij}^M$ is the amount by which the edges in W_l^{SA} are upranked in W^M. Thus, at every point l along the chain, some edges in W^M are upranked and the outcome is added to W_{l-1}. At the end, W_l

can be normalized by the length of the chain. The final W_l has an intuitive interpretation; it is the average of all the upranked W^M. Generally, many resampling techniques resample the expression data and re-infer the prediction matrix and use the average of all the inferred matrices for statistical stability. In the upranking strategy that we have used, rather than re-inferring the networks from method M we have used resampling within *SAprior*.

5.3.3 Incorporating complex combinatorial regulation

Inferelator [44] introduced a way to incoporate complex transcription factor interaction programs such as **AND**, **OR** and **XOR** in the network inference process. The introduced encoding is purported to be able to extract true combinatorial regulation; such regulation involves transcription factors interacting with each other to effect the regulation of the target gene. Since the methodology of [44] also uses multiple regression framework, it is straightforward to include the complex interaction term in *SAprior*.

For the purpose of illustraion, let us assume that $d = 2$, then incorporation of the interaction term would modify Eq 5.13 as follows.

$$ {}^{u}x_j = a_{0j} + \sum_{k=1}^{2} a_{kj}\,{}^{u}x_{\sigma_j(k)} + \beta min({}^{u}x_{\sigma_j(1)}, {}^{u}x_{\sigma_j(2)}) + \epsilon_{dj}, \quad u \in \{1, 2, \ldots, n\} \tag{5.20} $$

where β is the coefficient for the interaction term. Only second order interactions have been considered. If $d > 2$, interactions can be included for all pair of transcription factors. Thus, *SAprior* can be easily modified to include interaction terms for transcription factors.

5.4 Results and Discussions

We have conducted two sets of experiments with regards to this chapter. The first set of experiments perform a comparative analysis of three degree distribution prior incorporating techniques. For this purpose we compare our developed method *SAprior*-with edge frequency based aggregation against *Sheridan sfprior* and *GLASSO sfprior*. For the next set of experiments, we study the effect of *SAprior* with upranking strategy on 27 network inference methods. Next we discuss both the experimental setups.

5.4.1 Experimental setup 1: Comparison of scale free prior methods

We have compared the performances of *SAprior*, *Sheridan sfprior* and *GLASSO sfprior* using the 25, size 100 networks. *SAprior* has been used with four different degree distributions, uniform, binomial, exponential and scale free, and these are called, *SAprior Uniform*, *SAprior rnd*, *SAprior exp* and *SAprior sf* respectively. *Sheridan rndprior* and *GLASSO* have also been included in the analysis to assess the effects of adding scale free prior for *Sheridan* and *GLASSO*. We conducted both global and local performance analyses on all these methods, which are outlined next alongwith the resuls.

- **Global Performance** Fig. 5.1 shows the overall score averaged across all 25, size 100 networks. In terms of overall score, for all the prior incorporating methods, scale free prior works the best. With regards to *SAprior*, the order of performance for different priors in decreasing order is *SAprior sf*, *SAprior exp*, *SAprior rnd* and *SAprior Uniform*. As expected, uniform prior has significantly lower performance compared to the other priors. While the difference in performance between exponential and binomial priors is the least. In comparsion to *Sheridan*, *SAprior* has a slightly less increment

in performance between binomial and scale free priors. *GLASSO sfprior* exhibits the least improvement in performance. This might be due to the fact that the base model for comparison here is l_1 regularized *GLASSO* which is itself sparse.

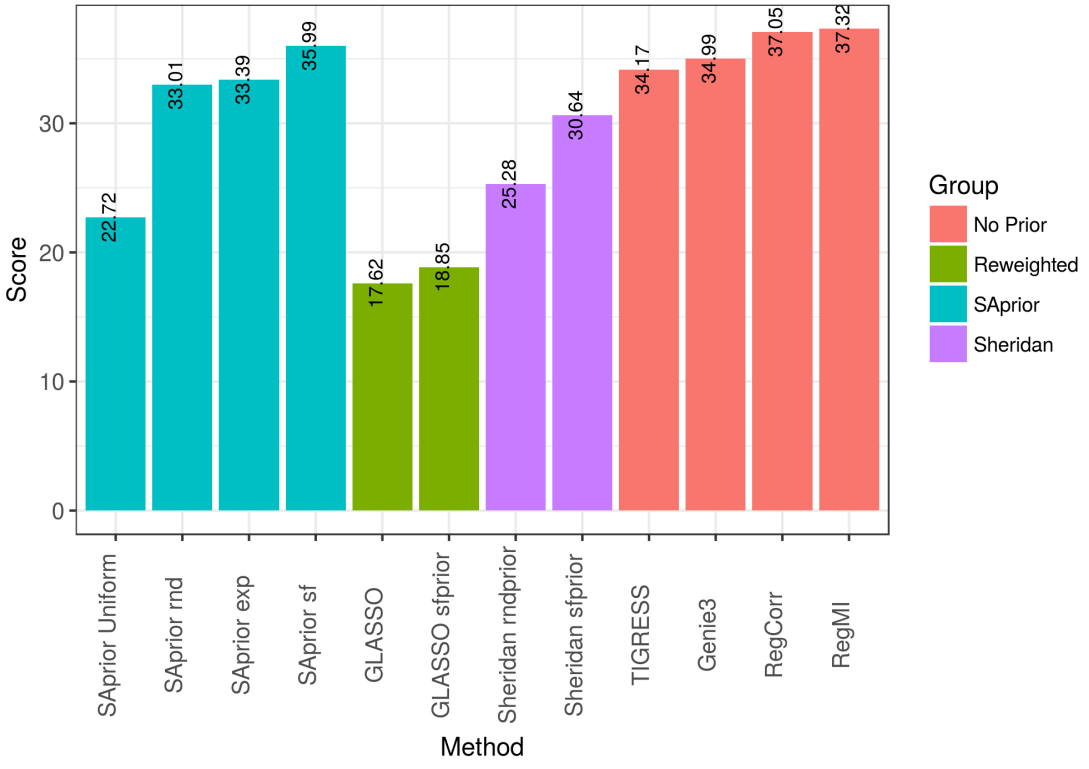


Fig. 5.1. Average Overall Score for studying scale-free priors.

The overall score averaged across the five network topologies, PIPO, DREAM4, DREAM3, EIPO and EIPO Modular for studying scale-free priors.

All of the above observations are more clearly visible in the average precision-recall curve for DREAM4 networks in Fig 5.2. *SAprior sf* lies above all the priors. The large difference between uniform and the other priors is quite evident. *GLASSO sfprior* dominates *GLASSO* up to a recall of 0.4, after which the curves seem indistinguishable. For *Sheridan*, scale free prior dominates for mid to high recall values whereas for low recall values binomial prior seems to slightly outperform.

Interestingly, *SAprior sf* is the best performer among all the prior-based methods. Whereas, it is overall the third best methods. It is evident from

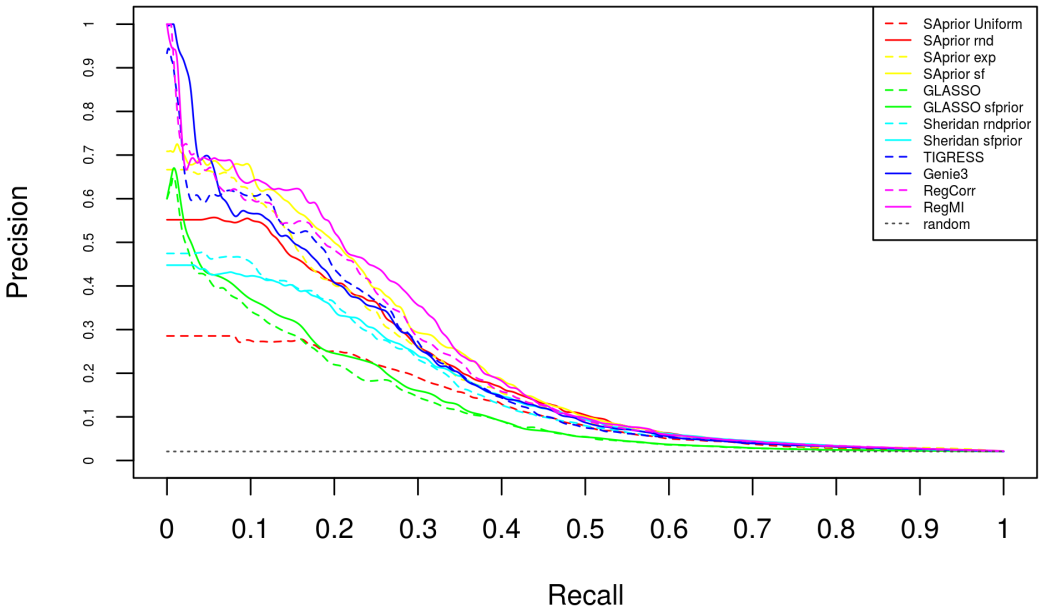


Fig. 5.2. Precision Recall curve for studying scale-free priors.

Precision recall curve averaged over the five DREAM4 networks for studying scale-free priors.

Figs. 5.1 and 5.2 that *SAprior sf* outperforms *Genie3* and *TIGRESS*, both of which are among the state of the art methods for gene network inference [11, 12]. Though, *SAprior sf* has lower overall score than *RegCorr* and*RegMI*. However, it can be seen from Fig. 5.2 that *SAprior sf* dominates *RegCorr* for most except for small recall values. For small recall values, *SAprior*, irrespective of the prior, has a flat curve not starting at the top left corner. This has been a consistent observation across methods that use edge frequency based aggregation. We see this behavior for *Sheridan* as well, and in some cases for *GLASSO* and *TIGRESS*. Additionally, we believe *SAprior* has potential for improvement since we have used a naive simulated annealing approach with unoptimized parameters. For instance, the number of iterations used was only 150, and the time step was also huge.

- **Effect on degree distribution** - To assess the effect of adding scale free prior on the estimation of indegree distribution, we examine the average

degree crossings in Fig. 5.3, where we see that *SAprior sf* better estimates the indegree distribution compared to all the other types of degree distributions. Whereas, for the case of *Sheridan*, scale-free prior dominates for all the indegrees except indegree 1, which is evident from Figs. 5.3.

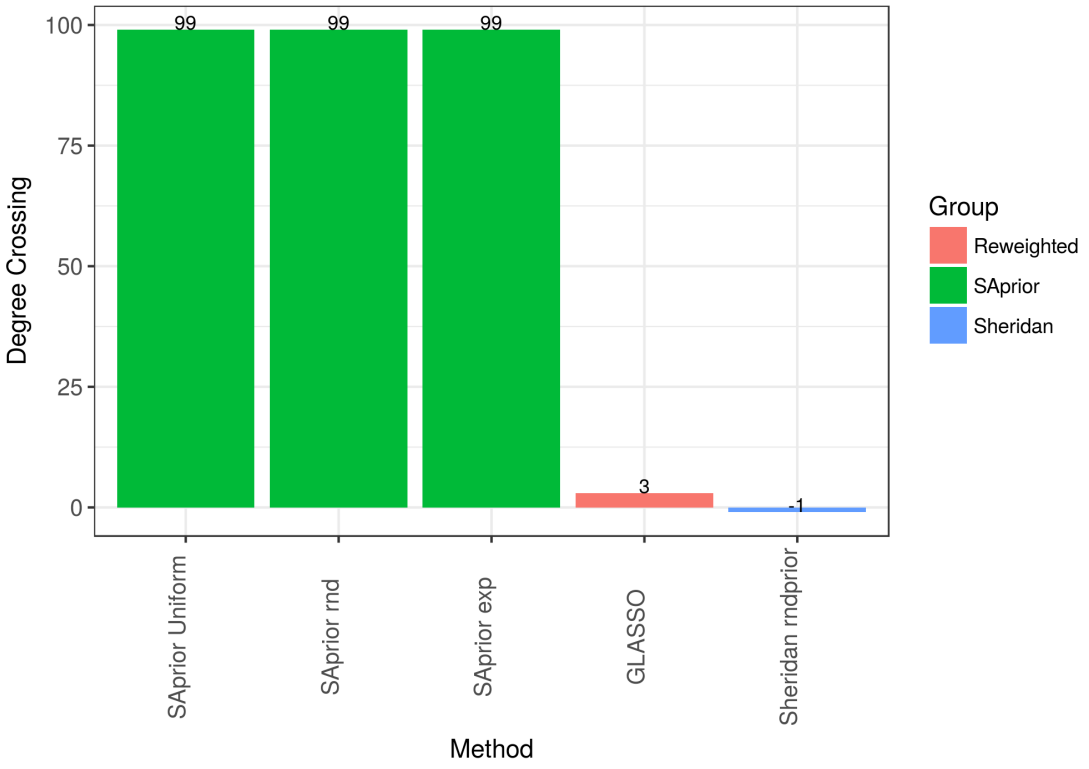


Fig. 5.3. Degree Crossing for Indegree for studying scale-free priors.

Degree crossing averaged across all the networks for studying scale-free priors. Within the context of the discussion in Section 2.3.2, with regards to degree crossing, a given method's version with scale-free prior is compared against the version without scale free prior or with some other prior. Thus, with scale-free prior the method would be M_1 and without scale-free prior or with some another distribution the method would then be M_2. So, the conclusions regarding degree crossing from 2.3.2 would hold accordingly. A value of $p-1$ for the degree crossing implies that M_1 dominates M_2. Any other positive value means that M_1 dominates M_2 up to a degree equal to the degree crossing value and after that M_2 dominates M_1. A negative value implies that M_2 dominates method M_1 up to a degree equal to the absolute value of the degree crossing and after that M_1 dominates. A value of $-(p-1)$ means that M_2 completely dominates M_1.

GLASSO sfprior has a rather limited effect however; it estimates only indegrees 1 to 3 better than *GLASSO*. However, if we observe the degree crossings only for PIPO networks, which have both indegree and outdegree as scale free, *GLASSO sfprior* now dominates *GLASSO*. For the DREAM3 and

DREAM4 networks which have been extracted from known topologies of biological networks and might have exponential indegree distribution, *GLASSO sfprior* has a limited capability of doing better than *GLASSO*. Interestingly, for DREAM3 networks, after indegree 2 exponential prior performs better than the scale free prior for *SAprior*. This network is expected to have exponential indegree and scale free outdegree. However, for EIPO and EIPO Modular, *SAprior sf* mostly dominates *SAprior exp*. One potential reason might be the deterministic nature of the degree distribution in Eq 5.11 and the truncation of the distribution at a maximum degree of N. N is one of the parameters being updated within the simulated annealing procedure, thus might be adapted depending upon the indegree distribution of the underlying network. Therefore, long tails might be clipped when the underlying network does not support them.

On outdegree distribution, *SAprior sf* and *Sheridan sfprior* seem to have a restricted dominance Fig. 5.4. This is yet another evidence for the dilution effect from indegree to outdegree. Both *SAprior sf* and *Sheridan sfprior* only dominate till outdegrees 6 or 7 with strategy 2. *GLASSO sfprior* has the opposite trend, where it dominates after outegree 5. However, if we look at the degree crossings for the PIPO networks, *GLASSO sfprior* dominates till outdegree 13, following a similar trend as *SAprior sf* and *Sheridan sfprior*. The experiments on indegree and outdegree distributions suggest that the prior introduced in [54], which has been implemented for *GLASSO sfprior*, is not robust to asymmetry in the indegree and outdegree distributions of the underlying network. It seems a reasonable obeservation, since this method gives an undirected network. Interestingly, *Sheridan sfprior*, which was also designed as an undirected method, seems more robust to the asymmetry in the indegree and outdegree distributions of the underlying network.

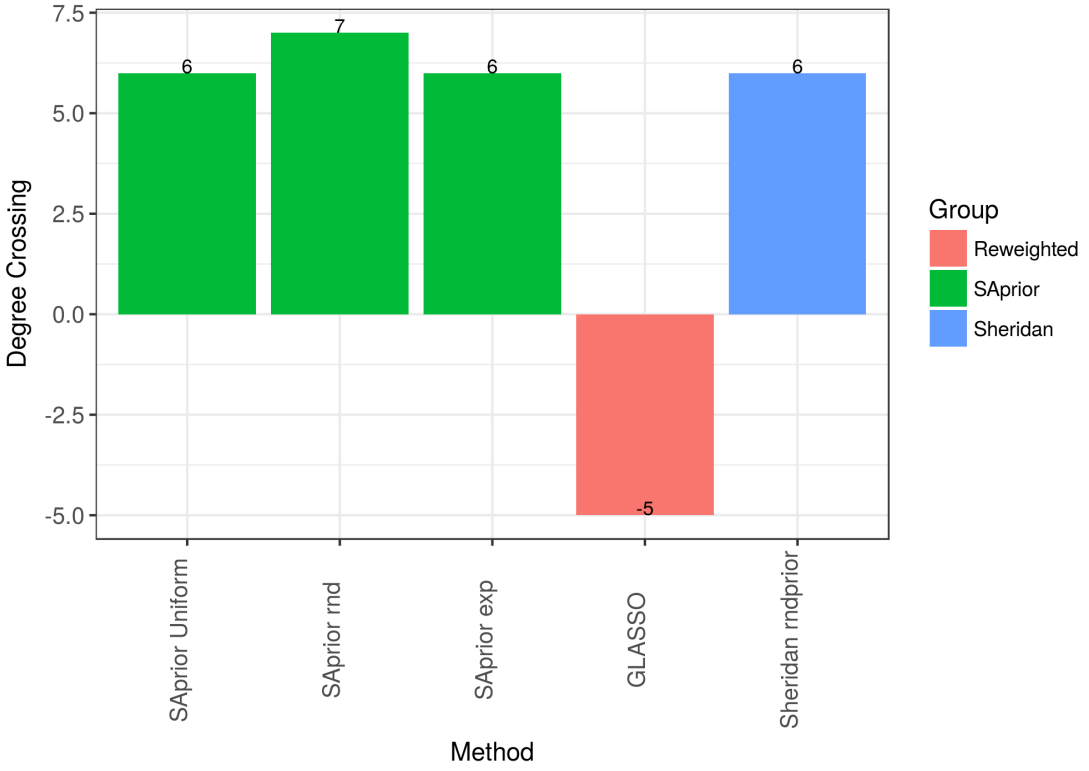


Fig. 5.4. Degree Crossing for outdegree for studying scale-free priors.

Degree crossing averaged across all the networks for studying scale-free priors. The inference strategy is the same as described in Section 2.3 and the caption for Fig. 5.3.

- **Effect on motif errors**

  Fig. 5.5 shows the change in bias for cascade and fan-out motif errors after adding scale free prior. The cascade error is consistently reduced for all the methods. Thus, incorporation of a scale free prior on the degree distribution leads to a reduction in cascade related false positives. *SAprior sf* has fewer fan-out erros compared to *SAprior uniform* and *SAprior rnd*. *GLASSO* exhibits lower fan-out errors as well with the addition of the scale-free prior. However, the fan-out error increases for *SAprior sf* against *SAprior exp* and also increases for *Sheridan sfprior*.

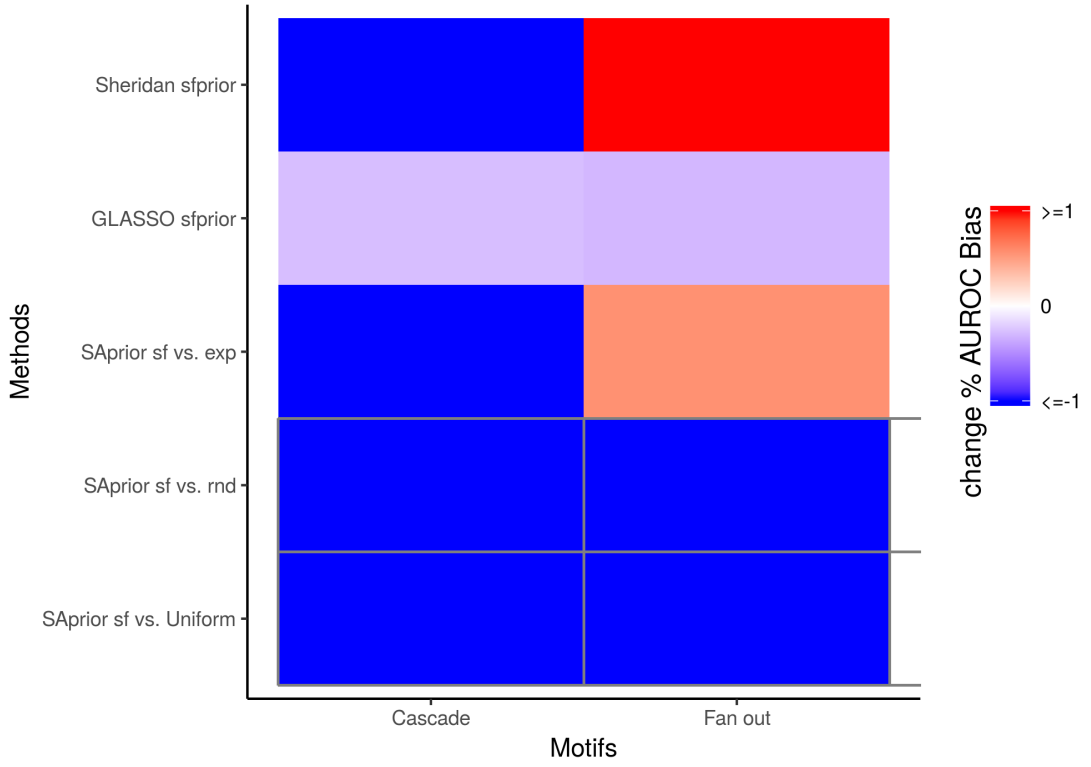


Fig. 5.5. Percentage Motif Bias for studying scale-free priors.

5.4.2 Experimental setup 2: SAprior upranking based combination

Within this experimental setup, we compare 27 network inference methods with and without *SAprior* upranking strategy. The 27 methods also include *Sheridan sfprior* and *Sheridan rndprior* with strategy 1 from Chapter 4. We have also used the version of *SAprior* augmented by the interaction term from Eq 5.20. Methods augmented by *SAprior* with the interaction term have *SA sf comb.* appended at the end of the name of the method.

- **Global Performance** Fig. 5.6 shows the overall score averaged across all 25, size 100 networks. The upranking strategy consistently leads to an increase in performance for all the methods. The magnitude of this increment decreases as the base performance of the method increases. This inverse relationship suggests that the extra information afforded by the upranking strategy using *SAprior* decreases for better performing methods since these

methods might have higher intersection with the predictions of *SAprior*. One of the interesting observations is that the performance increases even for *Sheridan sfprior* and *RegSheridan sfprior* methods, which have already incorporated prior information. For *Sheridan sfprior* this could be due to the directed nature of the predictions from *SAprior*. For *RegSheridan sfprior* which has an associated sense of directionality as well, one possible explanation could be that strategy 1 from 4 adds directionality without regard for the degree distributions. Perhaps, *SAprior* is introducing a degree distribution conscious sense of directionality.

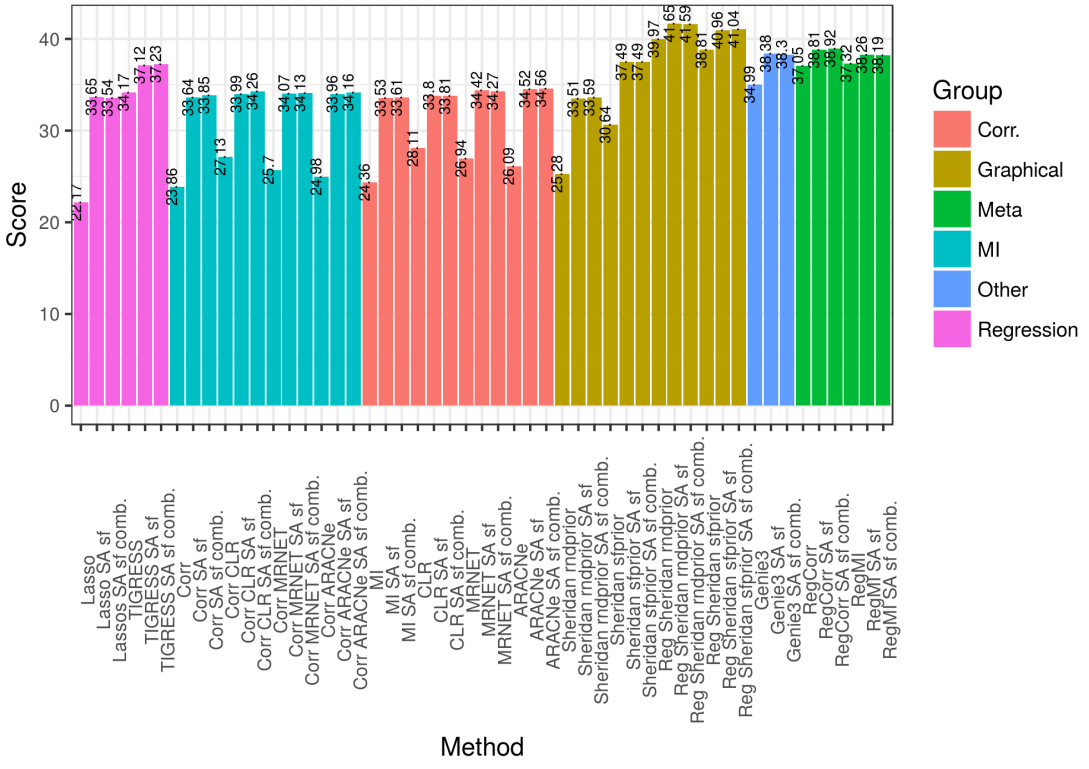
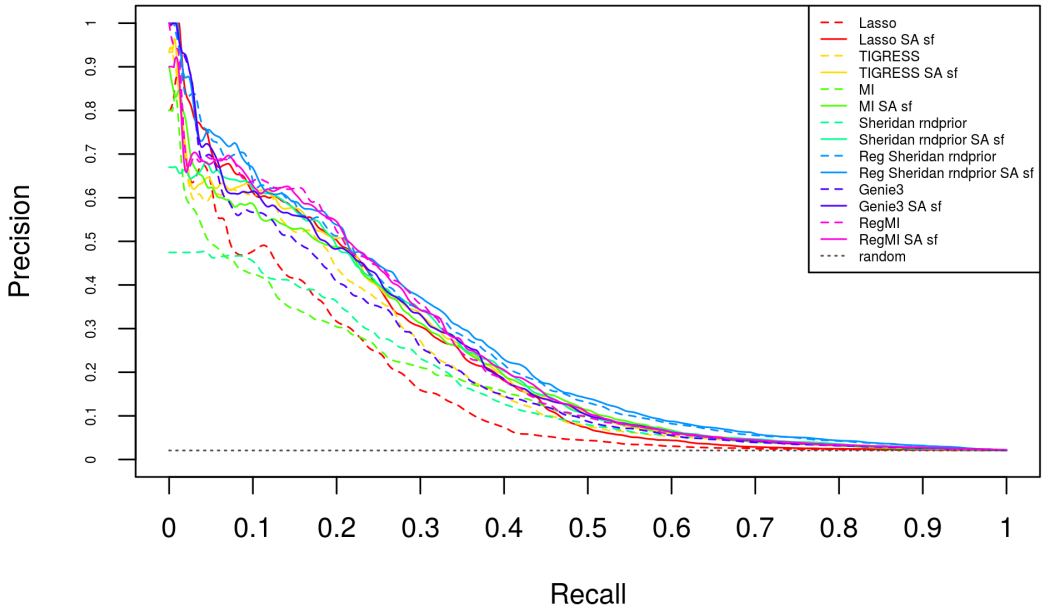


Fig. 5.6. Average Overall Score for studying *SAprior* upranking strategy.

The overall score averaged over all the networks for studying the effect of *SAprior* upranking strategy.

The effect of the upranking strategy is further visible in the average precision recall curve for the DREAM4 networks in Fig 5.7a. The huge improvements in performance is visible for most methods. Notably, for *Sheridan rndprior*, the curve at low recall values has shifted higher, thus leading to an increase in precision. As noted above, some methods show larger improvement in

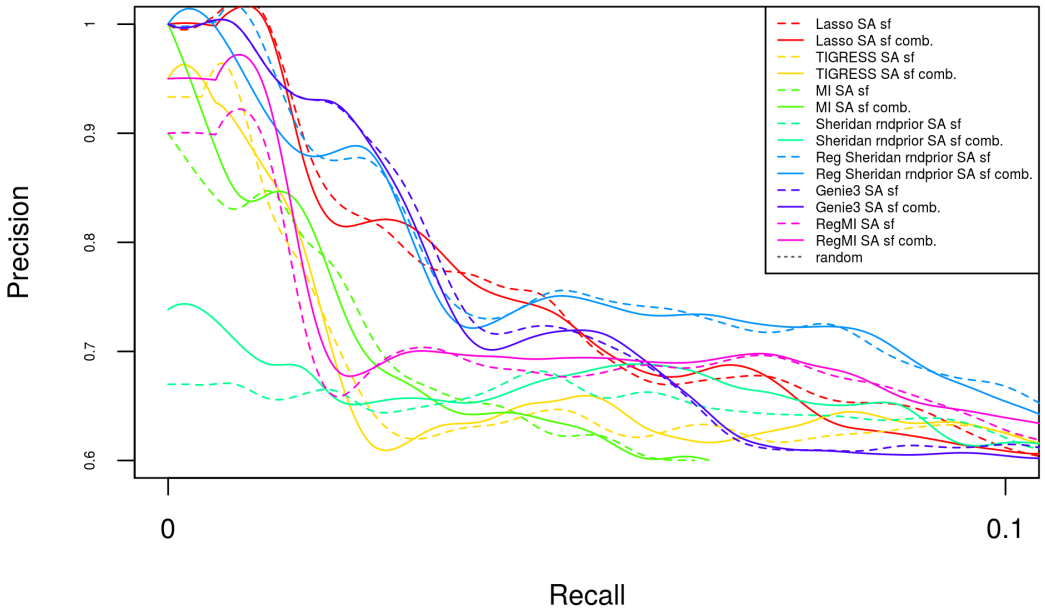
performance compared to the others. For instance, the curve for *RegSheridan sfprior* shifts by a lesser amount than for *MI*, *Lasso*, *TIGRESS*, *Sheridan rndprior* or *Genie3*.



(a)

Fig. 5.7. Precision Recall Curve studying *SAprior* upranking strategy. *(cont.)*

Addition of the interaction term introduced in Eq 5.20, leads to small improvements for some of the networks while for most networks the difference is visibly indistinguishable. Fig. 5.6 shows the score averaged across the DREAM3 networks, where we see that the interaction term has a small positive effect for *Corr* and *Sheridan* based methods. If we look closer at the zoomed-in average precision recall curve for the DREAM4 networks in Fig. 5.7b, the effect of the interaction term is clearly visible. For low recall values there is an increase in precision for methods such as *MI*, *RegMI* and *Sheridan rndprior*. However, the increment is not clearly observable in the final score. This could be attributed to the naive application of the simulated annealing strategy for *SAprior*. It might be the case that with a more



(b)

Fig. 5.7. Precision Recall Curve studying *SAprior* upranking strategy.

(a) Precision recall curve with SAprior upranking averaged over the five DREAM4 networks; (b) Precision recall curve with SAprior with the interaction term and upranking averaged over the five DREAM4 networks.

optimized simulated annealing procedure, the improvement in performance due to the interaction terms is larger and thus conclusively affects the final score.

- **Estimation of Degree Distribution** - For Majority of the methods, the degree crossing is 11 or higher, Fig 5.8. *Sheridan rndprior*, *Sheridan sfprior* and *RegSheridan rndprior* perform better over the entire range of indegrees with *SAprior*. The degree crossing for *CLR* is extremely high suggesting improved performance on the entire indegree range. Given that the performance of all network inference methods deteriorates exponentially with indegree, as seen in Chapter 3, we can conclude that the upranking strategy aides in extracting the indegree distribution for all the inference methods.

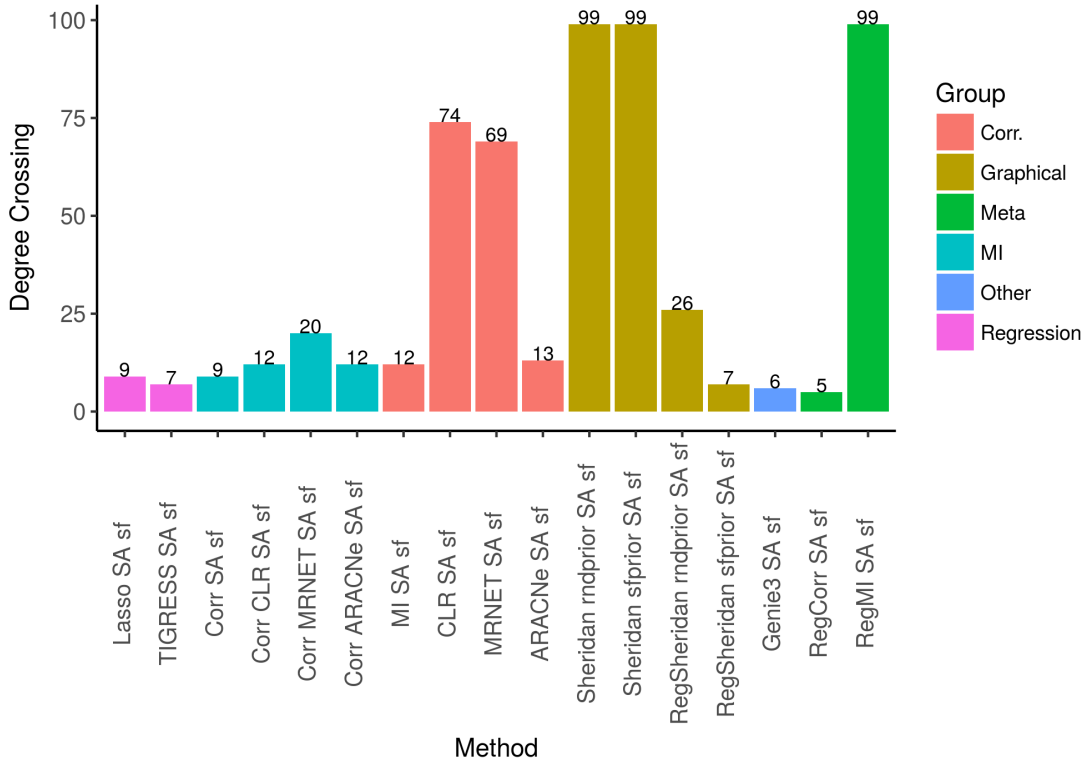


Fig. 5.8. Degree Crossing for Indegree for studying *SAprior* upranking strategy.

The degree crossing averaged across all the networks for comparing methods with and without SAprior upranking. Within the context of the discussion in Section 2.3.2, with regards to degree crossing, a given method's version with *SAprior* upranking is compared against the version without the same. Thus, with *SAprior* uprankingthe method would be M_1 and without it would then be M_2. So, the conclusions regarding degree crossing from 2.3.2 would hold accordingly. A value of $p-1$ for the degree crossing implies that M_1 dominates M_2. A positive value means that M_1 dominates M_2 up to a degree equal to the degree crossing value and after that M_2 dominates M_1. A negative value implies that M_2 dominates method M_1 up to a degree equal to the absolute value of the degree crossing and after that M_1 dominates. A value of $-(p-1)$ means that M_2 completely dominates M_1.

Outdegree crossing for all the methods is at outdegree 6 or 7, Fig. 5.9. The dilution effect has again limited the effect on outdegree compared to indegree. It has been shown in Chapter 2 that at higher outdegrees the dscore starts to saturate at the level of random guessing. Thus, Fig. 5.13 suggests that the upranking strategy increases the confidence for outdegrees which were being predicted better than random guessing for the base methods.

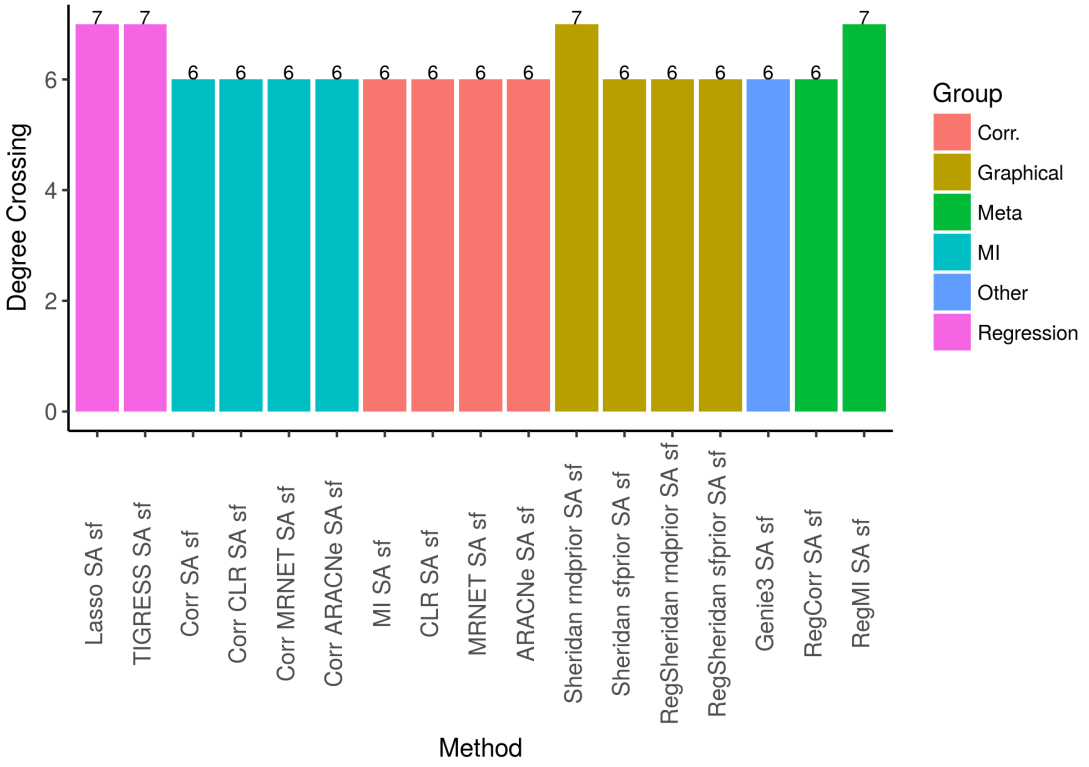


Fig. 5.9. Degree Crossing for Outdegree for studying *SAprior* upranking strategy.

The degree crossing averaged across all the networks for comparing methods with and without SAprior upranking. The inference strategy is the same as described in Section 2.3 and the caption for Fig. 5.8.

5.5 Future Work

We have shown that even a naive application of *SAprior* could be a good network inference method in isolation or in concert with other methods. This suggests one obvious line of potential future work- optimizing *SAprior* with respect to the involved parameters. In this regard, there are many parameters and settings that could be altered and played with. One of the things that could be done is to find ways to appropriately tune the values for the simulated annealing procedure. In the current form, we have used extremely crude estimates for the temperature parameters. Furthermore, the annealing process was stopped after a fixed number of iterations rather than checking the solution for some kind of stability. Such a stability criterion could be introduced to forego the need to select the number

of iterations manually. Another area for improvement could be the proposal distributions for updating k_{min}, γ_{in} and N. Distributions with better properties for our application could be found after trying combinations of different possibilites. Even the total cost function $TC(.)$ could be modified; potentially good cost functions available in the literature could be used or new customized cost functions could be designed. Lastly, in the current form, we had applied the combinatorial interaction term for indegrees up to 5. As we have seen that the inclusion of the interaction term has the potential for affording performance improvement in the recall region, inclusion of the term for larger indegree terms could be explored.

Besides the implementational concerns, the important issue is the methodological insights from the set of experiments that have been conducted. We have consistently seen that the inclusion of a prior on the degree distribution aides in the network inference task. However, the network inference task is still an unsolved problem. Degree distribution is just one of the many complex structural properties possessed by gene regulatory networks. Another similar property is the existence of network motif [17, 18] and these motifs have certain distributional properties. We already have included an approximate motif-based prior in a cursory manner. The combinatorial interaction introduced in [44] and used in Eq 5.20 tries to improve the methods' capability to infer Fan-in motifs. Even without optimization and limited implementation the term has proven to have potential.

This motivates the possibility of augmenting *SAprior* with other structural properties. We now propose a way of incorporating a cost-based constraint to ensure that cascade motif erros are reduced in the inferred network. We follow the strategy used in [39]. The method has been described in Chapter 3. The method uses a downranking strategy on a matrix of predictions obtained from knockout data. The predicted z-score matrix is appropriately thresholded to obtain a binary network. Potential cascade error edges are identified by collapsing the binary network to its condensation graph, and selecting the indirect edges for FFL type structures. All the edges in the binary network which correspond to the edges selected in the

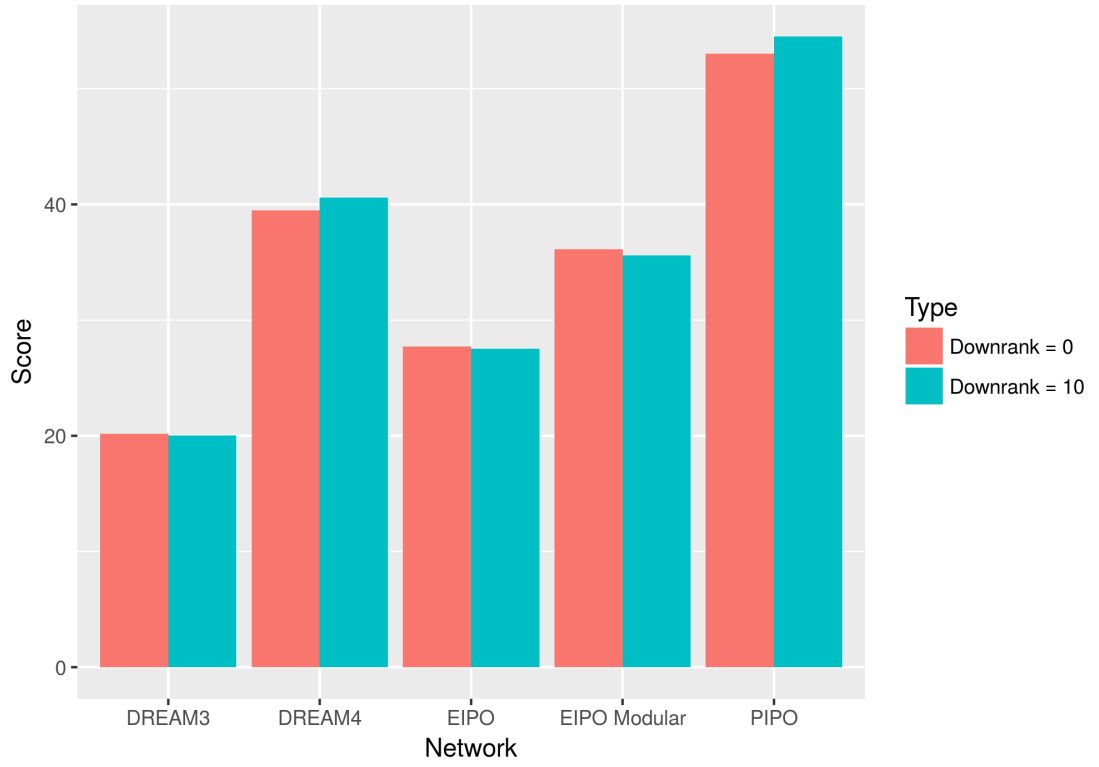
condensation graph, are downranked in thee final adjacency matrix. This particular step is informed by the observaation that gene regulatory networks are likely to have more cascade motifs than FFLs. We incorporate this scheme in *SAprior* by modifying the cost function as shown in Eq 5.5.

$$\hat{TC}(.) = TC(.) + \lambda_{casc.} \frac{Edg_{indirect}}{Edg_G} \tag{5.21}$$

where $\hat{TC}(.)$ is the final total cost function, Edg_G is the total number of edges in graph G obtained after solving the optimization problem in Eq 5.12, $Edg_{indirect}$ is the number of indirect edges in G identified using the strategy introduced in [39] and $\lambda_{casc.}$ is a constant. The additonal term in Eq 5.5 penalizes the number of indirect edges for every edge in G. $\lambda_{casc.}$ controls the strength of the penalization; larger the value of lambda, more heavily the indirect edges would be penalized and vice-versa. The new cost function is expected to drive *SAprior* towards solutions that have fewer number of indirect edges and thus lesser bias for cascade error.

Fig. 5.10 shows some preliminary results with this new cost function. We have arbitrarily used a value of 10 for $\lambda_{casc.}$ here. It is evident that for DREAM4 and PIPO networks, performance increases in terms of average score. We futher explore this improvement observed for DREAM4 networks in the average precision recall curve shown in Fig. 5.10b. *SAprior* with the new cost function is refered to as *SAprior Downrank* here. We can see that the curve for *SAprior Downrank* lies above the curve for *SAprior*. Moreover, at low recall values, *SAprior Downrank* starts at a higher precision value. These results suggest that the new cost function has potential for augmenting the network inference task. Further experimentation is required to strengthen these results.

Finally, we address the possibility of incorporating prior on both the indegree and outdegree distributions. The framework used in *Sheridan sfprior* for including the prior is easily generalizable to include the outdegree distribution as well [67]. Inspired by this observation and the performance of *SAprior* on the network inference task, we propose a bayesian framework for imposing a scale free prior on



(a)

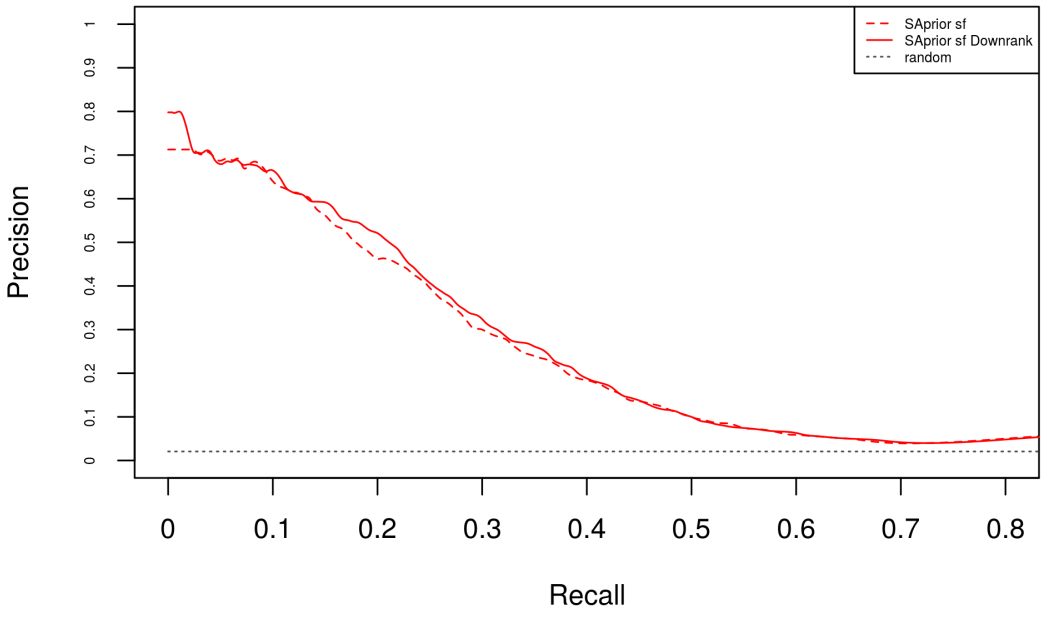
Fig. 5.10. Overall Score and PR curve for *SAprior Downrank. (cont.)*

both the indegree and outdegree distributions. Based on the discussion given in [67], the probability of any graph under a directed model could now be given as shown in Eq 5.22

$$P(G) = \prod_{ij \in E} \left(1 - (1 - s_i s_j)^{pK}\right) \prod_{ij \notin E} \left(1 - s_i s_j\right)^{pK} \tag{5.22}$$

where $s_i = \frac{{}^{OUT}\sigma_i^{-\mu_{out}}}{\sum_{k=1}^{p} {}^{OUT}\sigma_k^{-\mu_{out}}}$ and $s_j = \frac{{}^{IN}\sigma_i^{-\mu_{in}}}{\sum_{k=1}^{p} {}^{IN}\sigma_k^{-\mu_{in}}}$, μ_{out} is the Zipf exponent for the outdegree distribution and $\mu_{in}$is the Zipf exponent for the indegree distribution, all the other terms retain their meanings from Eq 5.10. The exponent of the scale free distribution can be estimated from the corresponding Zipf exponents as $\gamma_{in} = 1 + 1/\mu_{in}$ and $\gamma_{out} = 1 + 1/\mu_{out}$. Further, we introduce the likelihood function as shown in Eq 5.23

$$P(X \mid G) \propto \prod_{j=1}^{p} exp\left(- \frac{\sum_{u=1}^{n} ({}^{u}x_j - a_{0j} - \sum_{k=1}^{d_j} a_{kj} {}^{u}x_{\sigma_j(k)})^2}{2^{noise}\sigma_j} \right) \tag{5.23}$$



(b)

Fig. 5.10. Overall Score and PR curve for *SAprior Downrank*.

(a) The overall score for different topologies for comparing SAprior and SAprior downrank. (b) Precision recall curve averaged over five DREAM4 networks for comparing SAprior and SAprior downrank.

where d_j is the indegree of gene j under G and the rest of the variables and parameters retain their meanings from Eq 5.13. Eq 5.13 uses the PairWise Decoupling property for the network and thus decomposes the problem into p multivariate regression problems. Using the likelihood in Eq 5.23 and the prior in Eq 5.22, we can obtain an estimate for the gene network by sampling from the posterior using MCMC based sampling strategies. The likelihood can be obatined in closed form at every step of the MCMC chain by assuming approrpriate priors on the coefficients a_{kj} and ${}^{noise}\sigma_j$ or using the maximum likelihood estimate against these parameters as an approximation. Similar to *Sheridan sfprior* and *SAprior* an adjacency matrix can be estimated by aggregating the networks obtained at each point of the MCMC chain. Also, the obtained networks can also be combined with other methods using the upranking strategy.

Further, the PairWise Decoupling assumption can also be relaxed for Eq 5.23 and a full multivariate Gaussian noise model can be assumed. Another point of departure from the method of *Sheridan sfprior* would be that there is no need to ensure decomposability of the sampled graphs for this formulation.

5.6 Conclusion

We have given a brief overview of the state of some of the methods belonging to the class of techniques that leverage knowledge about the degree distribution for the network inference task. Adapting the framework proposed in one of the methods, we have devised a simulated annealing based method for incorporating degree distribution as a prior in the network inference task. Utilizing the metrics described in Chapter 2, we have conducted a comparative analysis of two degree distribution prior incorporating methods against our simulated annealing approach.

The results show that the scale-free distribution prior aides the inference task both globally and locally. *SAprior*, *Sheridan* and *GLASSO* exhibit better AUPR and AUROC values with the scale-free prior than the other priors. Inclusion of the scale-free prior also helps with the extraction of the degree distribution. *SAprior sf* dominates the other priors over the entire range of indegree. *Sheridan sfprior* and *GLASSO sfprior* also dominate, but over a limited range of indegree. A similar trend is seen with the outdegree distribution. Besides degree distribution, the scale-free prior also reduces the cascade motif error. Our proposed method, *SAprior*, outperforms the two other prior-based methods on 25 synthetically generated datasets. Further, on a breast cancer dataset, *SAprior sf* identifies a more connected network compared to *Sheridan*. These results suggest that *SAprior* needs to be extensively validated on experimental data.

We have also shown that the predictions from our simulated annealing based method can be seamlessly aggregated with the predictions from any other method. Such an aggregation has been shown to lead to performance enhancement for

all the methods considered in this work. Even methods which already have the scale-free prior included, exhibit an improvement in performance. These scale-free methods infer an undirected network; thus, the improvement in performance with *SAprior* could be accounted for by the directed nature of *SAprior*.

We have also proposed a framework for including local motif-inspired constraints to the simulated annealing method. One of the ways of contraining the solution tries to capture complex combinatorial regulation arising out of the interaction between transcription factors. Such a methodology would essentially have a strong bias for fan-out motifs. The other methodology tries to trade-off between cascade and FFL motifs. Preliminary results for both these methodologies have shown promise. Further research is required to establish the exact effect of these methodologies.

This chapter has thus shown that incoporating a prior on the degree distribution does aid the network inference task. However, the network inference task is still unsolved. Augmenting the scale-free prior with other structural properties might further help to improve the performance. Thus, efforts need to be directed in improving the already introduced frameworks for incorporating structural priors and better methodologies need to be developed.

Chapter 6

Conclusion

The aim of the present work has been two fold. To situate methods with integrated structural priors within the overall network inference framework and to conduct a comparative analysis of these methods on a common platform. For the former, we have indicated in Chapters 1 and 5 that subset of methods incorporating structural priors is quite small compared to the entire space of available methods. Most of the available degree distribution based methods belong to the class of graphical methods whether GGM or bayesian networks. Recently, some degree distribution incorporating bayesian network methods have emerged [50]. On the other hand, there have been many methods estimate GGMs with scale free degree distribution prior. However, these have not been benchmarked on a common platform with the most widely used network inference methods. Even the DREAM challenges have no participants that use GGMs. One potential reason is the limitation to undirected networks. Irrespective, these methods should be thoroughly benchmarked, as they might offer useful insights. In this work, we have benchmarked two GGM methods against some of the most widely used network inference methods. Further, we have demonstrated that these methods could be used in combination with other methods to offer high fidelity network predictions. In this work, we have also developed a simulated annealing based method for incorporating scale free degree distribution prior for network inference. Comparative analysis with GGM based methods has shown that this method is a better performer.

Another consistent theme in this study has been the importance of meta methods which try to leverage the strengths of multiple methods. This ideology has been repeatedly allured to in the DREAM challenges and leveraged by participants as well [59]. We have introduced a general and simple stratgey for combining a pairwise regression based startegy with any other method. The results for this strategy have shown promise for a certain class of methods. The results from the study of this strategy have reaffirmed the tenet that meta-methods or combination of multiple methods offer advantage in the inference task. The meta philosophy has even been used with the scale free prior method we have developed. The strategy for combining the scale free prior predictions with any other method is generalizable to any MCMC or simulated annealing based method. For instance, the upranking strategy of Chapter 5 can be easily used with the predictions generated by *Sheridan sfprior* as well.

One of the cumulative effects of the study of structural priors alongside a general analysis of network inference methods, has offered helpful insights. We have identified two motif based frameworks available in the literature that can be easily incorporated with the degree distribution prior. One method captures complex transcriptional control, which might induce a higher bias from Fan-out motifs. The other tries to distinguish cascade motifs from FFLs. We have used these methods with the simulated annealing based method we have developed. The results are promising and need further experimentation for strengthing the inferences made. This highlights meta analysis on the methodological insights; different methods can potentially offer rich insights for each other. This has been instruemental in the bayesian framework that we have proposed for incorporating both indegree and outdegree distributions as prior. This framework presents potential direction for future research.

Experimentation on true biological data is another line of future work. Synthetically generated data offers great variability in assessing the performance of gene network inference methods under different conditions. The insights gained are also accurate since the underlying network is precisely known. However, given the approximate nature of the expression data generating process, it is of paramount

importance to finally test any network inference method on biological datasets which have been consistently used by the research community. Within the context of this work, we have tested strategy 1 introduced in Chapter 3 on one of the real datasets from DREAM5 challenge. The scale free prior based methods, however, have only been tested and benchmarked on synthetic networks. Thus, any future work should quantify the performance of these methods on real biological data.

Finally, we end by coming to the question that motivated this work, "Do structural priors, specifically scale free degree distributions, help with network inference task?". We have answered this question within a limited context. Scale free priors indeed help with the network inference task; this answer is contingent upon the inferences made on the synthetic datasets being generalizable to larger, real biological networks. Further, there is a plethora of interwined questions that need to be explored. Scale free degree distribution is not the only property exhibited by gene regulatory networks. These networks have hierarchical modularity, exhibit motif structures, are highly robust and have other interesting properties. Another important question that needs to answered is whether a prior only on the degree distribution is sufficient to accomplish the inference task. It does not seem likely. It has been shown that the commonly used preferential attachment model for network growth might not be sufficent to capture the distribution of various motifs in the graph. We might need growth models that leverage motifs as primary units rather than individual genes. Within this context, the present work has made a small contribution by discussing the importance of including structural priors in the network inference task. Thus, future efforts should aim at working towards a more holistic approach by considering different structural properties in a collective manner for the network inference task.

Bibliography

1. Kesić S. Systems biology, emergence and antireductionism. Saudi Journal of Biological Sciences. 2016;23(5):584 – 591. doi:http://dx.doi.org/10.1016/j.sjbs.2015.06.015.

2. Barabasi AL, Oltvai ZN. Network biology: understanding the cell's functional organization. Nature reviews Genetics. 2004;5(2):101.

3. Davidson E, Levin M. Gene regulatory networks. Proceedings of the National Academy of Sciences of the United States of America. 2005;102(14):4935. doi:10.1073/pnas.0502024102.

4. Emmert-Streib F, Dehmer M, Haibe-Kains B. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. Frontiers in Cell and Developmental Biology. 2014;2:38. doi:10.3389/fcell.2014.00038.

5. MacNeil LT, Walhout AJ. Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. Genome research. 2011;21(5):645–657.

6. Stuart JM, Segal E, Koller D, Kim SK. A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. Science. 2003;302(5643):249–255. doi:10.1126/science.1087447.

7. de Matos Simoes R, Dehmer M, Emmert-Streib F. B-cell lymphoma gene regulatory networks: biological consistency among inference methods. Frontiers in genetics. 2013;4.

8. Madhamshettiwar PB, Maetschke SR, Davis MJ, Reverter A, Ragan MA. Gene regulatory network inference: evaluation and application to ovarian cancer allows the prioritization of drug targets. Genome medicine. 2012;4(5):41.

9. Emmert-Streib F, de Matos Simoes R, Mullan P, Haibe-Kains B, Dehmer M. The gene regulatory network for breast cancer: integrated regulatory landscape of cancer hallmarks. Frontiers in genetics. 2014;5:15.

10. Stolovitzky G, Prill RJ, Califano A. Lessons from the DREAM2 Challenges. Annals of the New York Academy of Sciences. 2009;1158(1):159–195.

11. Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G. Revealing strengths and weaknesses of methods for gene network inference. Proceedings of the National Academy of Sciences. 2010;107(14):6286–6291. doi:10.1073/pnas.0913357107.

12. Marbach D, Costello JC, Kuffner R, Vega NM, Prill RJ, Camacho DM, et al. Wisdom of crowds for robust gene network inference. Nat Meth. 2012;9(8):796–804. doi:10.1038/nmeth.2016.

13. Narendra V, Lytkin NI, Aliferis CF, Statnikov A. A comprehensive assessment of methods for de-novo reverse-engineering of genome-scale regulatory networks. Genomics. 2011;97(1):7–18.

14. Ma S, Kemmeren P, Gresham D, Statnikov A. De-novo learning of genome-scale regulatory networks in S. cerevisiae. Plos one. 2014;9(9):e106479.

15. Guelzim N, Bottani S, Bourgine P, Képès F. Topological and causal structure of the yeast transcriptional regulatory network. Nature genetics. 2002;31(1):60.

16. Erwin DH, Davidson EH. The evolution of hierarchical gene regulatory networks. Nature reviews Genetics. 2009;10(2):141.

17. Shen-Orr SS, Milo R, Mangan S, Alon U. Network motifs in the transcriptional regulation network of Escherichia coli. Nature genetics. 2002;31(1):64.

18. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, et al. Transcriptional regulatory networks in Saccharomyces cerevisiae. science. 2002;298(5594):799–804.

19. De Smet R, Marchal K. Advantages and limitations of current network inference methods. Nat Rev Micro. 2010;8(10):717–729. doi:10.1038/nrmicro2419.

20. Schaffter T, Marbach D, Floreano D. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics. 2011;27(16):2263–2270.

21. Gama-Castro S, Salgado H, Peralta-Gil M, Santos-Zavaleta A, Muniz-Rascado L, Solano-Lira H, et al. RegulonDB version 7.0: transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units (Gensor Units). Nucleic acids research. 2010;39(suppl_1):D98–D105.

22. Cantone I, Marucci L, Iorio F, Ricci MA, Belcastro V, Bansal M, et al. A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. Cell. 2009;137(1):172–181.

23. Chen N, Olvera-Cravioto M, et al. Directed random graphs with given degree distributions. Stochastic Systems. 2013;3(1):147–186.

24. Pinna A, Soranzo N, Hoeschele I, de la Fuente A. Simulating systems genetics data with SysGenSIM. Bioinformatics. 2011;27(17):2459–2462.

25. Jansen RC. Opinion: studying complex biological systems using multifactorial perturbation. Nature reviews Genetics. 2003;4(2):145.

26. Barrett T, Troup DB, Wilhite SE, Ledoux P, Evangelista C, Kim IF, et al. NCBI GEO: archive for functional genomics data sets—10 years on. Nucleic acids research. 2010;39(suppl_1):D1005–D1010.

27. Janzing D, Mooij J, Zhang K, Lemeire J, Zscheischler J, Daniušis P, et al. Information-geometric approach to inferring causal directions. Artificial Intelligence. 2012;182:1–31.

28. Alon U. Network motifs: theory and experimental approaches. Nature Reviews Genetics. 2007;8(6):450–461.

29. Zhang B, Horvath S. A general framework for weighted gene co-expression network analysis. Statistical applications in genetics and molecular biology. 2005;4(1).

30. Feizi S, Marbach D, Médard M, Kellis M. Network deconvolution as a general method to distinguish direct dependencies in networks. Nature biotechnology. 2013;31(8):726.

31. Stuart JM, Segal E, Koller D, Kim SK. A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. Science. 2003;302(5643):249–255. doi:10.1126/science.1087447.

32. Butte AJ, Tamayo P, Slonim D, Golub TR, Kohane IS. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. Proceedings of the National Academy of Sciences. 2000;97(22):12182–12186.

33. Butte AJ, Kohane IS. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In: Pac Symp Biocomput. vol. 5; 2000. p. 26.

34. Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G, et al. Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. PLOS Biology. 2007;5(1):1–13. doi:10.1371/journal.pbio.0050008.

35. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, et al. ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. BMC Bioinformatics. 2006;7(1):S7. doi:10.1186/1471-2105-7-S1-S7.

36. Meyer PE, Kontos K, Lafitte F, Bontempi G. Information-theoretic inference of large transcriptional regulatory networks. EURASIP journal on bioinformatics and systems biology. 2007;2007(1):1–9.

37. Wolfe CJ, Kohane IS, Butte AJ. Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. BMC Bioinformatics. 2005;6(1):227. doi:10.1186/1471-2105-6-227.

38. Ruyssinck J, Demeester P, Dhaene T, Saeys Y. Netter: re-ranking gene network inference predictions using structural network properties. BMC bioinformatics. 2016;17(1):76.

39. Pinna A, Soranzo N, De La Fuente A. From knockouts to networks: establishing direct cause-effect relationships through graph analysis. PloS one. 2010;5(10):e12912.

40. Xiong J, Zhou T. Gene regulatory network inference from multifactorial perturbation data using both regression and correlation analyses. PloS one. 2012;7(9):e43819.

41. Irrthum A, Wehenkel L, Geurts P, et al. Inferring regulatory networks from expression data using tree-based methods. PloS one. 2010;5(9):e12776.

42. Küffner R, Petri T, Tavakkolkhah P, Windhager L, Zimmer R. Inferring gene regulatory networks by ANOVA. Bioinformatics. 2012;28(10):1376–1382.

43. Haury AC, Mordelet F, Vera-Licona P, Vert JP. TIGRESS: trustful inference of gene regulation using stability selection. BMC systems biology. 2012;6(1):145.

44. Bonneau R, Reiss DJ, Shannon P, Facciotti M, Hood L, Baliga NS, et al. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. Genome biology. 2006;7(5):R36.

45. Yu J, Smith VA, Wang PP, Hartemink AJ, Jarvis ED. Advances to Bayesian network inference for generating causal networks from observational biological data. Bioinformatics. 2004;20(18):3594–3603.

46. Perrin BE, Ralaivola L, Mazurie A, Bottani S, Mallet J, d'Alche Buc F. Gene networks inference using dynamic Bayesian networks. Bioinformatics. 2003;19(suppl_2):ii138–ii148.

47. Liu F, Zhang SW, Guo WF, Wei ZG, Chen L. Inference of gene regulatory network based on local bayesian networks. PLoS computational biology. 2016;12(8):e1005024.

48. Fan Y, Wang X, Peng Q. Inference of Gene Regulatory Networks Using Bayesian Nonparametric Regression and Topology Information. Computational and mathematical methods in medicine. 2017;2017.

49. Vinh NX, Chetty M, Coppel R, Wangikar PP. GlobalMIT: learning globally optimal dynamic bayesian network with the mutual information test criterion. Bioinformatics. 2011;27(19):2765–2766.

50. Nair A, Chetty M, Wangikar PP. Improving gene regulatory network inference using network topology information. Molecular BioSystems. 2015;11(9):2449–2463.

51. Friedman J, Hastie T, Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. Biostatistics. 2008;9(3):432–441.

52. Meinshausen N, Bühlmann P. High-dimensional graphs and variable selection with the lasso. The annals of statistics. 2006; p. 1436–1462.

53. Peng J, Wang P, Zhou N, Zhu J. Partial correlation estimation by joint sparse regression models. Journal of the American Statistical Association. 2009;104(486):735–746.

54. Liu Q, Ihler AT. Learning Scale Free Networks by Reweighted L1 regularization. In: AISTATS; 2011. p. 40–48.

55. Defazio A, Caetano TS. A convex formulation for learning scale-free networks via submodular relaxation. In: Advances in Neural Information Processing Systems; 2012. p. 1250–1258.

56. Tang Q, Sun S, Xu J. Learning scale-free networks by dynamic node specific degree prior. In: International Conference on Machine Learning; 2015. p. 2247–2255.

57. Sheridan P, Kamimura T, Shimodaira H. A scale-free structure prior for graphical models with applications in functional genomics. PLoS One. 2010;5(11):e13580.

58. Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Methodological). 1996; p. 267–288.

59. Greenfield A, Madar A, Ostrer H, Bonneau R. DREAM4: Combining genetic and dynamic information to identify biological networks and dynamical models. PloS one. 2010;5(10):e13397.

60. Geier F, Timmer J, Fleck C. Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge. BMC systems biology. 2007;1(1):11.

61. Omranian N, Eloundou-Mbebi JM, Mueller-Roeber B, Nikoloski Z. Gene regulatory network inference using fused LASSO on multiple data sets. Scientific reports. 2016;6.

62. Mordelet F, Vert JP. SIRENE: supervised inference of regulatory networks. Bioinformatics. 2008;24(16):i76–i82.

63. Ud-Dean SM, Gunawan R. Ensemble inference and inferability of gene regulatory networks. PLoS One. 2014;9(8):e103812.

64. Zhu Y, Liu Z, Sun S. Learning Nonparametric Forest Graphical Models with Prior Information. arXiv preprint arXiv:151103796. 2015;.

65. Maruyama O, Shikita S. A scale-free structure prior for Bayesian inference of Gaussian graphical models. In: Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on. IEEE; 2014. p. 131–138.

66. Clauset A, Shalizi CR, Newman MEJ. Power-Law Distributions in Empirical Data. SIAM Review. 2009;51(4):661–703. doi:10.1137/070710111.

67. Goh KI, Kahng B, Kim D. Universal behavior of load distribution in scale-free networks. Physical Review Letters. 2001;87(27):278701.

68. Meyer PE, Lafitte F, Bontempi G. minet: AR/Bioconductor package for inferring large transcriptional networks using mutual information. BMC bioinformatics. 2008;9(1):461.

69. Friedman J, Hastie T, Tibshirani R. Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software. 2010;33(1):1–22.

Method Abbreviations

Method	**Abbreviation**
Correlation	*Corr*
Correlation with [34]	*Corr CLR*
Correlation with [36]	*Corr MRNET*
Correlation with [35]	*Corr ARACNe*
Mutual Information	*MI*
Mutual Information with [34]	*CLR*
Mutual Information with [36]	*MRNET*
Mutual Information with [35]	*ARACNe*
[51, 52]	*GLASSO*
[51, 52] with [54]	*GLASSO sfprior*
[57] with scale-free prior	*Sheridan sfprior*
[57] with binomial prior	*Sheridan rndprior*
[40]	*RegCorr*
[40] with Mutual Information	*RegMI*
[58]	*Lasso*
[43]	*TIGRESS*
[41]	*Genie3*

TABLE 1: **Method Abbreviations.**

Method	**Abbreviation**
Correlation	*RegCorr*
Correlation with [34]	*RegCorr CLR*
Correlation with [36]	*RegCorr MRNET*
Correlation with [35]	*RegCorr ARACNe*
Mutual Information	*RegMI*
Mutual Information with [34]	*RegCLR*
Mutual Information with [36]	*RegMRNET*
Mutual Information with [35]	*RegARACNe*
[51, 52]	*RegGLASSO*
[51, 52] with [54]	*RegGLASSO sfprior*
[57] with scale-free prior	*RegSheridan sfprior*
[57] with binomial prior	*RegSheridan rndprior*
[58]	*RegLasso*
[43]	*RegTIGRESS*
[41]	*RegGenie3*

TABLE 2: **Method Abbreviations with stratgey 1 in Chapter 4.**

Method	**Abbreviation**
[51, 52]	*GLASSO CLR*
[51, 52] with [54]	*GLASSO sfprior CLR*
[57] with scale-free prior	*Sheridan sfprior CLR*
[57] with binomial prior	*Sheridan rndprior CLR*
[58]	*Lasso CLR*
[43]	*TIGRESS CLR*
[41]	*Genie3 CLR*

TABLE 3: **Method Abbreviations with stratgey 2 in Chapter 4.**

Method	**Abbreviation**
SAprior with Uniform prior	*SAprior Uniform*
SAprior with binomial prior	*SAprior rnd*
SAprior with exponential prior	*SAprior exp*
SAprior with scale-free prior	*SAprior sf*

TABLE 4: **Method Abbreviations with *SAprior* in Chapter 5.**

Software Implementation of Methods

Method	Software	Author
Correlation	R	-
Mutual Information	R – Infotheo package	-
CLR	R – minet package	[68]
ARACNe	R – minet package	[68]
MRNET	R – minet package	[68]
Lasso	R – glmnet package	[69]
TIGRESS	R	Adapted from [43]
GLASSO	R – glasso package	[51]
GLASSO sfprior	R	Implemented as part of this work
Sheridan	C++	[57]
Genie3	R	[41]
RegCorr	R	Implemented as part of this work
RegMI	R	Implemented as part of this work
SAprior	C++	Implemented as part of this work

TABLE 5: **Details for softwares and packages used for implementing the different methods.**

Brief Bio-data

Tarun Mahajan obtained his B.Tech from the Department of Electronics and Communication Engineering, Indian Institute of Technology Roorkee in 2013. Before joining the masters program at IIT Delhi, he worked with the technology consulting firm iRunway India Pvt. Ltd. in Bangalore, India, where he primarily dealt with telecommunication and software patents. At IIT Delhi, besides his academic duties, he has represented the institute at the international synthetic biology competition iGEM (International Genetically Engineered Machine) as a member of team iGEM IIT Delhi. For iGEM 2016, he co-designed a reconfigurable oscillator to be implemented in the bacterium *E.coli*. The oscillator has a tunable frequency of operation which can be configured via light of a specific frequency leveraging optogenetics. The team won a silver at iGEM 2016. Tarun is the first author on the corresponding paper to be published in the iGEM 2016 collection of the journal PLOS ONE.

Tarun's research interests span synthetic biology, with specific focus on design of synthetic circuits, role of noise in synthetic circuit design and reconfigurability; systems biology, with focus on gene network inference and the application of gene network inference methodologies towards gaining understanding of biological processes.