Freie Universität Berlin

Association measures and prior information in the reconstruction of gene networks

Mahsa Ghanbari

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

Betreuer: Prof. Dr. Martin Vingron

Berlin
April 2015

Erstgutachter: Prof. Dr. Martin Vingron
Zweitgutachter: Prof. Dr. Hanspeter Herzel
Tag der Disputation: 20 July 2015

To the memory of
my beloved mother

Acknowledgements

First and foremost, I would like to thank my supervisor Martin Vingron for his scientific support and guidance during my PhD. I am grateful to him for giving me the opportunity to pursue my PhD in his group with exceptional scientists and friendly environment.

My special thanks go to Julia Lasserre, who contributed to the design of the analysis described in this thesis. I appreciated her help at work as well as her support and friendship outside of the work.

I would like to thank all my office mates Brian Caffrey, Matthew Huska, Wolfgang Kopp, Alessandro Mammana, Robert Schöpflin and Edgar Steiger for cheerful atmosphere and wonderful time we had together. Further, I wish to thank all the current and former members of the Computational Molecular Biology group at MPIMG for the friendly and motivating environment.

I would like to thank Juliane Perner, Brian Caffery and Matthew Huska for proofreading the thesis and giving their valuable comments. I wish to thank Kirsten Kelleher for her kind help through settling in Berlin.

I also want to thank the Computational Systems Biology research training group for financial support of my PhD study as well as its members for fruitful discussions in different meetings.

Finally, my warmest gratitude goes to my father Mohammadhossein, my sister Mozhdeh and my brother Mohammad for their unending overseas love and support.

Contents

Chapter 1

Introduction

The regulation of gene expression is the key process in the cell to adapt the cell in response to internal and external stimuli, allowing cells to have their own cell-type specific expression patterns (in multicellular organisms). Since genes encode for regulatory elements such as transcription factors (TFs) which in turn regulate other genes, there are complex interactions between genes through their products forming networks called gene regulatory networks(GRNs). In other words, GRNs describe the interactions between genes indirectly via their products. GRNs are usually represented as a graph, where the nodes of the graph represent genes and the edges represent the interactions between them.

Knowledge of GRNs can deepen our understanding of various diseases such as cancer where the development of the disease is not guided just by one gene but by a network of interacting genes. Furthermore, these networks help scientists in drug design and to find the targets of the drug.

The advent of high-throughput technologies such as DNA microarrays and RNA-Seq with the ability to measure the mRNA abundances of thousands of genes within a single experiment offers the opportunity to study interactions among thousands of genes in a living system. Under the assumption that mRNA abundance measurements of genes are predictive for their activity level, many researchers tried to find meaningful informative patterns in the gene expression data. For example genes showing similar patterns of expression across experimental conditions are more likely to be involved in common biological processes. Therefore by finding the association between genes one can gain further insight into the underlying interactions of genes and especially gene functions. Though many genes are coexpressed there are not necessarily direct interactions among these genes, as for example genes separated by one or more intermediaries (indirect relationships) may be highly coexpressed. It is therefore important to use algorithms

capable of inferring direct interactions among genes for the purpose of gene network inference.

GRN reconstruction from expression data is a challenging problem, not only because it suffers from high dimensionality and low sample size, as the number of genes is generally much larger than the biological samples, but also because biological measurements can be extremely noisy. A variety of computational methods have been suggested to address this problem including regression methods [1], graphical Gaussian models [2] and Bayesian Networks [3]. Despite considerable progress in the field, current methods still give relatively poor results due to the noisy and sparse nature of the data or are limited to small datasets. Hence, the problem is still an active field and much remains to be done to improve the reliability of the solutions without increasing the computational cost. The readers are referred to [4, 5] for comprehensive reviews on the field.

Another issue concerning GRN inference is to find the (direct) nonlinear interactions between genes. This is an important task since regulatory interactions are not necessarily linear [6] which is the assumption in many methods for GRN reconstruction. While (conditional) mutual information can detect (direct) nonlinear interactions, it is not trivial to estimate it from finite continuous data. Hence finding a method capable of capturing direct nonlinear associations is an essential task for inferring accurate GRNs.

In this thesis, we propose methods to tackle these problems concerning GRN reconstruction. In Chapter 2, we provide an overview of some association measures as well as some statistical models which use the association measures to find the (in)dependency structure among genes and to reconstruct GRNs. The methods will be the basis for models in the following chapters.

In Chapter 3, we investigate the performance of the methods introduced in Chapter 2 on different data sets and in different aspects to learn about their advantages as well as disadvantages.

In Chapter 4, we describe an algorithm called PriorPC which tackles the difficulties posed by gene expression data sets via the integration of prior knowledge. PriorPC is based on the PC algorithm, a popular methods for Bayesian network reconstruction which is known to depend strongly on the order in which nodes are presented. PriorPC exploits this flaw to include prior knowledge.

In Chapter 5, we introduce a novel approach to compute the empirical partial distance correlation, a generalization of the distance correlation with the ability to account for the effect of other variables. As a result, it can detect direct nonlinear interactions. The distance correlation is a recently proposed association measure capable of finding nonlinear relationships with an elegant way to estimate it from data. However, in the

context of multivariate analysis it is important to account for the influence of other variables in finding direct interactions.

In chapter 6, we provide a brief summary of the thesis.

1.1 The regulation of gene expression

Gene expression is the process by which the information coded in genes is used to produce functional gene products like RNAs and proteins. It starts with transcription, where the information in DNA is used to create RNA. While some of these RNA molecules can be the end product (non-coding RNAs), others (messenger RNAs) will be used as a template to produce proteins in the process of translation.

The expression level of genes or the abundance of RNAs and proteins in a cell is regulated at many stages. This allows cells to have their own cell-type specific expression patterns (in multicellular organisms) and to respond to environmental changes. Gene regulation can occur at all stages of gene expression with some differences between prokaryotic and eukaryotic cells. For example, in prokaryotic cells, the DNA floats in the cell cytoplasm and therefore transcription and translation occur almost simultaneously. In eukaryotic cells, the DNA is inside a nuclear membrane, where it is transcribed into RNA. The messenger RNA (mRNA) subsequently has to be transported to cytoplasm where it is translated into protein. As a result, transcription and translation processes are physically separated leading to a more complicated process. Furthermore, eukaryotic DNA is densely packed into chromatin and in the default state the tightly coiled structure of chromatin limits the access of the regulatory elements to the DNA. Therefore, in a process called chromatin remodeling, the cell's chromatin is made accessible in order for gene transcription to occur. The altering of local chromatin structure is performed by epigenetic modifications which lead to either the accessibility of regions of chromatin for binding of transcriptional activators, or condensing chromatin into a transcriptionally inactive state.

In both eukaryotes and prokaryotes, transcriptional regulation is considered as one of the most important mechanism of gene regulation. The main participants of this form of regulation are transcription factors (TFs). Transcription factors are proteins that activate or repress the transcription of target genes by binding to a DNA region (regulatory sequences). Regulation in eukaryotes is more complicated than prokaryotic and it requires the coordinated interactions of multiple proteins in a complex combinatorial mechanism.

In prokaryotes, binding of a TF to a promoter sequence determines whether or not RNA polymerase binds to promoter and initiates transcription of a particular gene. Promoters are regulatory sequences that are usually locate at the $5'$ of the transcriptional start site. Repressor TFs also bind to an operator, a region that is generally located downstream from and near the promoter, and inhibit the transcription.

In eukaryotes, the transcription process is a combinatorial mechanism involving both cis-acting elements, and trans-acting elements such as TFs and enhancers. Promoters are proximal DNA sequences that bind RNA polymerase for regulating gene expression. Enhancers are short regions of DNA that interact with regulatory proteins and TFs to promote expression of a distal or a proximal gene. In fact, activators bind to enhancer and this activator-enhancer complex can bend the DNA molecule so that additional transcription factors have better access to their bonding sites. In this way they recruit RNA polymerase II which then begins the process of transcription.

Post-transcriptional regulations can also control how much mRNA is translated into proteins. The translation of mRNA to proteins is also tightly controlled by some mechanisms and even after the translation there are some regulations such as the modification of proteins. However, gene expression data measures the mRNA levels and therefore these aspects of gene regulation are not reflected in the data.

1.2 Available biological data

The postgenomic era provides scientists with a huge amount of biological data sets which have proved to be valuable sources of information to discover the underlying interactions among genes. Although, the gene expression data is the main source of data used to infer GRNs, other data set like ChIP-x data provide valuable information that can help obtain a more accurately inferred network.

1.2.1 Gene expression data

Gene expression levels are the activity level of a gene measured as the amount of its resulting functional product. Since it is hard to measure the activity level of genes, the abundances of mRNA are often used as a proxy for gene activity. The two most widely used technologies to measure gene expression level are DNA microarray and RNA-seq experiments which have the ability to measure gene expression values of a large number of genes simultaneously. The raw data from both methods should go through some prepossessing analysis to provide the gene expression data. The gene expression data

are typically represented by an $p \times n$ matrix E, where p and n are the number of genes and the number of samples respectively. Each entry e_{ij} represents the expression level of gene i (the molecular abundance of the mRNA transcribed from gene i) in condition j. Expression data typically contain a large number of genes (on the order of hundreds or thousands) but only contain comparatively few samples n (on the order of tens or hundreds). Most of the standard learning methods have difficulty dealing with this "small n, large p" data setting noted as $p >> n$. In addition, expression data suffers from a high level of noise and when coupled with low sample size renders the analysis of data even more challenging.

DNA microarray data DNA Microarray technology is a high-throughput method which uses nucleic acid hybridization techniques to monitor the whole transcriptome on a single chip. A DNA microarray is a small solid surface containing thousands to millions of microscopic spots of gene specific sequences called probes. In order to measure the expression level of genes, the mRNA material is extracted form cell and labeled with a fluorescent dye. The labelled mRNA is then placed onto the slide where they hybridize (bind) to the probe with its complementary gene sequence. Fluorescence signals are then measured by scanning the microarray and are proportional to the concentration of mRNA.

RNA-Seq data RNA-Seq (RNA Sequencing) is a novel high-throughput method which utilizes next generation sequencing (NGS) to map and quantify the transcriptomes. In its protocol, the first step is the reverse transcription of RNA into cDNA samples. The cDNA samples are then used as the input to NGS to produce short sequence reads. The reads are then mapped to the reference genome. Finally, read counts obtained from mapping are used to estimate the gene expression levels.

1.2.2 ChIP-X data

Chromatin immunoprecipitation (ChIP) coupled with high-throughput techniques, such as microarray (ChIP-chip) and NGS (ChIP-Seq), is a method to study the interactions between specific proteins and a genomic DNA region. Specifically, it can be used to find transcription factor binding sites (TFBS) of a specific transcription factor (TF) along a DNA region (or the whole genome) and as a result to identify the potential targets of the TF.

In the ChIP-chip protocol, the protein of interest is first cross-linked with the DNA molecule. The DNA is then fragmented and an antibody specifically designed for the

protein is used to recover the DNA-protein complex. The complex is reverse cross-linked and the single stranded DNA is obtained, amplified, and denatured. The DNA strand is labeled with a fluorescent tag and hybridized over cDNA strands of known DNA positions arranged on a DNA array. Binding positions are then identified by measuring the fluorescence signal along the DNA.

In the ChIP-seq protocol, the protein of interest is first cross-linked with the DNA molecule. Following this, DNA is fragmented and an antibody specifically designed for the protein is used to recover the DNA-protein complex. Then, the fragments are sequenced and mapped to the reference genome. The read counts obtained from mapping can then be used to localize protein binding sites.

Chapter 2

Review of statistical models for reconstruction of gene networks

There is long history of using statistical methods to measure the associations among variables in many fields including biology. The Pearson correlation coefficient which quantifies linear associations, is probably the most well known method. However, the Pearson correlation is not an accurate way to measure nonlinear association which are ubiquitous in biology and especially in the context of GRNs where the regulatory relationships between genes are known to be nonlinear [6]. A more general measure is mutual information, which also quantifies nonlinear associations. However, reliably estimating mutual information from finite continuous data is a nontrivial task. Therefore, quantifying nonlinear associations is an active field of research and some new measures have been proposed recently which we introduce in this chapter.

We also describe some of the models, that are based on the concept of conditional independence. In these models, each gene or more precisely gene-activities (gene expression level) is considered as random variable and the aim is to find the (in)dependency structure among genes. The models differ on how they model the (in)dependencies between the variables.

2.1 Association measurements

In the following, we first briefly define the concept of marginal and conditional independence and then we introduce some association measurements that can be used to test the (conditional) independence between variables.

Two variables are called statistically independent if information about one does not change the probabilities of the other. Statistical independence indicates that there is no relation between two random variables. Consider two random variables X and Y with a joint probability distribution $f(x, y)$ (joint probability mass function for discrete variables and joint probability density function for continuous variables) and marginal distributions $f(x)$ and $f(y)$. Two variables X and Y are independent, denoted as $X \perp Y$, if and only if their joint probability distribution is the product of the marginals:

$$f(x, y) = f(x)f(y)$$

If two random variables X and Y are not independent they are dependent and is denoted as $X \not\perp Y$.

In a multivariate analysis, it is important to account for the influence of other variables on the relationship between two variables to detect direct interactions. We want to know whether the relationship between two variables can be explained away by a subset of other variables in the system. In other words, we want to distinguish between direct and indirect relationships. Here enters the concept of conditional independence. Two variables X and Y are called conditionally independent given a set of variables Z if, if we know the value of one variable, the knowledge about Z does not provide any further information about the other variable.

Formally, two variables X and Y are conditionally independent with respect to a probability distribution f given a set of variables Z, if $f(X, Y|Z) = f(X|Z)f(Y|Z)$, and denoted as $(X \perp Y|Z)$. The cardinality of the set Z is called the order of conditional independence. Marginal independence is the special case of conditional independence when there is no variable in the conditional set.

In reality, we have to estimate the independence of variables from observation data. A variety of methods have been proposed to quantify the (in)dependence between two random variables, each of which has its own advantages and disadvantages. In this section, we do not introduce independence tests based on the association measures which test the null hypothesis "$\mathcal{H}_0$:X and Y are independent (given Z)" against the alternative hypothesis "$\mathcal{H}_1$:X and Y are not independent (given Z)". We introduce them when it is needed.

2.1.1 Correlation and partial correlation

The Pearson correlation coefficient, commonly referred to as the correlation coefficient, is a widely used tool to measure the linear association between two variables X and

Y. It is defined as the covariance of the two variables divided by the product of their standard deviations:

$$r(X,Y) = \frac{cov(X,Y)}{\sigma_x \sigma_Y}.$$

The empirical correlation coefficient in the presence of n iid samples $(X_i, Y_i), i = 1, ..., n$ is computed as:

$$r(X,Y) = \frac{\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{N}(Y_i - \bar{Y})^2}}$$

where $\bar{X}$ and $\bar{Y}$ are the sample means of X and Y, respectively. The correlation coefficient takes values in $[-1, 1]$. Values of 1 and -1 indicate perfect (positive and negative) linear relation between the variables where one variable is completely determined by the other. If both variables are linearly independent, $r(X,Y) = 0$. The inverse statement is not necessarily true, only when the underlying distribution of variables are Gaussian.

In multivariate analysis, the partial correlation coefficient is the generalization of the correlation coefficients with the ability of controlling for other variables. It measures the correlation between two random variables after removing the effect of one or several other variables. The partial correlation coefficient between two random variables X and Y conditioning on Z is the correlation between the residuals of X and Y after they are regressed on the control variables Z. In other words, it is the correlation between the parts of X and Y that are uncorrelated with Z.

Partial correlation of order zero, i.e. when the conditional set is empty, is equal to correlation. Partial correlations of order $q > 0$ can be obtained by solving two corresponding regressions and then computing the correlation between the residuals. However, this approach is time consuming and in practice the two common methods to compute the partial correlation are the recursive formula and matrix inversion.

For $\mathcal{Z} \subset V \backslash \{X, Y\}$, the qth-order partial correlation can be obtained from $(q-1)$th-order partial correlation by using the following recursive formula:

$$\rho(X, Y|\mathcal{Z}) = \frac{\rho(X, Y|\mathcal{Z}\backslash\{Z_0\}) - \rho(X, Z_0|\mathcal{Z}\backslash\{Z_0\})\rho(Y, Z_0|\mathcal{Z}\backslash\{Z_0\})}{\sqrt{(1 - \rho^2(X, Z_0|\mathcal{Z}\backslash\{Z_0\})(1 - \rho^2(Y, Z_0|\mathcal{Z}\backslash\{Z_0\}))}}, \ for\ any\ Z_0 \subset \mathcal{Z}$$

The matrix inversion approach allows to compute all full order partial correlations at the same time. In this method, one should first compute the concentration matrix of the data. The concentration matrix Ω of p variables $V = \{X_1, X_2, ..., X_p\}$ is the inverse

of its covariance matrix Σ (when it is positive definite and therefore invertible):

$$\Omega = \Sigma^{-1}$$

Partial correlation coefficients between X_i and X_j $(i, j = 1, ..., p)$ given all other variables , i.e. $V \backslash \{X_i, X_j\}$, are then obtained by normalizing the off-diagonal entries of the concentration matrix $\Omega = (w_{ij})$:

$$\rho_{ij} = -\frac{w_{ij}}{\sqrt{w_{ii} w_{jj}}}$$

To obtain partial correlations by matrix inversion, the sample covariance matrix should be positive definite which only holds when the number of variables is lower than the number of samples. Therefore, in the cases like gene expression data with high variable/sample ratio different methods have been suggested to overcome this problem.

Two variables X and Y are conditionally independent in the multivariate Gaussian case if and only if the partial correlation between X and Y conditioned on Z is zero:

$$\rho_{X,Y|Z} = 0 \Longleftrightarrow X \perp Y|Z$$

As a result, if the distribution of $V = \{X_1, X_2, ..., X_p\}$ is multivariate Gaussian, two variables are conditionally independent (given the remaining variables) if and only if the corresponding entry in the concentration matrix is zero:

$$X_i \perp X_j | V \backslash \{X_i, X_j\} \Longleftrightarrow w_{ij} = 0$$

However, (partial) correlation captures only linear associations and its power is reduced when associations are nonlinear. In addition, zero (partial) correlation means independency just in the case of a Gaussian distribution. If the underlying distribution of the data is not Gaussian, other association measures that do not assume any particular distribution for the data can be more useful. However, partial correlation and correlation are both widely used even when normality of data is questionable.

2.1.2 Mutual information and conditional mutual information

Mutual information is another widely used measure of association between two variables [7]. MI is a fundamental concept of information theory defined by Shannon based on the concept of entropy. The entropy is the uncertainty of a single random variable. For two continuous random variables X and Y with marginal probability density function

$f(x)$ and $f(y)$ and joint probability distribution $f(x, y)$, marginal entropy $H(X)$ and joint entropy $H(X, Y)$ are defined as:

$$H(X) = -\int f(x)log(f(x))dx,$$

$$H(X, Y) = -\int f(x, y)log(f(x, y))dxdy,$$

where log is natural logarithm so that information is measured in natural units. Conditional entropy $H(X|Y)$ measures how much entropy variable X has if the value of a second random variable Y is known and is defined as:

$$\begin{aligned} H(X|Y) &= H(X, Y) - H(Y) \\ &= -\int f(y)H(X|Y = y)dy \end{aligned}$$

While entropy is a measure of the uncertainty about one variable, mutual information (MI) measures how much knowing one variable reduces uncertainty about the other. MI is non-negative and equal to zero when X and Y are independent. The MI between two variables X and Y is defined as:

$$\begin{aligned} MI(X, Y) &= \int\int f(x, y) \log \frac{f(x, y)}{f(x)f(y)} dxdy \\ &= H(Y) - H(Y|X) = H(X) - H(X|Y) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned}$$

To account for the influence of other variables in a multivariate system, conditional mutual information (CMI) between two variables X and Y given the value of a third variable Z is defined as:

$$\begin{aligned} CMI(X, Y|Z) &= \int\int f(x, y, z) \log \frac{f(z)f(x, y, z)}{f(x, z)f(y, z)} dxdy \\ &= H(X, Z) + H(Y, Z) - H(Z) - H(X, Y, Z) \\ &= H(X|Z) - H(X|Y, Z) \end{aligned}$$

CMI is always non-negative and can be smaller or greater than the corresponding MI. Further, CMI is zero if and only if X and Y are conditionally independent given Z.

$$X_i \perp X_j | V \backslash \{X_i, X_j\} \Longleftrightarrow CMI(x, y|z) = 0$$

MI and CMI are very promising tools to capture the association between variables owing to their capability of characterizing nonlinear dependency. However, to estimate MI, one has to estimate the probability density functions from a finite sample of n data points which is a nontrivial task.

There are several algorithms to estimate MI. For discrete data, the density functions can be estimated by simply counting the events. But gene expression data, which is the main data for inferring gene networks, are continuous. For continuous data the most popular strategies are based on a discretized model.

Estimation of MI is easier when the underlying distribution of variables are known. For example, under the assumption of Gaussian data, MI can be calculated as a function of covariance matrices:

$$MI(X, Y) = \frac{1}{2} \log \frac{|C(X)|.|C(Y)|}{|C(X, Y)|}$$

where C is the covariance matrix of variables, and $|C|$ is the determinant of matrix C. If X and Y are univariate with correlation coefficient r, then

$$MI(X, Y) = -\frac{1}{2} \log(1 - r^2).$$

Similarly, the CMI between X and Y given Z is a function of the partial correlation ρ between X and Y given Z:

$$CMI(X, Y|Z) = -\frac{1}{2} \log(1 - \rho^2)$$

If the underlying distribution of data is unknown, the estimation of MI is more complicated. The naive histogram-based method partitions the data into b bins and approximates the probabilities by the frequencies of occurrence in the bins. There are different methods for binning. The so called "equal width method" partitions the data into bins of equal size, resulting in different number of data points in each bin. This method suffers from a systematic error that overestimates the MI as a result of finite-size effects. A more sophisticated binning method, called "equal frequency", uses an

adaptive partitioning, where the bin size depends on the density of data points such that the marginal distributions are uniform. It has been shown that this method is superior to equal width method but at the cost of increasing the computation time. There are several other computationally demanding methods to estimate MI which is beyond the scope of the thesis [7, 8]. The choice of the estimator or the parameters for MI may influence the estimations considerably. Therefore, finding an accurate and stable estimation of MI from finite continuous data is an active field of research. More details about the influence of the choice of estimators of MI on the network inference problem can be found in [7].

2.1.3 Distance correlation and partial distance correlation

Distance correlation has emerged recently [9, 10] as a measure of association strength between random variables with the important property that it is equal to zero if and only if the random variables are statistically independent. Furthermore, it is defined for X and Y in arbitrary and not necessarily equal dimensions, rather than for univariate quantities.

Distance covariance between two variables X and Y is defined as the distance between the joint characteristic function $\phi_{X,Y}$ and the product of its marginal characteristic functions ϕ_X and ϕ_X in a special weighted space. If $X \in R^p$ and $Y \in R^q$ the distance covariance $dcov(x, y)$ is the non-negative square root of:

$$\begin{aligned} dcov^2(X,Y) &= ||\phi_{X,Y}(t,s) - \phi_X(t)\phi_Y(s)||_w^2 \\ &= \int_{R^{p+q}} |\phi_{X,Y}(t,s) - \phi_X(t)\phi_Y(s)|^2 w(t,s)dt\ ds \end{aligned}$$

where

$$\begin{aligned} w(t,s) &= (c_p c_q |t|_p^{1+p} |s|_q^{1+q}), \\ c_d &= \frac{\pi^{\frac{1+d}{2}}}{\Gamma(\frac{1+d}{2})}, \end{aligned}$$

and $\Gamma(.)$ is gamma function. Distance correlation is the standardized version of $dcov(X, Y)$ and defined as the non-negative square root of

$$dcor^2(X,Y) = \frac{dcov^2(X,Y)}{\sqrt{dcov^2(X,X)dcov^2(Y,Y)}}.$$

where $dcov^2(X,X)dcov^2(Y,Y) > 0$ and otherwise is equal to zero.

The empirical distance correlation for two random variables X and Y with given n iid samples can be calculated very easily. First, Euclidean distance matrices are calculated as $(a_{ij}) = (|X_i - X_j|)$ and $(b_{ij}) = (|Y_i - Y_j|)$. Then the transformed distance matrices $\hat{A}$ and $\hat{B}$ are obtained from the Euclidean distance matrices by subtracting the row/column means and adding the grand mean:

$$\hat{A}_{ij} = a_{ij} - \bar{a}_{i.} - \bar{a}_{.j} + \bar{a}$$

where $\bar{a}_{i.} = \frac{1}{n}\sum_{k=1}^{n} a_{ik}$, $\bar{a}_{.j} = \frac{1}{n}\sum_{k=1}^{n} a_{kj}$ and $\bar{a}_{i.} = \frac{1}{n^2}\sum_{i,j=1}^{n} a_{ij}$. The analogous definition is used for $\hat{B}$. These transformed distance matrices are called double centered distance matrices.

The sample distance covariance is then defined as the square root of

$$dcov_n^2(X,Y) = \frac{1}{n^2}\sum_{i,j=1}^{n} \hat{A}_{ij}\hat{B}_{ij}$$

and sample distance correlation dcor as the square root of

$$dcor_n^2(X,Y) = \frac{dcov_n^2(X,Y)}{\sqrt{dcov_n^2(X,X)dcov_n^2(Y,Y)}}.$$

For distributions with finite first moments the distance correlation takes values on $[0,1]$ and $\mathcal{R} = 0$ if and only if X and Y are independent. In the bivariate normal case $dcov(X,Y) \leq |cov(X,Y)|$,with equality when $|cov(X,Y)| = 1$

Recently, Szkely et al. [11] introduced a method to compute partial distance correlation. Partial distance correlation (analogous with partial correlation) controls for the effect of other variables in the systems on the association between two variables. Therefore, it can detect direct nonlinear interactions in multivariate analysis.

Since the squared distance covariance is not an inner product in the usual linear space, Szkely et al. introduce a new Hilbert space where the squared distance covariance is the inner product. They first define a new transformed matrix called $\mathcal{U}$-centered matrix. The (i,j)th entry of a $\mathcal{U}$-centered matrix $\tilde{A}$ for a $n \times n$ symmetric matrix $A = (a_{i,j})$ $(n > 2)$ with zero diagonal is defined as:

$$\tilde{A} = \begin{cases} a_{i,j} - \frac{1}{n-2}\sum_{l=1}^{n} a_{i,l} - \frac{1}{n-2}\sum_{k=1}^{n} a_{k,j} + \frac{1}{(n-1)(n-2)}\sum_{k,l=1}^{n} a_{k,l} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

If A and B are the Euclidean distance matrices of n iid samples (x_i, y_i) $i = 1, ..., n$ of two random vectors X and Y,

$$(\tilde{A}.\tilde{B}) = \frac{1}{n(n-3)} \sum_{i \neq j} \tilde{A_{i,j}}.\tilde{B_{i,j}}$$

is an unbiased estimator of squared population distance covariance [11].

They define a new Hilbert space $\mathcal{H}_n = \{\tilde{A} | A \in \mathcal{S}_n\}$ $(n > 4)$ where $\mathcal{S}_n$ is the linear span of all $n \times n$ distance matrices of samples $\{x_1, x_2, .., x_n\}$ in a Euclidean space R^p. They also define the inner product for each pair of elements $C = (C_{i,j})$ and $D = (D_{i,j})$ in the linear span of $\mathcal{H}_n$ as

$$(C.D) = \frac{1}{n(n-3)} \sum_{i \neq j} C_{i,j}.D_{i,j}$$

With this definition of inner product in Hilbert space $\mathcal{H}_n$, they use a projection operator to define partial distance covariance and partial distance correlation for random vectors in Euclidean spaces. Let $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ be elements of $\mathcal{H}_n$ corresponding to random samples x, y and z from vectors X, Y and Z, respectively. Further, consider

$$P_{z^\perp}(x) = \tilde{A} - \frac{(\tilde{A}.\tilde{C})}{(\tilde{C}.\tilde{C})}\tilde{A}$$

,

$$P_{z^\perp}(y) = \tilde{B} - \frac{(\tilde{B}.\tilde{C})}{(\tilde{C}.\tilde{C})}\tilde{B}$$

denote the orthogonal projections of $\tilde{A}(x)$ and $\tilde{B}(x)$ onto $(\tilde{C}(x))^\perp$, respectively. The sample partial distance covariance (pdCov) is defined as:

$$\begin{aligned} pdCov(x, y|z) =& (P_{z^\perp}(x).P_{z^\perp}(y)) \\ &= \frac{1}{n(n-3)} \sum_{i \neq j} (P_{z^\perp}(x))_{i,j} (P_{z^\perp}(y))_{i,j} \end{aligned}$$

The partial distance correlation is defined as cosine of the angle θ between the vectors $P_{z^\perp}(x)$ and $P_{z^\perp}(y)$ in the Hilbert space $\mathcal{H}_n$:

$$\begin{aligned} pdCor(x, y|z) &= cos\theta \\ &= \frac{(P_{z^\perp}(x).P_{z^\perp}(y))}{|P_{z^\perp}(x)|.|P_{z^\perp}(y)|}, \qquad |P_{z^\perp}(x)|.|P_{z^\perp}(y)| \neq 0, \end{aligned}$$

and otherwise it is equal to zero.

2.1.4 Heller, Heller and Gorfine (HHG) measure

Heller et al. [12] also proposed recently a test of independence between two multivariate random variables X and Y. The test is based on the pairwise distances between the sample values within X and Y respectively. i.e $\{d_X(x_i, x_j)|i, j \in \{1, ..., n\}\}$ and $\{d_Y(y_i, y_j)|i, j \in \{1, ..., n\}\}$ where $d(.,.)$ is the norm distance between two sample points.

As stated by the authors, the motivation of the test is that if X and Y are not independent and have a continuous joint density distribution then there exists a point (x_0, y_0) in the sample space of (X, Y) and radii R_x and R_y around x_0 and y_0 respectively, such that $f(x, y) \neq f(x)f(y)$ in the Cartesian product of balls around (x_0, y_0). Lets assume that we know the point (x_0, y_0) and the radii R_x and R_y. Further, consider the two dichotomous random variable $I\{d(x_0, X) \leq R_x\}$ and $I\{d(y_0, Y) \leq R_y\}$, where $I(.)$ is the indicator function. The table 2.1 shows the observed cross-classification of these two random variables for the n independent observations of (x_k, y_k) $k = 1, ..., n$ where

$$A_{11}(i, j) = \sum_{k=1}^{n} I\{d(x_0, x_k) \leq R_x\} I\{d(y_0, y_k) \leq R_y\}$$

and A_{12}, A_{21}, A_{22} defined similarly. Let $A_{m.}$, $A_{.m}$, $m = 1, 2$ be the sums of the rows or columns, respectively. Then the Pearson's chi-square test statistic or the likelihood ratio test statistic for contingency tables can quantify evidence against independence.

	$I\{d(y_0, .) \leq R_y\}$	$I\{d(y_0, .) > R_y\}$	
$I\{d(x_0, .) \leq R_x\}$	A_{11}	A_{12}	$A_{1.}$
$I\{d(x_0, .) > R_x\}$	A_{21}	A_{22}	$A_{2.}$
	$A_{.1}$	$A_{.2}$	n

TABLE 2.1: **The cross-classification of** $I\{d(x_0, X) \leq R_x\}$ **and** $I\{d(y_0, Y) \leq R_y\}$.

Since the point (x_0, y_0) and the radii R_x and R_y are not known, each sample point (x_i, y_i) in turn plays the role of (x_0, y_0) and for every sample point (x_j, y_j) $j \neq i$, R_x and R_y are defined as $d(x_i, x_j)$ and $d(y_i, y_j)$ respectively. Therefore, for each point $(x_i, y_j), j \neq i$ they define two random variables $I\{d(x_i, X) \leq d(x_i, x_j)\}$ and $I\{d(y_i, Y) \leq d(y_i, y_j)\}$ and obtain a 2×2 table of cross-classification of these two random variables for the remaining $n-2$ points as shown in table 2.2. Here, $A_{11}(i, j) = \sum_{k=1, k\neq i, k\neq j}^{n} I\{d(x_i, x_k) \leq d(x_i, x_j)\} I\{d(y_i, y_k) \leq d(y_i, y_j)\}$ and $A_{12}(i, j)$, $A_{21(i,j)}$, $A_{22}(i, j)$ defined similarly, and $A_{m.}(i, j)$, $A_{.m}(i, j)$, $m = 1, 2$ are the sum of the row or column, respectively.

To test the independence between two random variables X and Y, they defined the following statistic which aggregates the evidence against independence from all obtained

	$I\{d(y_i,.) \leq d(y_i,y_j)\}$	$I\{d(y_i,.) > d(y_i,y_j)\}$	
$I\{d(x_i,.) \leq d(x_i,x_j)\}$	$A_{11}(i,j)$	$A_{12}(i,j)$	$A_{1.}(i,j)$
$I\{d(x_i,.) > d(x_i,x_j)\}$	$A_{21}(i,j)$	$A_{22}(i,j)$	$A_{2.}(i,j)$
	$A_{.1}(i,j)$	$A_{.2}(i,j)$	$n-2$

TABLE 2.2: **The cross-classification of $I\{d(x_i,X) \leq d(x_i,x_j)\}$ and $I\{d(y_i,Y) \leq d(y_i,y_j)\}$.**

2×2 tables :

$$T = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} S(i,j)$$

where

$$S(i,j) = \frac{(n-2)\{A_{12}(i,j)A_{12}(i,j) - A_{11}(i,j)A_{22}(i,j)\}^2}{A_{1.}(i,j)A_{2.}(i,j)A_{.1}(i,j)A_{.2}(i,j)}$$

is the Pearson's chi square statistic for the 2×2 contingency table corresponding to (x_i, y_i). Note for i and j with 0 in at least one of the margins they set $S(i,j) = 0$.

2.1.5 Maximal Information Coefficient

The Maximal Information Coefficient (MIC) is another recently proposed measure of association between variables[13]. Reshef et al. designed MIC with the goal of satisfying two properties: generality and equitability. Generality means that the measure should be able to capture any kind of association (with sufficient sample size). Equability means to assign similar scores to equally noisy relationships independent of the association type.

The rational behind MIC is that if there is a relationship between two random variables X and Y, then a grid that partitions the data in the scatter plot of X and Y can encapsulate this relationship. Thus, for each possible grids of size m-by-n ($m \times n \leq \mathrm{N}^{0.6}$, where N is the number of samples) the largest possible mutual information $MI(X,Y)$ is computed. Then this value is normalized between 0 and 1 by dividing by the maximum achievable value for the grid of size m by n which is $log(min(m,n))$. The normalization step ensures a fair comparison between grids of different dimensions. The MIC is then defined as the maximum of the normalized mutual information values obtained from grids of different dimensions:

$$MIC(X,Y) = \max_{m \times n \leq B} \frac{MI(X,Y)}{log(min(m,n))}$$

where B is the the maximal resolution ($\mathrm{N}^{0.6}$). It is important to note that to find the maximum a heuristic approach has been used, since trying all possible binning schemes that satisfy $n_x \times n_y \leq \mathrm{N}^{0.6}$ is computationally infeasible even for small N.

Although the power (ability) of this method to identify known and novel relationships has been shown by applying it to various data in the original paper, some criticism was raised about the performance of MIC after the publication. In a comment [14], Simon and Tibshirani questioned the power of MIC by showing simulation results for different types of relationships demonstrating that MIC has lower power than DCOR for most relationships. In another comment [14] Gorfine et al. argued that the claim that non-equitable methods are less practical for data exploration is not true and both DCOR and their own HHG method are more powerful than the test based on MIC. In a recent paper[15], it has been proved that no non-trivial coefficient can exactly satisfy the equitability property as defined by Reshef et al. Recently, Reshef et al. addressed some criticisms and at the moment the debate about MIC and equitability is very active. Without going through all the debate which is beyond the scope of this thesis, we consider MIC in our analysis for the sake of comparison. However, in the context of multivariate analysis the drawback of MIC is that it is not clear how to extend it to the conditional case (in analogy to the partial correlation for correlation).

2.2 Relevance networks

Relevance networks or co-expression networks are a simple method to associate genes together. They look into all pairs of genes and associate those that have similar expression profiles throughout a set of different conditions and link them by an edge in the graph. This similarity score can be any association measures.

Although relevance networks are computationally efficient, in terms of identifying direct regulatory interactions they are not efficient since marginal independence alone cannot distinguish between direct or indirect associations. Therefore, they associate genes that only interact indirectly through one or more other genes.

In relevance networks, there are some methods to prune the reconstructed network of such false positives i.e. indirect interactions. The first method called ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks)[16] uses mutual information as association measurement and then prune the inferred network of false positives based on the so-called Data Processing Inequality (DPI). DPI states that if two variables X_1 and X_2 interact through a third variable X_3 and there is no alternative path from X_1 to X_2, then:

$$MI(X_1, X_2) < min(MI(X_1, X_3), MI(X_3, X_2))$$

Therefore, after computing pairwise mutual information, ARACNE evaluates interconnected triplets of variables and removes the link with the smallest associated mutual information.

Another extension to the relevance networks algorithm is the Context Likelihood of Relatedness (CLR) method [17]. It originally uses mutual information but in principle can be used for any association measure. Once association measures s_{ij} have been assigned for each pair of variables, a CLR score is derived, related to the empirical distribution of association scores s.

$$clr.score = \sqrt{\left(\frac{\mathbf{s}_{ij} - \mu_i}{\sigma_i}\right)^2 + \left(\frac{\mathbf{s}_{ij} - \mu_j}{\sigma_j}\right)^2}$$

where μ_i and σ_i (resp. μ_j and σ_j) are the mean and standard deviation of the association scores between X_i (resp. X_j) and all other variables.

MRNET is also an extension of the relevance networks [18]. It uses the minimum redundancy, maximum relevance (MRMR) feature selection criterion. MRNET originally was based on mutual information but in principle it can be used for any association measure. In MRNET each variable in turn plays the role of the target variable X_T and in a forward selection strategy the variable X_i with highest association score with the target is selected. At every subsequent step the variable with the highest association score with the target and, at the same time with the lowest average score with the already selected variables S is selected. In other words, the variable which maximizes $s_i = score(X_i, X_T) - \frac{1}{|S|}\sum_{x_k \in S} score(X_j, X_k)$ will be added to the currently selected variables. In a final step, the score for each pair of variables X_i and X_j is computed by taking the maximum between s_i and s_j.

2.3 Graphical models

Graphical models are representations of multivariate probabilistic models, where the conditional (in)dependencies between the random variables are expressed via a graph. Nodes of the graph correspond to random variables (genes) and the absence of edges between nodes represent conditional independencies between variables. There are two major classes of graphical models: namely, undirected graphical models, also known as Markov networks, where the edges of the graph have no direction, and directed graphical models, also known as Bayesian networks, where the edges have specific directions that could have causal interpretation under some assumptions.

Graphical models have gained much attention in the context of GRN inference due to their ability to distinguish between direct and indirect interaction. Since they use the conditional independence concept, they can eliminate indirect interactions among genes. This is very important since pairs of genes do not interact independently of all other genes and more or less all genes will be directly or indirectly dependent. However, it is not trivial to learn GMs from high dimensional small n, large p gene expression data and a wide variety of algorithms have been suggested to tackle this problem.

2.3.1 Gaussian graphical models

Gaussian graphical models (GGMs) are undirected graphical models that identify the conditional independence relations among the nodes under the assumption of a multivariate Gaussian distribution of the data. With this assumption, GGMs assess the conditional independencies among variables by terms of full-order (i.e. by conditioning on all other variables) partial correlation coefficients. The pairs of variables with zero partial correlation correspond to conditional independencies between the variables (they are independent of each other given the rest of the variables in the graph) and are removed from the graph. Therefore, the edge set of a GGM is defined by non-zero partial correlations.

To learn GGMs from the data, one should estimate the concentration matrix or equivalently the set of all full-order partial correlations from data. In practice, the sample concentration matrix is first computed and edges corresponding to significantly small values, indicating zero partial correlation, are removed from the graph. However, the sample concentration matrix requires the sample covariance matrix to be positive definite which only holds with probability one if and only if the number of variables is lower than the number of samples. This is problematic in the context of GRNs inference, where the number of genes are much higher than the number of samples. Inferring Gaussian graphical models (GGMs) in this issue, called "small n, large p" setting, is an ill-posed problem and different approaches have been suggested to cope with it.

2.3.1.1 Estimation of partial correlation in the $n >> p$ case

Different methods have been proposed to estimate the covariance matrix for data sets with high number of variables and low number of samples. The most straightforward method is the pseudoinverse approach. The pseudoinverse is the generalization of the inverse matrix and can be obtained by using singular value decomposition (SVD). Let $A = UDV^T$ be the SVD decomposition of the rectangular real matrix A, where U and V (V^T means V transposed) are orthogonal and D is a diagonal matrix whose entries

are zero except for the singular values which appear on the diagonal. In numerical computation, singular values smaller than a tolerance are also taken to be zero. Then the pseudoinverse, also known as Moore-Penrose or generalized inverse is obtained as:

$$A^{+} = VD^{+}U^{T}$$

where D^{+} is the pseudo inverse of D obtained by taking the reciprocal of non-zero singular values and then transposing the matrix.

Another way to circumvent the problem is to use limited-order partial correlation instead of full partial correlation [19]. A more sophisticated approach to overcome this issue is to introduce regularization to infer robust estimators of the covariance matrix. Specifically, the shrinkage techniques which combine two estimators into an overall better estimator. While there are many shrinkage methods, we use the approach suggested by Schäfer et al.[2].

2.3.2 Bayesian Networks

A BN is a graphical representation for probabilistic relationships among a set of random variables $V = \{X_1, ..., X_n\}$. The first component of a BN is its structure G, represented by a directed acyclic graph (DAG). A DAG is a graph containing only directed edges and no cycles, and the skeleton of a DAG is the DAG itself where directionality has been removed. Nodes correspond to the random variables in V and edges encode conditional dependencies over V. The second component of a BN is a set of distributions $\{P_i(X_i|\text{parents}(X_i, G))\}$ that are respectively conditioned on the parents of X_i in G, where a parent of X_i is a node X_j such that the edge $X_j \rightarrow X_i$ is in G. Together, G and $\{P_i\}$ define a joint probability distribution P over V as :

$$P(X_1, ..., X_n) = \prod_i P_i(X_i|\text{parents}(X_i, G)).$$

This follows from the conditional Markov assumption which states that each variable is independent of its non-descendants when conditioned on its parents.

A BN structure G entails a set of conditional independence relations that can be read from the G by the d-separation criterion. That is, if two variables X and Y are d-separated by a set Z, denoted as $X \perp_G Y|Z$, then $X \perp Y|Z$. Formally, two variables X and Y in a BN are d-separated given variable Z, if one of the following conditions holds:

- For all paths between X and Y, Z is a non-collider.

- Z is a collider and neither Z nor any of its descendent is observed.

A triplet of variables $X \rightarrow Z \leftarrow Y$, where X and Y are not connected is called v-structure and the center node Z is called collider. A DAG G and a probability distribution P generated by G are reciprocally faithful if and only if the independence relationships among the variables in V with respect to P are exactly those entailed by G by means of the d-separation criterion. The faithful assumption in BNs implies that there is an edge between nodes X_i and X_j in the skeleton of DAG G if and only if for all $Y \subset V \backslash \{X_i, X_j\}$, X_i and X_j are conditionally dependent given Y.

The application of BNs to reconstruct GRNs was pioneered by [3] and then became one of the most popular methods in this field. The main advantage of BNs is that edges are directed. However, this does not come free but at high computational cost.

It is also important to note that even with infinitely many samples, we can just learn the equivalence class of BNs. A class of BNs is called Markov equivalent if they represent the same statement of conditional independence and therefore they are statistically undistinguishable. Two graphs belong to the same equivalence class if and only if they have the same skeleton and the same set of v-structures. In the context of GRNs, this means that if we relate two genes in the graph it may not be clear which one is the regulator and which one is the target. This can be done just with perturbation experiments. An equivalence class of BNs can be represented by partially directed acyclic graph (PDAG), a DAG containing both directed and undirected edges with the same skeleton and the same set of v-structures as the DAGs in the equivalence class.

2.3.2.1 BNs structure learning

Learning methods to reconstruct the structure of BNs mostly fall into two categories: score-based methods and constraint-based methods [20, 21]. Score-based methods search the space of all possible DAGs to identify the network which maximizes a score indicating how well the DAG matches the given data. Constraint-based methods involve the repeated use of CI (conditional independence) tests. Both approaches have their own advantages and disadvantages. While score-based methods are favored when dealing with small dimensional data sets, their high computational cost made them intractable for large network. Constraint based methods are relatively fast but they are unstable which means that an error early on in the algorithm can cause many errors in the output i.e. the final graph. Recently, a new class called hybrid method has been suggested with the idea of combining the aforementioned methods to get the best of both [22].

Constraint-based methods Constraint-based methods involve the repeated use of CI tests to obtain the conditional independence relationships and use them as constraints to construct a PDAG representative of a BN equivalence class. Under the assumption of faithfulness, if there is no $S \subset V \backslash \{X, Y\}$ such that $X \perp Y|S)$ holds true, there is an edge between X and Y. The naïve algorithm decides on the presence of an edge by conditioning on all possible S. However, the naïve approach scales poorly and becomes infeasible for large networks due to the super exponential growth of the number of tests with respect to the number of nodes.

Since it is enough to find one S that $X \perp Y|S$ to remove the edge between X and Y, a more efficient way is to perform the CI tests such that skip unnecessary tests. Starting with low order tests and then proceeding to higher order tests is one reasonable way. In addition, if there exist such a S that $X \perp Y|S$, then there should be a S' which $X \perp Y|S'$ and all the variables in S' are adjacent to X or Y (or both). So to decide on the presence of an edges between X and Y, we only need to condition on variables which are still connected to X and Y and not those in different parts of the graph. This is the rational behind the PC algorithm [21] which we explain in more detail in section 2.3.3.

After finding the skeleton of the BN, direction will be assigned to the edges by using the identified separating sets. The first step to orient the edges is to identify potential v-structures in the graph. Therefore, for each pair X and Y with a common neighbor Z, the chain $X - Z - Y$ will be oriented into the v-structure $X \rightarrow Z \leftarrow Y$ if $Z \in S_{XY}$. The next step is to try to orient as many undirected edges as possible in the resulting PDAG following some rules. The first rule aims at avoiding the addition of new v-structures. Thus, orient $X - Z$ to $X \rightarrow Z$ in patterns such as $X - Z \rightarrow Y$ and X and Y are not adjacent. The second rule tries to avoid making cycles in the graph and orient all the edges that could potentially lead to a cycle. It is possible that the final result is PDAG with some undirected edges which can be oriented randomly.

Constraint-based methods are relatively fast and deterministic with a well defined stopping criterion. The main drawback of constraint-based algorithms is their poor robustness as they rely on an arbitrary significance level to test for independence. This means that small changes of the input i.e. single errors in the independence tests may lead to a large effects on the output of the algorithm i.e. the structure of the BN.

Score-based methods Score-based methods aim to find the network which optimizes a score among all possible networks. The score indicates how well the model can explain the data. Due to the very high number of possible network structures, exhaustive search to find the optimal network is not possible and therefore most existing learning methods

use standard heuristic search techniques with no guarantee to find a globally optimal solution.

There are several scoring functions. The simplest score is the likelihood of DAG G and a set of conditional probabilities θ after observing D:

$$L(G, \theta : D) = P(D|G, \theta)$$

Then one searches for the DAG G that maximizes the likelihood score. The likelihood score takes higher values for more complex structures with higher number of edges and therefore it is hard to compare networks with different number of edges. The standard solution for this is to penalize the likelihood according to model complexity. The Akaike Information Criterion (AIC) and the Bayesian information criterion (BIC) are both based on this idea and penalize the maximal likelihood of the model with respect to the number of model parameters. A more popular score is the Bayesian score which allows to include prior knowledge. It evaluates the posterior probability of DAG G given data D:

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)}$$

The term $p(D)$ is an average of data likelihoods over all possible models which can be neglected for relative model scoring. The $p(D|G)$ is the marginal likelihood and equals the full model likelihood averaged over all parameters of the local probability distributions, that is,

$$P(D|G) = \int_{\theta} P(D|G, \theta)P(\theta|G)d\theta$$

$p(G)$ is the prior on the structure of the network. This prior knowledge can be obtained from any source including, for example, a domain expert who can specify edges that are likely or not likely to be present in the network and other databases. However, one should transform these knowledge of known interaction into prior distributions that can be used in Bayesian framework.

Hybrid methods The third class of methods for learning the structure of BNs is a combination of constraint-based and score-based methods with the aim of overcoming their shortcomings while at the same time taking advantage of their strengths. The general idea is to find the skeleton based on a constraint-based method and use that as constraint on the DAGs considered in score-based methods.

2.3.3 PC algorithm

The PC algorithm is a constraint-based method which reduces the number of CI tests by avoiding unnecessary tests. The original PC algorithm consists of two main steps: building the skeleton of the graph and determining the orientation of the edges. In the remainder of this thesis, we will consider the skeleton only, and PC will stand for the first part of the original PC algorithm. The step for assigning direction for the edges in PC is explained in section 2.3.2.1.

PC takes as input a set of variables V and an ordering order(V) over V, and returns the skeleton of the graph G. Algorithm 1 shows PC in pseudo code. It starts with a complete undirected graph, where all the nodes in V are connected to one another, and edges are then removed iteratively based on CIs. For every ordered pair of adjacent nodes (X_i, X_j), all CIs $X_i \perp X_j | S$ where S is a subset of all nodes adjacent to X_i are computed in order to find a set S^* such that $(X_i \perp X_j | S^*)$ holds true.

Y is at first the empty set (zero-order test), then each variable X_d in turn following order(V) (first-order test), then all possible pairs of potentials variables (X_d, X_e) following order(V) (second-order test) and so on, until a S^* is identified or all possible conditions have been exhausted. If a S^* is found, then the edge between X_i and X_j is deleted. As the algorithm proceeds, the number of adjacent nodes decreases, and fewer and fewer tests are needed. Assuming a faithful distribution to G and perfect CI tests, PC correctly infers the skeleton of G [21], regardless of order(V).

2.4 Bootstrapping and bagging

Ensemble methods are methods to improve the stability and accuracy of learning algorithms by building some base models that are different from one another. There are two possibilities to create diversity in base models. First approach, called "heterogeneous ensemble method" applies different methods on the same data set. Second approach, called "homogeneous ensemble method", uses the same method but on some perturbed data sets obtained from the original data (most commonly by using bootstrapping). Both methods have been applied for network reconstruction[23].

Bagging (Bootstrap Aggregating) [24] is a homogeneous ensemble learning. Briefly, it draws multiple bootstrap data sets from the original data set using bootstrap. Each of these data sets is used to construct a base model. These are then aggregated into a final result.

Algorithm 1 PC

Require: a vertex set V, an ordering order(V), exact conditional independencies

form the complete undirected graph G' over V
$l = -1$; $G = G'$;
repeat
  $l = l + 1$
  repeat
    following order(V), select a pair (X_i, X_j) of adjacent nodes in G such that $|\text{adj}(G, i)\backslash\{j\}| \geq l$
    repeat
      following order(V), choose $Y \subseteq \text{adj}(G, i)\backslash\{j\}$ such that $|Y| = l$
      if $(X_i \perp X_j | S)$ is true **then**
        delete the edge (X_i, X_j)
        denote the new graph G
      end if
    until the edge (X_i, X_j) is deleted or all $Y \subseteq \text{adj}(G, i)\backslash\{j\}$ such that $|Y| = l$ have been exhausted
  until all pairs (X_i, X_j) of adjacent nodes in G such that $|\text{adj}(G, i)\backslash\{j\}| \geq l$ have been tested for conditional independence
until for each pair (X_i, X_j) of adjacent nodes in G, $|\text{adj}(G, i)\backslash\{j\}| < l$
return G

The bootstrap is a method to estimate the distribution of an estimator and as a result to derive several quantities of interest such as the estimator's variance and bias. It creates a bootstrap data set from the original data set of size n, by performing n multinomial trials where, in each trial, it draws one of the n samples. As a result, some of the original samples will not be added to the bootstrap data while others will be selected one or several times. In other words, bootstrap is a sampling method with replacement.

In bagging, M bootstrap data sets, which are created from the original data, are likely to to induce some differences among the base models while leaving their performances reasonably good. However, it is important to note that bagging is more useful when the base model learning algorithm is unstable, i.e., when small changes in the input data lead to large changes in the result returned by the algorithm. This is because stable methods tend to return similar result in spite of the differences among the bootstrap data sets. As a result, the ensemble returns the same result as almost all of its base models with no improvement over them. In the case of unstable methods, the ensemble is likely to perform better than the base models.

In the context of network reconstruction and specially GRNs reconstruction, we can use bagging and obtain an ensemble of networks. The frequency of the interactions (edges) in the ensemble of networks can be used as a proxy for the confidence of the interactions. Interactions that are present in nearly all networks are most likely real interactions while

interactions that are missing in nearly all networks are most likely absent in the real network.

Chapter 3

Comparison of different methods for network reconstruction

In this chapter, we assess the performance of different methods that we described in Chapter 2, namely relevance networks (with different association measures) and graphical models. While relevance networks are just based on the concept of independence, the more sophisticated graphical models try to find the direct interactions based on the concept of conditional independence. We investigate the effect of the number of samples as well as the effect of the noise in the data on the performance of the methods. This can help us to learn more about their performance in "$p >> n$" situation and when the data is noisy, as is typical for gene expression data.

3.1 Simulation of data

3.1.1 Gaussian data

Although gene expression data are not necessary multivariate Gaussian, many methods for inferring GRNs from expression data make such an assumption. Under this assumption, we can assess the performance of these methods under a controlled experiment by simulating Gaussian data. This allows us to learn the strengths of the methods as well as their pitfalls.

In order to simulate Gaussian data, we first generate a random DAG G. In the next step, a positive definite matrix from the skeleton of G is generated which then can be used as a covariance matrix for a multivariate Gaussian data. The inverse of this covariance matrix contains zeroes at the missing edges of the given graph G. With this covariance

matrix we can simulate Gaussian data with arbitrary means by using the R package mvtnorm.

3.1.2 Non Gaussian data

Since gene expression data need not follow a multivariate Gaussian, to assess the performance of the methods in a controlled but a more realistic situation we also simulate non Gaussian data. Similar to Gaussian data, first a random DAG is generated as well as a random covariance matrix from its skeleton. Then we generate multivariate Gaussian data from this covariance matrix. In the next step, we transform the Gaussian data using the cumulative Gaussian distribution into uniform distribution. Now, by using inverse transform method we can transform the uniform distribution to the distribution of our interest. The Inverse transform method states that if F^{-1} is the inverse cumulative distribution function (CDF) of any distribution then applying F^{-1} to a uniform random variable over the interval $(0, 1)$ $(U(0, 1))$ results in a random variable whose distribution is exactly F.

For example, assume X and Y have bivariate Gaussian distribution with a non zero correlation. Let Φ be the CDF of Gaussian distribution, then $u = \Phi(x)$ and $v = \Phi(y)$ have marginal uniform distributions, but are still correlated. If F^{-1} and G^{-1} are the inverse CDF of any two distribution, then $a = F^{-1}(u)$ and $b = G^{-1}(v)$ are two correlated variables with distribution F and G respectively. In this thesis, we use Beta and Gamma distributions.

In reality and especially in biological data, the data contains noise. Therefore, when needed in the simulation we also add Gaussian noise $\epsilon \sim N(0, \sigma^2)$ to the data.

3.1.3 DREAM Challenge data

The DREAM (Dialogue for Reverse Engineering Assessments and Methods) challenge is an annual reverse engineering competition with the aim of fair comparison of network inference methods[23, 25, 26]. Participants are asked to generate a network structure for each data set with a confidence score for each edge. In this thesis, we use the data sets provided by the DREAM challenge, editions DREAM3, DREAM4 and DREAM5. Each of these editions proposed several data sets varying in size, but also in number and type of variables. Participants were asked to generate a network structure for each data set with a confidence score for each edge. They also provide gold standard network for each data set for the evaluation of the methods.

3.2 Validation

To evaluate the performance of any network inference method, we need to know the true network or the gold standard (GS) to compare the inferred network with. Having the GS, we can quantify the correctly identified edges or true positives (TPs), the correctly missed edges or true negatives (TNs), the incorrectly identified edges or false positives (FPs) and the missed detection or false negatives (FNs).

With these values in hand, we can compute different quality measures for the inferred network including precision, recall, false positive rate and F_1-score (see below). While recall (sensitivity or true positive rate) is the fraction of inferred true edges among all true edges, the precision is the fraction of inferred true edges among all inferred edges:

$$Recall = \frac{TP}{TP + FN}$$
$$Precision = \frac{TP}{TP + FP}.$$

False positive rate (FPR) is also defined as the fraction of falsely identified edges out of all nonexistent edges:

$$FPR = \frac{FP}{FP + TN}.$$

The F_1-score is defined as the harmonic mean of precision and recall:

$$F_1 = 2\,\frac{precision \times recall}{precision + recall}.$$

The performance of network reconstruction algorithms can be depicted graphically by receiver operator characteristic (ROC) curve which plots FPR versus the recall and precision-recall (PR) curve which plots recall versus precision. The area under the ROC curve noted AUROC and the area under the PR curve noted AUPR are indicators of how good is an inferred network and is used to assess the performance of the algorithm.

One can build the ROC and PR curves by varying the parameter responsible for the sparsity of the network. However, to build smooth ROC and PR curves, the algorithm can provide as output a ranking of the edges by defining a score for each edge in the network. This allows us to avoid setting a threshold to decide on an edge between correspondence nodes by sorting the edges based on their scores and on growing the network starting from the highest score down to the lowest one.

3.3 Relevance networks with different association measurements

In this section, we evaluate the performance of the relevance networks in finding the association between random variables in different scenarios. First, we assess the performance in the "well-behaved" case $n >> p$, when we have many samples compared to the number of variables. Then we investigate the effect of number of samples and noise on the performance of the methods separately for Gaussian and non Gaussian data. Note that non Gaussian data is obtained from Gaussian data with transformation that we explained in section 2.1.

3.3.1 Performance of relevance methods with different number of samples

In this section, we assess the effect of the number of samples on the performance of relevance networks with different association measurements. We simulate different number of samples of Gaussian and non Gaussian data from a network with 50 nodes. For each association measurement we also applied the CLR and MRNET methods to reduce the number of false positives (indirect interactions). As mentioned before, to avoid setting a threshold to decide on an edge between correspondence nodes we sort the edges based on their scores (in absolute values) and on growing the network starting from the highest score down to the lowest one.

Figures 3.1 and 3.2 show the ROC curve and PR curve for different association measures in the well behaved case ($n >> p$) with 1000 samples of Gaussian data. Similarly, figures A.1 and A.2 in Appendix A show the results for non Gaussian case. The results show that all the association measurements are comparable in the case $n >> p$ and when there is no noise. In addition, PR curves indicate that for each association measurements, CLR and MRNET methods improve the result by removing some of the indirect interactions (false positives). In addition, the performance of the methods in Gaussian case versus non Gaussian case are comparable.

Figures 3.3 and 3.4 show the effect of the number of samples on the performance of relevance networks with different association measures in terms of AUPR and AUROC on Gaussian data. Figures 3.5 and 3.6 show the same results for non Gaussian data. Clearly, for each association measure the fewer the number of samples, the worse the performance.

The AUPR results indicate that when the number of sample is much less than the number of nodes (10 and 20 samples for 50 variables) the CLR and MRNET methods are comparable to the simple relevance networks.

3.3.2 Effect of the noise on the performance of relevance methods

In this section, we assess the effect of the noise on the performance of relevance networks with different association measurements. We simulate 1000 samples of Gaussian data and non Gaussian data separately from a network with 50 edges and then add Gaussian noise $\epsilon \sim N(0, \sigma^2)$ with different amount of variance σ^2. Figures 3.7 and 3.8 show the effect of noise on the performance of relevance networks in terms of AUPR and AUROC. Figures 3.9 and 3.10 show similar results for non Gaussian data.

Clearly, for each association measurement the higher the amount of noise, the worse the performance. However, MIC seems to be more sensitive to the noise.

3.3.3 Performance of relevance methods on DREAM challenge data

In this section we compare the performance of relevance networks with different association measures on the DREAM Challenge data. Figures 3.11 and 3.12 show the ROC and PR curves for DREAM3 (50 nodes and 100 nodes) and DREAM4 (10 nodes and 100 nodes), respectively.

The results show that correlation and dcor have the best performance while MIC has the worst performance. In fact, MIC is not better than random classification.

3.4 Graphical models

In this section, we evaluate the performance of graphical Gaussian models and compare the result of pseudo-inverse and shrinkage methods (see section 2.3) in different scenarios. First, we compare the performance of the methods in the "well-behaved" case $n >> p$ when we have many samples compared to the number of nodes. Then we investigate the effect of number of samples and noise on the performance of the methods separately for Gaussian and non Gaussian data.

3.4.1 Performance of graphical Gaussian model with different number of samples

In this section, we assess the effect of the number of samples on the performance of graphical Gaussian models. We simulate different number of samples of Gaussian and non Gaussian data from a network with 50 nodes. In the well behaved case ($n >> p$) with 1000 samples and no noise both methods work almost perfectly with AUROC and AUPR values close to 1 (see figures A.3 and A.4 in the appendix A).

Figures 3.13 and 3.14 show the effect of the number of samples on the performance of pseudo-inverse and regularized methods on Gaussian data and non Gaussian data, respectively. Clearly, for both method the fewer the number of samples, the worse the performance. However, when the number of samples is low the regularized method has a better performance, as expected.

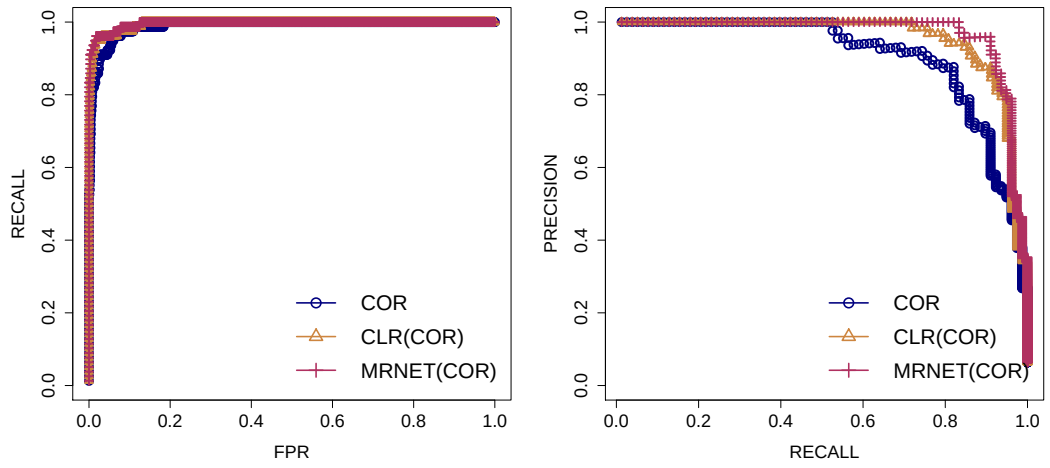
3.4.2 Effect of noise on the performance of graphical Gaussian model

In this section, we assess the effect of the noise on the performance of graphical Gaussian models. We simulate 1000 samples of Gaussian data and non Gaussian data separately from a network with 50 edges and then add Gaussian noise ϵ $N(0, \sigma^2)$ with different amount of variance σ^2. Figures 3.15 and 3.16 show the effect of noise on the AUPR and the AUROC for pseudo-inverse and regularized methods on Gaussian data and non Gaussian data respectively. The results show that when we have reasonable amount of samples (in this case 1000 samples for 50 variables) the noise does not have a strong impact on the results.

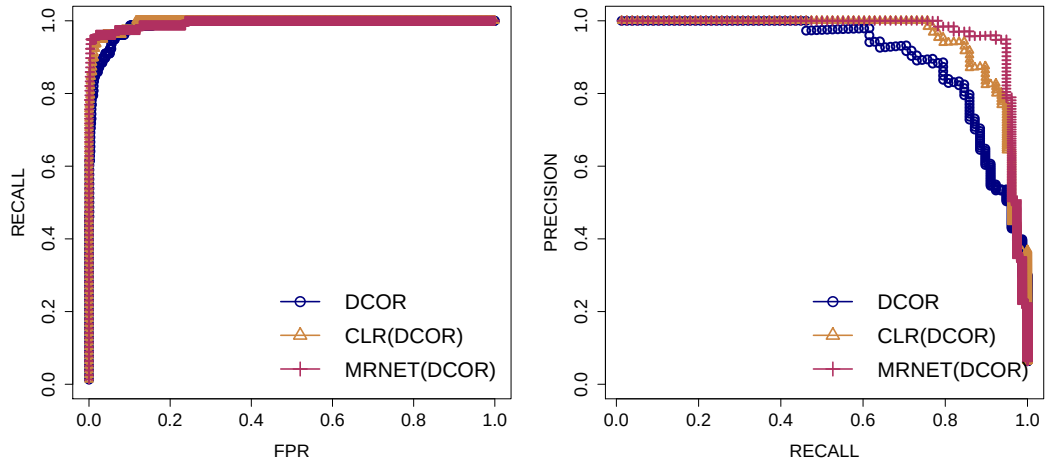
3.5 Conclusion

In this chapter, we assess the performance of relevance networks (with different association measures) and graphical models in different aspects. We used Gaussian and non Gaussian simulated data as well as data provided by the DREAM challenge. We investigate the effect of the number of samples as well as the effect of noise on the performance of the methods. The results show that in the well behaved case of having many samples and no noise all methods have a good performance. However, their performance decrease significantly in case of not having many samples and/or in facing high amount of noise, the typical case for gene expression data. This indicates that information in the gene expression data is not enough to decipher the complex interactions between genes.

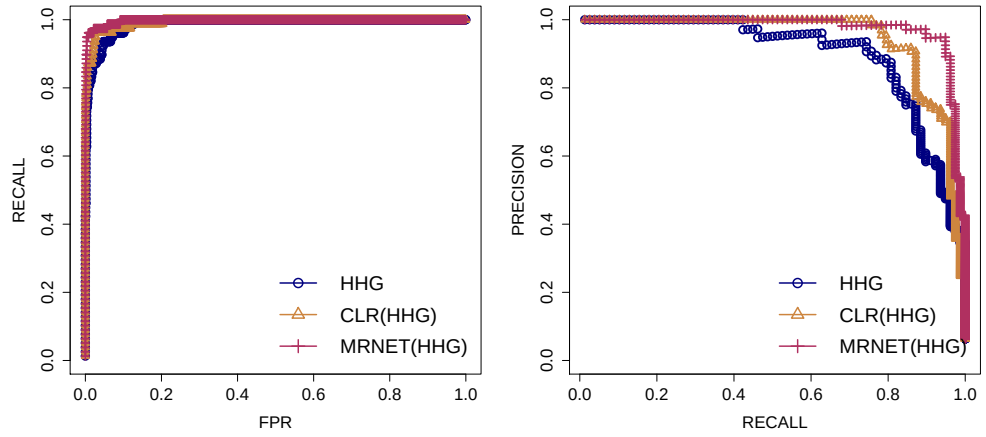
In the context of relevance networks, we compared the performance for different association measures. Among all association measures, correlation and distance correlation had the best performance while the MIC had the worst performance. In addition, for DREAM challenge data distance correlation performs well above other nonlinear associations. In the context of GGMs, in case of not having many samples the regularized method had a better performance compared to the pseudo inverse method, as expected.



(A) Correlation

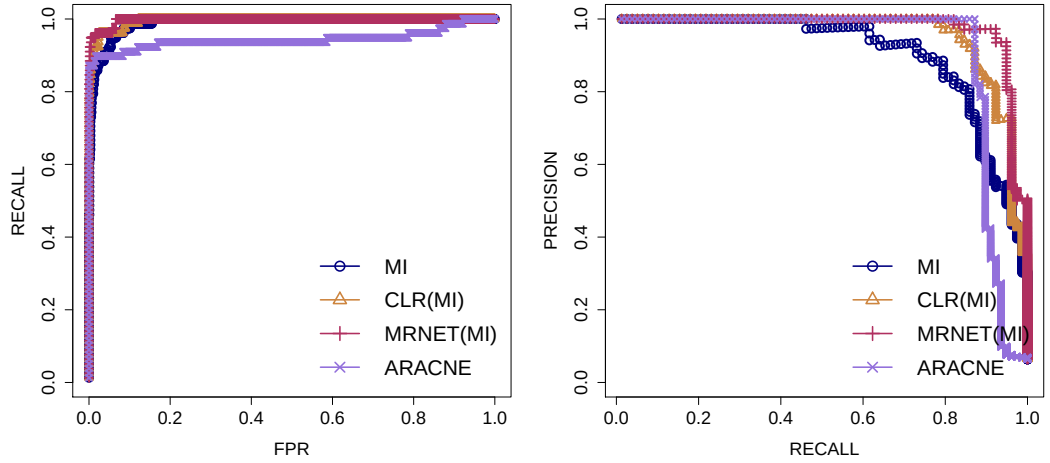


(B) Distance Correlation

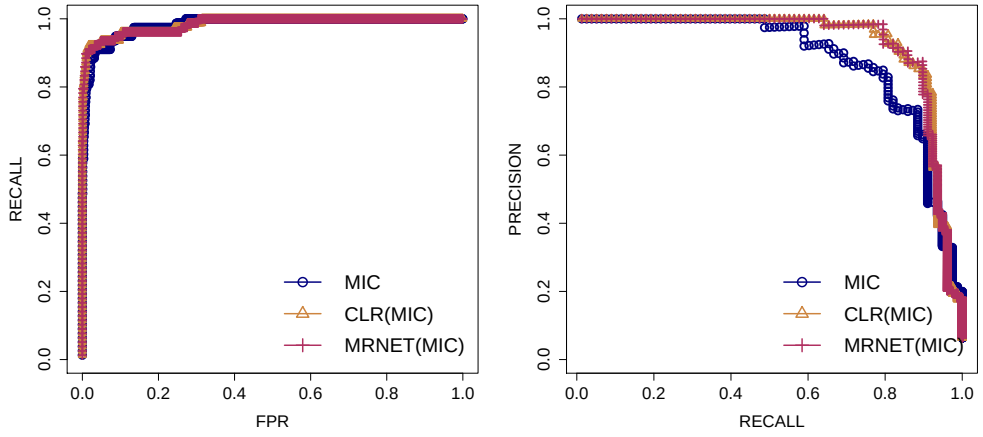


(C) HHG

FIGURE 3.1: **Performance of relevance networks on Gaussian data.** The data consist of 1000 samples simulated from a network with 50 nodes. The left subplots show the ROC curve, while the right subplots show the PR curve.

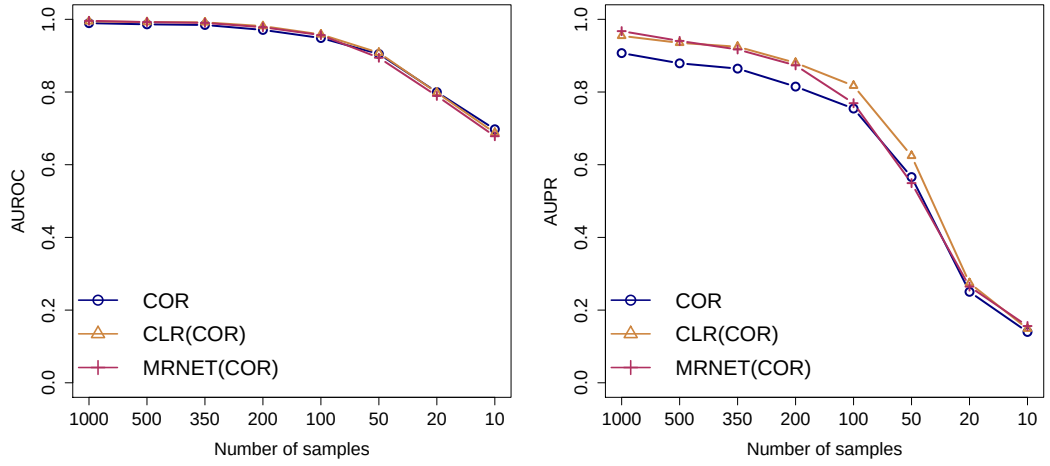


(A) Mutual Information

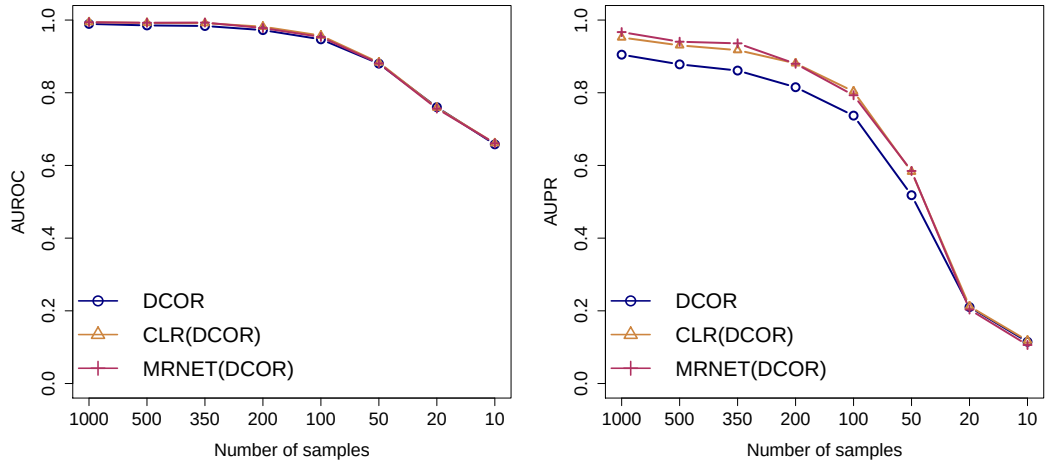


(B) MIC

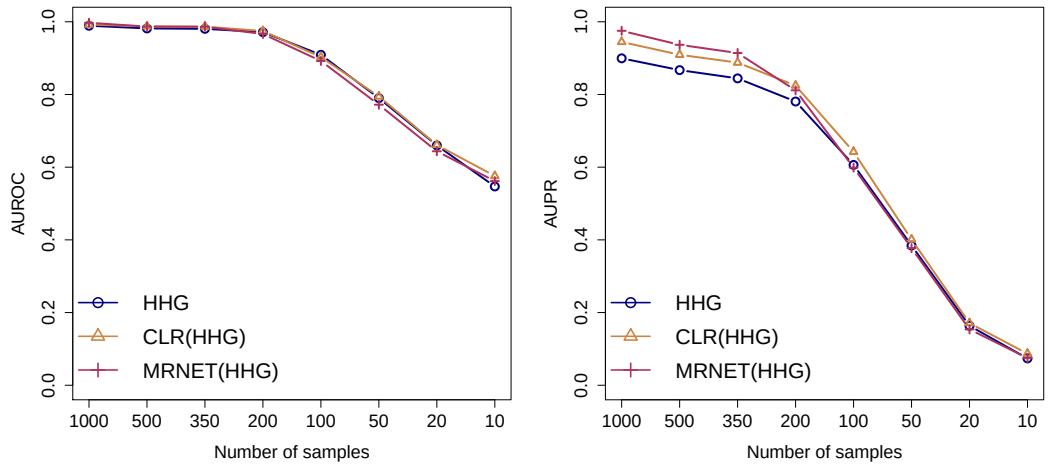
FIGURE 3.2: **Performance of relevance networks on Gaussian data.** The data consist of 1000 samples simulated from a network with 50 nodes. The left subplots show the ROC curve, while the right subplots show the PR curve.



(A) Correlation

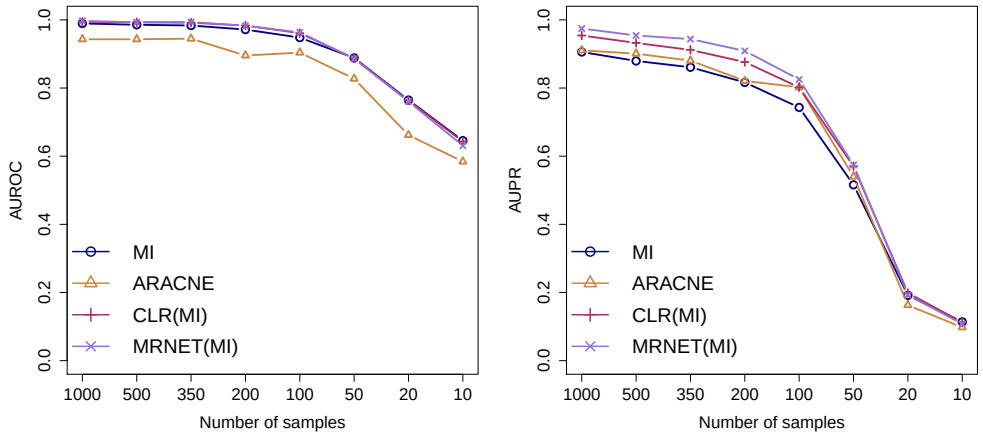


(B) Distance Correlation

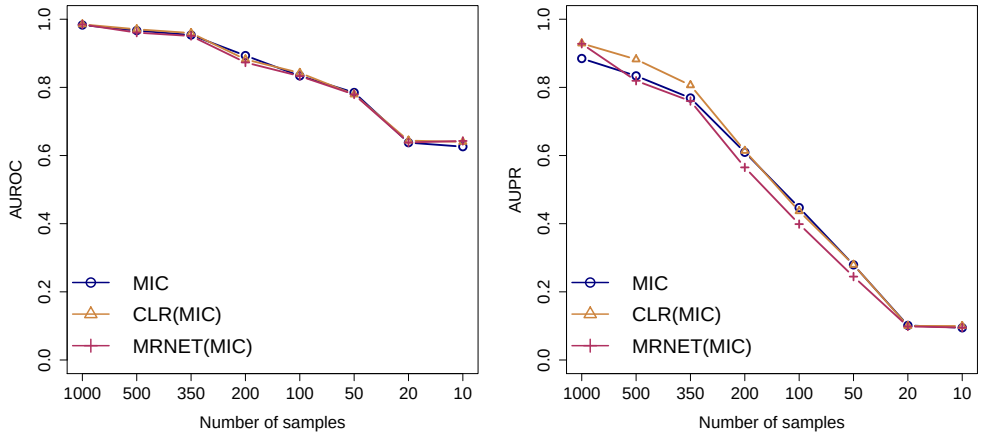


(C) HHG

FIGURE 3.3: **Effect of the number of samples on the performance of relevance networks on Gaussian data.** Different number of samples are simulated from a network with 50 nodes. The left subplots show the AUROC, while the right subplots show the AUPRC.

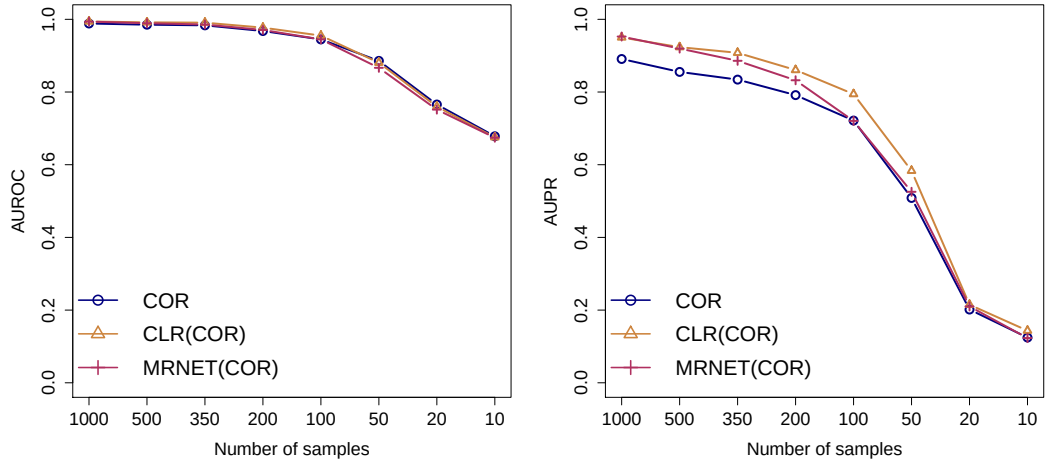


(A) Mutual Information

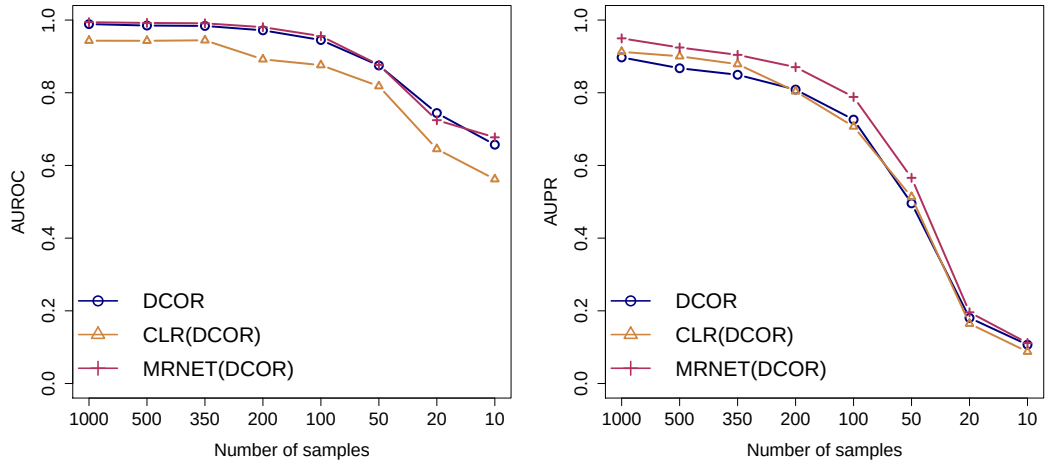


(B) MIC

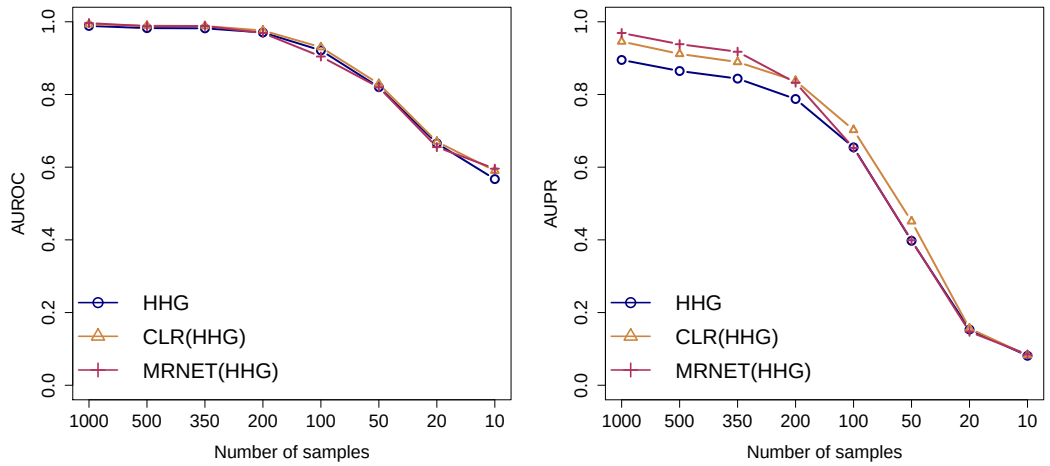
FIGURE 3.4: **Effect of the number of samples on the performance of relevance networks on Gaussian data.** Different number of samples are simulated from a network with 50 nodes. The left subplots show the AUROC, while the right subplots show the AUPRC.



(A) Correlation

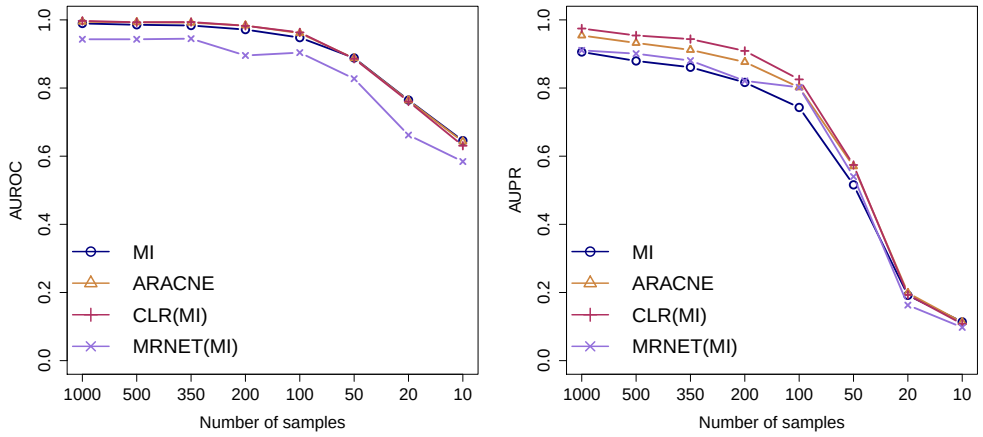


(B) Distance Correlation

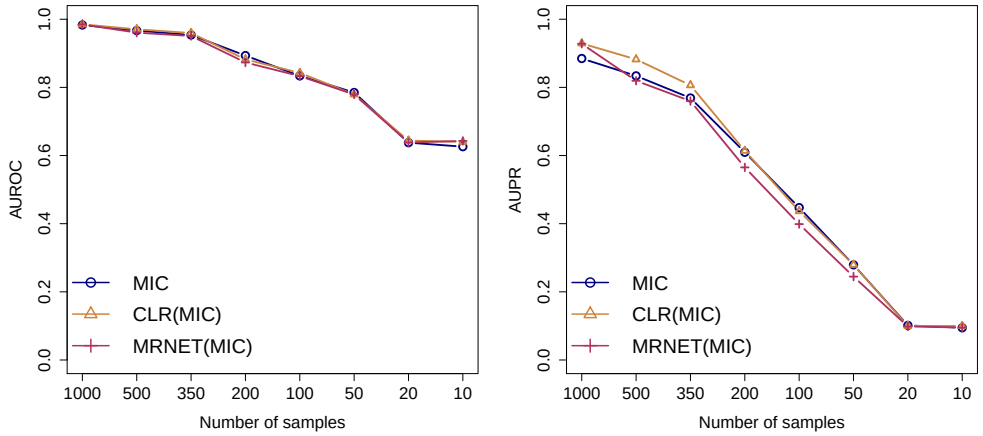


(C) HHG

FIGURE 3.5: **Effect of the number of samples on the performance of relevance networks on non Gaussian data.** Different number of samples are simulated from a network with 50 nodes. The left subplots show the AUROC, while the right subplots show the AUPRC.

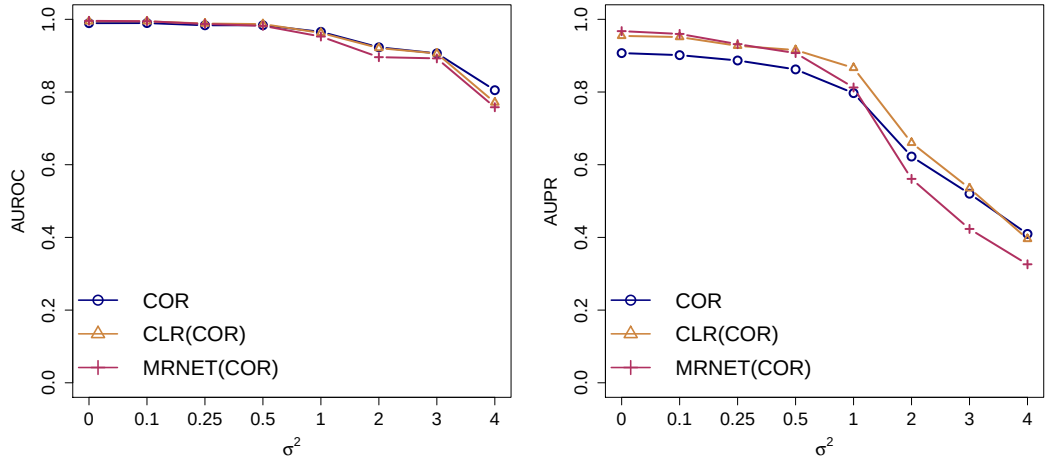


(A) HHG

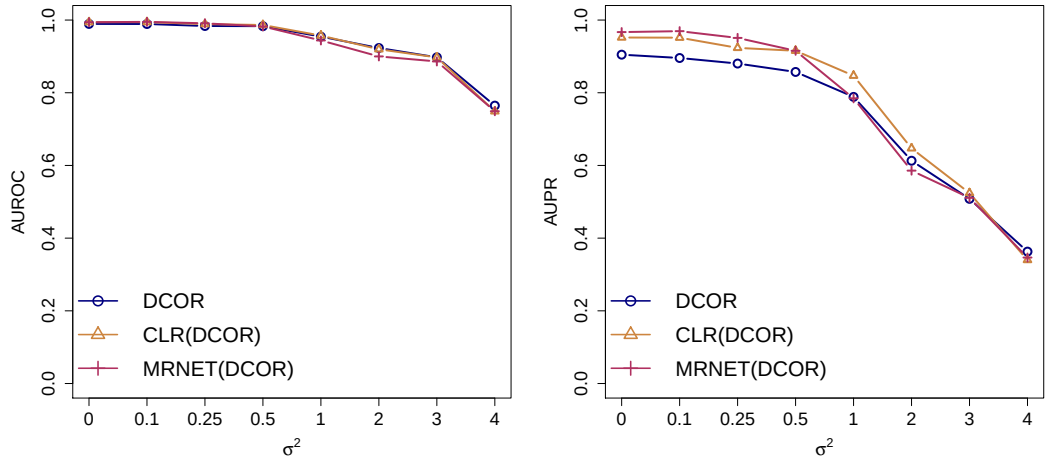


(B) MIC

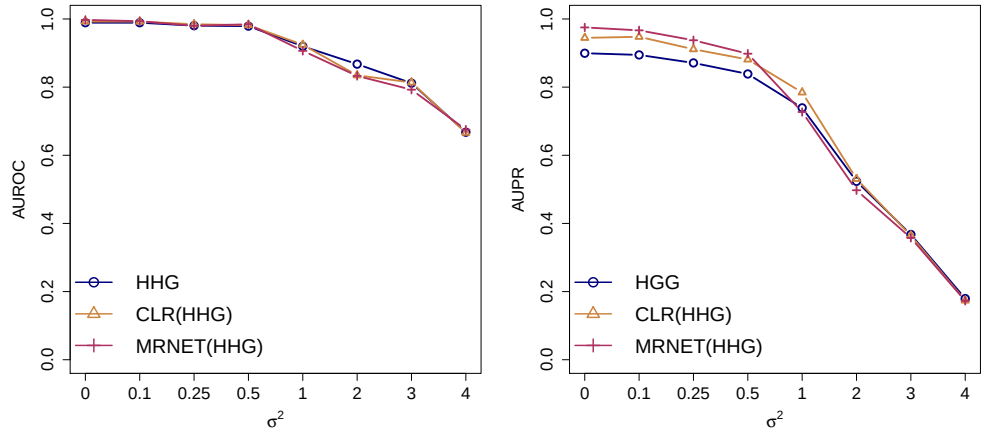
FIGURE 3.6: **Effect of the number of samples on the performance of relevance networks on non Gaussian data.** Different number of samples are simulated from a network with 50 nodes. The left subplots show the AUROC, while the right subplots show the AUPRC.



(A) Correlation

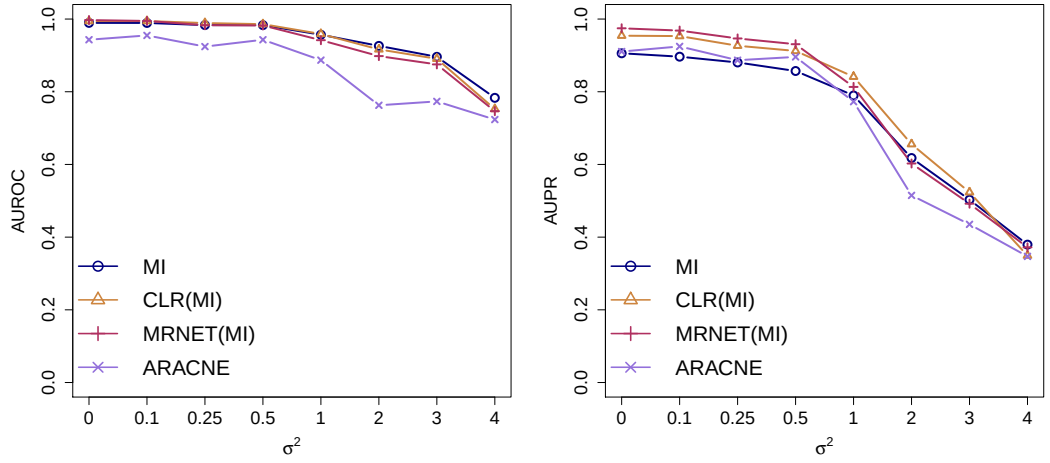


(B) Distance Correlation

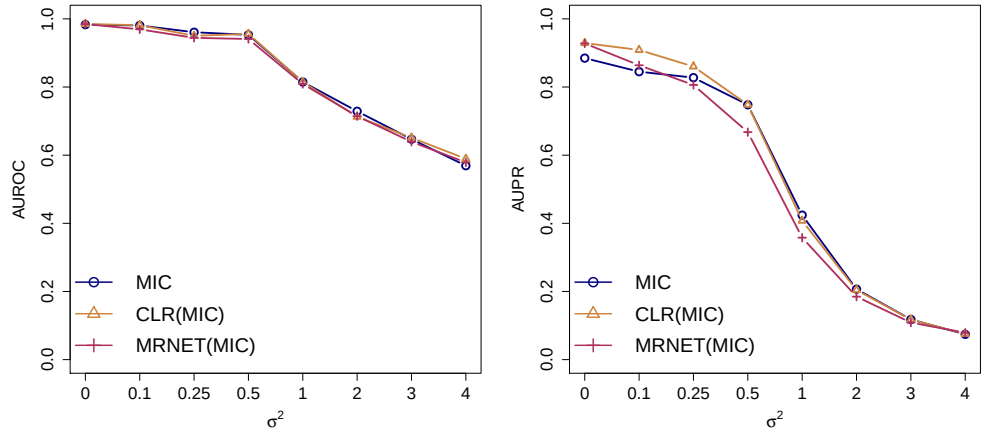


(C) HHG

FIGURE 3.7: **Effect of noise on the performance of relevance networks on Gaussian data.** 1000 samples are simulated from a network with 50 nodes and then Gaussian noise with different amount of variance σ^2 is added. The left subplots show the AUROC, while the right subplots show the AUPRC.

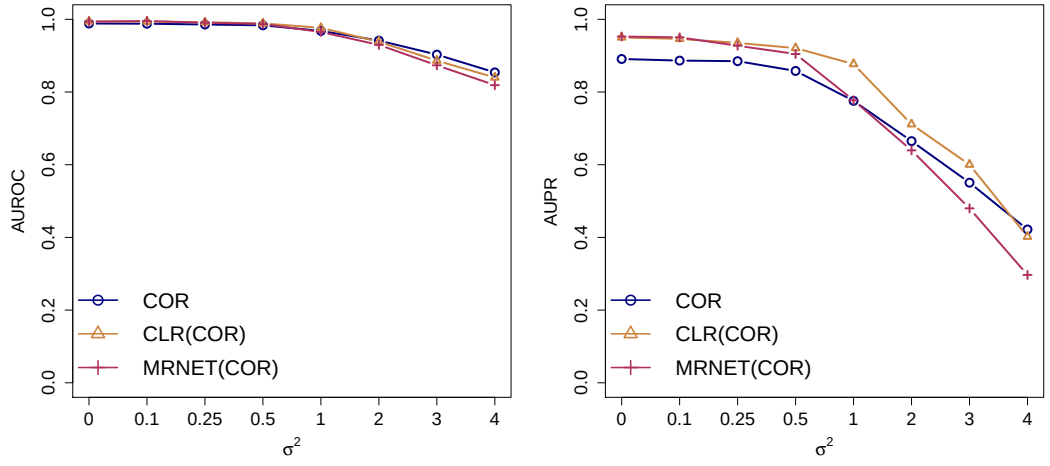


(A) Mutual Information

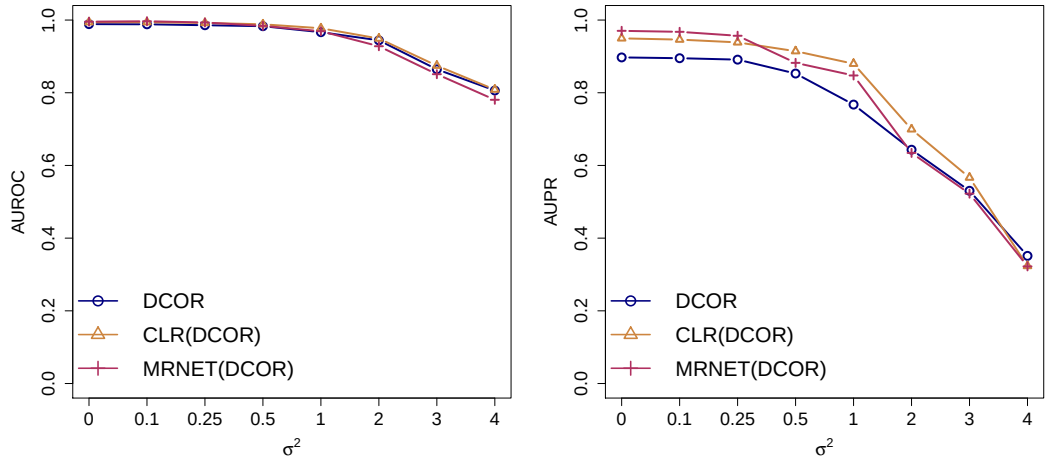


(B) MIC

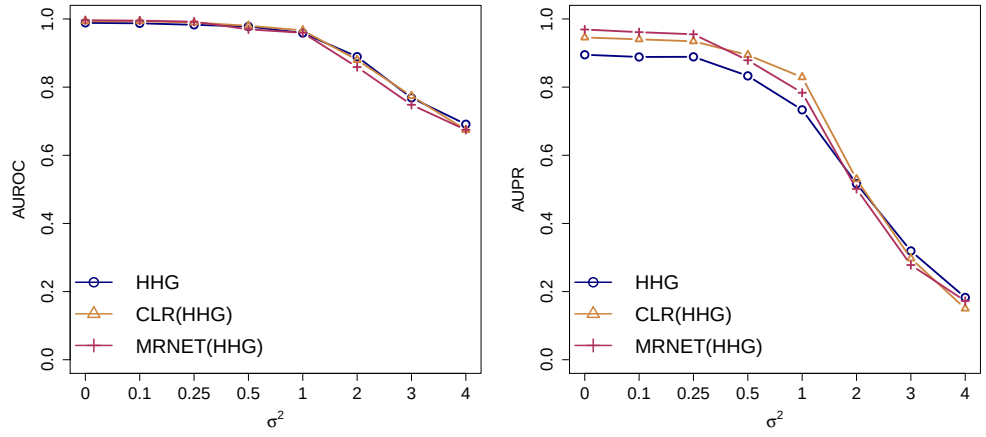
FIGURE 3.8: **Effect of noise on the performance of relevance networks on Gaussian data.** 1000 samples are simulated from a network with 50 nodes and then Gaussian noise with different amount of variance σ^2 is added. The left subplots show the AUROC, while the right subplots show the AUPRC.



(A) Correlation

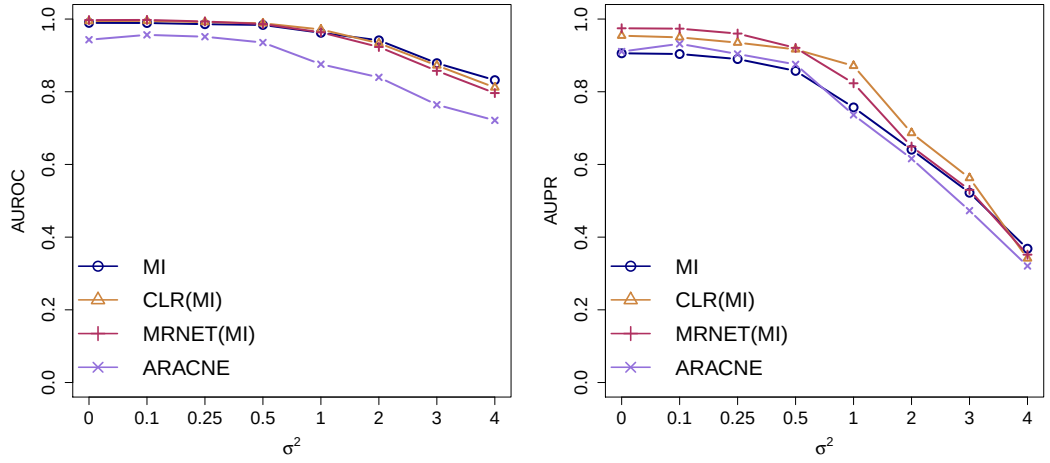


(B) Distance Correlation

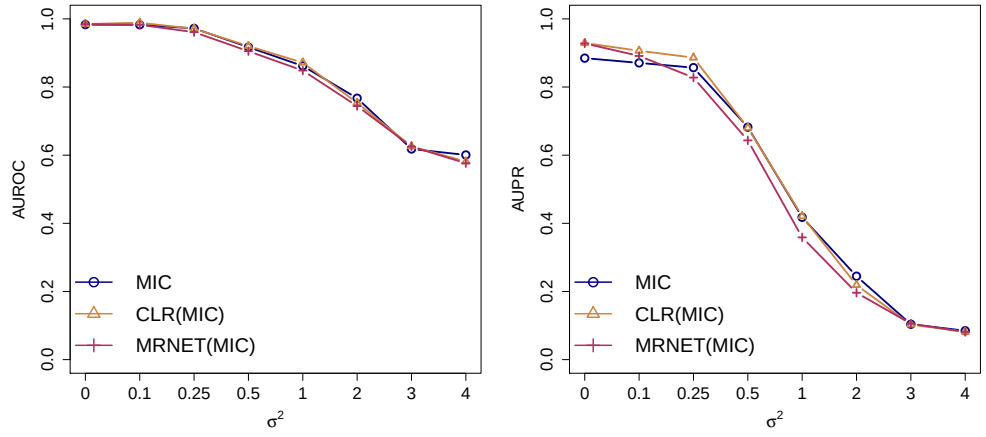


(C) HHG

FIGURE 3.9: **Effect of noise on the performance of relevance networks on non Gaussian data.** 1000 samples are simulated from a network with 50 nodes and then Gaussian noise with different amount of variance σ^2 is added. The left subplots show the AUROC, while the right subplots show the AUPRC.

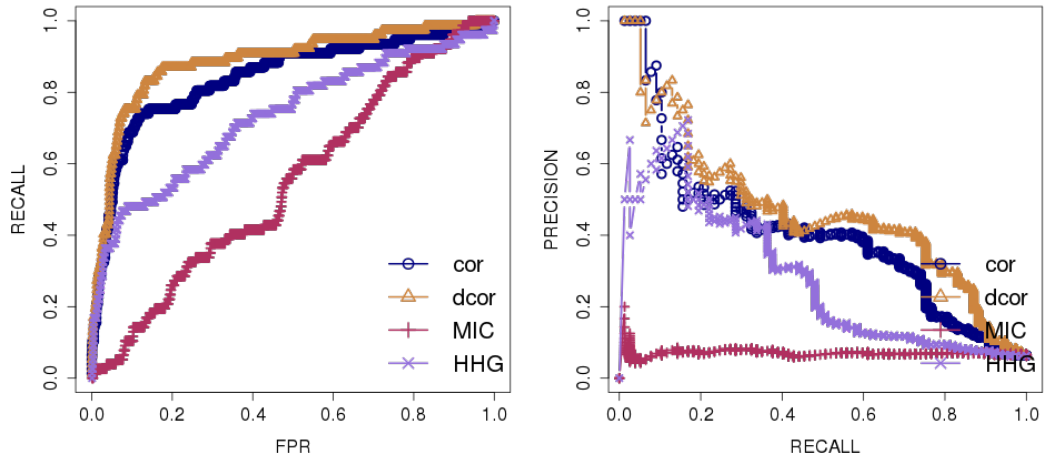


(A) Mutual Information

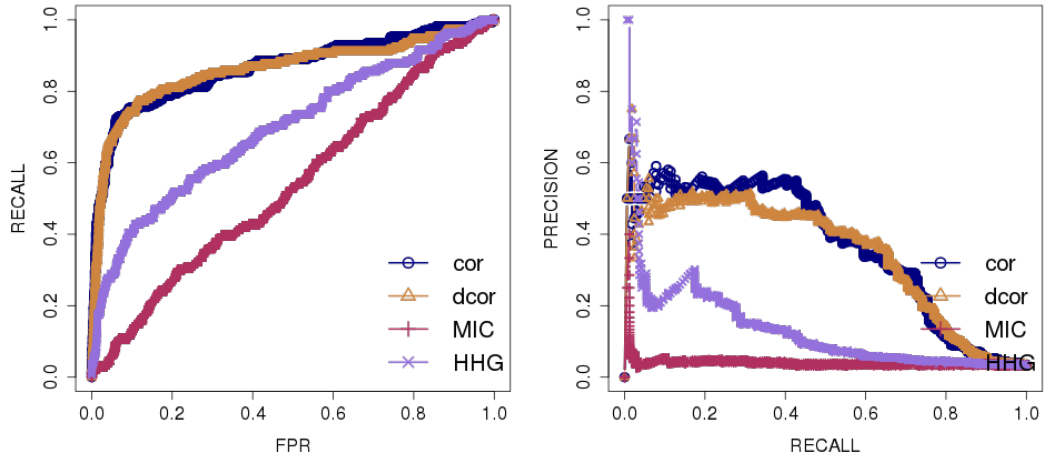


(B) MIC

FIGURE 3.10: **Effect of noise on the performance of relevance networks on non Gaussian data.** 1000 samples are simulated from a network with 50 nodes and then Gaussian noise with different amount of variance σ^2 is added. The left subplots show the AUROC, while the right subplots show the AUPRC.

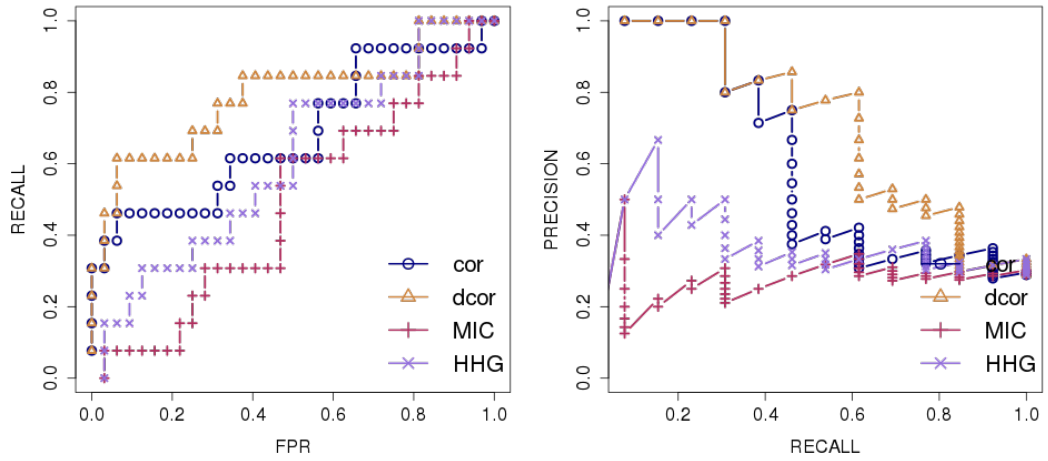


(A) 50 nodes

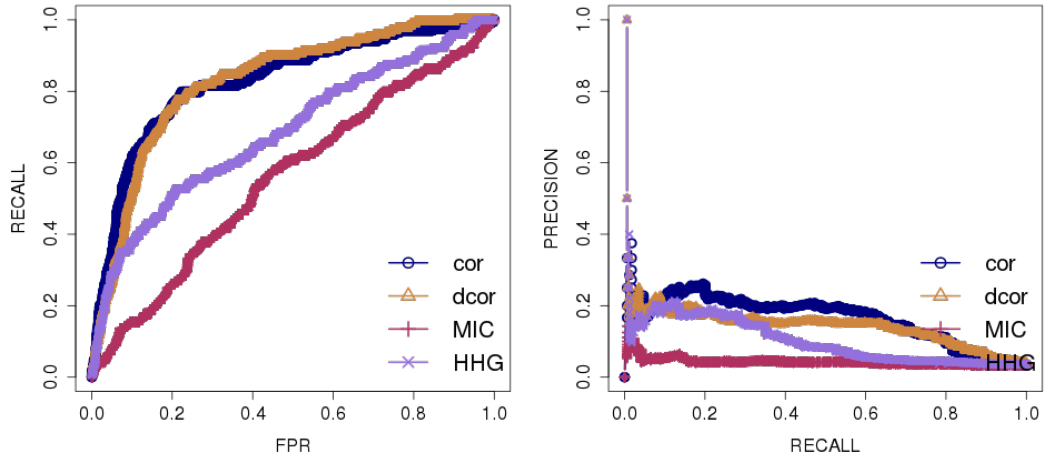


(B) 100 nodes

FIGURE 3.11: **Performance of relevance networks on DREAM3 challenge data.** The left subplots show the ROC curve, while the right subplots show the PR curve.



(A) 10 nodes



(B) 100 nodes

FIGURE 3.12: **Performance of relevance networks on DREAM4 challenge data.** The left subplots show the ROC curve, while the right subplots show the PR curve.

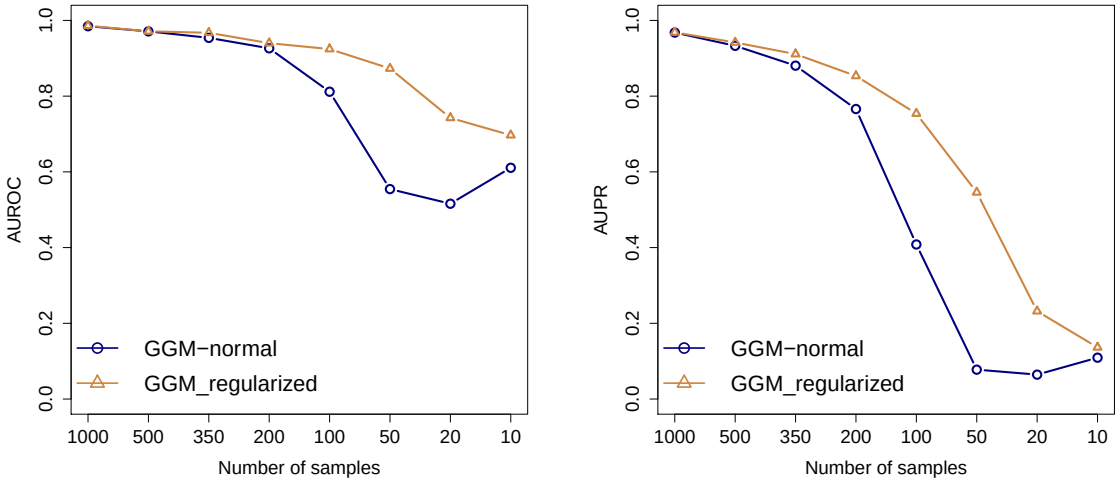


FIGURE 3.13: **Effect of the number of samples on the performance of GGM on Gaussian data.** Different number of samples are simulated from a network with 50 nodes. The left subplots show the AUROC, while the right subplots show the AUPRC.

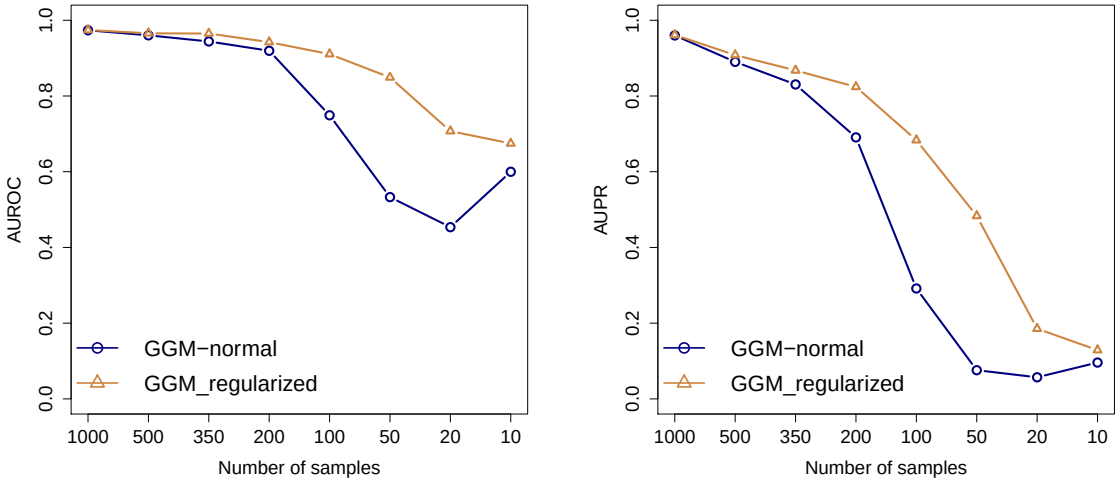


FIGURE 3.14: **Effect of the number of samples on the performance of GGM on non Gaussian data.** Different number of samples are simulated from a network with 50 nodes. The left subplots show the AUROC, while the right subplots show the AUPRC.

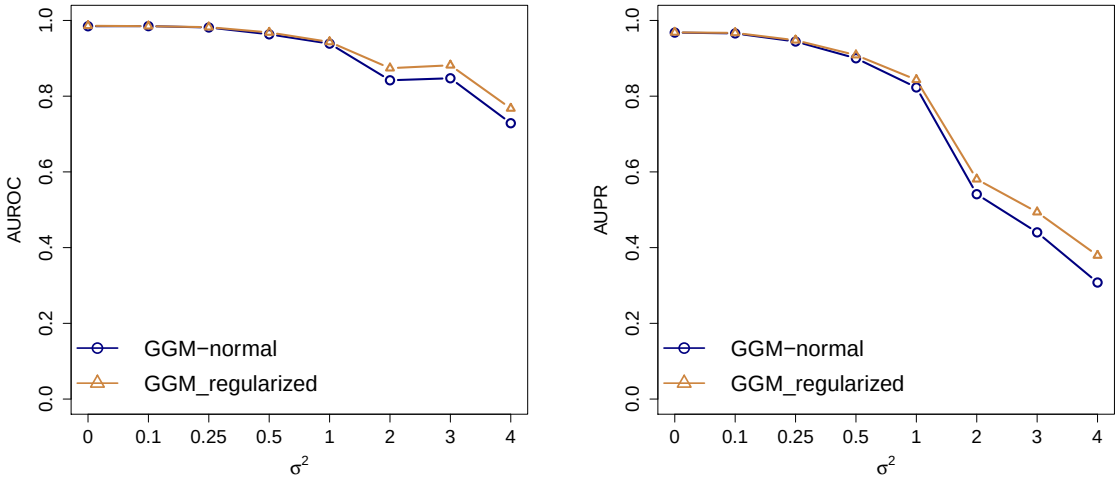


FIGURE 3.15: **Effect of noise on the performance of GGMs on Gaussian data.** 1000 samples are simulated from a network with 50 nodes and then Gaussian noise with different amount of variance σ^2 is added. The left subplots show the AUROC, while the right subplots show the AUPRC.

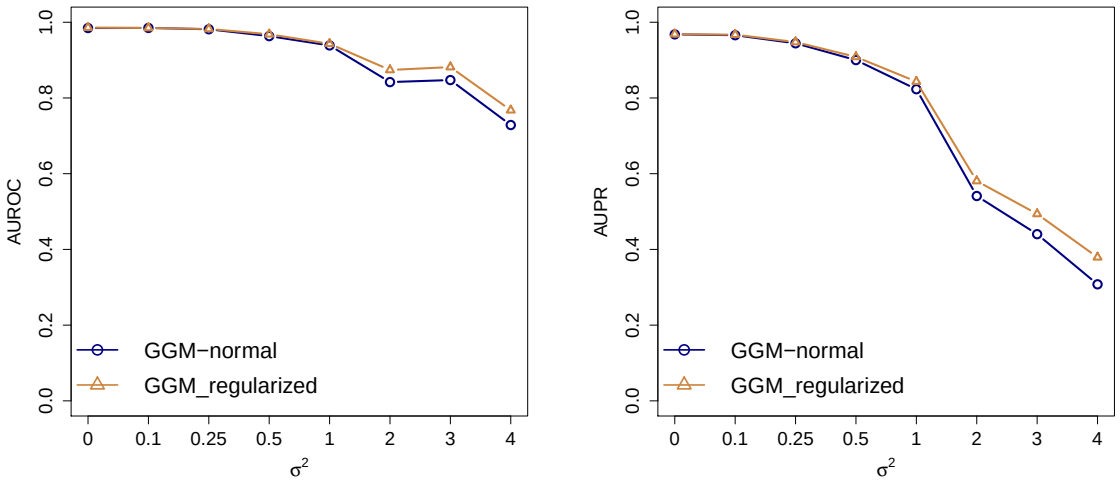


FIGURE 3.16: **Effect of noise on the performance of GGMs on non Gaussian data.** 1000 samples are simulated from a network with 50 nodes and then Gaussian noise with different amount of variance σ^2 is added. The left subplots show the AUROC, while the right subplots show the AUPRC.

Chapter 4

Reconstruction of gene networks using prior knowledge: PriorPC Algorithm

4.1 Introduction

GRN reconstruction from expression data is a challenging problem, not only because it suffers from high dimensionality and low sample size, as the number of genes is generally much larger than the biological samples, but also because biological measurements are extremely noisy. Exploiting other sources of knowledge is one reasonable way to address these issues. Recent advances in biology provide various data sources such as ChIP-seq data, pathway data and sequence data, each of which can shed more light on the cellular processes underlying GRNs. For instance ChIP-seq data can reveal potential target genes for transcription factors (TFs). Each of these sources is of course limited and noisy, and only gives a partial picture of gene regulation. However, taken together, they can help build a more robust description of the regulatory mechanisms, and reduce the effects of noise and sparsity in expression data. These pieces of information can be included in the process of GRN reconstruction in the form of prior knowledge, i.e. a subjective (but non-arbitrary) belief about how the network should look like. Hence, the use of prior information in network inference is a growing trend in computational biology [27–30].

Prior knowledge can be applied by discarding edges that are a priori unwanted, and enforcing edges that are a priori wanted. However we do not always have this level of confidence, particularly in biology where associations are difficult to establish. Another way is to set a prior to 1 when an edge is wanted, and to 0 when an edge is undesirable.

However not all sources of prior knowledge are reliable, and when combining several, there may be inconsistencies to resolve, so potential errors should be accounted for and uncertainty modeled. In addition, not all of the edges have the same level of confidence. For instance when using ChIP-seq data, not all potential target genes for a specific TF have the same probability to be functional. In this case, the binding affinity of TF to TF binding sites is a proper proxy for functionality which can be converted into a probability. We believe that soft priors, which represent the probability of existence of an edge, are better suited for our application.

Prior information about gene interactions in GRNs is typically converted into a prior knowledge matrix B, in which each entry b_{ij} represents the confidence about the existence of an interaction between two nodes X_i and X_j [28], where nodes represent genes. Entries in B range from 0 to 1, where 0 stands for the strongest belief in the absence of an edge and 1 for the strongest belief in the existence of an edge. If no information about the edge between X_i and X_j is available, b_{ij} is set to 0.5. How to include this prior matrix into the reconstruction process depends of course on the algorithm used to construct the GRN.

One of the most popular tools to model GRNs is Bayesian networks (BNs) and most algorithms that allow prior knowledge fall into the class of BNs. Indeed BNs can include prior information very naturally via a prior distribution over network structures. For instance Imoto et al. [27] define a prior distribution on network structures as a Gibbs distribution in which the prior knowledge is encoded via an energy function. Werhli et al. [28] have extended their work to integrate multiple sources of prior knowledge and for each source express the energy function as the absolute difference between the network structure and prior knowledge matrix. However, these algorithms are not applicable for large networks because of their complexity. Some other methods fall into the class of regularized regression where regularization is applied to regression methods to infer a limited number of edges, thereby favoring important ones [29].

The PC algorithm [21] (see section 2.3.3) is a popular constraint-based method which drastically reduces the number of Conditional independence (CI) tests by avoiding unnecessary ones, thereby allowing the reconstruction of larger networks. In fact, it has been shown [31] that PC scales well for sparse graphs and that, in the case where the number of nodes is much larger than the sample size, it is asymptotically consistent for finding the skeleton of a DAG, assuming the data follows a multivariate Gaussian distribution. However, by nature, the performance of PC relies heavily on the accuracy of its inner CI tests, which is not guaranteed in the presence of limited sample size and noisy data. If erroneous decisions are made, the output of PC depends on the order in which the variables are given.

In this work, we exploit the order dependency of PC to our advantage. We modify the original algorithm to include prior knowledge by favoring unwanted edges for early testing, and holding wanted edges out for late testing. The resulting algorithm is referred to as PriorPC. Prior knowledge is particularly advantageous when the quality of the CI tests is questionable, for example as mentioned above when data is high-dimensional and few samples are available, as is typical for gene regulatory networks.

Following Greenfield et al. [29], our method PriorPC is evaluated on one dataset containing *Bacillus subtilis* expression data [32], and two datasets from the DREAM challenge, and for the *E. coli* and *B. subtilis* datasets only the nodes that are linked to at least one other node in the gold standard are considered for evaluation. We compare our result to a recently published work [29] where they modify regression methods to incorporate the prior knowledge.

4.2 Methods

We explained the PC algorithm , or PC, in section 2.3.3. Although in PC algorithm ordering order(V) over the set of variables V determines in which order the CIs should be tested, it has no effect on the output if the CI tests are always correct. The standard choice, used in most implementations, is then the lexicographical ordering. In practice however, CI tests must be performed on the available dataset, containing a limited number of samples for all the nodes in V.

The use of small-sample-sized and noisy datasets (such as biological datasets) in CI tests can induce many false positives and false negatives. Moreover, in the presence of imperfect CI tests, the output of PC also depends on the significance level β, which allows to tune the sparsity of the resulting network but also increases the potential for errors. Because of these inevitable mistakes, edges may be wrongly removed or kept, thereby changing the adjacency structure and affecting the edges that are considered for deletion and the CI tests that are further performed. Therefore, the output of PC does depend on order(V), particularly when the number of nodes is large. This dependency has a cascading effect that can lead to a drastically different skeleton, rendering PC unstable. We use this weakness to our advantage and modify the ordering to include prior knowledge or/and data-based knowledge.

The distribution of the variables is assumed to be a multivariate Gaussian, so CIs can be inferred by testing for zero partial correlation [33]. Let $\text{cor}(X_i, X_j|Y)$ be the sample partial correlation between X_i and X_j given a set $Y \subseteq V \backslash \{X_i, X_j\}$, obtained from any method including regression, inversion of part of the covariance matrix or recursion,

and $z(X_i, X_j|Y) = \frac{1}{2}\log\left(\frac{1+\text{cor}(X_i,X_j|Y)}{1-\text{cor}(X_i,X_j|Y)}\right)$ the Fischer's z-transform. The null hypothesis $\mathcal{H}_0 : \text{cor}(X_i, X_j|Y) = 0$ is then rejected against the two-sided alternative $\mathcal{H}_A : \text{cor}(X_i, X_j|Y) \neq 0$ at significance level β if $|z(X_i, X_j|Y)|\ \sqrt{n - |Y| - 3} > \Phi^{-1}\left(1 - \frac{\beta}{2}\right)$, where Φ denotes the cumulative distribution function of a standard normal distribution [31]. In other word, PC uses the condition $|z(X_i, X_j|Y)|\ \sqrt{n - |Y| - 3} \leq t$ to decide whether $(X_i \perp X_j|Y)$ holds true where $t = \Phi^{-1}\left(1 - \frac{\beta}{2}\right)$. In this work we used the method developed in [2] to estimate the partial correlation since the expression data is high dimension low sample data.

The worst-case complexity of PC is $O(|V|^{maxo})$, where *maxo* is the maximum order reached in the algorithm. If we denote q the maximum number of neighbors of a node in G, then $maxo \in \{q-1, q\}$ [21]. Assuming sparsity, we can set the maximum order to the expected average degree of the network. We will use $q = 5$ for every algorithm presented in the thesis as GRNs are sparse networks.

4.2.1 PriorPC

PriorPC uses order(V) to inject information into the learning process. It first defines a confidence score for each edge representing the initial belief about existence of the edge. If we know a priori that some edges do not exist in the network, removing them in the early stages of the algorithm leads to more reliable neighborhoods and to a better set of CI tests in the rest of the algorithm. Similarly if we know a priori that some edges ought to be part of the network, keeping them as long as possible can lead to different neighborhoods and therefore to a different resulting skeleton. PriorPC uses confidence score to rearrange the CI tests such that edges which are less likely to be a real interaction are considered for CI testing first, while edges with a high belief to belong to the network are subjected to CI testing last. Note that, under the assumption of perfect CI tests, the outputs of PC and PriorPC concur.

4.2.1.1 Including prior knowledge

We introduce a confidence score for each edge indicating the initial belief of existence of the edge which can be simply the prior associated with the edge. However, we do not have prior for all edges and sometimes the prior is not correct and we need the support of data for the edge as well. We define data score d_{ij} as the normalized multiplication of two z-scores resulting from the deviation of the correlation $cor(X_i, X_j)$ from the two distributions of correlations $cor(X_i, .)$ and $cor(X_j, .)$. If $\mathbf{C}$ denotes the absolute

correlation matrix, the unnormalized score is obtained as:

$$e_{ij} = \left| \frac{\mathbf{C}_{ij} - \mu_i}{\sigma_i} \right| \times \left| \frac{\mathbf{C}_{ij} - \mu_j}{\sigma_j} \right|$$

where μ_i and σ_i (resp. μ_j and σ_j) are the mean and standard deviation of the correlation values between X_i (resp. X_j) and all other variables. This is similar to the CLR score (see section 2.2) [17]. The data score is then obtained as:

$$d_{ij} = \frac{e_{ij}}{\sqrt{e_{ii}}\sqrt{e_{jj}}} = \left| \frac{\mathbf{C}_{ij} - \mu_i}{\mathbf{C}_{ii} - \mu_i} \right| \times \left| \frac{\mathbf{C}_{ij} - \mu_j}{\mathbf{C}_{jj} - \mu_j} \right|$$

.

For the data score to be high, the observed correlation between X_i and X_j must be far from the average correlation involving X_i and from the average correlation involving X_j. We now define the confidence score s_{ij} of an edge $X_i - X_j$ as:

$$s_{ij} = \alpha \times b_{ij} + (1 - \alpha) \times d_{ij}$$

where $0 \leq \alpha \leq 1$, b_{ij} is the prior associated with the edge and is directly read from the prior matrix B, and d_{ij} is a data-based score. While b_{ij} encodes our belief in the existence of the edge, d_{ij} indicates how well the edge is supported by the data. To have a high confidence score, an edge must be supported by the prior or the data. Which source matters most depends on α.

4.2.1.2 Discarding the worst edges

Edges are ranked by decreasing confidence score s_{ij}. All edges after the top $N_E \simeq 3 \times |E|$, where $|E|$ is the number of expected edges, are discarded. This number stems from the idea that the network should be sparse, and from the three tier structure of the algorithm developed in the next section. This bold step replaces the zero-order CI tests in PC. Indeed, the zero-order CI tests can also be seen as a deletion step where edges are ordered by decreasing marginal correlation rather than confidence score, and deleted one by one until the CI test reaches the desired threshold. This step is also comparable to a high penalty on the number of edges.

4.2.1.3 3-tier structure

After discarding the worst edges, the remaining N_E edges are divided into three categories. We convert PC into a 3-tier algorithm, where in each tier a specific category of

edges is tested for CIs. We consider the top $\frac{1}{3}$ of N_E edges to be strong candidates, the bottom $\frac{1}{3}$ to be weak candidates, and the remaining $\frac{1}{3}$ to be average candidates. While PC runs all zero-order CI tests for all edges, then proceeds with the first-order CI tests and so on, PriorPC performs all CI tests of order 1 to 5 for all weak candidates first, then for average candidates, and finally for strong candidates.

If the confidence score of a candidate edge and the subsequent group in which it falls is a good indicator, 3-tier PC can remove more false edges, and faster. For instance, if there is a false edge $X_i - X_j$ for which $(X_i \perp X_j | Y = Y_1, Y_2)$ holds true, PC must perform several unnecessary first-order and second-order CI tests before getting to the relevant one. This is not only computationally expensive but also undesirable, because these unnecessary CI tests can cause multiple errors and lead to strong effects as discussed previously. Instead PriorPC removes the worst candidates at the very beginning, and the weak candidates earlier than the other candidates. This also leads to a more reliable neighborhood and CI tests when assessing strong candidates.

4.2.2 Bagging and edge ranking

PC and PriorPC do not naturally provide any score for the edges and therefore do not allow for ranking of the edges. To remedy that issue, we have chosen to apply bagging (see 2.4) and to post-rank the edges by their frequency of appearance. If K-fold bagging is applied to PC for example, K networks are obtained, and an edge can appear any number of times between 0 and K. This number is used to create a ranking a posteriori of the edges and to produce the ROC and PR curves. A consensus network can then be built by choosing a threshold and selecting the top edges only. Apart from the ranking, this step also has the advantage to produce more reliable results, as the consensus network contains less noise, regardless of the algorithm used.

We set K to 20 for all experiments. One could use the confidence score of edges to break the ties, however we rank them lexicographically. Note that, to produce a network in the first place, a threshold for the CI tests is required. As detailed in 4.3.8, this threshold was fixed to 0.1 for all experiments and optimized neither for PriorPC nor for each data set.

4.2.3 Synthetic prior knowledge

For each experiment and for each dataset, the prior information matrix B is simulated from the gold standard network available depending on the needs. To assign a true prior to an edge $X_i - X_j$, we check the existence of that edge in the gold standard network.

If the edge is present, the prior b_{ij} is randomly sampled from $(0.5, 1]$, otherwise b_{ij} is randomly sampled from $[0, 0.5)$. To assign a non-informative prior to $X_i - X_j$, b_{ij} is set to 0.5.

4.3 Results

4.3.1 Datasets

For the evaluation of the PriorPC, we used three different datasets. Two of them are from DREAM challenge. In the following, we explain the three datasets in more detail. Note, each dataset contains both time-series data and steady state data and we only use the steady state data. We used the gold standard of each data set to synthesize prior knowledge.

- A synthetic dataset from the DREAM4 competition [23, 25, 26]. The data consists of 100 genes where any gene can be a regulator. The gold standard contains 176 interactions. The normalization was done by the DREAM organizers.
- A real dataset from the DREAM5 competition [23]. The data includes a compendium of microarray experiments measuring the expression levels of 4,511 *E. coli* genes (344 of which are known transcription factors) under 805 different experimental conditions. Normalization was done using RMA [34]. DREAM5 challenge also provides a gold standard consisting of 2,066 established gene regulatory interactions.
- A set of 269 expression measurements of *B.subtilis* genes in response to a variety of conditions [32]. Greenfield et al. [29] normalized the data and compilated the overlapping probes into intensities and we used the data provided by them. The gold standard comes from SubtiWiki [35, 36] which is repository of information for *B.subtilis* contains 2422 interactions.

Note that PC is not feasible for large networks with a small threshold for the CI tests and we compare PriorPC to PC-lite. PC-lite is a variation of PC that removes edges with low correlation and keeps the N_E edges with the highest correlation instead of doing zero-order tests (the step of discarding the worst edges of PriorPC), and then applies PC to these edges only. As it is shown in 4.3.2 and 4.3.8 , PC-lite always outperforms PC. We set N_E to 600, 7000, 7000 for DREAM4, *E. coli* and *B.subtilis* respectively.

4.3.2 From PC to PriorPC

In this section, the results of the various steps taken between PC and PriorPC in order to see the effect of each step. Note that, in this section, bagging was not used because it would take too long for PC.

- The first step is to rank the edges based on their correlation and to use this ordering instead of the lexicographic ordering in PC. We refer to this algorithm as OPC (ordered PC).
- The second step is to remove edges with low correlation and keep the N_E edges with the highest correlation instead of doing zeroth-order tests, and apply PC to these edges only. We refer to this algorithm as PC-lite.
- Finally, we refer to the combination of OPC and PC-lite as OPC-lite.

In this experiment, the prior matrix B contains only true priors, i.e. priors sampled in $(0.5, 1]$ for present interactions and in $[0, 0.5)$ for absent interactions.

PC, OPC, PC-lite, OPC-lite and PriorPC with $\alpha = 0$ and $\alpha = 0.25$ were applied to the three datasets, all using the same threshold for the CI tests. Table 4.1 compares the number of true positives (TP), the number of false positives (FP) and the F1 scores obtained for each algorithm for a CI threshold of 0.1. The results show that the most effective steps are to discard the worst edges at the very beginning and to include prior knowledge.

The difference between PC and OPC lies strictly in how edges are ordered: by lexicographical order or correlation. Results are comparable on DREAM4 data, but OPC clearly wins for two datasets (*E. coli* and *B.subtilis*). Similarly, the difference between PC-lite, OPC-lite and PriorPC with $\alpha = 0$ lies (mostly but not strictly as PriorPC also has a tier-structure) in how edges are ordered: by lexicographical order, marginal correlation or data score d_{ij}. Results are comparable, with an advantage for PriorPC with $\alpha = 0$ on *B.subtilis* data.

The difference between PC and PC-lite, or between OPC and OPC-lite, lies strictly in the removal of the worst edges at the very beginning. The results are much better for PC-lite compared to PC, and for OPC-lite compared to OPC, in all three datasets.

The difference between PriorPC with $\alpha = 0$ and PriorPC with $\alpha = 0.25$ lies strictly in the use of prior knowledge (only with a coefficient of 0.25). Although PriorPC with $\alpha = 0$ was already consistently better than PC, OPC, PC-lite and OPC-lite, the results are further greatly improved with $\alpha = 0.25$, on all three datasets.

This section suggests that PriorPC wins on two aspects: the removal of the worst edges as a first step, and the use of prior knowledge. The first aspect is a nice result: it allows for faster processing and for larger networks, and clearly not at the expense of the accuracy. In fact removing these edges even helps the algorithm perform better tests. The second aspect shows that our inclusion of prior knowledge helps greatly, even when given a low weight.

	DREAM4			*E. coli*			*B.subtilis*		
	TP	FP	F1	TP	FP	F1	TP	FP	F1
PC	86	678	0.18	314	5592	0.07	592	6725	0.12
PC-lite	80	206	0.35	127	369	0.09	361	1154	0.18
OPC	86	676	0.18	284	4534	0.08	561	5602	0.13
OPC-lite	79	208	0.34	126	366	0.09	342	1056	0.17
PriorPC α=0	84	222	0.35	146	469	0.11	397	1199	0.19
PriorPC α=0.25	100	188	0.43	194	232	0.15	492	810	0.26

TABLE 4.1: **From PC to PriorPC**. Effect of all the various steps between PC and PriorPC. None of the methods were subjected to bagging. For PriorPC, all edges have a true prior. Two steps make a critical difference: using prior knowledge to rank the edges, and discarding straight away the worst edges.

4.3.3 Effect of the parameter α

The value of α determines the degree of influence of the prior knowledge in the ranking of the edges. While $\alpha = 1$ means ranking the edges using prior knowledge only, $\alpha = 0$ means using data only. Figure 4.1 shows the performance of PriorPC for different values of α. In this experiment, the prior matrix B contains only true priors, i.e. priors sampled in $(0.5, 1]$ for present interactions and in $[0, 0.5)$ for absent interactions.

PriorPC performs well above PC-lite, even though priors were simply sampled between [0,0.5) or (0.5,1]. Increasing the value of α leads to a better performance. This indicates that not all of the edges are well supported by the data and therefore increasing the effect of the prior improves the algorithm. This also emphasizes the value of integrating prior knowledge where data is sparse and noisy.

Note that PriorPC with $\alpha = 1$ does not perform perfectly. Indeed, the prior is used to reorder the CI tests, but it has no effect on the CI tests themselves. Therefore, it is not possible to reconstruct the real network unless data supports it.

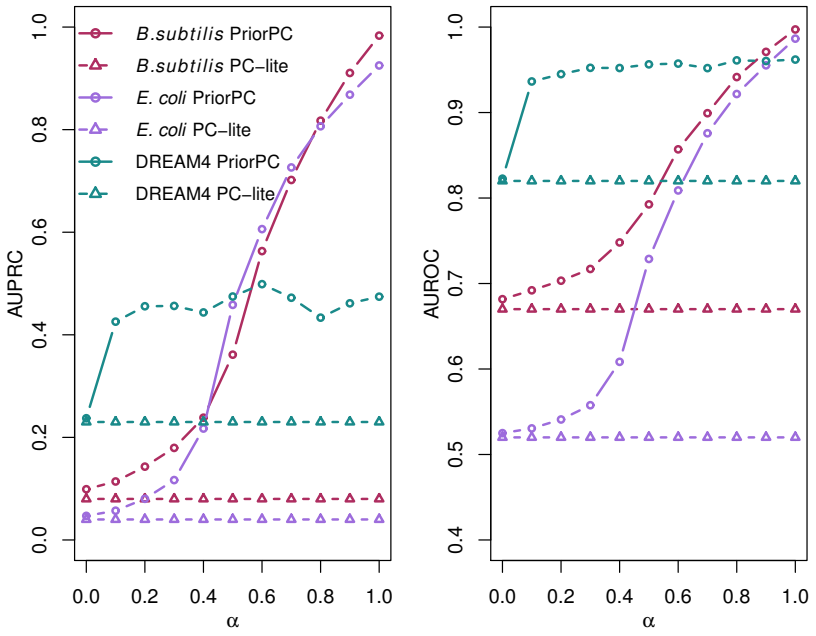


FIGURE 4.1: **Performance of PriorPC against α.** The left subplot shows AUPRC, while the right subplot shows the AUROC. PC-lite is plotted with triangles, while PriorPC is plotted with circles. The different colors represent the different datasets. For PriorPC, all edges have a true prior. PriorPC outperforms PC-lite and its performance increases with α.

4.3.4 Effect of the amount of prior knowledge

In order to assess the effect of the prior on the resulting network, the algorithm was given different amounts of prior knowledge. Initially, 5% of the edges were randomly selected and assigned a true prior as stated in section 4.2.3. For all other edges, the prior was set to 0.5. The percentage of the edges with a true prior was then gradually increased until it reached 100%. Figure 4.2 shows the results for $\alpha = 1$.

Here again, PriorPC performs well above PC-lite, even though priors were simply sampled between [0,0.5) or (0.5,1]. For each dataset, the more prior is included, the better the network can be recovered. This indicates that PriorPC is consistent.

4.3.5 Effect of the prior knowledge on the edges without prior

The prior, even if it is incomplete and only concerns a few edges, may influence the complete network. We refer to the edges that do not have a prior as neutral edges. To assess the influence of the prior on neutral edges, 5% of the edges were randomly sampled and assigned a true prior. This experiment was repeated for increasing percentages, until 80% edges were selected. The results were then compared with PC-lite but this time

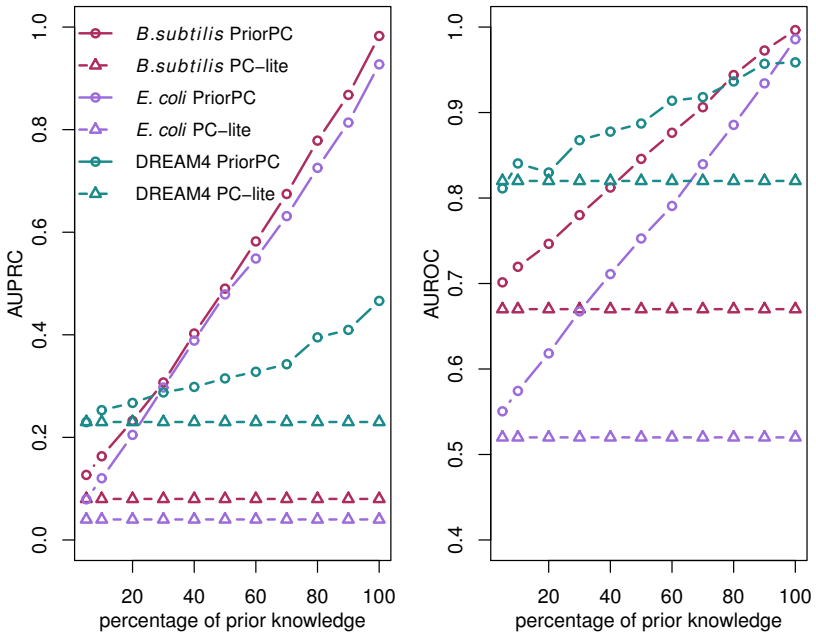


FIGURE 4.2: **Performance of PriorPC against the percentage of edges with a prior.** The left subplot shows the AUPRC, while the right subplot shows the AUROC. PC-lite is plotted with triangles, while PriorPC is plotted with circles. The different colors represent the different datasets. PriorPC outperforms PC-lite and its performance increases with the percentage of edges with a true prior.

separately for the neutral edges and for the edges with prior. Figure 4.3 shows the results for $\alpha = 1$.

The results show that for real data, parts of the network which are not subjected to the prior do not suffer from the prior. For DREAM4 data, using a high amount of prior leads to a performance decrease on the neutral edges, it is unclear why. The rest of the time, the performance is just as good as that of PC-lite.

4.3.6 Robustness to erroneous priors

Biological prior knowledge can come from different sources including ChIP-seq data, protein-protein interaction data and literature, which can all contain false information. Methods for integrating prior knowledge should therefore be robust to errors.

In order to assess the robustness of the algorithms to erroneous prior information, a true prior was assigned to all edges, and Gaussian noise was added (towards 0 for true edges, towards 1 for non-edges), with various standard deviations σ. Clearly, the effect of the amount of noise (σ) depends on the value of α. Figure 4.4 shows the effect of noise on the AUPRC and the AUROC for different values of α. The results indicate that PriorPC is robust to reasonable amounts of noise. Clearly, the higher the amount

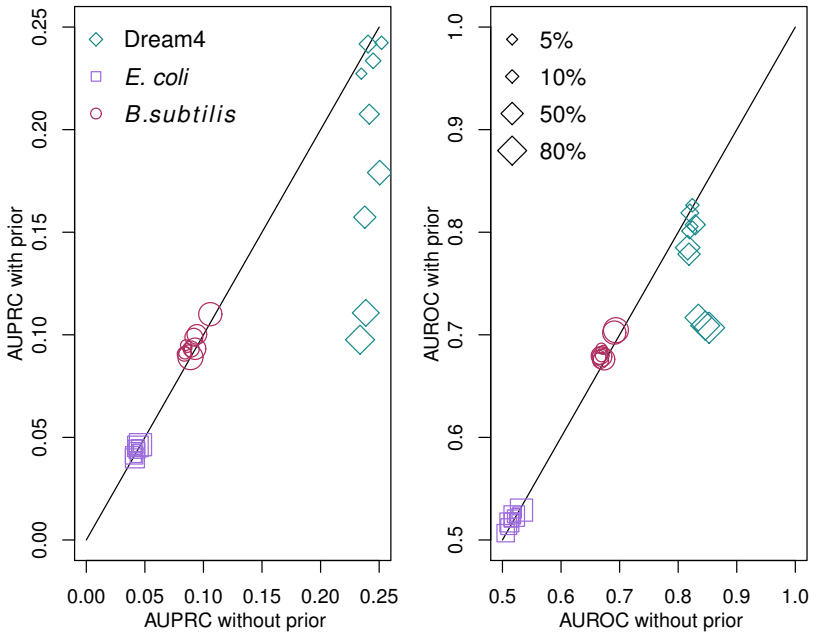
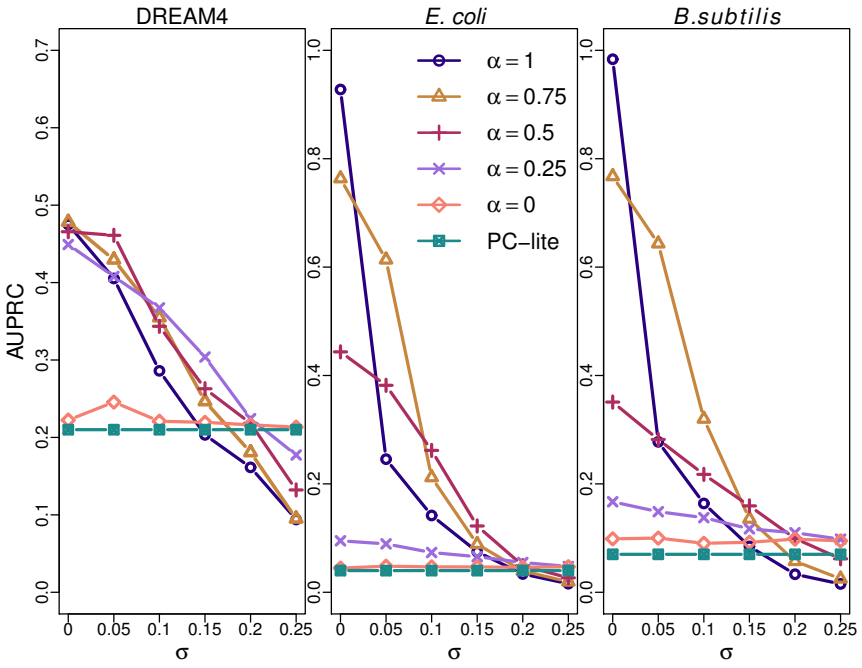


FIGURE 4.3: **Comparison between PC-lite and PriorPC on neutral edges.** Neutral edges are edges which are not subjected to prior knowledge. The left subplot shows the AUPRC, while the right subplot shows the AUROC. The x-axis shows the performance of PC-lite, the y-axis the performance of PriorPC. Each datapoint corresponds to a different amount of edges with a true prior from 5% to 80%. For PriorPC, $\alpha = 1$. Results are comparable, overall neutral edges are not negatively affected by the prior.

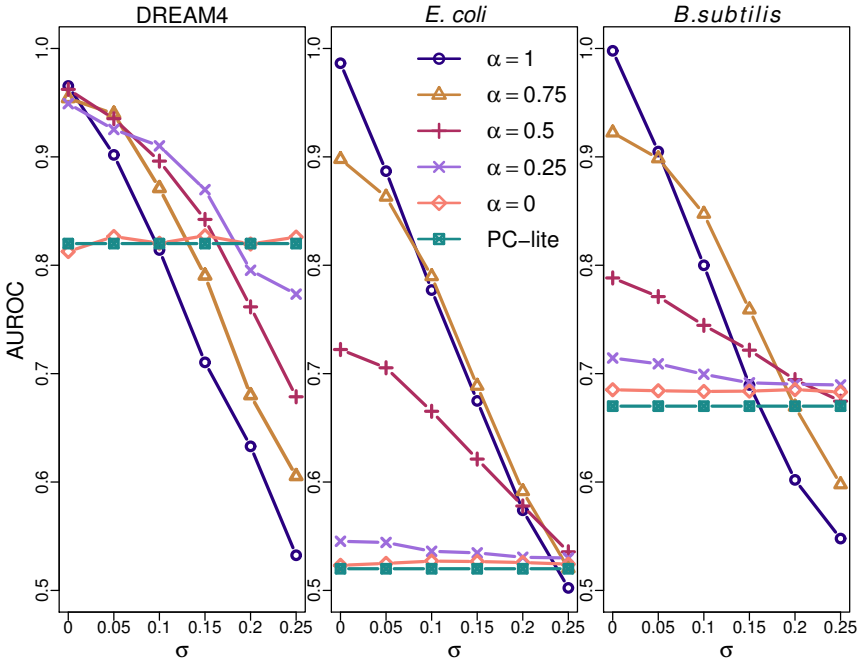
of noise, the worse the performance. Naturally, the results are less sensitive to noise for smaller values of α, which should be taken into account when choosing this parameter. Indeed, when α is small, PriorPC is still better than PC.

In addition, we followed the experimental set up given in [29] to assess the robustness of the algorithms to erroneous prior information. 50% of the true edges were randomly selected and assigned a random prior higher than 0.5. Then m edges from the remaining edges in the prior matrix were selected and their corresponding true prior values were flipped so that $b_{ij} = 1 - b_{ij}$ to introduce errors. Figure 4.5 shows the AUPRC and AUROC results for different ratio of true priors to false priors.

The results indicate that PriorPC is robust to reasonable amounts of error. Clearly, the higher the percentage of false priors, the worse the performance. Naturally, the results are less sensitive to errors for smaller values of α, which should be taken into account when choosing this parameter. Indeed, when α is small, PriorPC is still better than PC.



(A) AUROC



(B) AUPRC

FIGURE 4.4: **Performance of PriorPC against σ for various αs.** The first row shows the AUROC, while the second shows the AUPRC. All edges have a true prior. Gaussian noise is added to all priors with various standard deviations σ. The different colors represent the result for various αs . The performance of PC is plotted in green and with full squares for comparison. For small standard deviations, PriorPC performs better than PC-lite. This effect is not seen for large standard deviations since most priors are flipped.

4.3.7 Comparison of PriorPC to MEN and BBSR

Recently published work [29] suggests two methods to use prior knowledge. The first method called MEN (Modified Elastic NET) is a modification of Elastic Net where prior knowledge is expressed as a modifier of the $l1$ constraint incurred on each single regression coefficient. This leads to less shrinkage on the regression coefficient corresponding to a putative regulation.

The second method called BBSR (Bayesian best subset regression) is based on Bayesian regression with a modification of Zellner's g prior. In this framework the prior on the regression coefficients follows a multivariate Gaussian distribution centered at an initial guess with the empirical covariance matrix that is scaled by a chosen factor g, where g encodes the belief about the initial guess. They extend the original formulation of g and define a vector with one entry per predictor to allow for different levels of confidence for different entries in the initial guess. They use a criterion based on Bayesian Information Criterion (BIC) to select the final model. For both methods, bootstrapping is applied in order to provide a final ranking of the edges.

BBSR and MEN take as input both steady state data and time series data and the output is a matrix with confidence level for directed edges. For a fair comparison we just take the skeleton and assign the highest confidence of corresponding directed edges to undirected edge.

The prior used in BBSR and MEN is not probabilistic, instead it is a hard score stating the strength of belief in the presence of an edge, with 1 for belief and 0 for no belief (no belief in the sense of no opinion, which is similar to a probability of 0.5). The score 1 is assigned to the edges found in the gold standard network only. The rest of the edges are assigned the score 0. The two methods are compared with their respective core methods and with state-of-the-art algorithms which do not contain any prior information. In each case, the inclusion of prior knowledge improves the accuracy of the inferred network.

We compare PriorPC to these two methods. Tables 4.2 and 4.3 show the AUPRC and AUROC results, respectively, from MEN and BBSR for different (default) parameters corresponding to the low and high use of prior as well as the results of PriorPC for two different values of α. For the sake of comparison, we followed Greenfield et al. [29]: 50% of the true interactions in the gold standard network are selected and assigned a true prior (1 for MEN and BBSR, a random probability in $(0.5, 1]$ for PriorPC).

The results show that on average PriorPC performs as well as BBSR and MEN even without the use of time-series (TS) data and merely using soft prior. Note that none of PriorPC's parameters were tuned. PriorPC is also fast and one bootstrap takes 1:08,

	DREAM4	*E. coli*	*B.subtilis*	Using TS
MEN_low	0.48	0.201	0.218	yes
MEN_high	0.571	0.347	0.369	yes
BBSR_low	0.44	0.196	0.269	yes
BBSR_high	0.519	0.359	0.394	yes
PriorPC ($\alpha = 1$)	0.328	0.413	0.392	no
PriorPC ($\alpha = 0.75$)	0.341	0.336	0.303	no

TABLE 4.2: **Comparison of MEN, BBSR and PriorPC in terms of AUPRC.** For all three methods, 50% of the edges present in the gold standard network were randomly selected and assigned a true prior (1 for MEN and BBSR, a random probability in $(0.5, 1]$ for PriorPC). For PriorPC, α is given in brackets. MEN and BBSR also use time-series(TS) data. Results are comparable across the three algorithms.

	DREAM4	*E. coli*	*B.subtilis*	Using TS
MEN_low	0.908	0.768	0.828	yes
MEN_high	0.912	0.776	0.842	yes
BBSR_low	0.872	0.675	0.791	yes
BBSR_high	0.86	0.719	0.793	yes
PriorPC ($\alpha = 1$)	0.887	0.753	0.835	no
PriorPC ($\alpha = 0.75$)	0.885	0.71	0.801	no

TABLE 4.3: **Comparison of MEN, BBSR and PriorPC in terms of AUROC.** For all three methods, 50% of the edges present in the gold standard network were randomly selected and assigned a true prior (1 for MEN and BBSR, a random probability in $(0.5, 1]$ for PriorPC). For PriorPC, α is given in brackets. MEN and BBSR also use time-series(TS) data. Results are comparable across the three algorithms.

39:34, 6:01 minutes for DREAM4, *E. coli* and *B. subtilis* respectively, when $\alpha = 1$ (3.1GHz Intel Core).

4.3.8 Threshold for conditional independence test

Both PC and PriorPC require a threshold t for the CI tests. Since we apply bagging and use the frequency of occurrence to rank the edges, we choose t relatively small to have a graph contains a large number of edges. In this way, we keep more edges to compete during the bagging. However, the threshold should not be too small which leads to computational singularity and require a lot of time or simply not be feasible.

In order to see the effect of the threshold t, we apply PC, PC-lite and PriorPC (with different values of α for incorporation of prior knowledge) for a variety of thresholds. Figure 4.6 shows the ROC curve and PR curve for Dream4 data, when we change the threshold t, $t \in \{0.001, 0.01, 0.1, 0.5, 1, 2\}$. The result shows that PC-lite and PriorPC with any value of α outperform the PC and this improvement is not the effect of bagging.

We did not tune the parameter t for each data set and set the parameter to 0.1 for all datasets.

4.4 Conclusion

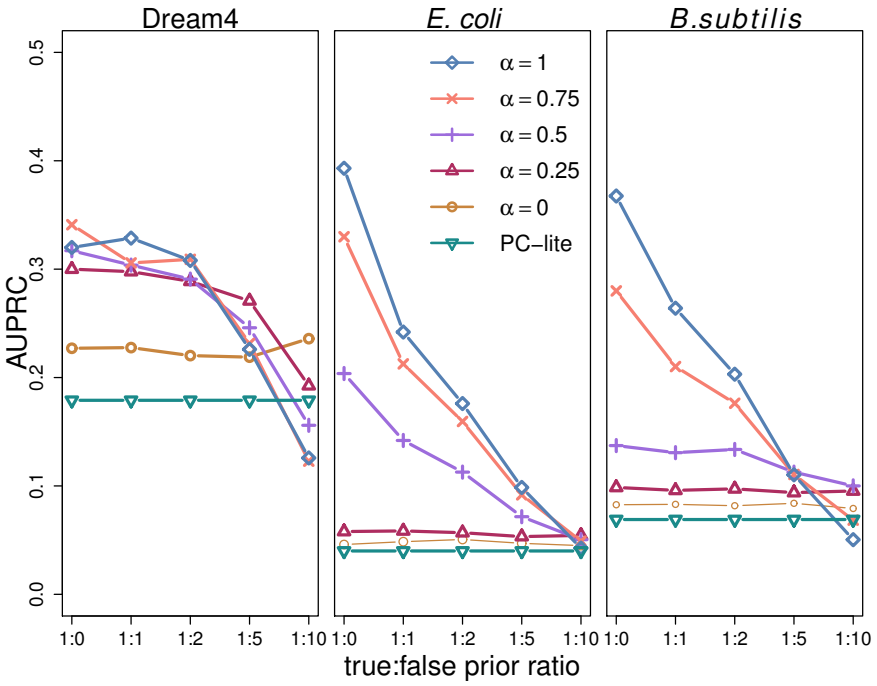
We presented PriorPC, a variation of the PC algorithm which takes advantage of the dependency of PC to the order in which variables are presented. This dependency is due to sparse and noisy data which affects negatively the performance of the CI tests. The larger the number of variables, the more impact the order has. This flaw is here exploited to integrate prior knowledge by rearranging the CI tests in order to favor less probable edges for early testing and to keep more likely edges for late testing.

PriorPC uses soft priors which assign to edges a probability of existence, rather than hard priors which give edges an existence state. We believe soft priors are more desirable as they can summarize the level of uncertainty the source associates with the edge, and the level of uncertainty associated with source itself.

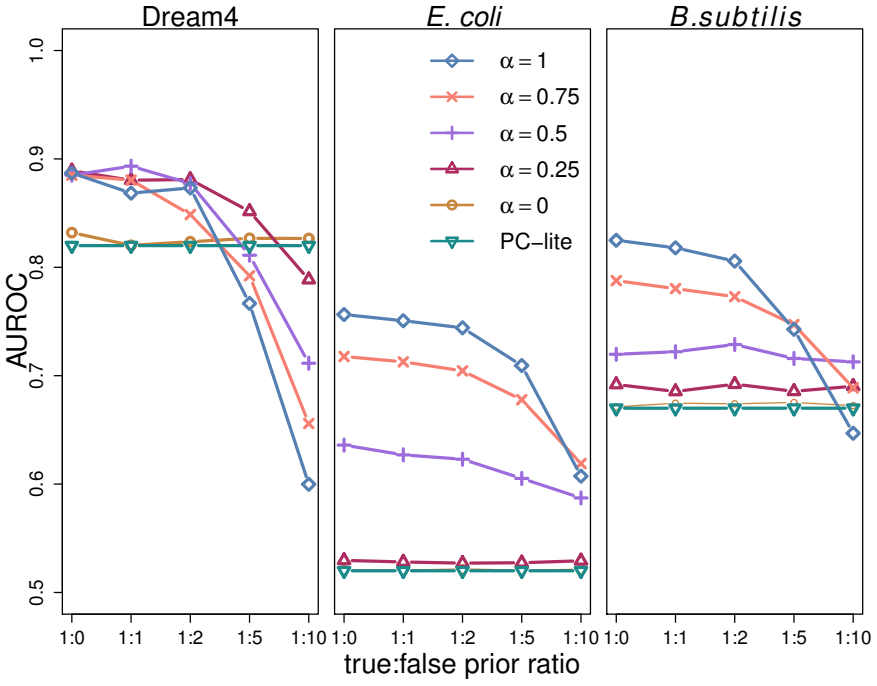
PriorPC is evaluated on three different datasets. Although parameters are never tuned at any point of the experiments, PriorPC produces a significant improvement in structural accuracy over PC for every dataset at hand. This improvement consistently increases with the amount of prior. Moreover, in the presence of partial prior knowledge, the part of the network that has no prior is not badly affected by the partial prior.

The robustness of the algorithm to noise in the prior matrix, which is not avoidable in the context of biological data, was tested. The results show that in the presence of noisy priors, PriorPC still performs better than PC up to a level of noise of 0.15. This transition level depends on how strong the dependency to prior knowledge is, i.e. how high α is. Similarly, if priors are flipped (i.e. false) rather than noisy, PriorPC performs better up to a ratio of true priors to false priors between 1:5 and 1:10. Again this ratio depends on α. In practice, if the reliability of the available prior knowledge is questionable, it is advisable to use a smaller value for α.

PriorPC is fast and scales well while most Bayesian network reconstruction methods which use prior knowledge are not feasible for large networks. These methods are mostly in the class of score-based methods and usually involve Markov-Chain-Monte-Carlo algorithm which is computationally expensive.



(A) AUROC



(B) AUPRC

FIGURE 4.5: **Performance of PriorPC against the ratio of true priors to false priors.** The first row shows the AUROC, while the second shows the AUPRC. A true prior is assigned to a random 50% of the edges present in the gold standard network. Different amounts of erroneous priors are then produced by flipping the true prior assigned to the remaining edges. The experiment is repeated for various values of α displayed in different colours. The performance of PC is plotted in green and with triangles for comparison. PriorPC performs better than PC-lite up to a ratio of true priors to false priors of 1:5.

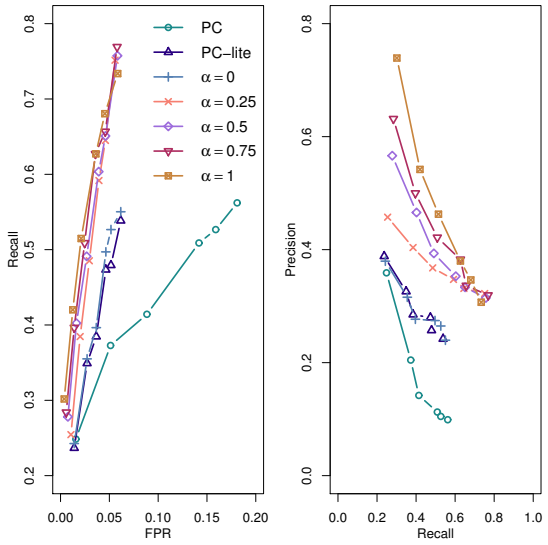


FIGURE 4.6: **Comparison of PC, PC-lite and PriorPC for DREAM4 data.** The left subplot shows the ROC curve, while the right subplot shows the PR curve for different tuning parameters. PriorPC (with any value of α) and PC-lite outperform the PC for all parameters.

Chapter 5

Partial distance correlation and its application in gene network reconstruction

5.1 Introduction

Recognizing direct relationships between variables is a substantial task in many problems including GRN inference. Although methods such as partial correlation and conditional mutual information (see section 2.1) are capable of finding direct interactions, the former is not able to find nonlinear relationships and estimation of the latter from continuous finite data is not trivial. Therefore, there is considerable interest in developing a method capable of detecting direct nonlinear relationships.

The recently proposed association measure, distance correlation (dcor) [9, 10] , is able to find nonlinear relationships. In addition, unlike mutual information, obtaining the empirical dcor is quite simple which makes it more appropriate than mutual information for the application on real data. In the context of inferring GRNs, one could use dcor as an association score in relevance networks as shown in the work by Guo et al. [37]. However, as we explained in section 2.2 relevance networks also associate variables that interact indirectly through one or more other variables and consequently they contain numerous indirect relationships. Although Guo et al. used CLR and MRNET methods to reduce the number of indirect interactions, a more proper method should take advantage of the concept of conditional independence analogous to partial correlation and conditional mutual information.

Recently, the authors of dcor introduced an approach to compute partial distance correlation (pdcor), the generalization of dcor which controls for the effect of other variables on the association between two variables in multivariate analysis. As we explained in Chapter 2, the definition is based on an unbiased distance covariance statistic using new distance matrices named $\mathcal{U}$-centered matrices. Even though they show that the statement of " $pdcor(X, Y; Z) = 0$ if and only if X and Y are conditionally independent given Z" does not hold true generally by providing numerical examples, it can be still beneficial to use pdcor for the purpose of network reconstruction. Especially in the non Gaussian case, where zero partial correlation is not equivalent to conditional independence, partial distance correlation which captures nonlinear associations can be more useful.

In this chapter, we suggest another approach to estimate partial distance correlation based on double centered matrices (mpdc). In this approach, we consider the squared distance correlation as an inner product in the Hilbert space of double centered matrices. Szekely et al. opposed this approach, stating that the difference of double centered distance matrices typically is not a double centered distance matrix of any sample, and therefore the projections do not have any interpretations. This was the reason that they defined an alternate type of centering, namely $\mathcal{U}$-centered matrices. In the Hilbert space of $\mathcal{U}$-centered matrices, the $\mathcal{U}$-centered distance covariance is the inner product. Furthermore, in this Hilbert space, all linear combinations, and in particular projections are $\mathcal{U}$-centered matrices.

We used pdcor and mpdc as an independence measure in the graphical models in analogy with partial correlation in Gaussian graphical models. We used simulated data and DREAM challenge data to assess the performance of these methods. We also compared their performance with the performance of relevance networks with dcor and cor as the association measures as well as graphical Gaussian models. Finally, we test the new measure in the context of the PC algorithm.

5.2 Methods

5.2.1 Partial distance correlation

As we explained in section 2.1.3, the distance covariance and distance correlation statistics are functions of the double centered distance matrices of the samples. The double centered distance matrix $\hat{A}$ for the variable X with n iid samples $\{x_1, x_2, ..., x_n\}$ is obtained from the matrix of Euclidean distance $(a_{ij}) = (|x_i - x_j|)$ by subtracting the

row/column means and adding the grand mean. As a result, all rows and columns in the double centered matrix $\hat{A}$ sum to zero.

Let $\hat{A}$, $\hat{B}$ and $\hat{C}$ be the double centered distance matrices corresponding to variables X,Y and Z obtained from n iid samples. Further, consider $V_{\hat{A}} = (\hat{A}_{.1}, \hat{A}_{.2}, ..., \hat{A}_{.n})$, $V_{\hat{B}} = (\hat{B}_{.1}, \hat{B}_{.2}, ..., \hat{B}_{.n})$ and $V_{\hat{C}} = (\hat{C}_{.1}, \hat{C}_{.2}, ..., \hat{C}_{.n})$ be the vector versions of double centered distance matrices $\hat{A}$, $\hat{B}$ and $\hat{C}$ respectively. Then the distance correlation between X and Y is defined as:

$$dcov_n^2(X,Y) = \frac{1}{n^2}\sum_{i,j=1}^{n} \hat{A}_{ij}\hat{B}_{ij}$$

which we can rewrite as:

$$dcov_n^2(X,Y) = cov(V_{\hat{A}}, V_{\hat{B}})$$

Therefore the form of the sample distance correlation offers an approach to compute the sample partial distance correlation by using partial correlation. Simply by regressing $V_{\hat{A}}$ and $V_{\hat{B}}$ to $V_{\hat{C}}$, we can obtain residuals r_x and r_y and then we can define the sample partial distance correlation as:

$$mpdc(X,Y|Z) = cor(r_x, r_y)$$

More formally, we can define an inner product in the Hilbert space of the double centered matrices as:

$$(\hat{A}.\hat{B}) = \frac{1}{n^2}\sum_{i\neq j} \hat{A_{i,j}}.\hat{B_{i,j}}$$

and obtain the projections $P_{z^\perp}(x)$ and $P_{z^\perp}(y)$ as:

$$P_{z^\perp}(x) = \hat{A} - \frac{(\hat{A}.\hat{C})}{(\hat{C}.\hat{C})}\hat{A}$$

,

$$P_{z^\perp}(y) = \hat{B} - \frac{(\hat{B}.\hat{C})}{(\hat{C}.\hat{C})}\hat{B}$$

Then the sample partial distance covariance and correlation are defined as:

$$\begin{aligned} mpdcov(x, y|z) =& (P_{Z^\perp}(X).P_{Z^\perp}(Y)) \\ &= \frac{1}{n^2}\sum_{i\neq j}(P_{Z^\perp}(X))_{i,j}(P_{Z^\perp}(Y))_{i,j} \end{aligned}$$

$$mpdc(X, Y|Z) = \frac{(P_{Z^\perp}(x).P_{Z^\perp}(Y))}{|P_{Z^\perp}(X)|.|P_{Z^\perp}(Y)|}, \qquad |P_{Z^\perp}(X)|.|P_{Z^\perp}(Y)| \neq 0,$$

5.2.2 Partial distance correlation as the independence measure in graphical models

One can use the partial distance correlation as the independence measure in the graphical models. In analogy with Gaussian graphical models, where independence relationships are assessed based on the full-order partial correlation, we can define full-order partial distance correlation models. However, in these models there is no assumption on the underlying distribution of the data.

A problem with full conditional models is that it is hard to reliably estimate them when the number of samples is smaller than the number of variables, for example when working with gene expression data. One way to cope with this issue is to use algorithms which use lower order conditional independence tests like PC algorithm. Models which are based on low-order conditional independence correct for the influence of some and not all the remaining variables.

The PC algorithm does not provide a score for the edges. As stated in section 4.2.2, one way to remedy this issue is to use bagging. However, since dcor is computed based on the distance matrices, bootstrapping is not an appropriate choice. Therefore, instead of using bootstrapping to build perturbed data, we use a single data set with different thresholds for the independence decisions (see section 2.4).

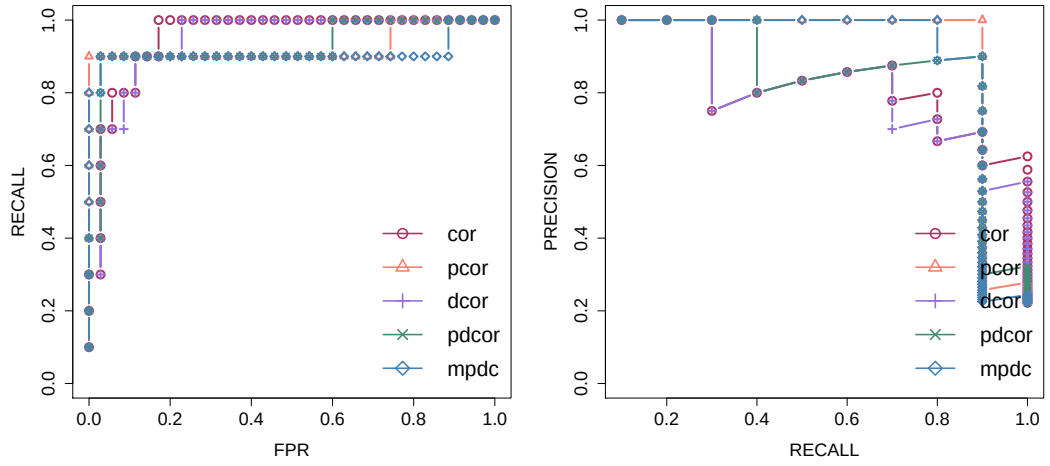
5.3 Results

5.3.1 Comparison of partial distance correlation with partial correlation

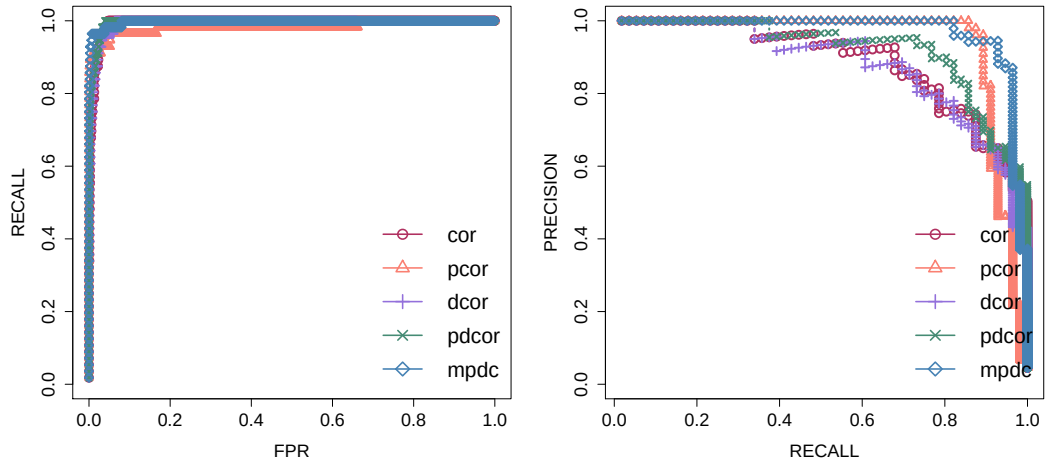
In this section, we compare the performance of partial distance correlation obtained from our methods (mpdc) to the one suggested by Szekely et al. (pdcor) as well as

to correlation and partial correlation. In the case of Gaussian data, the performance of partial distance correlation should be as good as partial correlation in order to be a proper substitution for partial correlation. Therefore, we simulate 300 samples of Gaussian data from networks with 10 and 50 nodes as described in section 3.1 and add Gaussian noise. We then evaluate the performance of different methods on the data.

Figure 5.1 shows the ROC and PR curves for different methods. The result indicates that mpdc performs better than pdcor. In addition, it is comparable to partial correlation on Gaussian data which is an important issue.



(A) 10 nodes



(B) 50 nodes

FIGURE 5.1: **Performance on simulated Gaussian data.** The left subplots show ROC curves, while the right subplots show PR curves.

5.3.2 Effect of the number of samples on the performance

In this section, we assess the effect of the number of samples on the performance of the full-order pdcor models (both mpdc and pdcor) with relevance networks using cor and dcor as association measures. We simulate a different number of samples of Gaussian data from networks with 10 and 50 nodes. Figure 5.2 shows the ROC and PR curves for different methods.

The results show that the performance of mpdc and pdcor are almost comparable except in the case where there are very few samples, where pdcor performs slightly better. In addition, in this case pcor and cor based methods also perform slightly better than both pdcor and mpdc. It is important to note that we used the regularized method for the estimation of pcor which was developed to cope with the $p >> n$ situation.

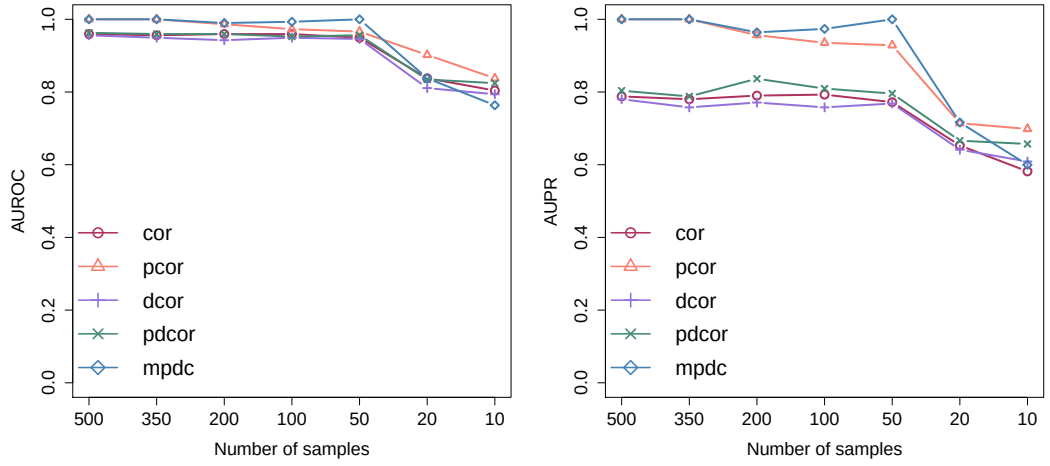
5.3.3 Comparison of methods in the presence of different amount of noise

In this section, we compare the performance of the full-order partial distance correlation models (both mpdc and pdcor) with relevance networks using cor and dcor as the association measures. We simulate 500 and 300 samples of Gaussian data from networks with 10 and 50 nodes, respectively. Then we add Gaussian noise $\epsilon \sim N(0, \sigma^2)$ with different amount of variance σ^2. Figure 5.3 shows the effect of noise on the AUPRC and the AUROC for different methods.

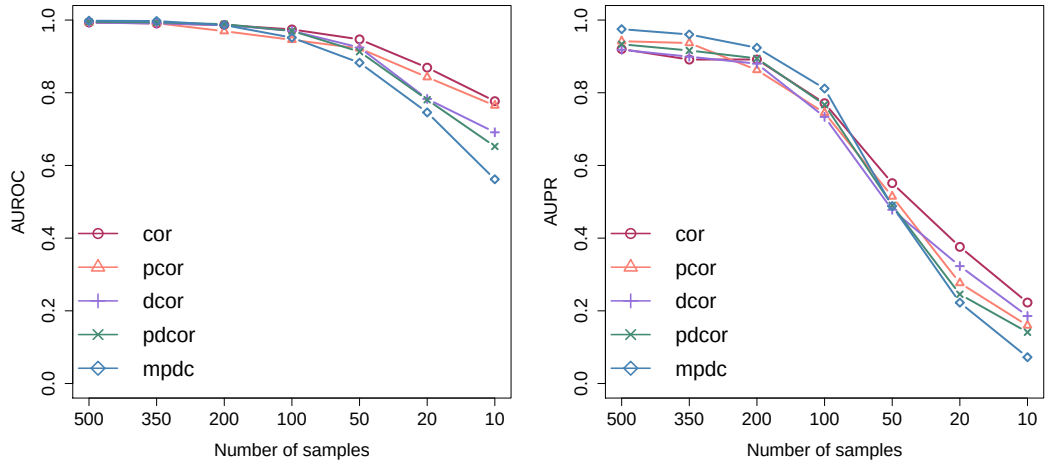
Clearly, for each method the higher the amount of noise, the worse the performance. While mpdc performs slightly better than pdcor in the presence of a low amount of noise, the performance is comparable for high amount of noise. Furthermore, for small amount of noise the performance of mpdc and pcor are comparable, while for high amount of noise pcor performs better.

5.3.4 Performance comparison on DREAM challenge data

In this section, we compared the performance of different methods on DREAM challenge data. Figures 5.4 and 5.5 show the ROC and PR curves for DREAM3 (10 nodes, 50 nodes and 100 nodes) and DREAM4 (10 nodes and 100 nodes) data sets, respectively. The results show that mpdc performs well above pdcor on all data sets. Yet, in comparison to pcor, both methods perform comparable for DREAM3 data sets with 50 and 100 nodes. For DREAM4 data sets, mpdc shows a better performance in the case of 10 nodes while pcor performs better in the case of 100 nodes.



(A) 10 nodes



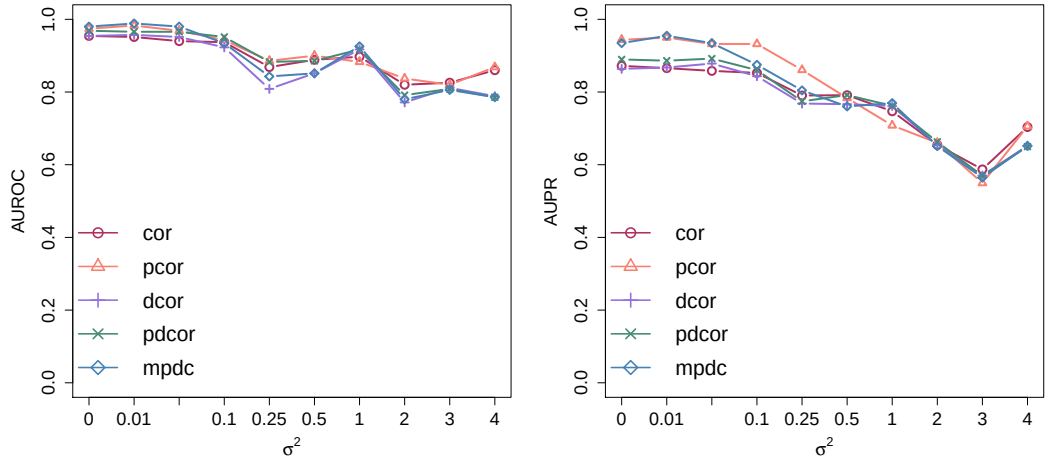
(B) 50 nodes

FIGURE 5.2: **Effect of the number of samples on the performance.** The left subplots show the AUROC reults, while the right subplots show the AUPR results.

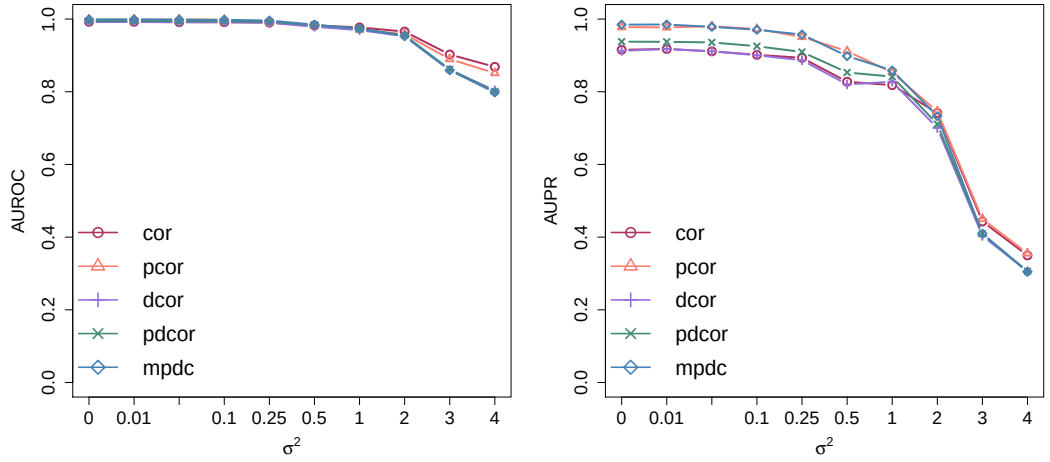
In the case of the DREAM3 data set with 10 nodes, while partial correlation performs better than pdcor, the pcor values are almost all zero as shown in Figure 5.6. As a result, it is hard to build the network from this score as almost all or none of the edges will appear in the network.

5.3.5 Performance of PC algorithm with pcor and pdcor as the independence tests

In this section, we assess the performance of the PC algorithm which uses mpdc as an independence test (PC-mpdc) and full-order mpdc methods. We simulate 50 Gaussian samples from a network with 50 nodes. Figure 5.7 shows the ROC and PR curves. The



(A) 10 nodes



(B) 50 nodes

FIGURE 5.3: **Effect of noise on the performance.** The left subplots show the AUROC results, while the right subplots show AUPR result.

result shows that the pefomance of both methods are comparable although PC-mpdc improves the result slightly compared to the full-order model.

5.4 Discussion

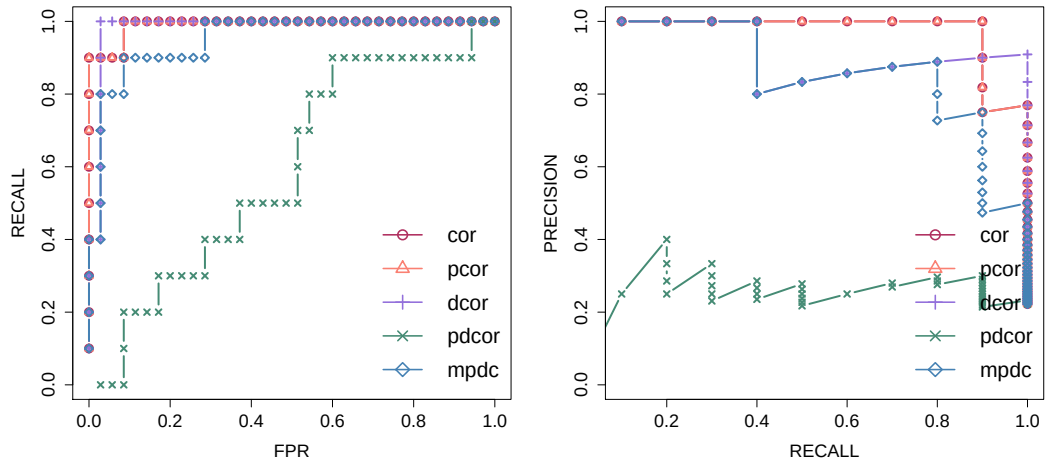
In this chapter, we introduced a new approach called mpdc that can be used to estimate sample partial distance correlation in a different way than the method proposed by the authors of dcor, which we call pdcor. Partial distance correlation is the generalization of dcor which controls for the effect of other variables in the system on the association between two variables (analogous to the partial correlation). Therefore, partial distance

correlation can detect direct nonlinear interactions which is an important task in the GRN inference.

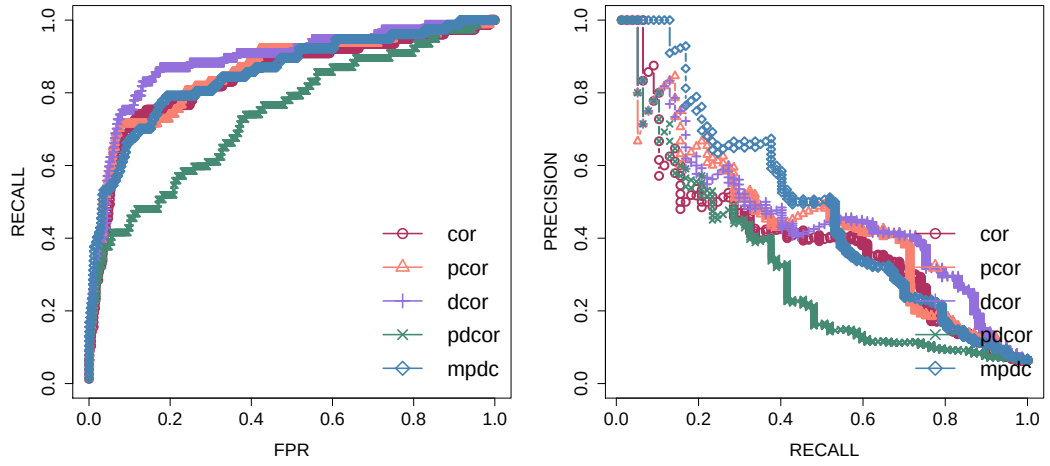
The performance of full-order pdcor and mpdc are evaluated on simulated data and DREAM challenge data and compared with the performance of relevance networks with dcor and cor as the association measure as well as graphical Gaussian models (full-order pcor). The results show that mpdc performs better than pdcor in all cases, although pdcor is mathematically better founded. The reason for this is not clear for us yet. In addition, mpdc performs better than dcor in terms of PR curves, indicating that mpdc is capable of removing the indirect interactions.

For the simulated Gaussian data, the performance of pcor and mpdc are comparable in most cases. However, in the cases with very high amount of noise and/or very few samples pcor performs slightly better than mpdc. It is important to note that mpdc is not designed for these cases in contrast to the regularized method which we used for the estimation of pdcor. For the DREAM challenge data, the performance of mpdc and pcor are comparable in general although there are some cases where one of them performs better. Therefore, more analysis is needed to confidently determine which method is better for GRN inference.

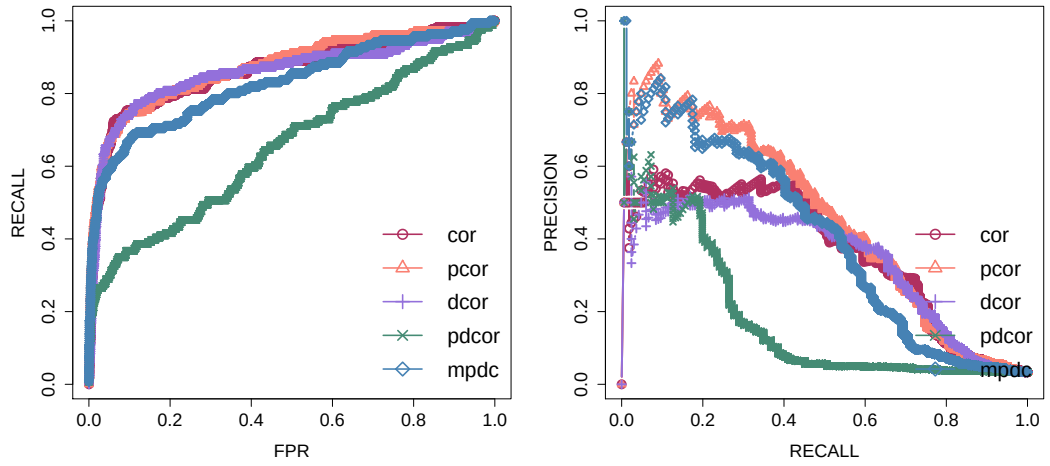
In the case of not having many samples, we also assess the performance of mpdc in the context of the PC algorithm (PC-mpdc). The result shows that PC-mpdc performs slightly better than the full-order model. In PC-mpdc, the scores of the edges are obtained from an ensemble method by varying the threshold to decide for independence relationships. It is important to note that since the result of the PC depends on these thresholds, this ensemble approach could underestimate the performance of the PC algorithm.



(A) 10 nodes

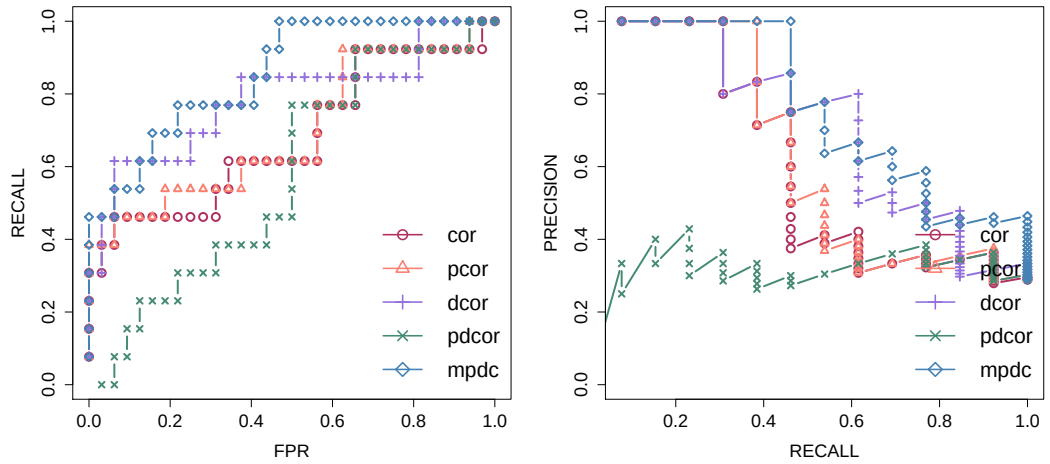


(B) 50 nodes

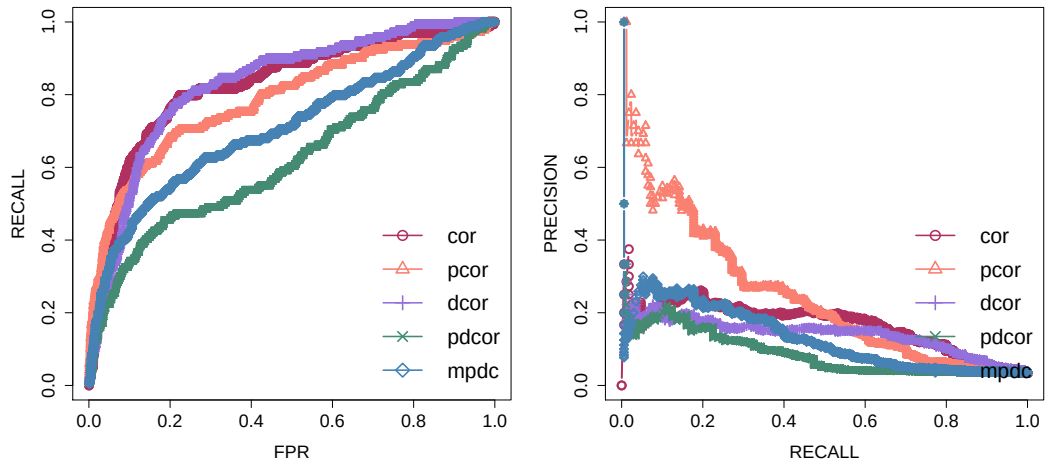


(C) 100 nodes

FIGURE 5.4: **Performance on DREAM3 challenge data.** The left subplots show the ROC curve, while the right subplots show the PR curve.



(A) 10 nodes



(B) 100 nodes

FIGURE 5.5: **Performance on DREAM4 challenge data.** The left subplots show the ROC curve, while the right subplots show the PR curve.

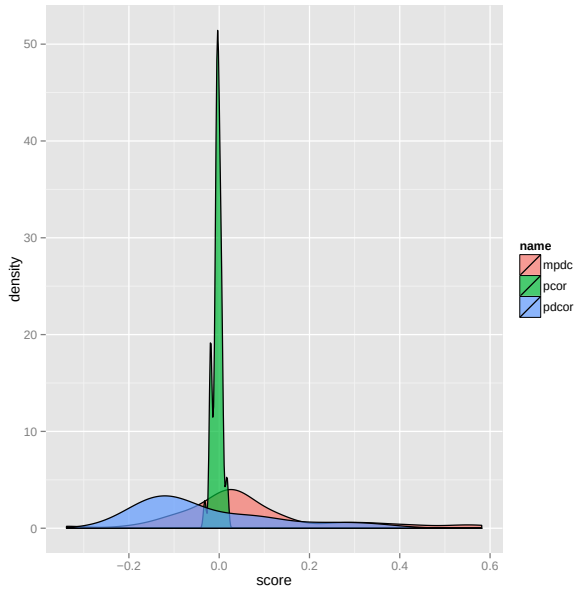


FIGURE 5.6: **Distribution of pcor and pdcor scores for DREAM3 with 10 nodes.)**

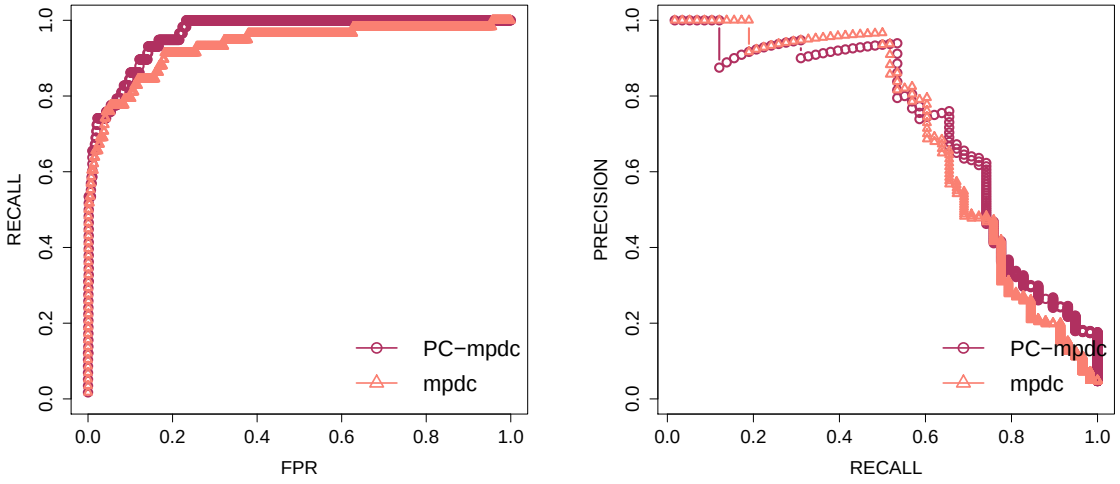


FIGURE 5.7: **Performance comparison of the PCmpdc with full-order mpdc.** The left subplots show the AUROC results, while the right subplots show AUPR result.

Chapter 6

Summary

One of the grand challenge of systems biology is to decipher the interactions among genes which is known as GRN reconstruction. Among all biological data, gene expression data obtained via measuring the abundances of mRNAs in the cell is the most widely available data used for this purpose. The "small n, large p" data setting of gene expression coupled with high amounts of noise in the data render the inference of GRNs from gene expression data a challenging task. Moreover, most methods of GRN inference rely on the assumption that regulatory interactions are linear, while in reality this is usually not the case. In this thesis, we propose methods to tackle these problems.

We started the thesis by describing models based on the concept of (conditional) independence in Chapter 2, namely relevance networks and graphical models. The objective of this kind of modeling is to find the (in)dependence structure among genes by means of (conditional) independence tests that we introduced. Relevance networks consider only the association between two genes and ignore the effect of other genes on the relationship between them. As a result they cannot distinguish between direct and indirect relationships. More sophisticated methods, such as graphical models address this issue by using the conditional independence concept and try to explain the association between genes by the presence of other genes and as a result to find the direct interactions.

In Chapter 3, we investigate the performance of models introduced in Chapter 2 for gene network reconstruction in different aspects. We used Gaussian and non Gaussian simulated data as well as data provided by the DREAM challenge. This helped us to learn more about the strengths and weaknesses of different methods. The results demonstrate that in the well behaved case of having many samples and no noise all methods are comparable. Although correlation and distance correlation perform better than other association measures in case of not having many samples and/or in facing high amount of noise, their performance also decreased significantly compared to the

well behaved case. This led us to the conclusion that information of gene expression data is not enough to decipher the complex interactions among genes and therefore exploiting other sources of knowledge is indispensable. Our development of PriorPC, an algorithm which also uses prior knowledge for reconstructing GRNs was motivated upon these grounds. In addition, distance correlation (dcor) performs well among nonlinear association measures but like other association measures is not able to detect direct interactions. This presented us with the motivation to generalize distance correlation for multivariate analysis with the ability to account for the effect of other variables (in analogy with partial correlation).

In Chapter 4, we present PriorPC, a variation of the PC algorithm that uses the prior knowledge in the form of soft prior which assign to edges a probability of existence. PriorPC takes advantage of the flaw of the PC algorithm, namely its dependency on the the order in which variables are presented to integrate prior knowledge. PriorPC modifies the PC algorithm by favoring unwanted edges for early testing, and holding wanted edges back for late testing. Prior knowledge is particularly advantageous when the quality of the CI tests is questionable, like the case when data is high-dimensional and few samples are available (the typical form of gene expression data). PriorPC produces a significant improvement in structural accuracy over PC for every dataset at hand. This improvement consistently increases with the amount of prior. Moreover, in the presence of partial prior knowledge, the region of the network that has no prior is not badly affected by the partial prior. PriorPC is fast and scales well while most Bayesian network reconstruction methods which use prior knowledge are not feasible for large networks. This is an important issue in with the application to biological data since in reality we are dealing with thousands genes. It is also robust to noise in the prior, which is not avoidable in the context of biological data.

In this thesis, we used synthetic priors. It would be interesting to see how performance changes when using real priors. Prior knowledge can be obtained from different sources including experimental data like ChIP-seq data and even information derived from relevant literature. All theses sources of information can be included in a prior knowledge matrix representing the aggregated belief about gene interactions.

In Chapter 5, we address the problem of finding direct nonlinear association between variables by proposing a new way to compute the partial distance correlation (mpdc). Most of methods for inferring GRNs are not able to find direct nonlinear relationships and therefore detecting direct nonlinear relationships is an important task which can improve the accuracy of the inferred networks. Distance correlation is a new association measure with the ability to detect nonlinear association. The advantage of this measure over mutual information, the classical way of finding nonlinear association, is that there

is a simple way to estimate it from data. However, for the purpose of GRN inference it is important to find the direct interactions among genes. Although the authors of distance correlation proposed a way to compute partial distance correlation (pdcor), we show that in the context of network reconstruction our approach performs better than their method.

Based on the observations from the analysis done in Chapter 5, mpdc performs better than pdcor, even though the pdcor is well defined with mathematical background. The reason is not clear for us yet. In addition, we compared the performance of mpdc with performance of a regularized method for estimation of partial correlation which has designed to cope with $p >> n$ situation. Therefore, the new challenge is to find a similar approach for partial distance correlation in case of $p >> n$.

List of Figures

List of Tables

Abbreviations

AUPRC	**A**rea **U**nder the **P**recision **R**ecall **C**urve
AUROC	**A**rea **U**nder the **R**OC **C**urve
BN	**B**ayesian **N**etwork
cDNA	**C**omplementary **DNA**
ChIP	**C**hromatin **I**mmunoprecipitation
CI	**C**onditional **I**ndependence
CLR	**C**ontext **L**ikelihood of **R**elatedness
DAG	**D**irected **A**cyclic **G**raph
DNA	**N**ucleotide **N**ucleic **A**cid
DREAM	**D**ialogue for **R**everse **E**ngineering **A**ssessments and **M**ethods
GRN	**G**ene **R**egulatory **N**etwork
GGM	**G**raphical **G**aussian **M**odel
FP	**F**alse **P**ositive
FN	**F**alse **N**egative
HHG	**H**eller-**H**eller-**G**orfine
mRNA	**M**essenger **RNA**
MI	**M**utual **I**nformation
MIC	**M**aximal **I**nformation **C**oefficient
NGS	**N**ext **G**eneration **S**equencing
PDAG	**P**artially **D**irected **A**cyclic **G**raph
PR	**P**recision **R**ecall
RNA	**R**ibosomal **N**ucleic **A**cid
RNA-Seq	**R**NA **S**equencing
ROC	**R**eceiver **O**perator **C**haracteristic
SVD	**S**ingular **V**alue **D**ecomposition

TF	**T**ranscription **F**actor
TFBS	**T**ranscription **F**actors **B**inding **S**ites
TN	**T**rue **N**egative
TP	**T**rue **P**ositive

Symbols

$\tilde{A}$	$\mathcal{U}$-centered matrix
$\hat{A}$	double centered distance matrix
$X \perp Y$	X and Y are independent
$X \not\perp Y$	X and Y are not independent
$cov(X, Y)$	covariance of X and Y
$r(X, Y)$	correlation of X and Y
$dcov(X, Y)$	distance covariance of X and Y
$dcor(X, Y)$	distance correlation of X and Y
$MI(X, Y)$	mutual information of X and Y
$X \perp_G Y \vert Z$	X and Y are d-separated by a set Z

Bibliography

[1] Anne-Claire Haury, Fantine Mordelet, Paola Vera-Licona, and Jean-Philippe Vert. Tigress: Trustful inference of gene regulation using stability selection. *BMC Systems Biology*, 6:145, 2012. URL http://dblp.uni-trier.de/db/journals/bmcsb/bmcsb6.html#HauryMVV12.

[2] Juliane Schäfer and Korbinian Strimmer. A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics. *Statistical Applications in Genetics and Molecular Biology, The Berkeley Electronic Press*, 4 (1), 2005. URL http://www.bepress.com/sagmb/vol4/iss1/art32/.

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

[4] Florian Markowetz and Rainer Spang. Inferring cellular networks – a review. *BMC Bioinformatics*, 8:S5, 2007. URL http://www.biomedcentral.com/1471-2105/8/S6/S5.

[5] Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780, 2008. URL http://scholar.google.com/scholar.bib?q=info:pUfCnmlzpOYJ:scholar.google.com/&output=citation&scisig=AAGBfm0AAAAAUkd49ci5QLX_gXqZ5dgB1EfP6IP7kpvD&scisf=4&hl=en&scircf=1.

[6] Helena Brunel, Joan-Josep Gallardo-Chacón, Alfonso Buil, Montserrat Vallverdú, José Manuel Soria, Pere Caminal, and Alexandre Perera. Miss: a non-linear methodology based on mutual information for genetic association studies in both population and sib-pairs analysis. *Bioinformatics*, 26(15):1811–1818, 2010. URL http://dblp.uni-trier.de/db/journals/bioinformatics/bioinformatics26.html#BrunelGBVSCP10.

[7] C. Olsen, P. E. Meyer, and G. Bontempi. On the impact of entropy estimation on transcriptional regulatory network inference based on mutual information. *EURASIP journal on bioinformatics & systems biology*, 2009. ISSN 1687-4145. URL http://view.ncbi.nlm.nih.gov/pubmed/19148299.

[8] Hanspeter Herzel and Ivo Große. Measuring correlations in symbol sequences. *Physica A: Statistical Mechanics and its Applications*, 216(4):518 – 542, 1995. ISSN 0378-4371. doi: http://dx.doi.org/10.1016/0378-4371(95)00104-F. URL http://www.sciencedirect.com/science/article/pii/037843719500104F.

[9] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *Ann. Statist.*, 35(6):2769–2794, 12 2007. doi: 10.1214/009053607000000505. URL http://dx.doi.org/10.1214/009053607000000505.

[10] Gábor J. Székely and Maria L. Rizzo. Brownian distance covariance. *Ann. Appl. Stat.*, 3(4):1236–1265, 12 2009. doi: 10.1214/09-AOAS312. URL http://dx.doi.org/10.1214/09-AOAS312.

[11] Gábor J. Székely and Maria L. Rizzo. Partial distance correlation with methods for dissimilarities. *Ann. Statist.*, 42(6):2382–2412, 12 2014. doi: 10.1214/14-AOS1255. URL http://dx.doi.org/10.1214/14-AOS1255.

[12] Ruth Heller, Yair Heller, and Malka Gorfine. A consistent multivariate test of association based on ranks of distances. *Biometrika*, 100(2):503–510, 2013. URL http://EconPapers.repec.org/RePEc:oup:biomet:v:100:y:2013:i:2:p:503-510.

[13] David N. Reshef, Yakir A. Reshef, Hilary K. Finucane, Sharon R. Grossman, Gilean McVean, Peter J. Turnbaugh, Eric S. Lander, Michael Mitzenmacher, and Pardis C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011. doi: 10.1126/science.1205438. URL http://www.sciencemag.org/content/334/6062/1518.abstract.

[14] Noah Simon and Robert Tibshirani. Comment on " detecting novel associations in large data sets" by reshef et al. January 2012. URL http://www-stat.stanford.edu/~{}tibs/reshef/.

[15] Justin B. Kinney and Gurinder S. Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*,

111(9):3354–3359, 2014. doi: 10.1073/pnas.1309933111. URL http://www.pnas.org/content/111/9/3354.abstract.

[16] Adam A. Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(S-1), 2006. URL http://dblp.uni-trier.de/db/journals/bmcbi/bmcbi7S.html#MargolinNBWSFC06.

[17] Jeremiah J. Faith, Boris Hayete, Joshua T. Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J. Collins, and Timothy S. Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, 5:8, 2007.

[18] Patrick Emmanuel Meyer, Kevin Kontos, Frédéric Lafitte, and Gianluca Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J. Bioinformatics and Systems Biology*, 2007, 2007. URL http://dblp.uni-trier.de/db/journals/ejbsb/ejbsb2007.html#MeyerKLB07.

[19] Robert Castelo and Alberto Roverato. Reverse engineering molecular regulatory networks from microarray data with qp-graphs. *Journal of Computational Biology*, 16(2):213–227, 2009. URL http://dblp.uni-trier.de/db/journals/jcb/jcb16.html#CasteloR09.

[20] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.

[21] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2nd edition, 2000.

[22] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. In *Machine Learning*, page 2006, 2006.

[23] Daniel Marbach, James C Costello, Robert Kü ffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, Manolis Kellis, James J Collins, Gustavo Stolovitzky, the DREAM5 Consortium, and Yvan Saeys. Wisdom of crowds for robust gene network inference. *NATURE METHODS*, 9(8):796–804, 2012. ISSN 1548-7091. URL http://dx.doi.org/10.1038/NMETH.2016.

[24] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996. ISSN 0885-6125. doi: 10.1023/A:1018054314350. URL http://dx.doi.org/10.1023/A:1018054314350.

[25] Daniel Marbach, Thomas Schaffter, Claudio Mattiussi, and Dario Floreano. Generating Realistic In Silico Gene Networks for Performance Assessment of Reverse Engineering Methods. *Journal of Computational Biology*, 16(2):229–239, 2009. doi: 10.1089/cmb.2008.09TT. WingX.

[26] Robert J. Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K. Sorger, Leonidas G. Alexopoulos, Xiaowei Xue, Neil D. Clarke, Gregoire Altan-Bonnet, and Gustavo Stolovitzky. Towards a rigorous assessment of systems biology models: The dream3 challenges. *PLoS ONE*, 5(2):e9202, 02 2010. doi: 10.1371/journal.pone.0009202. URL http://dx.doi.org/10.1371%2Fjournal.pone.0009202.

[27] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*, pages 104–113, Aug 2003. doi: 10.1109/CSB.2003.1227309.

[28] Adriano V Werhli and Dirk Husmeier. Reconstructing gene regulatory networks with bayesian networks by combining expression data with multiple sources of prior knowledge. *Stat. Appl. Genet. Mol. Biol.*, 6:Article15, 2007. doi: 10.2202/1544-6115.1282.

[29] Alex Greenfield, Christoph Hafemeister, and Richard Bonneau. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics*, 29(8):1060–1067, 2013. doi: 10.1093/bioinformatics/btt099. URL http://bioinformatics.oxfordjournals.org/content/29/8/1060.abstract.

[30] Sach Mukherjee and Terence P. Speed. Network inference using informative priors. *Proceedings of the National Academy of Sciences*, 105(38):14313–14318, 2008. doi: 10.1073/pnas.0802272105. URL http://www.pnas.org/content/105/38/14313.abstract.

[31] Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *J. Mach. Learn. Res.*, 8:613–636, May 2007. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1248659.1248681.

[32] Pierre Nicolas, Ulrike Mäder, Etienne Dervyn, Tatiana Rochat, Aurélie Leduc, Nathalie Pigeonneau, Elena Bidnenko, Elodie Marchadier, Mark Hoebeke, Stéphane Aymerich, Dörte Becher, Paola Bisicchia, Eric Botella, Olivier Delumeau, Geoff Doherty, Emma L. Denham, Mark J. Fogg, Vincent Fromion, Anne Goelzer, Annette Hansen, Elisabeth Härtig, Colin R. Harwood, Georg Homuth, Hanne Jarmer, Matthieu Jules, Edda Klipp, Ludovic Le Chat, François Lecointe, Peter Lewis, Wolfram Liebermeister, Anika March, Ruben A. T. Mars, Priyanka Nannapaneni, David Noone, Susanne Pohl, Bernd Rinn, Frank Rügheimer, Praveen K. Sappa, Franck Samson, Marc Schaffer, Benno Schwikowski, Leif Steil, Jörg Stülke, Thomas Wiegert, Kevin M. Devine, Anthony J. Wilkinson, Jan Maarten van Dijl, Michael Hecker, Uwe Völker, Philippe Bessières, and Philippe Noirot. Condition-dependent transcriptome reveals high-level regulatory architecture in bacillus subtilis. *Science*, 335(6072):1103–1106, 2012. doi: 10.1126/science.1206848. URL http://www.sciencemag.org/content/335/6072/1103.abstract.

[33] Steffen L. Lauritzen. *Graphical Models*. Oxford University Press, 1996. ISBN 0-19-852219-3.

[34] Rafael A. Irizarry, Bridget Hobbs, Francois Collin, Yasmin D. Beazer-Barclay, Kristen J. Antonellis, Uwe Scherf, and Terence P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

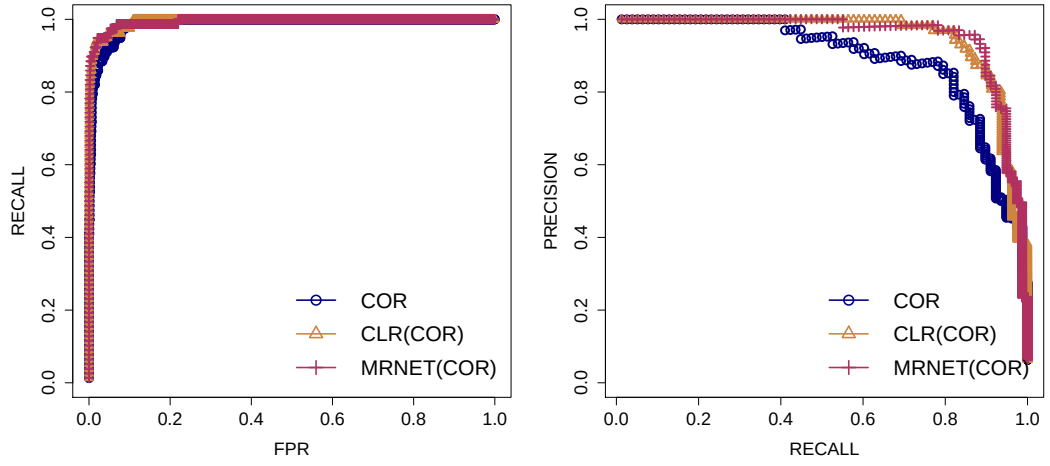
[35] Lope A. Flórez, Sebastian F. Roppel, Arne G. Schmeisky, Christoph R. Lammers, and Jörg Stülke. A community-curated consensual annotation that is continuously updated: the Bacillus subtilis centred wiki SubtiWiki. *Database : the journal of biological databases and curation*, 2009(0):bap012+, January 2009. ISSN 1758-0463. doi: 10.1093/database/bap012. URL http://dx.doi.org/10.1093/database/bap012.

[36] Christoph R. Lammers, Lope A. Flórez, Arne G. Schmeisky, Sebastian F. Roppel, Ulrike Mäder, Leendert Hamoen, and Jörg Stülke. Connecting parts with processes: SubtiWiki and SubtiPathways integrate gene and pathway annotation for Bacillus subtilis. *Microbiology*, 156(3):849–859, March 2010. ISSN 1465-2080. doi: 10.1099/mic.0.035790-0. URL http://dx.doi.org/10.1099/mic.0.035790-0.

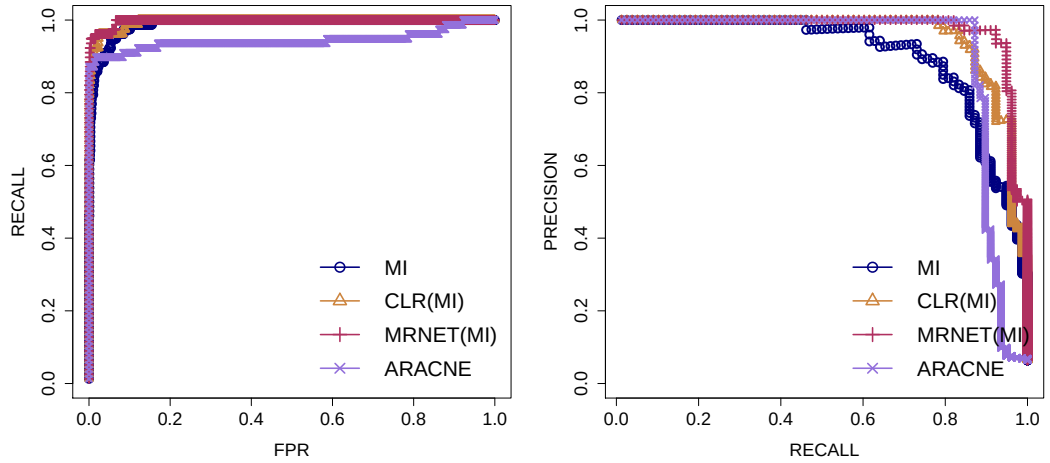
[37] Xiaobo Guo, Ye Zhang, Wenhao Hu, Haizhu Tan, and Xueqin Wang. Inferring nonlinear gene regulatory networks from gene expression data based on distance correlation. *PloS one*, 9(2):e87446, 2014. ISSN 1932-6203. doi: 10.1371/journal.pone.0087446. URL http://europepmc.org/articles/PMC3925093.

Appendix A

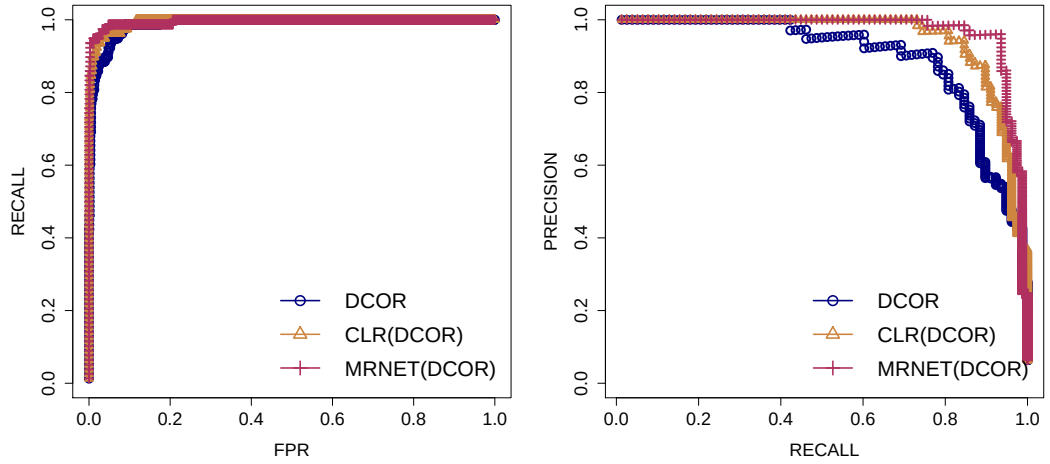
Supplementary Figures



(A) Correlation

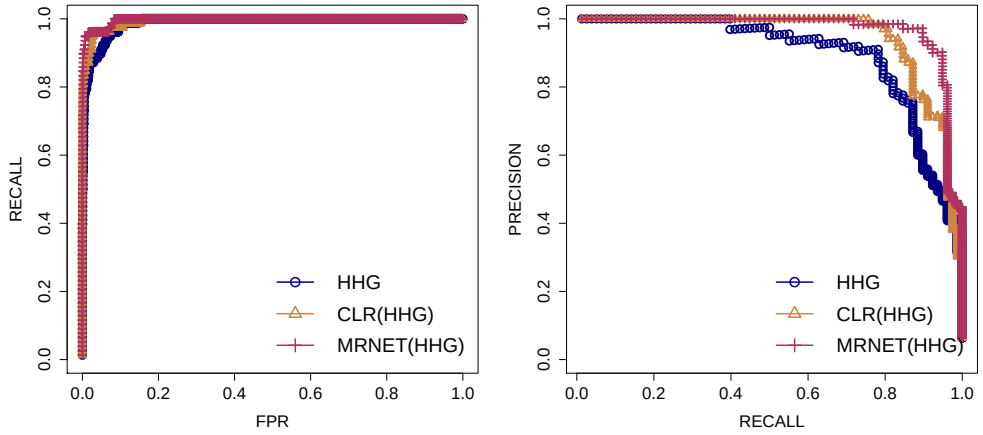


(B) Mutual Information

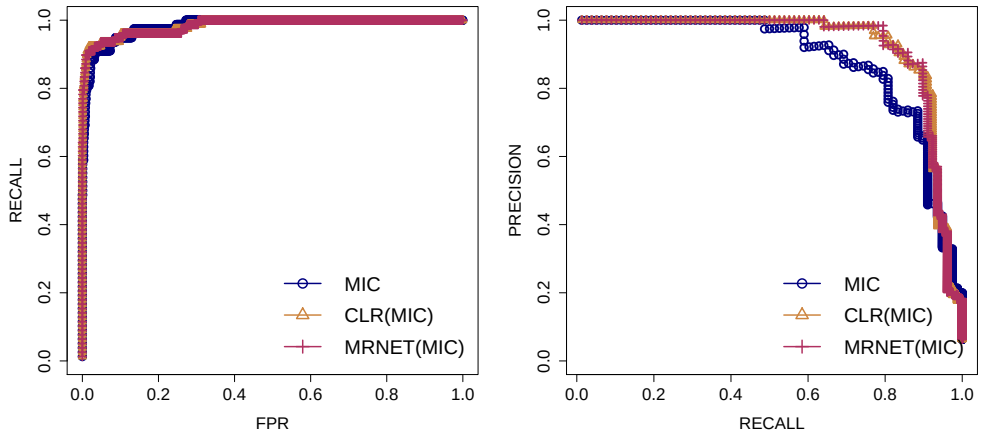


(C) Distance Correlation

FIGURE A.1: **Performance of relevance networks on non Gaussian data.** The data consist of 1000 samples simulated from a network with 50 nodes. The left subplots show the ROC curve, while the right subplots show the PR curve.



(A) HHG



(B) MIC

FIGURE A.2: **Performance of relevance networks on non Gaussian data.** The data consist of 1000 samples simulated from a network with 50 nodes. The left subplots show the ROC curve, while the right subplots show the PR curve.

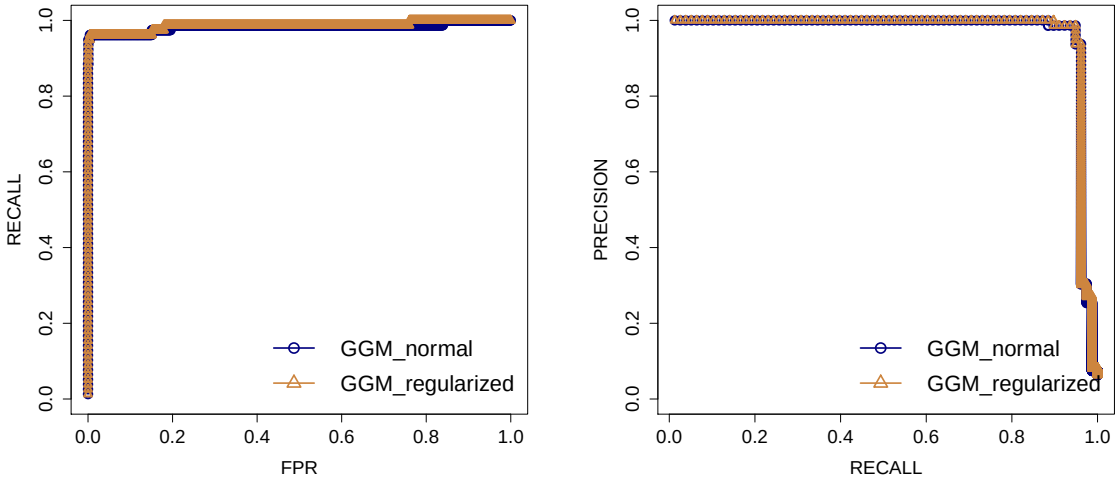


FIGURE A.3: **Performance of pseudo-inverse and shrinkage methods for obtaining GGMs on Gaussian data.**. The left subplot shows the AUROC, while the right subplot shows the AUPRC.

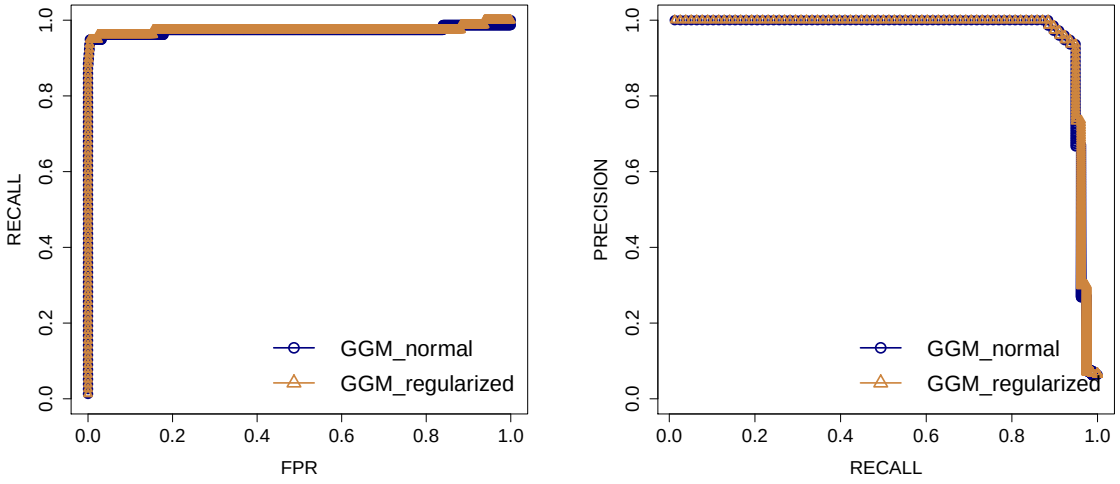


FIGURE A.4: **Performance of pseudo-inverse and shrinkage methods for obtaining GGMs on non Gaussian data.**for a network with 50 nodes and 1000 samples. The left subplot shows the AUROC, while the right subplot shows the AUPRC.

Appendix B

Zusammenfassung

Die Rekonstruktion von Gennetzwerken ("Gene Regulatory Networks", GRNs) aus Genexpressionsdaten ist eine anspruchsvolle Problemstellung, deren Lösung wichtig ist für das Verständnis der komplexen Regulationsmechanismen in der Zelle. Erschwert wird die Aufgabe einerseits durch die hohe Anzahl von Genen, deren Interaktionen man aus wenigen Experimenten schätzen möchte, und andererseits durch die fehlerbehafteten Messwerte der Genexpression. In der vorliegenden Arbeit wird zuerst untersucht, welche Auswirkungen die Anzahl der Experimente sowie die Stärke des Rauschens auf die Ergebnisse der statistischen Auswertung hat. Es zeigt sich, dass eine zu geringe Anzahl von Experimenten bei allen Methoden zu wesentlich schlechteren Ergebnissen führt. Ebenso führt höheres Rauschen in den Daten bei allen Methoden zu schlechteren Ergebnissen.

Ein naheliegender Ausweg liegt in der Nutzung zusätzlicher Informationen ("prior knowledge"), um die Rekonstruktion des Gennetzwerkes zu unterstützen und so die Probleme mit Datenmenge oder –qualität wenigstens teilweise zu kompensieren. Wir entwickeln hierzu den PriorPC-Algorithmus, ein neues Verfahren, das auf dem bekannten PC-Algorithmus zur Rekonstruktion eines Netzwerkes basiert. Obwohl weit verbreitet, ist über den PC-Algorithmus bekannt, dass die Qualität der Resultate von der Reihenfolge, in der die Eingabedaten abgearbeitet werden, abhängt. PriorPC verwandelt diesen Nachteil in eine Stärke, indem in die Reihenfolge der Abarbeitung das verfügbare Vorwissen einfließt. Wir zeigen hier an simulierten sowie an echten Daten, dass der

PriorPC-Algorithmus mit Vorwissen Netzwerke besser rekonstruieren kann als der einfache PC-Algorithmus. PriorPC ist außerdem schnell und auch für große Probleme, wie echte experimentelle Datensätze, einsetzbar.

Eine weitere Herausforderung der Netzwerkrekonstruktion besteht in der Aufdeckung (direkter) nicht-linearer Beziehungen zwischen Genen. Vor Kurzem wurde das neue Assoziationsmaß der Distanzkorrelation eingeführt, welches eine leistungsfähige Methode zur Identifikation nicht-linearer Zusammenhänge darstellt. In der vorliegenden Arbeit schlagen wir mit der partiellen Distanzkorrelation eine Verallgemeinerung dieser Methode vor, welche für Einflüsse anderer Variablen korrigiert und so nicht-lineare Zusammenhänge findet sowie direkte von indirekten Beziehungen unterscheidet.

Appendix C

Ehrenwortliche Erklarung

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe.

Berlin, April 2015