Comparative Evaluation of different Graphical Models for the Analysis of Gene Expression Data

Doctoral Thesis

Submitted to

the Department of Statistics

of the University Dortmund

in Fulfilment of

the Requirements for the Degree of

'Doktor der Naturwissenschaften'

by

Marco Grzegorczyk

from Waltrop

Dortmund, June 2006

Supervisor: Prof. Dr. Wolfgang Urfer

2^{nd} Referee: Prof. Dr. Claus Weihs

Date of oral examination: 24 August 2006

Acknowledgements

First of all, I would like to acknowledge the strong support, encouragement, and inspiration of my supervisor Prof. Dr. Wolfgang Urfer who guided me throughout my research.

In addition to my supervisor, I am deeply grateful to my colleagues at work Klaus Jung and Nina Kirschbaum for their encouragement and lots of useful discussions during the years as well as to Brigitte Koths, Eva Brune, and Magdalena Thöne for their help.

Outside University Dortmund, I would like to thank Dr. Dirk Husmeier and Adriano Werhli (BIOSS, Edinburgh, Scotland). I have benefited greatly from Dr. Husmeier‘s experience and advices as well as from interesting discussions we had during my visit at the BIOSS institute. Especially with Dr. Dirk Husmeier‘s PhD-student Adriano Werhli I had a very good collaboration during the last year of my reserach.

Further thanks go to Dr. Nick Fieller and Clare Foyle (Department of Probability and Statistics, University of Sheffield, United Kingdom) for improving the English of my technical reports.

My thanks also go to Dr. Karsten Quast (Boehringer Ingelheim Pharma GmbH & Co. KG, Biberach, Germany) who made available microarray gene expression for my research.

Especially, I am deeply grateful for the financial support of my research by the Forschungsband: 'Molekulare Aspekte der der Biowissenschaften/ Biologisch-Chemische Mikrostrukturtechniken für biomolekulare Physiologie' Dortmund as well as for the financial funding provided by the Graduiertenkolleg Statistik which enabled me to participate in serveral interesting conferences and workshops.

Last but not least, on a personal note, I would like to thank all members of my family for their encouragement and support.

Dortmund, June 2006 Marco Grzegorczyk

Contents

1. Introduction

During the last decade the development of high-throughput postgenomic biotechnologies has resulted in the production of exponentially expanding quantities of biological data, such as genomic, proteomic, and metabolomic expression data. Along with the increasing amount of available data, a lot of novel statistical methods for analysing this new type of data have been developed and proposed in the literature.
One of the main and probably first addressed issues is to compare the expressions in different tissue types, such as healthy and cancerous cells, to detect which cell components are differentially expressed and therefore possibly associated with different phenotypes, such as diseases. In the context of gene expression data this kind of analysis is referred to as 'Determination of differentially expressed genes' and its main goal is to identify the genes whose disregulation, e.g. up- or down-regulation, leads to diseases. The practical advantage of identifying these genes is straighforward: Differentially expressed genes can be seen as candidates for useful points of application for pharmaceutical treatments and/or diagnostic tests. Interesting publications on this field of research are given by: [37], [38], [48], and [17] among others.

Another as important but more fundamental issue in the context of gene expression data is to understand how the expressions of genes are regulated, and to identify the relationships and interations between genes. Traditional approaches to systems biology and functional genomics are based on mathematical description of putative pathways in terms of coupled differential equations with the objective to obtain a deeper understanding of the exact nature of the regulatory circuits and their regulation

mechanisms. However, the availability of high-throughput postgenomic data of different nature has recently prompted substantial interest in reverse engineering the network and pathways in an inferential way from the data themselves. So, the idea behind this kind of analysis is not to compare the expressions of genes in different tissues, but to extract the relationships between the genes from data taken from one special cell type. The final goal is to understand the mechanisms that sustain life and/or cause genetic diseases by pathologically relevant mutations.

From a molecular biological point of view it is known that such complex systems can be analysed as a whole entity only. Because such complex systems exhibit a behaviour that is hardly explainable from the properties of its individual molecules, and consequently, modelling single components separately does not lead to useful results. For that reason several more and less sophisticated machine learning reverse engineering methods, such as Bayesian networks (e.g. [22]), Gaussian graphical models (e.g. [43]), and Relevance networks (e.g. [7]), have been applied to gene (protein and metabolite) expression data. But although such reverse engineering approaches raise the question of how much confidence one can have in networks reconstructed from sparse and noisy gene expression data, only serveral publications about assessing the accuracy of reverse engineering can be found in the literature. One of the first evaluations was carried out by [44]. The authors simulated a complex biological system at different levels of organisation, involving behaviour, neural anatomy, and gene expression of songbirds. They then tried to infer the structure of the known true genetic network from the simulated gene expression data with Bayesian networks. In a related study, [26] evaluated the accuracy of reverse engineering gene regulatory networks with Bayesian networks from data simulated from realistic molecular biological pathways, where the latter were modelled with a system of coupled differential equations. This network was also used in an earlier study by [54], who investigated the inference accuracy of deterministic linear and log-linear models. While all three publications certainly shed some light on the accuracy of reverse engineering in systems biology, they only investigated a particular inference method and hence do not allow a cross-method comparison between different

machine learning approaches.
In order to adress this shortcoming, an evaluation study was carried out by Pournara (see [39]). The author compared Gaussian graphical models and Bayesian networks on synthetic data generated from networks with random structures and different gene regulation mechanisms, where the latter differed with respect to the cooperative or competitive interactions between transcription factors regulating the same gene.

The focus of the research presented in this doctoral thesis was motivated by and is based on the ideas of Pournara (see [39]), but improves this earlier work in lots of important aspects. So for example a further learning method: the approach of Relevance networks has been included and, in order to capture uncertainty inherent in learning from sparse and usually noisy expression data, the most modern machine learning algorithms, such as MCMC sampling schemes for Bayesian networks or a novel shrinkage estimator for Gaussian graphical models, have been implemented and applied. Another important improvement follows from the fact that the present cross-method comparison is not only based on synthetic data generated from random network structures, but also includes real expression data gathered from the cytometric protein signalling network which is well-known and described in detail in the systems biology literature. In addition, certainly further synthetic data sets had to be generated, but thereby the same biologically realistic network topology was utilised. Finally, as more and more often interventional data, that is data in which the expression of some nodes (genes, proteins, etc.) are up- and/or downregulated by experimental conditions, are collected in systems biology, it has been distinguished between pure observational and interventional test data sets. Especially, Bayesian networks can deal with and benefit from interventions, so that much more clues about the causal direction of the interactions between the nodes can be revealed.

In this context a further aspect of this doctoral thesis is a detailed comparison between two different stochastic models (scoring metrics) for Bayesian networks. A lot

of publications can be found in the literature which describe the application of one of these models to expression data, and usually there is only few or even no justification for the choice of the stochastic model. Researchers using the continuous Gaussian Bayesian network model which can not model any non-linear relationships in the data often simply argue that a discretisation would incur too much information loss. On the other hand researchers using the discrete multinomial Bayesian network model argue that there may be non-linearity in the data, so that the more flexible modelling tool is preferable. But as no comparison between the performance of these two stochastic models for Bayesian networks can be found in the literature either, the practical meaning of this theoretical difference has never been investigated in detail. To fill this gap the cross-method evaluation mentioned above was extended to such a Bayesian network specific cross-model comparison (see Section 5.6).

Another aspect of this thesis is a case study dealing with the identification of interacting genes in a gene expression data set taken from healthy human kidney cells. The data set was made available by the German company Boehringer Ingelheim Pharma GmbH & Co. KG (Biberach, Germany) for a confidential analysis, and in accord with the company a strategy for the analysis was mapped out. The strategy and some results of this analysis of real expression data using Bayesian network methodology are presented in Chapter 4.

Because no satisfactory software program for the different Bayesian network machine learning approaches and models is available to date, all algorithms described in this thesis have been implemented using the programming language provided by the software package Matlab developed by the Mathworks company. Much time had to be spend on these implementations, as Markov Chain Monte Carlo sampling of Bayesian network is a complex task. Especially, Bayesian network inference via MCMC is computational very expensive, so that efficient implementations had to be used. Although the developed library of self-written Matlab programs is not a part of this thesis, it is planed on making it freely available on the internet sometime.

This doctoral thesis is organised as follows:
Chapter 2 provides brief introductions to the most fundamental concepts and methods of molecular biological research, such as introductions to cell biology and systems biology (2.1), to Affymetrix GeneChip biotechnology (2.2), and to gene regulatory networks from a biological point of view (2.3).

Afterwards in Chapter 3 detailed statistical descriptions of the different graphical models (reverse engineering machine learning methods) which have been compared in the evaluation study are given. More precisely, after two sections devoted to the Information bottleneck algorithm for data discretisation (3.1) and a general introduction to gene networks from a more mathematical point of view (3.2), the statistical theory behind the reverse engineering machine learning methods: Relevance networks (3.3), Gaussian graphical models (3.4), and Bayesian networks (3.5) are presented, whereby for each method the most modern learning algorithms are included. The theory chapter closes with two sections describing the concept of the Receiver Operator Characteristic (3.6) and two synthetic data generators (3.7).

Details on the kidney cell gene expression data analysis are given in Chapter 4. Especially, some results are presented that have not been published yet.

Chapter 5 is dedicated to the comparative evaluation study, that is the main part of this thesis. After a more detailed motivation of the study (5.1), follows a description of the cytometric signalling network (5.2). Subsequently, an example data set is used to illustrate of some statistical aspects (see 5.3), which then - due to space limitations - had to be omitted in the subsequent sections. Finally, the actual evaluation study is described in the last three sections. Precisely, after Section 5.4 presenting the applied parameter settings, there is a presentation of some results of a kind of screening experiment that has been used to extract two more special questions which are then issued in the nest two sections. Section 5.6 is dedicated to an extensive cross-method comparison between Relevance networks, Gaussian graphical models and Bayesian networks. And Section 5.7 compares the two different stochastic models for Bayesian networks in detail. Finally,

the most important results of the screening experiments as well as the two more detailed cross-method comparisons are briefly summarised in Section 5.8.

Chapter 6 concludes the thesis with a discussion of the results and gives an outlook to future work.

2. Molecular biological concepts and methods

This second chapter provides some brief introductions to the most fundamental concepts and methods of molecular biological research. Thereby not only the well-known fundamentals of cell biology are described, but also some methods and principles of modern research in the field of sytems biology are adressed. Due to space limitations the descriptions are restricted to the most important aspects, which are necessary and sufficient for understanding the basic idea of reverse engineering regulatory networks from expression data.

2.1. Cell biology and systems biology

Within this section a quick introduction to systems biology (functional genomics) is given. The introduction focuses on some molecular biological aspects that are relevant for the statistical analysis of expression data. Thereby all biological aspects are predominantly seen as information transfer processes only.

All processes within biological cells are regulated by interactions between DNA, mRNA, proteins, and metabolites. These cellular processes are necessary to enable the cells of an living organism to differentiate to specialised cells during development, and to response to different environmental conditions during lifetime. The characteristics (*phenotype*) of an organism depend on the phenotypes of its cells, and the information

necessary for the development of the phenotype of a cell is encoded in molecular units referred to as *genes*.

Omitting almost all biological details, the human DNA (*desoxyribonucleic acid*) consists of 23 double stranded DNA molecules which are organized as chromosomes and carry the complete genetic information (genome) of a living cell. Thereby the different DNA sequences on the chromosomes can be seen as molecular units and are referred to as *genes*. Each of these genes contains the information (code) for synthesizing functional molecular units called *proteins*. The synthesis of a protein from a gene is regulated by control mechanisms at different stages, such as transcription, RNA splicing, translation and post-translational modifications. The process of building a mRNA (*messenger ribonucleic acid*) copy of the code of a gene is called *transcription*, and is started by the binding of a *transcription factor* to the DNA sequence of that gene. During the *transcription* the instruction for the creation of a protein is transferred from DNA to mRNA. Subsequently, in the *translation* process the mRNA copy is used for the synthesis of a new protein. Thereby to each codon (triple of nucleotides) on the mRNA a special amino acid is matched, and the resulting composed sequence of amino acids is the synthesised protein. For short, each gene is a sequence on a DNA molecule, from which a complex molecular machinery within cells can read information necessary for manufacturing a particular type of protein. The mRNA is a transcribed copy of the information carried by a DNA sequence (gene) which is used to move the information contained in the DNA to the translation machinery, where finally the corresponding protein is synthesised.

The proteins which are synthesised by genes not only serve as transcription factors which bind to regulatory sites of other genes, but also as enzymes for metabolic reactions, or as components of signal pathways. So, proteins can be seen as the main functional components within living cells. Especially, the binding of proteins to the cis-regulatory domain of other genes usually leads to the synthesis of other specific proteins by these genes, so that in the end complex molecular pathways come into being. These pathways regulate the major functions of living cells, whereby especially the proteins are necessary for cell life processes. Although genes do not interact directly

with each other within these pathways, their *expressions* (activities) indirectly, that is over the synthesis of proteins, dictate the expressions of other genes. More precisely, it depends on the concentrations of the different synthesised proteins as well as on the presence of metabolites in the cells to which extend the expression of another gene is influenced. So, the expression of each different gene is a complex process, regulated through both indirect interactions with other gene‘s expressions and other cell component‘s concentrations.
Almost all cells in a living organism contain the same *genome* (set of genes), nonetheless the syntesised protein concentrations and so the phenotype can be totally different. For this reason it is clear that the cellular regulation strongly depends on the expressions of the genes (*gene profile*).

The biological principle of *systems biology* in general is to understand the relationships between all these cell components, and to explain which responses are given by these regulatory mechanisms to different cellular conditions. Thereby it is rarely possible to measure all cell components simultaneously. Usually only gene expressions, or protein concentrations, or metabolom concentrations can be measured by biological experiments. But as these cellular components interact (at least) indirectly with each other, such measurements are very useful to shed some light into the cellular regulatory meachanisms. In particular, the gene expressions in each living cell regulate the production of proteins, that is the final expression of the genetic information, which then regulates almost all cellular processes in biological systems.

One of the most powerful technologies for monitoring the amount of mRNA transcripts for ten thousands of genes within a cell is briefly described in the next Section 2.2. And the last Section 2.3 is dedicated to *gene networks* which can be used in molecular biology to describe the complex regulatory mechanisms exclusively on the gene level, whereby proteins and metabolites are omitted from consideration. More precise and detailed descriptions of the basic molecular biology concepts can be found in [1] and [4].

2.2. Affymetrix genechips

A lot of different technologies have been recently developed for gathering gene expression data. One of the most powerful and famous such biotechnologies are *Affymetrix genechips*. In this section a very brief description of Affymetrix genechips, which were introduced in 1996 (see [31]), is given.

Affymetrix genechips belong to the type of high density oligonucleotide microarrays which are printed using a lithographic masking process, and allow the monitoring of expression levels for ten thousands of genes in a cell simultaneously. The idea behind oligonucleotide arrays is to measure the amount of mRNA transcripts in a solution extracted from a cell, whereby the cell is the experimental sample of interest in this context. As each mRNA transcript is the copy of a DNA sequence (gene) which contains the code for a protein which was obviously going to be synthesised within the experimental cell, the amount of mRNA transcripts reflects the expression (activity) of the corresponding gene.

A high density oligonucleotided array consists of ten thousands of orderly arranged *spots*, each containing many copies of a unique probe, that is a set of chemically synthesised short cDNA (*complementary desoxyribonucleic acid*) sequences (synthetic *oligonucleotides*) that can bind specific target mRNA molecules in a solution. Thereby, as for each sequenced gene specific cDNA target molecules are known, each spot can be designed to bind only the specific mRNA molecule transcripts which belong to one single gene. Usually, the collection of the gene specific spots on an array is arranged, so that all spots on the array collectively represent the entire genome of a cell. For example, Affymetrix chips are designed for representing the genome of human cells.

If mRNA transcripts are isolated from an experimental cell, labelled with fluorescent tags, and hybridized to such oligonucleotide arrays, the cDNA on each spot binds its complementary mRNA target-molecules, that is the labelled mRNA transcripts of

the gene for which the spot was designed. Afterwards due to the fluorescent tags, for each spot the amount of transcript in the solution that was binded can be scanned using a fluorescence camera. The measured fluorescent intensities reflect the amount of binding and so the amount of mRNA transcripts in the solution extracted from the experimental cell. Ideally, there is a one-to-one correspondance between array spots and genes, so that each of these measurements represents the detected amount of a specific mRNA transcript, which in turn can be interpreted as the expression of one specific gene. For short, Affymetrix genechips can be used to measure the expression levels of the genes in a human cell by measuring the amount of mRNA transcripts in it.

However, as there is also nonspecific background binding, that is binding between the cDNA on the array and mismatching mRNA from the experimental cell, nonspecific background binding can falsify the measurements. Therefore, usually each gene is represented by 14-20 spots on the array and each of these spots not only contains cDNA oligonucleotides being a *perfect match* (PM) to the corresponding mRNA, but also *mismatching* (MM) cDNA oligonucleotides. Thereby, mismatching cDNA is synthetically created by substituting one single nucleotide in the central position of the corresponding perfect match cDNA sequence of nucleotides. The difference between the amount of binding to the perfect match (PM) and to the mismatching (MM) cDNA can be interpreted as the amount of binding being exclusively due to specific binding, as the amount of background binding should be the same for the perfect match and the mismatching cDNA. Usually, the average of the differences between the amounts of binding to the perfect match and the mismatching cDNA for all (14-20) spots, being designed for the same gene, are outputed as the final expression level measurement of this gene.

Further details on the technological aspects involved in microarrays can be found in [31], [30], and [35].

2.3. Gene networks

A *gene network* is a graphical representation of interactions between genes. Thereby these relations between genes are usually represented as if the expressions of genes would directly affect the expression levels of other genes. It is not explicitly mentioned that the interactions between genes are actually mediated by proteins, metabolites or other protein-metabolite-complexes as described in Section 2.1. From this point of view, gene networks must be interpreted as a very rough simplification of the real molecular biological regulatory mechanisms within cells. But on the other hand, gene networks are capable of representing the indirect interactions between genes, that is the final effect of the activity of one gene to the other‘s activities, whereby the exact detailed molecular biological mechanisms are omitted. And since these omitted molecular biological mechanisms are often still unknown anyway, at least the known relations on the gene levels can be described in a concise way. So, especially if bearing in mind that all relations between gene expressions always depend on proteins and metabolites, gene networks can be interpreted as if the mediating interactions on the proteome and metabolome level were implicitely represented within them. The focus of gene network representations is simply on how changing expressions of genes are related to the expressions of other genes without raising the claim to describe the mediating paths.

Therefore, and especially as often only gene expression data are available, e.g. from microarray based measurements (see Section 2.2), gene networks can be seen as a first but important step towards uncovering the complete biochemical regulatory mechanisms in cells. Once the relations between the genes are known, biologists can search for the mediating paths between these indirect gene-gene relations, and add the newly-discovered regulatory details on the proteome and metabolome level to the gene networks already available.

In analogy *protein* and *metabolite networks* are graphical representations of interacting proteins or metabolites, whereby all intermediating components are omitted in the graphical representations.

3. Statistical theory

In this chapter the mathematical details for all methods, applied for the present doctoral thesis, are given. First, in Section 3.1 the Information bottleneck algorithm for data discretisation is described. Afterwards a brief introduction to graphical models in general is given in Section 3.2. Different graphical models and the corresponding machine learning methods for reverse engineering gene regulatory networks with these models are presented in Sections 3.3, 3.4, and 3.5. As Bayesian networks are usually sampled with Markov Chain Monte Carlo (MCMC) simulations Subsection 3.5.3 deals with two different MCMC sampling schemes. The next Section 3.6 focuses on some ROC curve based criteria for assessing the goodness of performance of reverse engineering methods. Finally, two simulation methods for generating synthetic network data are described in Section 3.7.

3.1. The Information bottleneck algorithm

Although discretisation of data always incurs a certain information loss, it is often necessary to discretise data sets for applying statistical methods which are based on discrete observations, such as discrete Bayesian networks with multinomial node distributions. In gene expression data the discretisation usually contains three values, under-expressed ('-1'), not differentially expressed ('0') and over-expressed ('+1'), depending on whether the expression rate is significantly lower than, similar to, or higher than 'control', respectively. One simple way to discretise the values of continuous variables to these three levels is the application of *quantile discretisation*. For each

domain variable X_i $(i = 1, ...n)$ the lowest third of the values is labeled state '-1', the next third is labeled state '0' and the highest third is labeled state '+1'. More generally, quantile-discretisation can be used to discretise continuous variables into any number of discrete levels.

Nevertheless, more suitable is the application of an information-preserving discretisation procedure, i.e. a discretisation procedure which retains as much information about the dependencies between the domain variables as possible. Instead of considering each variable independently during the discretisation, information-preserving algorithms choose discretisation levels by considering all domain variables simultaneously. This section focuses on an extension of the agglomerative Information bottleneck algorithm which was first applied in the context of gene expression data by [22]. This algorithm chooses levels for each variable in terms of the mutual information between pairs of variables. The goal is to minimize the total pairwise information loss.

As the mutual information is defined for discrete variables only, the Information bottleneck algorithm requires the application of an initial discretisation procedure first. For the remainder of this section is it assumed that the continuous domain variables X_i $(i = 1, ...n)$ have been independently discretised into $M \in \mathbf{N}$ (M>3) levels using *quantile discretisation*.

The non-negative pairwise mutual information $MI(X, Y)$ between two (discrete) variables X and Y each with M different discrete levels out of the set $\{1, ..., M\}$ is defined as follows:

$$MI(X,Y) = \sum_{i=1}^{M}\sum_{j=1}^{M} P(X = i, Y = j) \cdot log_2\left(\frac{P(X = i, Y = j)}{P(X = i) \cdot P(Y = j)}\right) \tag{3.1}$$

and can be empirically estimated by replacing the theoretical probabilities through the corresponding portions of a sample of size m: $(x_u, y_u)(u = 1, ..., m)$ from the joint probability distribution $P^{X,Y}$ of X and Y.

For i and $j \in \{1, ..., M\}$ is valid:

$$\widehat{P(X = i)} = \frac{|\{u \in \{1, ..., m\} | x_u = i\}|}{m} \tag{3.2}$$

$$\widehat{P(X = i, Y = j)} = \frac{|\{u \in \{1, ..., m\} | (x_u, y_u) = (i, j)\}|}{m} \tag{3.3}$$

Plugging-in these empirical estimators given in Formulae (3.2) and (3.3) in Formula (3.1) leads to the estimator $\widehat{MI(X, Y)}$, whereby it is necessary to define: $0 \cdot log_2(0) := 0$.

The *mutual information score* $\widehat{S_w}$ for a variable X_w out of the set $\{X_1, ..., X_n\}$ is defined as the sum of the (empirical) pairwise mutual information values between X_w and the other $n - 1$ variables:

$$\widehat{S_w} = \widehat{S(X_w)} = \sum_{\substack{v=1 \\ v \neq w}}^{n} \widehat{MI(X_w, X_v)} \tag{3.4}$$

The Information bottleneck algorithm is a stepwise procedure consisting of two loops. Stepwise (outer loop) for each variable (inner loop) some neighbouring pair of discretisation levels are coalesced into one single level, reducing the number of discretisation levels of all variables by one in each step of the outer loop. More precisely, the inner loop iterates over each of the variables $X_1, ..., X_n$ to determine for each of these variables simultaneously which single coalescence of neighbouring levels reduces the mutual information score (between that variable and the n-1 others to be also discretised in this outer step) the least. The outer loop is finished when for each variable all observations have been discretised into three levels.

In the first step, coalescing the neighbouring discrete levels u and $u + 1$ ($u \in \{1, ..., M - 1\}$) of variable X_w to a new discrete level U leads to the following mutual

information loss:

$$L(X_w, u) = \sum_{\substack{v=1 \\ v \neq w}}^{n} \sum_{j=1}^{M} P(X_w \in U, X_v = j) \cdot log_2\left(\frac{P(X_w \in U, X_v = j)}{P(X_w \in U) \cdot P(X_v = j)}\right)$$
$$- \sum_{\substack{v=1 \\ v \neq w}}^{n} \sum_{j=1}^{M} \sum_{t=0}^{1} P(X_w = u + t, X_v = j) \cdot log_2\left(\frac{P(X_w = u + t, X_v = j)}{P(X_w = u + t) \cdot P(X_v = j)}\right) \tag{3.5}$$

For each variable the algorithm coalesces the levels u_0 and $u_0 + 1$ with $L(X_w, u_0) \leq L(X_w, u)$ for all $u \in \{1, ..., M - 1\}$ at the end of the inner loop, before continuing with the next step of the outer loop. Consequently, as the coalescing is implemented at the end of the inner loop, the results of the outer loop steps do not depend on the order in which the variables are considered in the inner loop.
Stepwise (outer loop) the number of discrete levels for each variable reduces by one until only three levels per variable remain. For further information see [22].

3.2. Introduction to graphical models

Traditional approaches to systems biology are based on mathematical description of putative pathways in terms of coupled differential equations with the objective to obtain a deeper understanding of the exact nature of the regulatory circuits and their regulation mechanisms. However, the availability of high-throughput postgenomic data of different nature has recently prompted substantial interest in reverse engineering the network and pathways in an inferential way from the data themselves. That is, one major goal of modern biology research is to take large sets of biological data, usually correlational, and eludicate functional interactions between elements in a causal pathway network. Such efforts have led to developments and applications of different graphical model machine mearning inference methods to predict biological pathways. Usually, the goal is to learn about a model of the system not for prediction but for discovering the domain structure.

For example, one may want to understand the mechanism by which genes in a cell produce proteins, which in turn cause other genes to express themselves or prevent them from doing so. Hence, from a mathematical point of view the following situation is given. There are different biological variables X_i (i=1,...,n), e.g. each measuring the expression of a certain gene in a cell, and a measured sample of size m of these domain variables is available $(x_{1j}, ..., x_{nj})$ (j=1,...,m). That is x_{ij} is the value (e.g. expression) of the i-th domain variable (e.g. gene or protein) in the j-th sample. The statistical goal is to find all dependencies between these n domain variables. Lots of different graphical modelling frameworks for discovering these dependencies have been proposed in the literature. And although they are based on different statistical aspects and ideas, most of them lead to the same kind of result. That is, a *mathematical graph* representing all dependencies between the n domain variables. Such a mathematical graph consists of a set of nodes, whereby each node corresponds to a domain variable. And the edges between the nodes of the graph correspond to probabilistic dependencies between the domain variables. More precisely, the domain variables $X_1, ..., X_n$ can be considered as in one-to-one correspondence with the nodes $X_1, ..., X_n$ in a graph and the graphical structure (set of edges) determines the relationships between them. For that reason the terms 'variables' and 'nodes' can be used interchangeably in the context of graphical models.

The meaning of the edges is different for different modelling frameworks, and depends on the theoretical statistical idea behind these models. For example, in a Relevance network framework (see 3.3) an edge may simply mean that the corresponding two nodes (variables) are strongly correlated, while the same edge in a Bayesian network model usually has a much more complicated meaning (see 3.5). Different graphical model frameworks are described in detail in the next sections. Since all graphical models can be combined with different learning algorithms they will be synonymously referred to as (reverse engineering) machine learning methods in this thesis.

Especially, in Bayesian network methodology it can be distinguished between directed and undirected edges in a mathematical graph. That is using a Bayesian network model

for reverse engineering a gene network from an expression data set usually leads not only to connections (undirected edges) between the domain nodes, but also to some arcs (directed edges). These edges, pointing from one node to another, can be interpreted as causal relations. If there is an edge pointing from node A to node B, one may conclude that gene A causes gene B, e.g. gene A activates or inhibits gene B. On the other hand, an undirected edge indicates that the corresponding two genes are related in some joint biological regulation process or interaction, but there is no possibility to conclude about causality.

3.3. Relevance networks

The method of Relevance networks, proposed by [6], is exclusively based on pairwise association scores, and therefore represents a very simple machine learning method for reverse engineering regulatory networks. A suitable association score is computed for all pairs of domain variables X_i and X_j $(i, j \in \{i = 1, .., n\})$ from the values observed for these variables. Thereby, in the context of gene regulatory networks each variable usually corresponds to a gene, and its values to the measured expressions of this gene.

The authors propose the mutual information (see 3.1) as an appropriate association score. This requires a discretisation of the data, which can be carried out with the Information bottleneck algorithm (see 3.1). Alternatively, for continuous data the standard Pearson correlation coefficient can be used:

$$corr(x, y) = \frac{1/m \sum_{i=1}^{m}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{1/m \sum_{i=1}^{m}(x_i - \bar{x})^2}\sqrt{1/m \sum_{i=1}^{m}(y_i - \bar{y})^2}} \tag{3.6}$$

where $x = (x_1, ..., x_m)$ and $y = (y_1, ..., y_m)$ are the m-dimensional observations of two different domain variables with empirical means $\bar{x}$ and $\bar{y}$.

With regard to the graphical model representation the domain variables are interpreted

as the nodes of a network. So, as already pointed out before, they can be seen as in one-to-one correspondance to the network nodes (see 3.2). Each variable X_i represents a network node X_i (i=1,...,n).

The association scores are compared with a threshold parameter, and the nodes whose pairwise association score exceeds this threshold are linked by an undirected edge. In statistical terminology such a Relevance network based on the Pearson correlation is referred to as a 'covariance graph'. The threshold parameter can be estimated by a randomization test so as to keep the number of false positive edges below an a priori specified tolerance level. However, this approach is usually too conservative in that it discards too many true positive edges, and hence some explorative modification of the threshold parameter is usually required. For further information see [6].

3.4. Gaussian graphical models

A more complex machine learning method is given by Gaussian graphical models (GGMs). These models are based on the assumption that the observed data for the domain variables (i.e. nodes in the network) are distributed according to a multivariate Gaussian distribution $N(\mu, \Sigma)$. The (i,j)-th element $\Sigma_{i,j}$ in the covarianve matrix Σ is proportional to the correlation coefficient between node X_i and X_j. But a high correlation coefficient between two nodes must not necessarily indicate a direct causal association. Not rarely a high correlation coefficient may be due to an indirect association only, e.g. both nodes may depend on another network node. Consequently, a high correlation coefficient between two variables provides only weak evidence for a direct association. And actually only the direct dependencies between nodes are of interest for the construction of regulatory networks. To avoid this shortcoming of Relevance network methodology (see 3.3), partial correlations are considered in Gaussian graphical models instead. That is, the strength of a direct association between two nodes X_i and X_j is measured by the partial correlation coefficient $\pi_{i,j}$ which describes the correlation between these nodes conditional on all the other network nodes. From the theory of nor-

mal distributions it is known that the partial correlation coefficients $\pi_{i,j}$ can be easily computed from the inverse $\Omega = \Sigma^{-1}$ of the covariance matrix Σ [12]. More precisely, it holds:

$$\pi_{i,j} = \frac{-\omega_{i,j}}{\sqrt{\omega_{i,i} \cdot \omega_{j,j}}}, \tag{3.7}$$

whereby $\omega_{i,j}$ are the elements of matrix Ω.

Hence, in order to reconstruct a Gaussian graphical model from a given data set D, one typically employs the following procedure. From the data set D, the empirical covariance matrix is estimated and inverted to obtain Ω, subsequently the entries $\pi_{i,j}$ of the partial correlation matrix Π can be computed using Formula (3.7). Afterwards the interpretation is as follows: Small elements $\pi_{i,j}$ in the resulting partial correlation matrix Π correspond to weak partial correlations, and the corresponding nodes become not connected by an edge. On the other hand, high entries correspond to strong partial correlations, so that there is reason to believe that a *direct* association between the corresponding two nodes.

The disadvantage of this procedure, is that the empirical covariance matrix can only be inverted if the number of observations exceeds the number of nodes in the network. This condition is usually not satisfied for many real applications in systems biology, such as reverse engineering gene regulatory networks with microarray data.

In order to learn a Gaussian graphical model from a data set in which the number of variables exceeds the number of observations, i.e. a singular covariance matrix is given, [42] and [43] introduced the following modified schemes:

3.4.1. Small sample point estimation of partial correlations

In [42] the authors propose three conceptually simple methods to obtain estimates of the partial correlation coefficient matrix Π for Gaussian graphical models from sparse data, that is data, where the number of nodes exceeds the number of observations ($n > m$).

They show that these estimators can be used to infer regulatory networks with high accuracy from such sparse data.

- First, they propose to use the Moore-Penrose pseudoinverse, which is a generalisation of the standard matrix inverse, but can also be applied to singular matrices. That is the inverse of the covariance matrix Σ^{-1} is simply replaced by the Moore-Penrose inverse Σ^{+}. For non-singular matrices (given if $n < m$), the Moore-Penrose pseudoinverse is equal to the standard matrix inverse Σ^{-1}. And for singular matrices it can be computed as follows: $\Sigma^{+} = V(E^T E)^{-1} E U^T$, where $\Sigma = UEV^T$ is the singular value decomposition representation of the covariance matrix Σ. The final estimator $\widehat{\Omega}_1$ for Ω is given by:

$$\widehat{\Omega}_1 = \left(\widehat{\Sigma}\right)^{+}. \tag{3.8}$$

  The subsequent transformation of $\widehat{\Omega}_1$ (see Formula (3.7)) leads to the *observed partial correlation estimator*: $\widehat{\Pi}_1$.

- Second, they propose to estimate the covariance matrix Σ by using the mean of the covariance estimators $\widehat{\Sigma(D_b)}$ for B (e.g. $B = 1000$) different bootstrap samples with replacement $D_1, ..., D_B$ generated from the original data set D. Subsequently the mean of these bootstrap covariance matrix estimators can be inverted using the Moore Penrose inverse. The resulting estimator for Ω is given by:

$$\widehat{\Omega}_2 = \left(\frac{1}{B} \cdot \sum_{b=1}^{B} \widehat{\Sigma(D_b)}\right)^{+}. \tag{3.9}$$

  Using the Transformation 3.7 yields the *partial bagged correlation estimator*: $\widehat{\Pi}_2$.

- Third, they propose to invert each bootstrap sample estimate $\Sigma(D_b)$ (b=1,...,B) using the Moore Penrose inverse, and to estimate Ω by the mean of theses bootstrap estimates. The resulting estimator $\widehat{\Omega}_3$ is given by:

$$\widehat{\Omega}_3 = \frac{1}{B} \cdot \sum_{b=1}^{B} \widehat{\Sigma(D_b)}^{+}. \tag{3.10}$$

Using the Transformation 3.7 yields the *bagged partial correlation estimator*: $\widehat{\Pi}_3$.

3.4.2. Regularized estimation of partial correlations

In a more recent publication ([43]) the same authors present an alternative novel regularized covariance estimator (*shrinkage covariance estimator*) which is based on the concept of shrinkage and exploits the Ledoit Wolf lemma [29] for analytic calculation of the optimal shrinkage. This novel shrinkage estimator $\widehat{\Sigma}_4$ is guaranteed to be non-singular, so that it can be inverted to obtain a new estimator $\widehat{\Omega_4} = (\widehat{\Sigma_4})^{-1}$ for the matrix Ω, and is based on the following theoretical idea. It is known that the (unconstrained) maximum likelihood estimator $\widehat{\Sigma}_{ML}$ for the covariance matrix Σ has a high variance if the number of nodes exceeds the number of observations ($n > m$). On the other hand there are lots of possible constrained estimators that have a certain bias but a much lower variance. The shrinkage approach combines the maximum likelihood estimator with one of these constrained estimators $\widehat{\Sigma}_C$ in a weighted average:

$$\widehat{\Sigma}_4 = (1 - \lambda)\widehat{\Sigma}_{ML} + \lambda\widehat{\Sigma}_C, \tag{3.11}$$

where $\lambda \in [0, 1]$ denotes the shrinkage intensity. The authors show that this regularized estimator outperforms both single estimators in terms of accuracy and statistical efficiency. Furthermore they show that the Ledoit Wolf lemma can be used to estimate the otimal shrinkage intensity λ, and recommend to restrict the constrained estimator $\widehat{\Sigma}_C$ by assuming that the network variables (nodes) are pairwise uncorrelated ($\Sigma_{i,k} = 0$ for $i \neq k$) but may have unequal variances ($\Sigma_{i,i} \neq \Sigma_{k,k}$ for $i \neq k$). Omitting the technical details which can be found in [43] the authors show that the optimal *shrinkage covariance estimator* $\widehat{\Sigma}_4 = (\widehat{\Sigma}_4)_{i,j}$ is then given by:

$$(\widehat{\Sigma}_4)_{i,j} = \left\{ \begin{array}{ll} s_{ii}^2, & i = j \\ s_{ij}^2 \cdot min\left\{1, max\left\{0, 1 - \widehat{\lambda^\star})\right\}\right\}, & i \neq j \end{array} \right\} \tag{3.12}$$

whereby the optimal shrinkage is given by:

$$\widehat{\lambda^\star} = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \widehat{Var(r_{ij}^2)}}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (r_{ij}^2)^2}. \tag{3.13}$$

In Formula (3.12) s_{ij}^2 is the empirical covariance between variables X_i and X_j:

$$s_{ij}^2 = \frac{1}{m-1} \sum_{k=1}^{m} (x_{ik} - \overline{x_i})(x_{jk} - \overline{x_j}) \tag{3.14}$$

and r_{ij}^2 is the corresponding empirical correlation:

$$r_{ij}^2 = \frac{s_{ij}^2}{\sqrt{s_{ii}^2 \cdot s_{jj}^2}} \tag{3.15}$$

The variances of the correlations in Formula (3.13) can be estimated as follows:

$$\widehat{Var(r_{ij}^2)} = \frac{m}{(m-1)^3} \sum_{k=1}^{m} (w_{kij} - \overline{w_{ij}})^2 \tag{3.16}$$

$$\text{with: } w_{kij} = \left(\sqrt{s_{ii}^2 \cdot s_{jj}^2}\right)^{-1} \cdot (x_{ki} - \overline{x_i})(x_{kj} - \overline{x_j})$$

$$i, j \in \{1, ..., n\} \text{ and } k \in \{1, ..., m\} \text{ and } \overline{w_{ij}} = \tfrac{1}{m} \textstyle\sum_{k=1}^{m} w_{kij}.$$

In these equations x_{ik} (k=1,...,m) is the k-th observation of the i-th domain variable X_i (i=1,...,n).

Computing the inverse of $\widehat{\Sigma}_4$ and applying Formula (3.7) as usual leads to the *shrinkage* estimator $\widehat{\Pi}_4$ of the partial correlation coefficient matrix Π.

3.5. Bayesian networks

The most sopisticated machine learning method (graphical model) for reverse engineering gene regulatory networks that was applied within this doctoral thesis is the Bayesian network (BN) approach. Unlike the other modelling frameworks Bayesian networks (BNs) permit stochastic, combinatorial and non-linear relationships among domain variables. The probabilitic nature of these networks is capable of handling noise inherent in the biological process. Beyond it, a Bayesian network approach towards modelling regulatory networks is attractive because of its solid basis in statistics, which enables to deal with stochastic aspects of biological systems. Consequently, BNs are interpretable and flexible models for representing probabilistic relationships between multiple interacting variables. At a qualitative level, the graphical structure of Bayesian network describes the relationships between the domain variables in the form of conditional independence relations (see 3.5.1). At a quantitative level, (local) relationships between variables are described by (conditional) probability distributions (see 3.5.2). Formally, a BN is defined by a graphical structure G, a family of (conditional) probability distributions F, and their parameters q, which together specify a joint distribution over all domain variables. Two different Markov Chain Monte Carlo (MCMC) schemes for learning Bayesian networks (BNs) from data by a model-averaging approach are presented in Subsection 3.5.3.

3.5.1. Introduction to Bayesian networks

A Bayesian network is defined by a triple (G,F,q), whereby G is the graphical structure, F is a family of probability distributions, and q the set of parameters for the family F. This subsection focuses on the graphical structure G.

The graphical structure G of a Bayesian network consists of a set of n nodes $X_1, ..., X_n$ and a set of directed edges between these nodes. Thereby as usual each node X_i represents the corresponding random domain variable X_i (i=1,...,n), while the directed edges indicate conditional dependence relations between these variables. If there is a directed edge pointing from node X to node Y, then X is called a *parent* (node) of Y, and Y

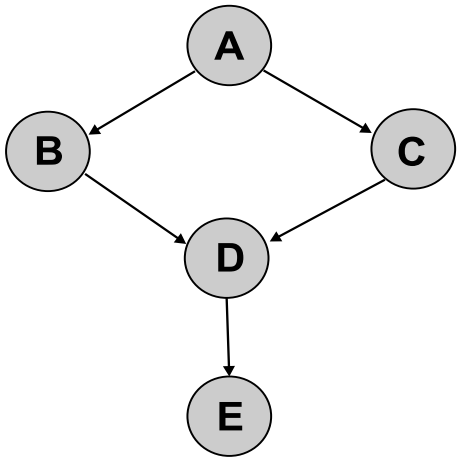


Figure 3.1.: Example of a Bayesian network (DAG) with 5 nodes

is called a *child* (node) of X. And if a node Z can be reached by following a path of directed edges, starting at node X, Z is called a *descendant* of X, while X is called an *ancestor* of Z. An important feature of these graphical structures of Bayesian networks (BNs) is that there is no path of directed edges leading from a node to itself. That is no node can be one of its own ancestors or descendants. This means that the graphical structure has to be a directed acyclic graph, called DAG, without paths, such as $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_1$. Due to this characteristic of G the joint probability distribution $P(X_1, ..., X_n)$ in Bayesian networks can be factorised into a product of simpler distributions (see below). For example in Figure 3.1, where an example of a simple DAG for a Bayesian network over five domain variables is given, the nodes B and C are parents of node D, node E is a child of node D, and node E is a descendant of all four other nodes. That is A, B, C, and D are ancestors of node D. The set of all parents $pa(X)$ of a node X, is simply defined as the set of all nodes from which an edge points to node X. If node X and node Y have one or more common children, Y is called a *coparent* of X and vice versa. Considering again the DAG in Figure 3.1, node B and node C are coparents of each other, as they have a common child D. The parent sets are given by: $pa(A) = \{\}$, $pa(B) = pa(C) = \{A\}$, $pa(D) = \{B, C\}$, and $pa(E) = \{D\}$.

The dependency structure in Bayesian networks is based on the concept of the *Markov blanket*. That is, the conditional distribution of a variable X_i, given the other n-1 variables, just depends on the nodes in the Markov blanket $M(X_i)$ of node X_i. And the Markov blanket $M(.)$ of a node is the set of its children, parents, and coparents. That is for i=1,...,n:

$$P(X_i|X_1, ..., X_{i-1}, X_{i+1}, ..., X_n) = P(X_i|M(X_i)).$$

More precisely conditioning on its children in $M(X_i)$ renders X_i independent from its other descendants, and conditioning on its parents in $M(X_i)$ renders X_i independent from its other ancestors. Furthermore X_i depends on a coparent only, if on one of their common children is also conditioned. Otherwise X_i and the corresponding coparent are independent of each other. As a first consequence, this yields that node X_i, given all its ancestors $X_{i1}, .., X_{il}$, just depends on its parents $pa(X_i)$. That is for i=1,...,n:

$$P(X_i|X_{i1}, ..., X_{il}) = P(X_i|pa(X_i)).$$

Using the theorem of the total probability on a suitable ordering of the domain variables, implied through a permutation σ:

$$P(X_{\sigma(1)}, ..., X_{\sigma(n)}) = P(X_{\sigma(1)}) \cdot \prod_{i=2}^{n} P(X_{\sigma(i)}|X_{\sigma(i-1)}, ..., X_{\sigma(1)})$$

the application of these characteristcs leads to the following factorisation of the joint probability distribution in Bayesian networks:

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|pa(X_i)) \tag{3.17}$$

For a detailed derivation of this formula see [27]. The factors in the product in Formula (3.17) are referred to as *local probability distributions*. For the DAG represented in Figure 3.1 the application of Formula (3.17) yields the following factorisation of the joint probability distribution:

$$P(A, B, C, D, E) = P(A) \cdot P(B|A) \cdot P(C|A) \cdot P(D|B, C) \cdot P(E|D)$$

The main advantage of the factorisation is that the joint probability distribution of the domain variables $X_1, ..., X_n$ becomes a product of simpler conditional probability distributions.

It can be summarised that directed acyclic graphs (DAGs) imply sets of (in-)dependence assumptions for Bayesian networks. But more than one DAG can imply exactly the same set of (in-)dependencies. For example the following two DAGs (G_1): $X_1 \rightarrow X_2$ and (G_2): $X_1 \leftarrow X_2$ over the two nodes domain X_1, X_2 both imply that the variables X_1 and X_2 are not stochastically independent. This leads to identical probability distributions for both DAGs:

$$\begin{aligned} P(X_1, X_2|G_1) = P(X_1) \cdot P(X_2|X_1) = P(X_1, X_2) = P(X_2) \cdot P(X_1|X_2) \\ = P(X_1, X_2|G_2) \end{aligned}$$

This means that the graphs G_1 and G_2 only show alternative possibilities of describing the same set of conditional independence relations. Consequently, the independence assumptions of a Bayesian network can not be uniquely represented by DAGs. If two DAGs over the same domain assert the same set of independence assumptions among the variables, those graphs are said to be *equivalent*. This relation of graph equivalence imposes a set of *equivalence classes* over DAGs. The directed acyclic graphs within an equivalence class have the same underlying undirected graph, but may disagree on the direction of some of the edges.

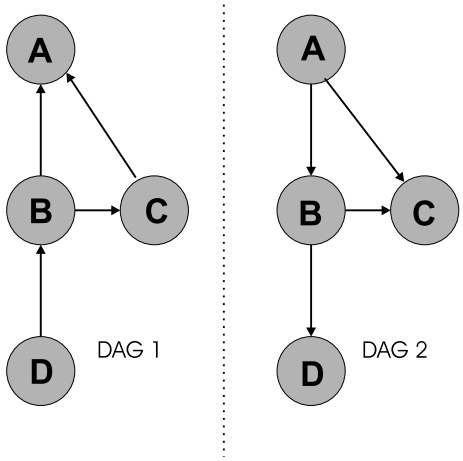


Figure 3.2.: Example of two equivalent DAGs

Figure 3.2 shows two DAGs over the domain $\{A, B, C, D\}$ which are equivalent, although there are three edges with opposite orientation. But in the end both graphs represent two independence relations only. The variables A and D as well as the variables C and D are independent conditional on variable B. All the other pairs of variables are immediately connected by an edge, and are therefore not stochastically independent. This can also be seen when using the factorisation rule of the joint probability distribution (see Formula (3.17)) for both DAGs:

$$P(A, B, C, D|DAG_1) = P(A|B, C) \cdot P(B|D) \cdot P(C|B) \cdot P(D)$$
$$P(A, B, C, D|DAG_2) = P(A) \cdot P(B|A) \cdot P(C|A, B) \cdot P(D|B)$$

because both factorisations can be easily transformed to: $P(A, B, C) \cdot P(D|B)$.
[49] proof that two directed acyclic graphs are equivalent if and only if they have the same *skeleton* and the same set of *v-structures*. The skeleton of a directed acyclic graph (DAG) is defined as the undirected graph resulting from ignoring all edge directions. And a v-structure denotes a configuration of two directed edges converging on the same

node without an edge between the parents (see [8]). More precisely, a v-structure in a DAG is an ordered triple of pairwise different nodes (X_i, X_j, X_k) with $(i, j, k) \in \{1, ..., n\}$ such that: (1) the DAG contains the directed edges $X_i \rightarrow X_j$ and $X_k \rightarrow X_j$, and (2) there is no edge between X_i and X_k. The two DAGs in Figure 3.2 have the same v-structures, namely none. The DAG in Figure 3.1 has the following set of v-structures: (B, D, C) and (C, D, B).

The two well-known scoring metrics for Bayesian networks (DAGs) derived in the literature (see 3.5.2) are *score-equivalent*, that is, lead to the same scores for equivalent DAGs. Consequently, using these metrics only equivalence classes can be learnt from data not individual DAGs within each class. But this restriction is not disavantageous. Far from it, score-equivalence is desirable, because equivalent DAGs assert the same set of conditional independencies, and therefore must be seen as equally expressive. Thus, the application of a non-score-equivalent criteria which arbitrarily prefers a DAG of an equivalence class is not useful. [8] shows that equivalence classes of DAGs can be uniquely characterised and represented using *completed partially directed acyclic graphs* (CPDAGs). CPDAGs contain both directed and undirected edges and cyclic in the sense that they contain no directed cycles. Every directed edge $X \rightarrow Y$ of a CPDAG denotes that all DAGs of this class contain this edge, while every undirected edge $X - Y$ in this CPDAG-representation denotes that some DAGs contain the directed edge $X \rightarrow Y$, while others contain the oppositely orientated edge $X \leftarrow Y$. Given a directed acyclic graph G the CPDAG representation of its equivalence class can be constructed efficiently. With respect to the v-structures it has to be decided for every directed edge if it is *reversible* or not. An edge of G is not reversible (*compelled*) if and only if this directed edge is present in every DAG G‘ equivalent to G. Otherwise the edge is reversible. Every edge participating in a v-structure is non reversible. But not every non reversible edge necessarily participates in a v-structure, because the reversal of such an edge can lead to other v-structures. As an example consider the edges of the DAG in Figure 3.1. As mentioned above the v-structures of this DAG are given by: (B, D, C) and (C, D, B). So, the edges $B \rightarrow D$ and $C \rightarrow D$ are compelled. Furthermore, although not participating in a v-structure,

the edge $D \rightarrow E$ is compelled, because its reversal would lead to four new v-structures, namely: (E, D, C), (C, D, E), (B, D, E), and (E, D, B). Table 3.1 lists the edges of all three DAG members of the corresponding equivalence class, and Figure 3.3 shows the CPDAG representation of this class, in which the edges $A \rightarrow B$ and $A \rightarrow C$ become undirected. The simultaneous reversal of both edges $A \rightarrow B$ and $A \rightarrow C$ of the DAG in Figure 3.1 would lead to new v-structures (B, A, C) and (C, A, B), and so to a DAG that is not equivalent to the three DAGs in Table 3.1. An algorithm that takes as input a DAG, and outputs the CPDAG representation of the equivalence class to which that DAG belongs, can be found in [9].

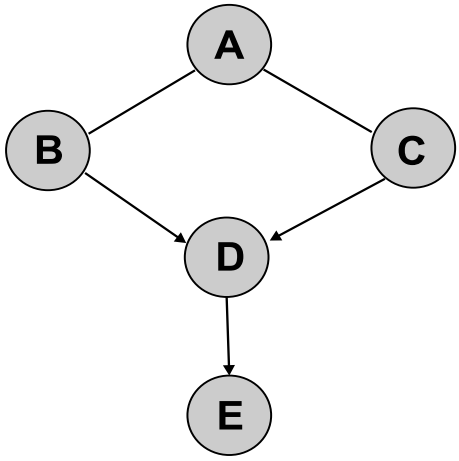


Figure 3.3.: The CPDAG of the DAG presented in Figure 3.1.

Graph	Reversible edges		Compelled edges		
CPDAG (Figufe 3.3)	$A - B$	$A - C$	$B \rightarrow D$	$C \rightarrow D$	$D \rightarrow E$
DAG_1 (Figure 3.1)	$A \rightarrow B$	$A \rightarrow C$	$B \rightarrow D$	$C \rightarrow D$	$D \rightarrow E$
DAG_2	$A \leftarrow B$	$A \rightarrow C$	$B \rightarrow D$	$C \rightarrow D$	$D \rightarrow E$
DAG_3	$A \rightarrow B$	$A \leftarrow C$	$B \rightarrow D$	$C \rightarrow D$	$D \rightarrow E$

Table 3.1.: Representation of the three DAGs of an equivalence class

3.5.2. Stochastic models for Bayesian networks

After having described Bayesian networks at a qualitative level through directed acyclic graphs (DAGs) in the last subsection, the two major stochastic models for Bayesian networks are presented now. These parametric models specify the distributional form F and the parameters q of the local probability distributions $P(X_i|pa(X_i))$ (i=1,...,n). That is, they assert a distribution to each domain node X_i in dependence of its parent nodes $pa(X_i)$. Thereby the set of parent nodes is implied through DAGs in Bayesian network methodology. Those local probability distributions together specify the joint probability distribution of all domain variables $P(X_1, ..., X_n)$ when the factorisation rule (see Formula (3.17)) is applied. Consequently, given a data set these parametric models can be used to score DAGs with respect to their posterior probabilities $P(DAG|data, F, q)$. Neglecting the parametrical parameters F and q the posterior probability of a directed acyclic graph G_0 given a data set D can be represented as follows:

$$P(G_0|D) = \frac{P(G_0, D)}{P(D)} = \frac{P(D|G_0) \cdot P(G_0)}{\sum_{G \in \Omega} P(D|G) \cdot P(G)}, \tag{3.18}$$

whereby P(G) ($G \in \Omega$) is the prior probability over the space Ω of all possible DAGs for the domain $X_1, ..., X_n$. $P(D|G)$ is the marginal likelihood, that is the probability of the graph G given the observed data D.

As the number of possible directed acyclic graphs (DAGs), that is the cardinality of the set Ω, grows exponentially with the number of domain nodes n, the denominator on the right hand side of Formula (3.18), which is a sum over the whole model space Ω, is not tractable for high n(>6). But the denominator does not depend on the directed acyclic graph G_0 itself. So, it is sufficient to consider the numerator of Formula (3.18) only, as it is proportional to the posterior probability (score) of G_0:

$$P(G_0|D) \propto P(D|G_0) \cdot P(G_0).$$

When the marginal likelihood of the data is interpreted as the integral over all possible parameter values q for the Bayesian network model (G_0, F, q), it can be derived:

$$P(D|G_0) \cdot P(G_0) = P(G_0) \cdot \int f(D, q|G_0))dq = \int f(D|q, G_0)f(q|G_0)dq, \qquad (3.19)$$

whereby the parameter vector $q = q(F, G_0)$, and especially its dimension depends on the distributional form F as well as on the graph G_0 which specifies the dependencies between the domain variables. Asserting a stochastic model specifies the functional form of the likelihood $f(D|q, G_0)$ and thereby the parameter space. The interpretation of the likelihood as an integral over the parameter space protects against data over-fitting, as it includes a penalty for model complexity. As the likelihood can be seen as an average probability of generating the data D over all possible parameter vectors q, it balances the ability of the Bayesian network model to explain the data with the ability to do so economically. Instead of estimating a single parameter-setting, e.g. the maximum-likelihood estimator of q, all possible parameter vectors are permitted in the prior distribution $f(q|G_0)$. Such an approach is well known in the field of statistics, and referred to as a Bayesian modelling approach (BMA). What follows is the definition of two different stochastic models (scoring metrics) that can be asserted and realised within this BMA setting (see Formula (3.19)). Thereby the term $P(D|G_0) \cdot P(G_0)$ in Formula (3.19) is denoted as the *score* of G_0. Furthermore, it is assumed that a data set D of m independent observation vectors $\vec{D_{.j}}$ of the n domain variables is given. $D = (\vec{D_{.1}}, ..., \vec{D_{.m}})$ whereby the matrix element D_{ij} is the value of the i-th domain variable in the j-th observation. The simplest prior distribution over DAGs $P(G)$ with $G \in \Omega$ is a uniform distribution. That is each DAG G has the same probability $P(G) = \frac{1}{|\Omega|}$. In this case the prior probabilities cancel out in Formula (3.18). Consequently, the graph prior could be ignored in all further derivations.

An alternative common prior over DAGs is the following one: For all $G \in \Omega$:

$$P(G) = \frac{1}{\pi} \prod_{i=1}^{n} \begin{pmatrix} n-1 \\ |pa(X_i)| \end{pmatrix}^{-1} \tag{3.20}$$

where π is a normalisation constant. This prior depends on the cardinalities of the parent sets in the DAG and is given by a product, where each factor corresponds to a domain variable. So, this prior over DAGs can be interpreted, as if it is decomposed into a product of priors over parent sets. One prior for each variable. Thereby the priors over parent sets depend on the cardinalities of these sets only. And for the prior over parent sets holds that the probability that a domain node has k parents is the same for each cardinality $k = 0, 1, 2, ...$, because

$$\sum_{pa(X_i):|pa(X_i)|=k} \begin{pmatrix} n-1 \\ |pa(X_i)| \end{pmatrix}^{-1} = 1.$$

This prior penalises DAGs in which the domain variables have parent sets of high cardinalities without loosing interpretability. Implicitly, it is assumed that the vector of cardinalities of parent sets $(|pa(X_1)|, ..., |pa(X_n)|)$ is uniformly distributed. With regard to the MCMC procedure (see Subsection 3.5.3.2 and Subsection 3.5.3.1) it is important to mention that not only the prior given in Formula (3.20) can be decomposed into a product where each factor corresponds to a domain variable, but also the uniform prior. The latter one can be simply represented as the following product: $P(G) = \prod_{i=1}^{n} |\Omega|^{-\frac{1}{n}}$. The factors in the factorisations of both graph priors are referred to as *local parent set priors* and will be denoted $\mathrm{P(pa(X_i))}$ in following formulae.

In the next two subsections two stochastic models for Bayesian networks will be described. For both models it will be shown that the likelihoods $P(D|G)$ for any directed acyclic graph G can also be decomposed into products. That is in analogy to the factorisation in Formula (3.17) the likelihoods will be of the form:

$$P(D|G) = \prod_{i=1}^{n} P(X_i = D_{X_i} | pa(X_i) = D_{pa(X_i)}) = \prod_{i=1}^{n} Score(X_i | D, pa(X_i)) \tag{3.21}$$

Thereby D_{X_i} and $D_{pa(X_i)}$ represent the data set D reduced to the indicated variables, that is the relevant data for that particular factor. The likelihood factors are referred to as *local scores*. In the following two subsections the graph prior won't be factorised. But instead of multiplying the graph prior to the likelihood (see Formula 3.21), the graph prior P(G) can be factorised, so that there is one factor (local parent set prior) P(pa(X_i)) for each local score $Score(X_i|D, pa(X_i))$ (i=1,...,n). The possibility of factorising the graph prior is especially important with regard to the MCMC sampling schemes.

3.5.2.1. Discrete multinomial Bayesian scoring metric

The first parametric model for Bayesian networks is the *discrete multinomial model* which asserts a multinomial distribution to each domain variable. The resulting scores are usually referred to as the *BDe* scores. Although, as multinomial distributions can deal with discrete observations only, it is necessary to discretise the data D in advance, the BDe score is a very flexible modelling tool which allows to model non-linear relationships and interactions between the domain variables. So, there is a certain trade off between the information loss incurred through data discretisation and modelling flexibility. Assuming that the domain variables are discrete with r possible realisations, respectively have been discretised accordingly, it can be assumed that each local probability distribution $P(X_i|pa(X_i))$ (i=1,...,n) is a collection of multinomial distributions, one distribution for each possible realisation of the parent variables in $pa(X_i)$, that is for each *configuration* of the parent variables. Then it can be defined for i=1,...,n:

$$P(X_i = k|pa(X_i) = j) = \theta_{ijk}. \tag{3.22}$$

In other words θ_{ijk} is the probability that domain variable X_i takes on its k-th value (k=1,...,r), given the j-th parent configuration of $pa(X_i)$ (j=1,...,r_i). The values r_i, that is the number of different parent configurations, depend on the cardinalities of $pa(X_i)$, and are given by $r_i = r^{|pa(X_i)|}$.

For the parameters holds $0 \leq \theta_{ijk} \leq 1$ and

$$\sum_{k=1}^{r} \theta_{ijk} = 1.$$

For convenience, it is useful to define additionally:

$$\theta = (\theta_1, ..., \theta_n)$$

whereby $\theta_i = (\theta_{ijk})_{k=1,...,r}^{j=1,...,r_i}$, so that θ_i are the parameters for the local probability distribution of the i-th domain variable X_i.

This multinomial distribution assumption combined with the assumption of having m independent observations of the domain $D_{.1}, ..., D_{.m}$ in a data set D can be used to obtain the following presentation of the likelihood $f(D|\theta, G_0)$:

$$f(D|\theta, G_0) = \prod_{l=1}^{m} f(D_{.l}|\theta, G_0) = \prod_{l=1}^{m} \prod_{i=1}^{n} P(X_i = D_{il}|\theta_i, pa(X_i) = \psi_{il}), \tag{3.23}$$

whereby D_{il} represents the value of the i-th variable in the l-th observation, and ψ_{il} represents the configuration of $pa(X_i)$ in the l-th case. The latter decomposition is due to the factorisation rule (see Formula (3.17)) in Bayesian networks. By grouping terms, this term can be rewritten as follows:

$$f(D|\theta, G_0) = \prod_{i=1}^{n} \prod_{j=1}^{r_i} \prod_{k=1}^{r} \theta_{ijk}^{N_{ijk}}, \tag{3.24}$$

where N_{ijk} is the number of observations in D in which variable X_i has the value k and the configuration of $pa(X_i)$ is j. Substituting this into the BMA approach Formula (3.19) leads to:

$$P(D|G_0) \cdot P(G_0) = P(G_0) \cdot \int \prod_{i=1}^{n} \prod_{j=1}^{r_i} \prod_{k=1}^{r} \theta_{ijk}^{N_{ijk}} \cdot f(\theta|G_0) d\theta \tag{3.25}$$

Assuming *global* and *local parameter independence* (see [24] for further details) what means for the parameter prior:

$$f(\theta|G_0) = \prod_{i=1}^{n} f(\theta_i|pa(X_i)) = \prod_{i=1}^{n}\prod_{j=1}^{r_i} f(\theta_{ij1},...,\theta_{ijr}),$$

yields:

$$P(D|G_0)\cdot P(G_0) = P(G_0)\cdot \int \prod_{i=1}^{n}\prod_{j=1}^{r_i}\prod_{k=1}^{r} \theta_{ijk}^{N_{ijk}} \cdot \prod_{i=1}^{n}\prod_{j=1}^{r_i} f(\theta_{ij1},...,\theta_{ijr}) d(\theta_{ij1},...,\theta_{ijr})$$

By using the independence of the terms, this integral of products can be converted to a product of integrals:

$$P(D|G_0)\cdot P(G_0) = P(G_0)\cdot \prod_{i=1}^{n}\prod_{j=1}^{r_i} \int \left(\prod_{k=1}^{r} \theta_{ijk}^{N_{ijk}}\right) \cdot f(\theta_{ij1},...,\theta_{ijr}) d(\theta_{ij1},...,\theta_{ijr}) \quad (3.26)$$

[24] show that the *Dirichlet distribution* which is the conjugate prior for the multinomial distribution leads to the analytical tractability of this integral, that is a closed-form solution of Formula (3.26).

For i=1,..,n and j=1,...,r_i the Dirichlet prior $(\theta_{ij1},...,\theta_{ijr}) \sim DIR(\alpha_{ij1},...,\alpha_{ijr})$ is given by:

$$f(\theta_{ij1},...,\theta_{ijr}) = \prod_{k=1}^{r} \theta_{ijk}^{\alpha_{ijk}-1} \cdot \frac{\Gamma\left(\sum_{k=1}^{r}\alpha_{ijk}\right)}{\prod_{k=1}^{r}\Gamma\left(\alpha_{ijk}\right)},$$

where α_{ijk} are unknown hyperparameteres and $\Gamma(.)$ is the gamma function. Using this Dirichlet prior in Formula (3.26) leads to $P(D|G_0)\cdot P(G_0)$

$$= P(G_0)\cdot \prod_{i=1}^{n}\prod_{j=1}^{r_i} \int \left(\theta_{ijk}^{N_{ijk}}\right) \prod_{k=1}^{r} \theta_{ijk}^{\alpha_{ijk}-1} \cdot \frac{\Gamma\left(\sum_{k=1}^{r}\alpha_{ijk}\right)}{\prod_{k=1}^{r}\Gamma\left(\alpha_{ijk}\right)} d(\theta_{ij1},...,\theta_{ijr}) \quad (3.27)$$

[10] derive the following closed-form solution of this product of multiple integrals:

$$P(D|G_0) \cdot P(G_0) = P(G_0) \cdot \prod_{i=1}^{n} \prod_{j=1}^{r_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij} + \alpha_{ij})} \cdot \prod_{k=1}^{r} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{3.28}$$

where $N_{ij} = \sum_{k=1}^{r} N_{ijk}$ and $\alpha_{ij} = \sum_{k=1}^{r} \alpha_{ijk}$. The unknown hyperparameters α_{ijk} can be interpreted as pseudo-counts (see [34] for further details). That is α_{ijk} can be interpreted as the number of imaginary observations in which the event $X_i = k$ and $pa(X_i) = j$ has occured (in some virtual database). Especially Formula (3.28) is the factorisation of the likelihood (see Formula (3.21)) for the discrete multinomial Bayesian network BDe model. The local scores are given by:

$$Score(X_i|D, pa(X_i)) = \prod_{j=1}^{r_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij} + \alpha_{ij})} \cdot \prod_{k=1}^{r} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})}$$

[5] proves that the following choice of the hyperparameters:

$$\alpha_{ijk} = \alpha \cdot \frac{1}{r \cdot r_j}, \tag{3.29}$$

whereby $\alpha > 0$ is referred to as *total prior precision*, leads to *score-equivalence*. As discussed in Subsection 3.5.1 score-equivalence means that DAGs that assert the same set of independence relations among the domain variables obtain the same likelihood score, that is are equally strong supported by the data. Usually, the total prior precision α is set equal to 1 what renders the prior distribution over the parameters uninformative, as it leads to relatively low hyperparameters α_{ijk} (which can be interpreted as pseudo counts).

It can be summarised that the discrete multinomial BDe scoring metric for Bayesian network models (G, F, q) over a domain of discrete variables has a closed-form solution that can be computed using Formula (3.28).

3.5.2.2. Continuous Gaussian Bayesian scoring metric

The second parametric model for Bayesian networks is the *continuous Gaussian model* which asserts a Gaussian distribution to each domain variable. The resulting scores are usually referred to as the *BGe* scores for Bayesian networks. More precisely, using the Gaussian BGe-scores each domain variable X_i is interpreted as a normally distributed random variable, whose mean value $E[X_i]$ depends on the values of its parent variables. That is, if a DAG G is given in which node X_i has u parent nodes $X_{i_1}, ..., X_{i_u}$, the distribution of X_i is given by:

$$X_i \sim N(\mu_i + \sum_{j=1}^{n} b_{ij} \cdot (x_j - \mu_j), \sigma_i^2), \tag{3.30}$$

where μ_i is the unconditional mean of X_i, σ_i^2 is the conditional variance of X_i given the realisations $X_1 = x_1$,...,$X_n = x_n$, and the coefficients b_{ij} reflect the strengths of the dependencies between X_i and the other domain variables. Thereby holds $b_{ij} = 0$ if $j \notin \{i_1, ..., i_u\}$, so that the realisation x_i of X_i is interpreted in dependence of the realisations $x_{i_1}, ..., x_{i_u}$ of the parent variables $X_{i_1}, ..., X_{i_u}$ only. In other words each coefficient $b_{ij} \neq 0$ represents an edge in the directed acylic graph (DAG) G which points from node X_j to node X_i.
The coefficients b_{ij} and the conditional variances σ_i^2 can be used to compute the *precison matrix* W of the joint multivariate Gaussian distribution of the n domain variables with the following recursive formula [18]:

Recursive Transformation:

- Set $W(1) = 1$ and define $\vec{b_i}$ as the following column vector of length i-1: $\vec{b_i} = (b_{1,i}, ..., b_{i-1,i})^T$ (i=1,...,n).

- For i=1,...,n-1 compute $W(i+1)$ from $W(i)$, σ_i^2, and $\vec{b}_{i+1}$ as follows:

$$W(i+1) = \begin{pmatrix} W(i) + \left(\vec{b}_{i+1} \cdot \vec{b}_{i+1}^T \cdot \sigma_{i+1}^2\right) & -\vec{b}_{i+1} \cdot \sigma_{i+1}^2 \\ \\ -\vec{b}_{i+1}^T \sigma_{i+1}^2 & \sigma_{i+1}^2 \end{pmatrix}$$

$W(n)$ is the precison matrix W for the joint Gaussian distribution of the domain variables $X_1, ..., X_n$. As usual the covariance matrix Σ is the inverse of the precison matrix: $\Sigma = W^{-1}$. Additionally defining the unconditional mean vector $\mu = (\mu_1, ..., \mu_n)^T$, the joint Gaussian distribution is given by: $(X_1, ..., X_n) \sim N(\mu, \Sigma)$. In analogy to the derivation of the BDe score (see Subsection 3.5.2.1) [18] derive a scoring metric for Gaussian Bayesian networks. As their derivation is extensive and complicated, only the main steps are presented in this subsection. First of all, they assume that the prior distribution over the unknown parameter vector μ is a Gaussian distribution with mean μ_0 and precison matrix $\nu \cdot W$ with $\nu > 0$, whereby the matrix W in turn is Whishart distributed with $\alpha > n + 1$ degrees of freedom and precison matrix T_0. That is:

- $P(\mu = \mu^\star | \mu_0, \nu W) = (2\pi)^{-\frac{n}{2}} \cdot |\nu \cdot W|^{\frac{1}{2}} \cdot e^{-\frac{1}{2} \cdot (\mu^\star - \mu_0)^T \nu W (\mu^\star - \mu_0)}$
- $P(W = W^\star | T_0) = c(n, \alpha) \cdot |T_0|^{\frac{\alpha}{2}} \cdot |W^\star|^{\frac{\alpha - n - 1}{2}} \cdot e^{-\frac{1}{2} \cdot trace(T_0 \cdot W^\star)}$.

Thereby $|.|$ is the determinant and $trace(.)$ is the sum of the diagonal elements of the input matrix. The factors $c(n, \alpha)$ are given by:

$$c(n, \alpha) = \left(2^{\frac{\alpha \cdot n}{2}} \cdot \pi^{\frac{n(n-1)}{4}} \cdot \prod_{i=1}^{n} \Gamma\left(\frac{\alpha + 1 - i}{2}\right) \right)^{-1}. \tag{3.31}$$

The matrx T_0, the vector μ_0 as well as the degrees of freedom α and the factor ν are unknown parameters that have to be specified in advance and can be used to include some background knowledge about the domain. The assessement of these parameters is briefly discussed at the end of this subsection.

Subsequently, [18] show that this normal-Wishart prior assumption is sufficient for deriving a score for *complete* Gaussian Bayesian networks, that is for DAGs with as

many edges as possible representing that all domain variables are pairwise dependent. Such a complete DAG is for example given if each b_{ij} in Formula (3.30) is unequal to zero for i<j. An as $b_{ij} \neq 0$ reflects an edge pointing from variable X_j to X_i, this in turn means that domain variable X_i (i=2,...,n) has the parent nodes $pa(X_i) = \{X_1, ..., X_{i-1}\}$ with X_1 having no parent nodes. Alternative complete DAGs - all lying in the same equivalence class - can be obtained by permutating the order of the domain variables to $X_{\sigma(1)}, ..., X_{\sigma(n)}$ and setting $pa(X_{\sigma(i)}) = \{X_{\sigma(1)}, ..., X_{\sigma(i-1)}\}$ afterwards.

The BGe-score of such a *complete* DAG G_C derived in [18] is then given by:

$$P(D|G_C) \cdot P(G_C) = (2\pi)^{-\frac{n \cdot m}{2}} \cdot (\frac{\nu}{\nu + m})^{\frac{n}{2}} \cdot \frac{c(n, \alpha)}{c(n, \alpha + m)} \cdot |T_0|^{\frac{\alpha}{2}} \cdot |T_m|^{-\frac{\alpha+m}{2}}, \qquad (3.32)$$

whereby m is the number of independent observations in the data set D, the function $c(.,.)$ is defined in Formula (3.31), and the matrix T_m is given by:

$$T_m = T_0 + \sum_{j=1}^{m} (D_{.j} - \overline{D}) \cdot (D_{.j} - \overline{D})^T + \frac{\nu \cdot m}{\nu + m} \cdot (\mu_0 - \overline{x}) \cdot (\mu_0 - \overline{x})^T \qquad (3.33)$$

In Formula (3.33) $\overline{D}$ is the mean vector of the m observation vectors $D_{.j}$ in the data set D.

Subsequently, [18] show that it is possible to derive the BGe score for any DAG under two fairly weak assumptions of parameter independence: $P(\sigma_1^2, ..., \sigma_n^2, \vec{b_1}, .., \vec{b_n}|G) = \prod_{i=1}^{n} P(\sigma_i^2, \vec{b_i}|G)$, and parameter modularity: $P(\sigma_i^2, \vec{b_i}|G) = P(\sigma_i^2, \vec{b_i}|pa(X_i))$. So, parameter independence means that the unknown parameters of the local probability distributions (see Formula (3.30)) are stochastically independent of each other, and parameter modularity means that the prior distribution of the parameters of these local probability distributions depend on the parent variables only.

Under these two assumptions can be derived (see [18]) that the BGe score of any Gaussian Bayesian network G_0 can be computed as follows:

$$P(D|G_0) \cdot P(G_0) = P(G_0) \cdot \prod_{i=1}^{n} \frac{P(D^{(X_i, pa(X_i))}|G_C)}{P(D^{(pa(X_i))}|G_C)} \tag{3.34}$$

where $D^{(pa(X_i))}$ is the data set D restricted to the variables in $pa(X_i)$, and $D^{(X_i, pa(X_i))}$ is the data set D restricted to the variables in $pa(X_i) \cup X_i$. G_C represents a complete DAG over the variables to which the corresponding data set $D^{(.)}$ is restricted. Therefore, Formula (3.32) can be used on reduced data sets to compute the BGe score of any DAG G_0 over the domain $X_1, ..., X_n$. Formula (3.34) provides the factorisation of the likelihood (see Formula (3.21)) for the continuous Gaussian Bayesian network BGe model. The local scores are given by:

$$Score(X_i|D, pa(X_i)) = \frac{P(D^{(X_i, pa(X_i))}|G_C)}{P(D^{(pa(X_i))}|G_C)}$$

In addition, [18] give a heuristic method for encoding prior knowledge about the domain when assessing the unknown prior parameters T_0 and μ_0. They recommend to build a prior Gaussian Bayesian network with respect to the user‘s knowledge. For example, a Bayesian network without any edges in which every variable has a standard Gaussian $N(0, 1)$ distribution. This prior network, that is the specification of the network parameters μ, b_{ij}, and σ_i^2, which in turn specify the parameters of the multivariate normal $N(\mu, \Sigma)$ over the domain, can be used to obtain the following reasonable prior parameters:

- $\mu_0 = \mu$
- $T_0 = \frac{\nu(\alpha - n - 1)}{\nu + 1} \cdot \Sigma$

The parameters $\nu > 0$ and $\alpha > n + 1$ are referred to as the *user‘s equivalent sample sizes* for μ_0 and T_0. The higher these equivalent sample size parameters are selected the more information is implied through the prior network.

From a Bayesian Statistics point of view, when there isn‘t any prior knowledge, the prior parameters should be choosen as uninformative as possible to avoid parameter over-fitting. On the other hand, an inadequately specified prior network can lead to some serious bias of the results, so that it is not useful to specify an absolutely unrealistic prior network. E.g. if the measured variables of a domain can have positive realisations only with very low variances, it is surely inadequate to assume a prior network of independent Gaussian distributed variables with mean zero and a very high variance. So, there is a certain trade-off between overfit and bias. The author of this thesis holds the view that an uninformative network prior must satisfy the following conditions: the means of all variables as well as the variances are equal, and all correlations between the domain variables are zero, so that it is assumed that each domain variable is independently and identically N(μ,σ^2) distributed. Furthermore, the equivalent sample sizes should be set as small as possible, when there is no prior knowledge about the domain. But it is not clear how to specify the two parameters of the Gaussian distribution.

Because of some systematic differences in the means and variances of the variables of the synthetically generated data for the comparative evaluation study (see Section 5), it was decided to normalise each data set before analysing it. Consequently, it holds for each test data set used in the study that all its variables have mean zero and a variance of one. Although it must be seen a little critical, after a good deal of thought, it was decided to set the two prior parameters of the BGe Bayesian network scoring metric correspondingly. That is $\mu = 0$ and $\sigma^2 = 1$. The justification of this approach is as follows: Firstly, mainly the correlations between the domain variables contain information about possible network edges. All correlations are set to zero in the prior network described above, so that the most important prior parameters (with regard to the extraction of gene networks) are choosen uninformative. Secondly, since the equivalent sample sizes will be choosen as small as possible, it can be concluded that the prior network has not much influence on the results at all, so the prior network does not protect much against overfitting anyway. (The validity of this second point

was empirically checked on lots of test data sets.) Thirdly, this approach guarantees that the parameter prior is equally informative for all test data sets. That is, there is no effect of the true realisation‘s nature on the learning performance. Especially, since the functional relationships between the variables are more or less arbitrarily specified before generating synthetic data, it would be left to chance how much the prior network of independent standard Gaussian distributions fits the nature of the real test data.

However, for interventional data sets these two prior parameters can not be specified as adequately as for pure observational data. For pure observational data it does not depend on whether a domain variable is considered as a child or parent node: in both cases its empirical variance and its empirical mean (as well as its empirical covariances) are the same. But this is not true for intervened nodes in interventional data sets (see Subsection 3.5.5). In the case of interventional data it is necessary to exclude some realisations of the network domain whenever an intervened node is considered as a child node, and its local score given a parent set is computed. More precisely, all realisations, where this intervened child node was activated or inhibited by experimental conditions, must be excluded when its local score given its parent set is computed. So, the empirical means, variances, and covariances of all domain variables depend on the remaining realisations only. Consequently, the means and variances of all domain variables differ with respect to the node that is considered as child node. Either the child node is a non-intervened node so that all realisations of the network domain can be used to comute these empirical characteristics, or the child node is an intervened node, so that certain realisations have to be excluded from the computations of these characteristics.

Nevertheless, for such interventional data sets exactly the same prior parameters $\mu = 0$ and $\sigma^2 = 1$ were selected. As a consequence when an intervened node is scored, there is automatically a certain discrepancy between the two prior parameters and the corresponding empirical parameters - even when the interventional data set has been normalised.

Although it was not expected, it will be seen in Section 5.6 that the learning performance

of the BGe Bayesian network model on interventional data is sensitive to the prior network in some extreme cases. So, in real applications it is advisable to select the prior network with extreme caution.

3.5.3. MCMC sampling of Bayesian networks

In this section two different Markov Chain Monte Carlo (MCMC) methods for sampling directed acyclic graphs (DAGs) G_0 of Bayesian networks from the posterior distribution $P(G_0|D)$ (see 3.18) are presented. For all stochastic models of the corresponding Bayesian networks both MCMC methods can be used for sampling. Therefore we assume within this section that a Bayesian network is given by the triple (G_0, F, q), whereby as described in Section 3.5.2 G_0 is a DAG, F describes the distributional form, and $q = q(F, G_0)$ is the corresponding parameter vector.

Based on an independent sample $D = (D_{.1}, ..., D_{.m})$ of the joint probability distribution of the domain variables $P(X_1, ..., X_n)$ the objective of interest is learning the network structure behind the variables. In the context of Bayesian networks one possible method of learning is to search for the DAG that is most supported by the data D. Statistically this means to determine, on the basis of the data D, the DAG whose independency assumptions best represents the mechanism that generated the data. This DAG G^* maximises the posterior distribution and therefore satisfies: $P(G^*|D) \geq P(G|D)$ for all directed acylic graphs G. A comparison of several heuristic search procedures, such as *Greedy-Search* algorithms, can be found in [23]. But biological expression data are usually sparse, that is, the amount of data is small relative to the number of parameters of a Bayesian network model. Therefore data over-fitting must be expected, if trying to represent the dependency structure behind the variables by one single DAG. And especially the DAG G^* that maximises the posterior distribution possibly gives no adequate, but an over-fitted insight into the relations between the domain variables. Consequently, it is more appropriate to report conclusions from more than one DAG.

Since direct sampling from the posterior distribution:

$$P(G_0|D) = \frac{P(G_0, D)}{P(D)} = \frac{P(D|G_0) \cdot P(G_0)}{\sum_{G \in \Omega} P(D|G) \cdot P(G)},$$

is intractable due to the intractability of the normalisation factor in the denominator, Markov Chain Monte Carlo (MCMC) schemes can be adopted to generate samples from this posterior distribution. Two different MCMC sampling schemes defined in the space of DAGs (Structure-MCMC by [32]) and node orders (Order-MCMC by [15]) have been proposed in the literature and will be presented in the following two subsections.

In general, a Markov Chain Monte Carlo (MCMC) sampling scheme can be used to generate a sample $s_1, s_2, ...$ from a discrete target distribution $P^\star(.)$ with state space S ($|S| < \infty$). This is accomplished by constructing a Markov Chain in the space S whose distribution converges to the desired posterior distribution $P^\star(.)$ as stationary one. The MCMC simulation scheme consists of evaluating at each step an acceptance probability with which a new state can replace the current state.
More precisely, the general mechanism of a Markov Chain $(M_n)_{n \in N}$ with state space S is given by:

$$P(M_{n+1} = x) = \sum_{y \in S} T(x|y) \cdot P(M_n = y) \tag{3.35}$$

for all $x \in S$ and $n \in N$. Thereby T(x|y) is the *transition kernel* which denotes the probability of a transition from state y to state x. In addition an initial distribution $P(M_1 = z)$ (z$\in$ S) is defined. If $T(x, x) > 0$ for all states x, and if for all $x, y \in S$ there exists an integer k, so that $P(M_{n+k} = x|M_n = y) > 0$, it is guaranteed that the distribution of $(M_n)_{n \in N}$ converges to a stationary one P_∞. That is, for all $z \in S$ holds: $P(M_n = z) \to P_\infty(z)$ for $n \to \infty$. This is due to the fact that these conditions of *irreducibility* and *aperiodicity* are sufficient conditions for *ergodicity* of the Markov Chain (M_n). And ergodicity is a sufficient condition for stationarity for $n \to \infty$. See [19] for further details. The stationary distribution is determined by the transition kernel

$T(.|.)$, and does not depend on the initial distribution. The *equation of detailed balance*:

$$\frac{T(x|y)}{T(y|x)} = \frac{P^\star(x)}{P^\star(y)} \tag{3.36}$$

for all states x and y is a sufficient condition for that the stationary distribution is the desired posterior distribution: $P_\infty(.) = P^\star(.)$. Equation (3.36) can be easily fulfilled when decomposing the transitions at each 'time-index' n into two parts.

In a first step a new state x for M_{n+1} is proposed with a *proposal probability* $Q(x|y)$ which depends on the current state y of M_n. Thereby the new state x has to be unequal to the current state y. Afterwards, in the second step the new state x is accepted with an acceptance probability $A(x|y)$ as new state of the Markov Chain at $n+1$. If it is not accepted, the new state at $n+1$ is set equal to the current state y. This procedure is reiterated for all $n > 0$.

The transition probabilities are then given by: $T(x|y) = Q(x|y) \cdot A(x|y)$ for all $x, y \in S$ with $x \neq y$ and

$$T(x|x) = \sum_{\substack{y \in S \\ y \neq x}} (1 - A(y|x)) \cdot Q(y|x) \tag{3.37}$$

The equation of detailed balance is accomplished if the acceptance probability is choosen as follows: $A(x|y) = min\{1, R(x|y)\}$, whereby

$$R(x|y) = \frac{P^\star(x) \cdot Q(y|x)}{P^\star(y) \cdot Q(x|y)} \tag{3.38}$$

As $R(x|y)$ is equal to $\frac{1}{R(y|x)}$, it immediately follows that $R(x|y) > 1 \Leftrightarrow R(y|x) < 1$ and consequently hold the following two equivalence relations:

- $A(x|y) = 1 \Leftrightarrow A(y|x) = \frac{1}{R(x|y)}$
- $A(x|y) = R(x|y) \Leftrightarrow A(y|x) = 1$

This means for the ratio of transition probabilities:

$$\frac{T(x|y)}{T(y|x)} = \frac{Q(x|y) \cdot A(x|y)}{Q(y|x) \cdot A(y|x)} = \frac{Q(x|y)}{Q(y|x)} \cdot R(x|y) = \frac{Q(x|y)}{Q(y|x)} \cdot \frac{P^\star(x) \cdot Q(y|x)}{P^\star(y) \cdot Q(x|y)} = \frac{P^\star(x)}{P^\star(y)}$$

so that the equation of detailed balance (see Formula (3.36)) is fulfilled.

The proposal probabilities $Q(.|.)$ which have to be defined in advance depend on the design of the transitions in the state space S, that is on the particular MCMC sampling scheme. So, they will be described together with the two MCMC sampling schemes in the following two subsections.

3.5.3.1. Structure-MCMC

The Structure-MCMC approach of [32] is a Markov Chain Monte Carlo (MCMC) sampling scheme that generates a sample of DAGs $G_1, G_2, G_3, ...$ from the posterior distribution $P^\star(.) = P(.|D)$. So, the state space S is the set of all valid DAGs. The proposal probabilities $Q(G|G^\star)$ are defined as follows:

$$Q(G|G^\star) = \left\{ \begin{array}{cc} \frac{1}{|\Pi(G^\star|)} & , G \in \Pi(G^\star) \\ 0 & , G \notin \Pi(G^\star) \end{array} \right\}$$

Thereby $\Pi(G^\star)$ denotes the *neighbourhood* of $G^\star$, that is the collection of all DAGs that can be reached from $G^\star$ by deletion, addition or reversal of one single edge. $|\Pi(G^\star)|$ is the cardinality of this collection. As the new graph G has to be an acyclic one too, it has to be checked which edges can be added to $G^\star$, and which edges can be reversed in $G^\star$, without violating the acyclicity-constraint. Some details on how to determine these edges are given in C. Appendix III.

[32] show that these proposal probabilities lead to an ergodic Markov Chain in the space of all valid DAGs if the acceptance probabilities are set to $A(G|G^\star) = min\{1, R(G|G^\star)\}$, where $R(.|.)$ was defined in (3.38).

Adding and removing of edges is needed for reaching ergodicity. Edge reversals just lead to a faster convergence of the Markov Chain as shown in [20]. However, [32] point out

that the reversal of reversible edges directly leads to a DAG within the same equivalence class (for details see Section 3.5.1). They conclude that one of the drawbacks of using a Markov Chain in the space of DAGs is that the Markov Chain may visit equivalence classes proportionally to their sizes (in terms of how many DAG members they have). In order to alleviate this problem, they recommend to allow reversals of compelled edges only. This can be accomplished by modifying the neigbourhoods with regard to this restriction (see C. Appendix III). Furthermore, a reasonable approach adopted in most applications of Bayesian networks to the reverse engineering of gene (or protein) regulatory networks is to impose a limit on the cardinality of the sizes of parent node sets. This limit is referred to as *fan-in*. Each domain node in a DAG can have at most fan-in parent nodes then. The practical advantage of the restriction on the maximum number of edges converging on a node is a reduction of the computational complexity, which improves the convergence of the Markov Chain in the Structure-MCMC simulation. Fan-in restrictions can be justified in the context of expression data as many experimental results have shown that the expression of a gene is usually controlled by a comparatively small number of active regulator genes, while on the other hand regulator-genes themselves seem to be nearly unrestricted in the number of genes they regulate. The imputation of such a fan-in restriction leads to a further reduction of a DAG's neighbourhood. All DAGs that contain nodes with too many parents, that is more than the fan-in value, have to be removed from the respective neighbourhoods (see C. Appendix III).
The main advantage of the proposal probabilities mentioned above is that it is efficient to compute $R(G|G^\star)$, when G and $G^\star$ differ by a single edge only. Inserting the proposal probabilities $Q(G|G^\star)$ in the formula for $R(G|G^\star)$ (see Formula 3.38) leads to:

$$R(G|G^\star) = \frac{P(G|D) \cdot \frac{1}{|\Pi(G^\star)|}}{P(G^\star|D) \cdot \frac{1}{|\Pi(G)|}} = \frac{\frac{P(D|G) \cdot P(G)}{P(D)} \cdot \frac{1}{|\Pi(G^\star)|}}{\frac{P(D|G^\star) \cdot P(G^\star)}{P(D)} \cdot \frac{1}{|\Pi(G)|}} = \frac{P(D|G)}{P(D|G^\star)} \cdot \frac{P(G)}{P(G^\star)} \cdot \frac{|\Pi(G^\star)|}{|\Pi(G)|}$$

As the likelihoods $P(D|G)$ and $P(D|G^\star)$ can be factorised into products of local scores (see Formula (3.21)), and as the priors $P(G)$ and $P(G^\star)$ can be factorised into products of local parent set priors (see Formula (3.5.2)), it follows that for DAGs that differ by one edge only, most of the factors cancel out. If for example X_i has the parent set

$pa(X_i) = \{X_j, X_k\}$ in $G^\star$ and the edge $X_j \to X_i$ is deleted in G, then $R(G|G^\star)$ is given by:

$$R(G|G^\star) = \frac{P(pa(X_i) = \{X_k\})}{P(pa(X_i) = \{X_j, X_k\})} \cdot \frac{Score(X_i|D, \{X_k\})}{Score(X_i|D, \{X_j, X_k\})} \cdot \frac{|\Pi(G^\star)|}{|\Pi(G)|}$$

The Structure-MCMC approach of [32] for sampling DAGs from their posterior distribution can be summarised as follows:

- Initialisation: Choose an arbitrary DAG G_1 and set $M_1 = G_1$. For example initialise the Markov Chain by the *empty* DAG containing no edges.
- Iteration: For i=1,2,3,...: Given a realisation $G_i = G^\star$ of M_i randomly choose a neighbour DAG G of $G^\star$ from the proposal distribution:

  $$Q(G|G^\star) = \left\{ \begin{array}{cc} \frac{1}{|\Pi(G^\star)|)} & , G \in \Pi(G^\star) \\ 0 & , G \notin \Pi(G^\star) \end{array} \right\}$$

  Accept the randomly choosen DAG G with the acceptance probability $A(G|G^\star) = min\{1, R(G|G^\star)\}$. If G is accepted, set $M_{i+1} = G$. Otherwise leave the Markov Chain unchanged, that is, set $M_{i+1} = G^\star$.

As it takes 'some time' until the Markov Chain converges to its stationary distribution, the idea is to sample from the Chain for 'long enough' to ensure that it has reached its stationary distribution. The time until then is called the *burn-in time* and the DAGs sampled in this 'time' are usually rejected and thrown away. Any further sample can be seen as sample from the posterior distribution. But as there is no sufficent condition that guarantees that convergence is reached, there is need for some convergence diagnostics, such as *trace plot diagnostics*.

3.5.3.2. Order-MCMC

The Order-MCMC approach of [15] is a Markov Chain Monte Carlo (MCMC) sampling scheme that generates a sample of domain node orderings $O_1, O_2, O_3, ...$ from the posterior distribution $P^\star(.) = P(.|D)$ over node orderings. So, the state space S is the set of all $n!$ possible orderings of the domain nodes. Afterwards in a second step a sample of DAGs $G_1, G_2, G_3, ...$ can be obtained by sampling DAGs out of the sampled node orderings.

Each ordering $O = (X_{\sigma(1)}, ..., X_{\sigma(n)})$ of the domain variables $X_1, ..., X_n$ is implied through a permutation σ. The meaning of such an ordering in the context of Order-MCMC is as follows: O represents the set of all DAGs for which holds $X_{\sigma(i)} \notin pa(X_{\sigma(j)})$ if $\sigma(j)$ preceds $\sigma(i)$ in the permutation vector $\sigma = (\sigma(1), ..., \sigma(n))$. Only if $X_{\sigma(j)}$ succeeds $X_{\sigma(i)}$ in σ the relation $X_{\sigma(i)} \in pa(X_{\sigma(j)})$ is valid. That is the j-th variable $X_{\sigma(j)}$ in the ordering O is not allowed to have parents that are standing to the right of $X_{\sigma(j)}$ in O. The valid parent sets of $X_{\sigma(j)}$ are restricted to variables that are standing to the left. Consequently, node $X_{\sigma(1)}$ must have the empty parent set $pa(X_{\sigma(1)}) = \emptyset$, node $X_{\sigma(2)}$ can have either the empty parent set $\emptyset$ or the set $\{X_{\sigma(1)}\}$, node $X_{\sigma(3)}$ is allowed to have one of the following 4 parent-sets: $\emptyset$, $\{X_{\sigma(1)}\}$, $\{X_{\sigma(2)}\}$, $\{X_{\sigma(1)}, X_{\sigma(2)}\}$, etc.

The likelihood $P(D|O)$ of a given node ordering O can be computed efficiently, as the selection of the parent-set for one node with respect to O does not lead to any additional restrictions for another. That is for each node its parent set can be selected independently with respect to the ordering. Or in other words, as long as the restrictions implied through the ordering O are regarded for each node, it is guaranteed that no cycles will come into being. Therefore the likelihood can be obtained by summing the products of *local scores* and *local parent set priors* for each domain node over the set of all valid parent-sets, and then multiplying these sums. Thereby it is important that the prior over DAGs as well as the likelihood of a DAG can be decomposed into a product where each factor corresponds to a node. As described in Section 3.5.2, this holds for both priors. Furthermore, it could be seen from Formula (3.28) and Formula (3.34) that such

a factorisation of the likelihood into local scores is given for the discrete multinomial and the continuous Gaussian Bayesian network model. Consequently, the likelihood of the node ordering $O = (X_{\sigma(1)}, ..., X_{\sigma(n)})$ is given by:

$$P(D|O) = \prod_{i=1}^{n} \sum_{P \in V_i(\sigma)} P(pa(X_i) = P) \cdot Score(X_i|pa(X_i) = P, D) \tag{3.39}$$

whereby $V_i(\sigma)$ denotes the system of all parent sets that are valid for domain variable X_i with respect to the given ordering. If a fan-in restriction is imposed, the systems $V_i(\sigma)$ are restricted to sets of cardinalities not higher than the fan-in restriction.

The idea of Order-MCMC is to construct a Markov Chain that converges to the posterior-probability over node orderings, that is $P(O|D)$. This can be accomplished by using the construction presented in Section 3.5.3. The equation of detailed balance (see Formula (3.36)) states that the Markov Chain converges to the posterior probability, if for the ratio of transition probabilities holds:

$$\frac{T(O|O^\star)}{T(O^\star|O)} = \frac{P(O|D)}{P(O^\star|D)} \tag{3.40}$$

where O is a node ordering that can be reached from the current node ordering $O^\star$. And the equation of detailed balance in turn can be easily fulfilled by decomposing the transition probabilities into products of proposal and acceptance probabilities: $T(O_2|O_1) = Q(O_2|O_1) \cdot A(O_2|O_1)$ Thereby the acceptance probabilities depend on the proposal probabilities (see Formula (3.38)), which in turn depend on the way the transitions are designed in the space of node orderings. [15] recommend to use a simple *flip-operator* which exchanges one node for another in the ordering. This leads to the following proposal probabilities:

$$Q(O|O^\star) = \left\{ \begin{array}{cc} \frac{2}{n \cdot (n-1)} & , O \in \Pi(O^\star) \\ 0 & , O \notin \Pi(O^\star) \end{array} \right\}$$

Thereby $\Pi(O^\star)$ is the set of all node orderings $O^\dagger = (X_{\sigma^\dagger(1)}, ..., X_{\sigma^\dagger(n)})$ that can be

reached from $O^\star$ by flipping two nodes in $O^\star$, and leaving all other nodes in the ordering unchanged. More presisely, the ordering $O^\dagger = (X_{\sigma^\dagger(1)}, ..., X_{\sigma^\dagger(n)})$ can be reached from $O^\star = (X_{\sigma^\star(1)}, ..., X_{\sigma^\star(n)})$ if and only if for the corresponding permutations $\sigma^\dagger$ and $\sigma^\star$ holds: $|\{1, ..., n\} : \sigma^\star(i) = \sigma^\dagger(i)| = n - 2$. Moreover, as $\sigma^\dagger$ and $\sigma^\star$ are permutations, it follows that there is exactly one pair $(j, k) \in \{1, ..., n\}^2$ of integers with: $\sigma^\star(j) = \sigma^\dagger(k)$ and $\sigma^\star(k) = \sigma^\dagger(j)$. The proposal probability $Q(.|O^\star)$ is a uniform distribution over all $\frac{n\cdot(n-1)}{2}$ possibilities of exchanging two nodes for each other in $O^\star$.

To guarantee convergence to the desired posterior distribution, the acceptance probabilities must be set to $A(O|O^\star) = min\{1, R(O|O^\star)\}$, where $R(.|.)$ was defined in Formula (3.38) (see Subsection 3.5.3 for further details). If a uniform prior over all $n!$ possible node orderings is assumed, that is $P(O) = \frac{1}{n!}$ for every ordering O, the term $R(O|O^\star)$ is given by:

$$R(O|O^\star) = \frac{P(O|D) \cdot \frac{2}{n\cdot(n-1)}}{P(O^\star|D) \cdot \frac{2}{n\cdot(n-1)}} = \frac{\frac{P(D|O)\cdot P(O)}{P(D)}}{\frac{P(D|O^\star)\cdot P(O^\star)}{P(D)}} == \frac{P(D|O)}{P(D|O^\star)}$$

The likelihoods $P(O|D)$ and $P(O^\star|D)$ can be computed using Formula (3.39). But as the orderings $O = O(\sigma)$ and $O^\star = O(\sigma^\star)$ differ by the exchange of two nodes $X_{\sigma(j)}$ and $X_{\sigma(k)}$ only, the factors for the nodes that precede $X_{\sigma(j)}$ or succeed $X_{\sigma(k)}$ in both orderings do not change in Formula (3.39), that is $V_i(\sigma) = V_i(\sigma^\star)$ for $i < j$ as well as for $i > k$. Consequently, the ratio of likelihoods reduces to:

$$\frac{P(D|O)}{P(D|O^\star)} = \prod_{i=j}^{k} \frac{\sum_{P\in V_i(\sigma)} P(pa(X_i) = P) \cdot Score(X_i|pa(X_i) = P, D)}{\sum_{P\in V_i(\sigma^\star)} P(pa(X_i) = P) \cdot Score(X_i|pa(X_i) = P, D)} \tag{3.41}$$

To avoid unnecessary computations it is advisable to precompute for each domain node the scores of all its possible parent sets in advance, that is before starting the MCMC-Order simulation. So, instead of computing lots of local scores again and again within each Order-MCMC step, for each node the local scores of its valid parent sets can be searched in precomputed lists and summed up in the MCMC steps accordingly.

When a sufficiently restrictive fan-in K is imposed, the computational complexity of this pre-processing is of order n^{K+1}. However, for domains with lots of nodes, that is high number of variables n, or unrestricted fan-in, it is not an useful way to store all these local scores. This is due to the fact, that for searching the valid parent-sets in extensive precomputed and stored lists, lots of computational time is needed, so that the Order-MCMC algorithm becoms too slow then. [15] recommed some heuristic computational tricks for such domains with lots of variables.

The Order-MCMC approach of [15] for sampling node orderings from their posterior distribution can be summarised as follows:

- Initialisation: Choose an arbitrary node ordering O_1 and set $M_1 = O_1$. For example initialise the Markov Chain by a randomly determined node ordering.
- Iteration: For i=1,2,3,...: Given a realisation $O_i = O^\star$ of M_i randomly choose an ordering O out of the set $\Pi(O^\star)$ (which consists of all node orderings that can be reached from $O^\star$ by exchanging two nodes for each other and leaving all other nodes in the ordering $O^\star$ unchanged) from the proposal distribution:

$$Q(O|O^\star) = \left\{ \begin{array}{cc} \frac{2}{n\cdot(n-1)} & , O \in \Pi(O^\star) \\ 0 & , O \notin \Pi(O^\star) \end{array} \right\}$$

  Accept the randomly choosen ordering O with the acceptance probability $A(O|O^\star) = min\{1, R(O|O^\star)\}$. If O is accepted, set $M_{i+1} = O$. Otherwise leave the Markov Chain unchanged, that is, set $M_{i+1} = O^\star$.

In analogy to the Structure-MCMC approach it takes 'some time' until the Markov Chain converges to its stationary distribution. So, one has to sample from the Chain for 'long enough' to ensure that it has reached its stationary distribution. After the *burn-in time* the DAGs can be seen as sampled from the posterior distribution. So, Order-MCMC ouptputs a sample of node orderings $O_1, ...O_M$ which, if convergence of the Markov Chain was actually reached, is a sample from the posterior distribution over

node ordering $P(O|D)$.

The idea of Order-MCMC is to use this sample to obtain a sample of DAGs. That is for each sampled ordering $O_i = O_i(\sigma^i)$ a DAG G_i can be sampled out of the posterior distribution $P(G|O_i(\sigma^i), D)$, that is the posterior distribution over DAGs given the ordering O_i and the data D. Conditional on the node ordering, for each node its parent set can be sampled independently with respect to its valid parent-sets in O_i. So, for each domain node X_i its parent set can be sampled out of the following posterior distributions:

$$P(pa(X_i) = P_0|O_i(\sigma^i), D) = \frac{I_{\{V_i(\sigma^i)\}}(P_0) \cdot P(pa(X_i)=P_0) \cdot Score(X_i|pa(X_i)=P_0,D)}{\sum_{P \in V_i(\sigma^i)} P(pa(X_i)=P) \cdot Score(X_i|pa(X_i)=P,D)}$$

Thereby the indicator function $I_{\{V_i(\sigma^i)\}}(P_0)$ is equal to one if the condition $P_0 \in V_i(\sigma^i)$ is true, and zero otherwise. Subsequently, the complete DAG can be obtained straightforwardly: For each domain variable and each of its parent nodes there is one edge pointing from the parent nodes to the node itself. Due to the ondition on the node ordering the final DAG that consists of all these edges is guaranteed to be an acyclic one. So for each sampled node ordering a DAG can be obtained, so that in the end a DAG sample $G_1, ..., G_M$ becomes extracted from the outputed sample of node orderings $O_1 = O_1(\sigma^i), ..., O_M = O_M(\sigma^M)$.

Although [15] show that Order-MCMC is superior to Structure-MCMC with regard to convergence and mixing of the resulting Markov Chain, there is a substantial drawback of the Order-MCMC sampling scheme. Using Order-MCMC the prior over DAGs, which has usually a substantial influence on the posterior probabilities and the outcome of the inference, cannot be defined explicitly. More precisely, the assumption that each ordering O has the same prior probability $P(O)$ leads to a change of the form of the originally determined prior over DAGs $P(G)$. DAGs that are consistent with more orderings are more likely than DAGs consistent with fewer orderings. For instance, the DAG without any edge can be sampled out of all $n!$ node orderings, while a DAG of the type $X_{\sigma(1)} \rightarrow X_{\sigma(2)} \rightarrow .... \rightarrow X_{\sigma(n)}$ can be sampled out of one single node ordering, namely

$O = (X_{\sigma(1)}, ..., X_{\sigma(n)})$, only. [15] argue that 'the standard priors over DAGs are often used not because they are particulary appropriate for a task, but rather because they are simple and easy to work with' and justify their approach with the argument that 'a DAG that is consistent with more orderings makes fewer assumptions about causal ordering, and therefore should be more likely a priori'. In addition [15] present results of simulation studies for domains with small n which show that the bias in the prior over DAGs implied through Order-MCMC is of minor degree only.

3.5.4. Relation-Features

Either the Structure-MCMC sampling scheme (see Subsection 3.5.3.1) or the Order-MCMC sampling scheme (see Subsection 3.5.3.2) can be used to obtain a sample of DAGs from the posterior distribution $P(G|D)$ over directed acyclic graphs. Once such a DAG sample $G_1, ..., G_M$ is present, it is useful to search for 'features' that are common to most of the DAGs in the sample. Informally, a 'feature' indicates the presence of a particular directed or undirected edge or a particular set of edges in a DAG or its CPDAG representation. Thus 'features' can be seen as structural properties of Bayesian networks. Especially, the posterior probabilities of these 'features' given the data D are quantities of interest.

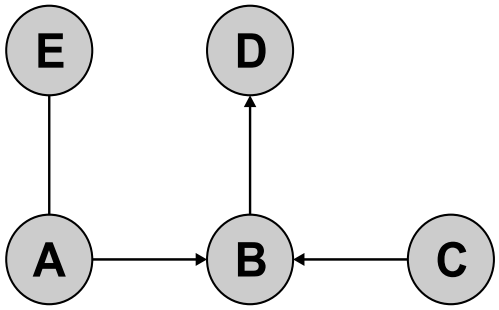


Figure 3.4.: A CPDAG for a domain with 5 variables $\{A, B, C, D, E\}$

[14] recommend to convert each sampled DAG into its CPDAG (see Subsection 3.5.1)

first. How to convert between DAGs and CPDAGs is described in detail in [9]. Subsequently, it can be searched for 'features' in the extracted CPDAG sample, whereby each CPDAG consists of directed and undirected edges. For the remainder of this section it is assumed that the DAG sample $G_1, ..., G_M$ has been converted into the corresponding CPDAG sample. For example, the CPDAG presented in Figure 3.4 contains the following *relation-features*:

- a directed edge from A to B: '$A \to B$'
- an undirected edge between A and E: 'A—E'
- a directed path from A to D: '$A \to B \to D$'
- etc.

More formally, a feature F is a binary indicator-variable over the space of DAGs, which is 1 if the feature is present in a certain DAG, and 0 otherwise. Thereby the features are usually characterised through properties of the corresponding CPDAG.

$$F : \Omega \to \Omega^\star \to \{0, 1\}$$

where Ω represents the space of DAGs and $\Omega^\star$ is the space of CPDAGs.

The most important features are the following ones. Thereby X and Y are two nodes in a CPDAG G.

- **Order-relation-features** If the CPDAG G contains a directed path, that is a path from node X to node Y in which all edges are directed, then X and Y are in *order-relation*. More precisely, there is an order-relation '$F_{\triangleright}(X, Y) = 1$' in G if and only if X is an ancestor of Y in G. In Figure 3.4 the order-relations are given by $F_{\triangleright}(A, B) = 1$, $F_{\triangleright}(C, B) = 1$, $F_{\triangleright}(B, D) = 1$, and $F_{\triangleright}(A, D) = 1$. There are no other order-relations. Therefore all other order-relation-features are of measure zero. In

the context of gene expression data order-relations can be seen as indications for causation. $F_{\triangleright}(X, Y) = 1$ means that X is a cause of Y.

- **Markov-relation-features** The nodes X and Y are in *Markov-relation* in G, if there is an (directed or undirected) edge between them, or if they have a common child node Z, that is if there are two directed edges '$X \rightarrow Z$' and '$Y \rightarrow Z$' converging on the same common child node Z. Markov-relations are symmetric, that is the relationship $F_M(X, Y) = F_M(Y, X)$ holds. In Figure 3.4 there are 8 Markov-relations implied through (directed and undirected) edge connections: $F_M(A, E) = 1$, $F_M(E, A) = 1$, $F_M(A, B) = 1$, $F_M(B, A) = 1$, $F_M(B, C) = 1$, $F_M(C, B) = 1$, $F_M(B, D) = 1$, and $F_M(D, B) = 1$ as well as two additional Markov-relations since nodes A and C have a common child, namely B: $F_M(A, C) = 1$ and $F_M(C, A) = 1$. In the context of gene expression data a Markov-relation indicates that the two genes are related in some joint biological regulation process or interaction.

- **Directed- and Undirected-edge-relation-features** The nodes X and Y are in *directed-edge-relation* in G if there is a directed edge '$X \rightarrow Y$' from X to Y. Accordingly, they are in *undirected-edge-relation* in G if there is an undirected edge 'X—Y' between X and Y in G. Directed-edge-relations are special cases of order-relations. In Figure 3.4 there are two undirected-edge-relations: $F_{-}(A, E) = 1$ and $F_{-}(E, A) = 1$ as well as three directed-edge-relations: $F_{\rightarrow}(A, B) = 1$, $F_{\rightarrow}(B, D) = 1$, and $F_{\rightarrow}(C, B) = 1$.

- **Individual-edge-relation-features** There is an *individual edge relation* from X to Y in G, if there is eiter a directed edge from X to Y or an undirected edge between X and Y. In Figure 3.4 there are five individual-edge-relations: $F_{\succ}(A, B) = 1$, $F_{\succ}(C, B) = 1$, $F_{\succ}(B, D) = 1$, $F_{\succ}(A, E) = 1$, and $F_{\succ}(E, A) = 1$.

The next question is to what extend the data D support a particular feature F. If the data are sparse there can be many DAGs (CPDAGs) that explain the data equally well.

Unfortunately, these DAGs (CPDAGs) can have very different sets of edges. Consequently, there are DAGs (CPDAGs) that contain a feature while others do not. Therefore, for every feature of interest F one has to estimate the posterior probability $P(F|D)$ of this feature given the data D. This probability is also called the *confidence* of the feature and is given by:

$$P(F|D) = \sum_{G \in \Omega} F(G) \cdot P(G|D) \tag{3.42}$$

Because exact computation of the posterior probability is impractical due to the fact that the number of valid DAGs in Ω is exponential in the number of variables n, this posterior probability has to be estimated by the aid of the sample $G_1, ..., G_M$. An estimator is given by the fraction of DAGs (CPDAGs) that contain the feature of interest. For each feature F the corresponding estimator is given by:

$$\widehat{P(F|D)} = \frac{1}{M} \sum_{i=1}^{M} F(G_i) \tag{3.43}$$

Although pairwise relation-features give some insight into the biological phenomena captured by the data, the view remains limited to pairwise relations. Therefore, [14] discuss how to identify broader structures with the aid of individual confidences of the Markov-relation-features. Within their framework pairwise relations are brought together with the aim to extract sub-graphs with a high concentration of Markov-relation-features of high confidence. Their *score-based approach* is presented in A. Appendix I.

3.5.5. Analysing interventional data with Bayesian networks

Although most of the available biological expression and pathway data bases are passively observed (that is so called *observational data*), sometimes experimenters can manipulate single domain variables in some experiments and observe the resulting values of the other domain variables. Such data are called *ideal interventional data*. Since in these interventional experiments the manipulated variables are usually either inhibited

(down-regulated) or activated (up-regulated) by the experimenter, e.g. through special experimental conditions, their values are not stochastic any longer. From a theoretical point of view this means that the values of these variables are deterministically assigned specific values. Thus these values obtained by experimental intervention can not depend on the values of the other domain variables. But on the other hand these assigned values can influence the values of other domain variables. Consequently, the intervened data points are extremely useful for discovering causal relationships (directed edges). Under fairly weak conditions a combination of observational and ideal interventional data can be analysed using Bayesian networks. These conditions are described in detail in [51] for the *BDe scoring metric* and in [50] for the *BGe scoring metric*. Only two little modifications are necessary.

First, in the likelihoods, which are products of local scores (see Formula (3.21)), each local score becomes restricted to those relevant data points where the variable itself was not intervened.

That is the likelihood for pure observational data (see Formula (3.21)):

$$P(D|G) = \prod_{i=1}^{n} P(X_i = D_{X_i}|pa(X_i) = D_{pa(X_i)}) = \prod_{i=1}^{n} Score(X_i|D, pa(X_i))$$

has to be replaced by:

$$P(D^-|G) = \prod_{i=1}^{n} P(X_i = D_{X_i}^{-i}|pa(X_i) = D_{pa(X_i)}^{-i}) = \prod_{i=1}^{n} Score(X_i|D^{-i}, pa(X_i))$$

D^- is a set of n data sets D^{-i} (one for each domain variable X_i) in which the intervened observations of the indicated variable are deleted from D. More precisely, $D_{X_i}^{-i}$ consists of the observations of domain variable X_i, where X_i itself was not intervened. Accordingly, $D_{pa(X_i)}^{-i}$ consists of the observations of the variables in the parent set $pa(X_i)$, where their common child node X_i was not intervened.

Secondly, the definition of equivalence classes (see Subsection 3.5.1) must be changed. While for pure observational data sets two DAGs assert the same set of independency assumptions among the domain variables, if and only if they have the same skeleton

and the same set of v-structures (see Subsection 3.5.1), this definition of 'equivalence' does not make sense for data sets which are a mixture of observational and ideal interventional measurements.

To see that, it is useful to consider a very simple example. If in a domain with two variables A and B, node B is set to the deterministic value b through experimental condition, then this manipulation can not influence the distribution of node A in DAG G_1: '$A \rightarrow B$', as A does not depend on B in G_1. That is the probability $P(A = a, B = b|G_1, B = b)$ reduces to $P(A = a)$. On the other hand in DAG G_2: '$A \leftarrow B$' the manipulation of node B causes a change in the distribution of node A, as A depends on B. That is $P(A = a, B = b|G_2, B = b)$ is equal to $P(A = a|B = b)$. Thus although both DAGs are equivalent with the same CPDAG representation 'A—B', the (in-)dependence relations differ for interventional data points. While for the non-interventional observations holds $P(A, B|G_1) = P(A, B|G_2)$, the conditional distributions $P(A, B|G_1, B = b)$ and $P(A, B|G_2, B = b)$ are different.

[47] show that two DAGs assert the same set of (in-)dependence assumptions among the variables for a mixture of observational and ideal interventional data if and only if they are *equivalent*, that is have the same skeleton and the same set of v-structures, and additionaly the same set of parents for each domain variable which was manipulated in at least one observation. The resulting equivalence classes are referred to as *transition-sequence equivalence* or *TS equivalence* classes. Two DAGs that assert the same set of independency assumptions among the variables for a mixture of observational and ideal interventional data are said to be *TS-equivalent*.

All edges being connected with an intervened node become automatically directed in the CPDAG representation, if the concept of *TS equivalence* is used. As a consequence new v-structures come into being and further edges not entering or leaving an intervened node become directed too. Thus, usually much more information about causality can be gained from interventional data. The algorithms of [9] that can be used to convert DAGs to CPDAGs and vice versa, if the usual concept of *equivalence* is used, can be

easily adapted to the concept of *TS equivalence*. In the original DAG for each node, that was intervened at least in one observation, two *dummy nodes* have to be added as parents of this node. That is for each intervened node two dummy nodes and for both dummy nodes one directed dummy edge pointing from the dummy node to the intervened node are added. Subsequently, the new DAG including dummy nodes and directed dummy edges can be converted into its CPDAG representation, using the algorithm of [9]. Due to the addition of directed dummy edges new v-structures result, and thus edges of the original DAG become directed due to the new v-structures. Finally, the deletion of all dummy nodes and dummy edges from the extracted CPDAG-representation results in the CPDAG representation of the original DAG without any dummy node in terms of *TS equivalence*.

How to obtain the TS equivalence CPDAG for the simple network '$A \rightarrow B$', in which node B was intervened, is illustrated in Figure 3.5. Due to the addition of two dummy parent nodes D_1 and D_2 for the intervened node B, the edge from A to B participates in four v-structures, e.g. (A, B, D_1), and so becomes directed. After removing the dummy components, the edge is left directed. If node B hadn't been an intervened one, the dummy nodes wouldn't have been added to the two real domain nodes, and the edge would have been an undirected one in the CPDAG representation, as there would not have been any v-structure.

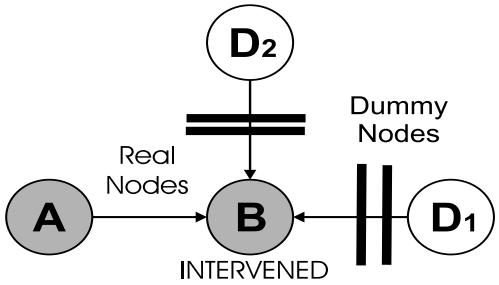


Figure 3.5.: Converting DAGs into CPDAGs for TS equivalence

In analogy the DAG '$A \leftarrow B$' becomes converted to the CPDAG '$A \leftarrow B$' if TS equivalence is applied, as the reversal of the edge would lead to new v-structures, so its direction

is not *reversible* either.

3.6. Measures for goodness of performance

Various reverse engineering machine learning methods which can be used to infer the architecture of biochemical pathways and regulatory networks have been proposed in the literature. The most important and widely used methods have been described at the beginning of this chapter. Usually these methods are applied to biological data, whereby the real regulatory mechanisms are unknown. So, they can be used to generate new hypotheses about biological phenomena, which either can be confirmed by the information in biological data bases or traditional molecular biology experiments, or otherwise must be seen with caution. Although the statistical model behind the machine learning method may find relations that are supported by the available exprseeion data, there is no guarantee whether these findings represent real biological relationships or not. Therefore it is necessary to test the performance of a machine learning method before applying it to real biological expression data, where the regulatory relationships are unknown. Such a performance test can be done by generating data from a known synthetic network and searching with the machine learning method for the relationships in the data. Different methods of generating synthetic regulatory network data are described in Section 3.7. Afterwards, the findings can be compared with the relations in the known true network topology (graph), and the 'gooddness' of the method's performance can be evaluated. Alternatively, the findings derived from real biological data can be compared with a biologically accepted 'true gold standard network (graph)', that is a regulatory network that can be considered as a reliable one with respect to the molecular biological 'up-to-date' knowledge.

This section deals with the concept of *ROC curves* and *AUROC values* that can be used to evaluate the *ranking quality* of machine learning methods. In the context of graphical models each possible edge of the domain obtains a *confidence-score*, e.g. a

		$A \rightarrow B$	$A \leftarrow$ B	A—B	A B
UGE	TP	1	1	1	0
	FP	0	0	0	0
	TN	0	0	0	0
	FN	0	0	0	1
DGE	TP	1	0	1	0
	FP	0	1	1	0
	TN	1	0	0	1
	FN	0	1	0	1

Table 3.2.: Comparison between the undirected graph evaluation (UGE) and the directed graph evaluation (DGE) for a true directed edge from node A to node B: $A \rightarrow B$. The top row shows the learnt edges. TP stands for true positive count, FP stands for false positive count, TN stands for true negative count, and FN stands for false negative count.

posterior probability, that indicates the confidence of its presence given the model and the data. So, all possible edges can be ranked with respect to their confidence-scores. *ROC curves* visualise the distribution of the true edges within this ranking in terms of *sensitivity* and *specificity*, so that the performance can be visually evaluated. More precisely, ROC curves visualise which fraction of the true edges can be found if accepting different fractions of false edge findings. *AUROC values*, which can be computed from such ROC curves, summarise the 'goodness of ranking' in integer values. The concept of ROC curves and AUROC values was originally introduced in signal detection theory (see [13]), and was first applied in the context of learning graphical models by [26].

In analogy to the theoretical graphical models presented at the beginning of this chapter, it is assumed that the true regulatory network G_{true} (or at least a gold standard network) for a set of n domain variables $X_1, ..., X_n$ is given. The true network is a graph which consists of directed edges information e_{ij} $(i, j \in \{1, ..., n\})$. e_{ij} indicates a directed edge pointing from domain node X_i to node X_j, and $e_{ij} = 1$ means that this edge is present, while $e_{ij} = 0$ means that there is no edge from X_i to X_j in G_{true}. Inference methods usually ouput confidence-scores instead. That is for each directed edge e_{ik} a confidence score $\psi(e_{ik})$, such that the confidence increases with increasing values of

$\psi(e_{ik})$, is outputed. Using a cut-point for the outputed confidence-scores the inference method's output can be discretised into a graph G, where each edge is either present or absent. As the inference method applied to learning G_{true} are based on different models, this discretisation may lead to an undirected, a directed, or a partially directed graph G.

To assess the performance of the learning method in terms of ROC curves, two different criteria can be applied. The first approach, referred to as the *undirected graph evaluation* (UGE), discards the information about the edge directions altogether. To this end, G_{true} is replaced by its skeleton, where the skeleton of a general graph is defined as the graph in which two nodes are connected by an undirected edge whenever these nodes are connected by any type of edge in the original graph. More precisely, each directed edge information e_{ik} is simply replaced by the undirected edge information $e^{\star}_{ik} = max\{e_{ik}, e_{ki}\} \in \{0, 1\}$. The methods based on undirected edges, such as Relevance networks and Gaussian graphical models, output confidence-scores for undirected edges only. That is each pair of confidence-scores $(\psi(e_{ik}), \psi(e_{ki}))$ with $\psi(e_{ik}) = \psi(e_{ki})$ can be directly compared with the corresponding undirected edge information pair $(e^{\star}_{ik}, e^{\star}_{ki})$. For Bayesian networks, which are based on directed eges, the posterior probabilities of the symmetric *undirected-edge-relation-features* can be used to build pairs of confidence-scores for the UGE criteria. The second approach, referred to as the *directed graph evaluation* (DGE), compares the learnt graph G with the original graph G_{true}. Thereby for the Relevance networks and the Gaussian graphical models a learnt undirected edge ($\psi(e_{ik})$,$\psi(e_{ki})$) with $\psi(e_{ik}) = \psi(e_{ki})$ is interpreted as a superposition of two directed edges, pointing in opposite directions. That is $\psi(e_{ik})$ is compared with e_{ik} and $\psi(e_{ki})$ is compared with e_{ki} at the same time. Consequently, even if there is an edge between X_i and X_k in G_{true}, there is always a false positive finding, as $e_{ik} \neq e_{ki}$. For Bayesian networks, which are based on directed eges, the posterior probabilities of the *individual-edge-relation-features* can be used as confidence-scores $\psi(e_{ik})$. A comparison of the two scoring schemes is shown in Tables 3.2 and 3.3.

What follows is the exact mathematical description of ROC curves and AUROC values.

		$A \rightarrow B$	$A \leftarrow$ B	A—B	A B
UGE	TP	0	0	0	0
	FP	0	0	0	0
	TN	0	0	0	1
	FN	1	1	1	0
DGE	TP	0	0	0	0
	FP	1	1	0	0
	TN	1	1	0	2
	FN	0	0	2	0

Table 3.3.: Comparison between the undirected graph evaluation (UGE) and the directed graph evaluation (DGE) if there is no connecting edge between node A and node B in the true graph: '$A\ B$'. The top row shows the learnt edges. TP stands for true positive count, FP stands for false positive count, TN stands for true negative count, and FN stands for false negative count.

This methodology can be used for both criteria: for the UGE as well as for the DGE scheme.

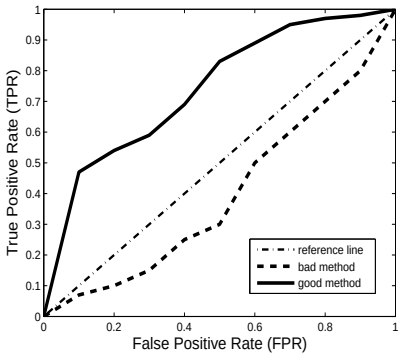


Figure 3.6.: Example of ROC curves

As mentioned above e_{ik} denotes an directed edge from node X_i to node X_k and the inference method outputs a confidence-score $\psi(e_{ik})$ for each edge. Let $\epsilon(\theta) = \{e_{ik} | \psi(e_{ik}) > \theta\}$ denote the set of all edges whose confidence-scores exceed a given confidence-threshold θ. For a given threshold θ the number of true positive (TP), false positive (FP), and false negative (FN) edge findings can be counted, and the *true positive rate*

$TPR = TP/(TP + FN)$ and the *false positive rate* $FPR = FP/(TN + FP)$ can be computed. The true positive rate TPR is also referred to as *sensitivity*, and the false positive rate FPR is also referred to as *inverse specificity*.

But rather than selecting an arbitrary value for the threshold θ, this procedure can be repeated for several values of θ and the ensuing TPR scores can be plotted against the corresponding FPR scores. This gives the *receiver operator charcteristic* (ROC) curves. Loosely speaking, such ROC curves show which rate of erroneously learnt edges (FPR) must be accepted to obtain a desired recovery rate of true positive edges (TPR). As an example, the ROC curves of two different methods for learning the relationships in a domain are given in Figure 3.6. The thin diagonal dashed line is a reference line. It corresponds to a 'virtual' learning method that asserts the same confidence-score to all possible edges. Consequently, it either outputs that no edge is present at all (TPR=0 and FPR=0) or it outputs that all edges are present (TPR=1 and FPR=1). And there are no further grades in between. Alternatively, the 'virtual' method can be interpreted as a 'random' predictor method which assigns random confidence-scores to the edges. From this point of view the diagonal dashed ROC curve can be interpreted as the expected ROC curve for such a random method. The dashed thick line corresponds to a 'bad' learning method as the TPR (sensitivity) is lower than the FPR (inverse specificity). The thick solid line corresponds to a 'good' learning method as the TPR rates are higher than the FPR rates for all thresholds. Especially, for the same FPR rate (x-axis)the 'good' method recovers more true edges than the 'bad' method and the 'random' method.

Different AUROC values can be computed from such ROC curves. Firstly, it is often useful to compute the complete area under the ROC curve (AUROC_1), where larger values indicate better performances. However, the right change of the inverse specificity (FPR) is usually of no practical interest as the number of false positive (FP) counts, in absolute terms, would be unreasonably high. For this reason it is sometimes better to

AUROC-UGE	Area under the whole ROC curve, obtained from undirected edges
AUROC_ϵ-UGE	Area under the left part of the ROC curve with FPR$<\epsilon$, obtained from undirected edges
AUROC-DGE	Area under the whole ROC curve, obtained from directed edges
AUROC_ϵ-DGE	Area under the left part of the ROC curve with FPR$<\epsilon$, obtained from directed edges

Table 3.4.: Figures of merit for evaluating the performance of a method.

compute the area under the ROC curve up to a small, pre-specified upper limit on the FPR: $FPR < \epsilon$. This yields the AUROC_ϵ score. For example, $\epsilon = 0.1$, corresponds to an upper bound on the false positive rate (FPR) of 10 percent. In the end, this leads to four different 'figures of merit' for assessing the performance of a machine learning inference method, which are summarised in Table 3.4. However, in the comparative evaluation study (see Chapter 5) the performances of the different machine learning methods will be mainly measured in terms of AUROC_1 values (representing the complete areas under the ROC curves), as the specified threshold $\epsilon = 0.1$ is arbitrarily selected, and so may distort the results in favour of some machine learning methods.

The computation of the area under the ROC curve can be done by numerical integration, e.g. the *trapezodial method*. Using trapezoidal numerical integration for the ROC curves in Figure 3.6 the 'bad' method obtains the value: $\text{AUROC}_1 = 0.397$, and the 'good' method obtains the value AUROC_1=0.741. The reference line leads to the reference value AUROC_1 =0.5. So, these AUROC values reveal that the 'good' method clearly outperforms the 'bad' and the 'random' learning method.

3.7. Generating synthetic network data

In this subsection two methods of generating data from a synthetic regulatory network are presented. As before, it is assumed that a domain with n variables $X_1, ..., X_n$ is

given, and that the 'qualitative structure' of the true regulatory network for the domain is known. In this context 'qualitative structure' means that only the complete directed edge information e_{ij} $(i, j \in \{1, ..., n\})$ is known, while the exact relationships (regulatory mechanisms) are unknown. That is, e_{ij} indicates a directed edge pointing from domain node X_i to node X_j, and it is known, whether this edge is present $(e_{ij} = 1)$ or not $(e_{ij} = 0)$. Furthermore, it is assumed that the edge information belongs to a DAG, that is a directed acyclic graph. Given this qualitative information (the DAG) of the network, the regulatory mechanisms can be implemented as a Bayesian network with Gaussian scoring metric (BGe) (see Subsection 3.7.1) or as a steady-state-approximation to a system of coupled differential equations (see Subsection 3.7.2). Although the former method of generating data is surely less biologically realistic, it is useful to include such data, as they can be learnt much more easily than data sets generated with more complicated methods. Furthermore, it is assumed that at most three edges can point on the same domain node, that is for each node X_i the cardinality of the set $\{X_i | e_{ij} = 1\}$ is restricted to a so called *fan-in* of size three.

Especially in Netbuilder, the noise level was specified in terms of 'dynamic' noise instead of adding some 'observational' ('experimental') noise (in the sense of erroneous or improper expression measurements) after having generated the data. That is, the realisations of each domain node X having k parent nodes P_1,...,P_k is the sum of the 'signals' transmitted from these parent nodes in form of a functional relationship: f(P_1,...,P_k) and some additional random noise ϵ. Afterwards X transmits its realisation of the form f(P_1,...,P_k)+ϵ to its child nodes. As mentioned before, alternatively, it could have been specified that X transmits exclusively its determinstc part f(P_1,...,P_k) (without the noise variable ϵ) to its child nodes. Then ϵ could be interpreted as subsequently added 'observational' noise. But as such noise weakens direct as well as indirect associations between the domain variables in equal measure, it was decided to use 'dynamic' noise instead. Different levels of noise were specified in the Netbuilder data generator, while in the Gaussian data generator no different noise levels were distinguished.

3.7.1. Bayesian network data generator

Within the *Bayesian network data generator* non-interventional observations for each domain node can be sampled from the following univariate Gaussian distributions:

$$X_i \sim N(\sum_{j=1}^{n} b_{ij}(x_j - \mu_j), \sigma_i^2) \tag{3.44}$$

Thereby x_j represents the value of the j-th domain variable X_j. All variances σ_i^2 are set to 0.01, and the regression coefficients b_{ij} are independently sampled from uniform distributions over the intervals $[-2, -0.5] \cup [0.5, 2]$ if there is an edge pointing from node X_j to node X_i ($e_{ji} = 1$). Otherwise, that is if there is no edge from X_j to X_i ($e_{ji} = 0$), the corresponding regression coefficients are set equal to zero. It follows from Formula (3.44) that the means $E[X_j] = \mu_j$ of all variables are 0. From a statistical point of view, after having sampled the regression coefficients, the Gaussian distribution of domain variable X_i in Formula (3.44) can be interpreted as the following conditional distribution: $P(X_i | \{X_j = x_j | j \in \{1, ..., n\} : e_{ji} = 1\})$. Subsequently, observations can be sampled from these distributions.

But if the distribution of a variable X_i depends on the realisations of other nodes, that is if the set $\{X_j : e_{ji} = 1\}$ is non-empty, the realisations of those other domain variables have to be sampled in advance. But as the 'qualitative structure' of the network is assumed to be a DAG, that is a directed graph without any cycles, a simple recursive algorithm can be used to sample observations for the parents of each node beforehand. Mathematically more precisely, the nodes have to be sorted *topologically* first, so that a node ordering $X_{\sigma(1)}, ..., X_{\sigma(n)}$ results, in which every node exceeds ist parents nodes (see [9])). Afterwards the observations of the variables can be sampled with respect to the topological ordering.

Intervened observations can be sampled as follows: The values of inhibited (down-regulated) variables are sampled from a Gaussian $N(0|0.01)$ distribution and the val-

ues for activated (up-regulated) variables are sampled from the Gaussian distribution of X_i conditional on the set $\{X_i < q_{0.025} \vee X_i > q_{0.975}\}$, whereby q_α represents the α -quantile of X_i's unconditional distribution. Thereby, the parameters of the unconditional Gaussian distribution of X_i can be computed from the sampled regression coefficients b_{ij} and variance-parameters $\sigma_i^2 = 0.01$ of the conditional distributions in Formula (3.44) using the *recursive transformation* algorithm presented at the beginning of Subsection 3.5.2.2. In the notation intoduced in Subsection 3.5.2.2 the unconditional Gaussian distribution of domain variable X_i is $N(0, \Sigma_{i,i})$. Whereby $\Sigma_{i,i} = (W^{-1})_{i,i}$ is the i-th diagonal element of the covariance matrix, which in turn is the i-th diagonal element of the inverse of the precison matrix W. See Subsection 3.5.2.2 for further details.

Sampling from a $N(0|0.01)$ can be interpreted as sampling a 'weak' expression (signal), while sampling from the conditional $P(X_i|X_i < q_{0.025} \vee X_i > q_{0.975})$ leads to signals that are 'stronger' than 95 percent of the pure observational signals of the unconditional distribution of X_i. Especially, for intervened observations of X_i the values are sampled independently from the realisations of the set $\{X_j|e_{ji} = 1\}$. That is, although the values of intervened observations may influence the realisations of other domain variables, if there are edges pointing away from them, they themselves do not depend on any other node, even if there are edges pointing on them.

Finally, the Bayesian network data generator can be modified, so that non-linear and interacting regulations can be modelled. This can be done by simply adding some non-linear regression terms to Formula (3.44). For variables without parents nothing changes, for variables with one or three parents Formula (3.44) is replaced by:

$$X_i \sim N(\sum_{j=1}^{n}(1-p) \cdot b_{ij}(x_j - \mu_j) + \sum_{j=1}^{n} p \cdot (-1) \cdot b_{ij} \cdot |x_j - \mu_j|, \sigma_i^2), \tag{3.45}$$

And for variables with two parents both coefficients $b_{ij} \neq 0$ are forced to have the same sign.

Afterwards Formula (3.44) can be replaced for:

$$X_i \sim N(\sum_{j=1}^{n}(1-p)\cdot b_{ij}(x_j-\mu_j)+p\cdot(-1)\sum_{j\neq k} f(b_{ij},b_{ik},x_j,x_k),\sigma_i^2), \tag{3.46}$$

whereby

$$f(b_{ij},b_{ik},x_j,x_k) = I_{\{b_{ij}\cdot b_{ik}>0\}}\sqrt{|x_j-\mu_j|}\cdot\sqrt{|x_k-\mu_k|}\cdot sign((x_j-\mu_j)\cdot(x_k-\mu_k)) \tag{3.47}$$

In Formula (3.45) and Formula (3.46) the user defined parameter $p \in [0,1]$ represents the strength of the non-linearity. For $p = 0$ there is no non-linearity and for $p = 1$ there is exclusively non-linear-regulation. The non-linearity is obtained by adding either squareroot or absolut value terms. These terms have to be used instead of the more 'usual' quadratic and product terms, because the latter ones do not lead to values that are comparable to the values of the linear effects. $I_{\{.\}}$ is the indicator function. Due to the non-linear terms holds that for $p > 0$ the means μ_j of all variables with parents become unequal to zero. So, they must be computed numerically in advance. But instead of computing these means they can also be estimated by the sample mean of some million observations generated independently for each variable. But here again, if a variable X_i depends on the realisations of other nodes, that is if the set $\{X_j : e_{ji} = 1\}$ is non-empty, the realisations (means) of those variables have to be sampled (estimated) in advance. The following algorithm can be used:

Procedure for estimating the theoretical means:

1. Sort the domain nodes *topologically* with respect to the given DAG, that is find an ordering $X_{\sigma(1)},...,X_{\sigma(n)}$ of the domain nodes, in which each node exceeds its parent nodes. Thereby σ is a permutation of the set $\{1,...,n\}$. An algorithm for sorting the nodes of a DAG topologically is given in [9].

2. Define a vector V of length n, in which the estimated means can be stored, and define a (n,M)-matrix X, in which the M realisations for each of the n domain

variables can be stored.

Initialise both the vector V and the matrix M with zero entries.

- For i=1,...,n:

  Sample M realisations of variable $X_{\sigma(i)}$. Thereby it depends on the cardinality of the set $\left\{X_{\sigma(i)} : e_{j\sigma(i)} = 1\right\}$, whether the Gaussian distribution in Formula (3.44), (3.45), or (3.46) must be used for sampling. However, in the corresponding formula replace for $j = 1, ..., n$ the mean μ_j by the j-th entry of the vector V.

  - Then, for k=1,...,M:

    Replace the observation x_j by the (j,k)-th element of X in the formula, and sample a value x for variable $X_{\sigma(i)}$ from the resulted distribution.

    Store the sampled value x as ($\sigma(i)$,k)-th entry of matrix X.

- Compute the empirical mean of the $\sigma(i)$-th row of X, and store it as the $\sigma(i)$-th entry of the vector V.

3. Output the vector V and output the matrix X. The j-th element of V is an estimation of the parameter μ_j.

If the estimations in vector V are based on a very high number of sampled observations M, e.g. M=10 million, the estimations become good enough, so that the unknown true parameters μ_j can be replaced by these estimations. Afterwards, a synthetic data set can be generated using the conditional Gaussian distributions in Formulae (3.44), (3.45), and (3.46).

As for $p > 0$ different non-linear terms are added to the original equation, the joint distribution of the variables $P(X_1, ..., X_n)$ as well as the unconditional distributions $P(X_i)$ are no longer Gaussian distributions. Only the conditional distributions $P(X_i | \{X_j = x_j | e_{ji} = 1\})$ are still Gaussian. Consequently, for $p > 0$ intervened obser-

vations can not be sampled as described above. But such observations can be generated using the following procedure instead:

Procedure for sampling interventional non-linear data:

1. Input the vector V and the matrix X which were outputed from the procedure for estimating the theoretical means (see above).

2. For each domain variable X_i use the i-th entry of V as an estimation of its mean $\mu_i = E[X_i]$ as before, and compute the empirical 0.025-quantil $q_{0.025}$ and the empirical 0.975-quantil $q_{0.975}$ of the set $\{X_{i,1}, ..., X_{i,M}\}$, that is the i-th row of X.

3. Sample inhibited observations of X_i from the distribution $N(V(i), 0.01)$, and activated observations from a discrete uniform distribution over the set: $\{x \in \{X_{i,1}, ..., X_{i,M}\} \,|x < q_{0.025} \vee x > q_{0.975}\}$.

This procedure guarantees that the signals of inhibited observations become very weak, as they are sampled from a Gaussian distribution around the estimated mean $\widehat{\mu_i} = V(i)$. And as it can be seen from the Formulae (3.44), (3.45), and (3.46), the signals 'transmitted' to other nodes are always given by the deviations between the realisation x_i and the mean μ_i, which was replaced by the estimation $V(i)$. Hence the transmitted signal of inhibited observations $x_i - V(i)$ has zero mean and variance 0.01. The signals of activated nodes become strong, as they are sampled from a set of observations that are 'stronger' than 95 percent of the pure observational signals in average.

3.7.2. Netbuilder data generator

As a second synthetic data generator the software package *Netbuilder* (see [52, 53]) can be used. It can be assumed that the data sets generated with Netbuilder are much more biologically realistic than the data sets from the Gaussian network generator. Netbuilder is an interactive graphical tool for representing and simulating genetic regulatory networks in multicellular organisms. It models the co-regulation between interacting

genes with the *sigma-pi calculus* which corresponds to a steady-state-approximation to the system of coupled differential equations. In this approach genes are modelled as *sigma-pi units*, which were introduced by [40] as nodes in higher order neural networks to avoid linear separability constraints associated with first-order neural networks. Boolean functions and logic gates can be expressed in a sigma-pi formalism, and their input and output are not restricted to boolean values. Sigma-pi units are combinatorial, so simpler units connected can lead to a very complex module.

So far gene regulatory processes and systems have usually been modelled with a chemical kinetic approach based on enzyme-substrate intercation, that is a detailed mathematical description of the individual chemical reactions that form a biochemical pathway. See B. Appendix II for some more details. But as the number of parameters necessary to specify such systems is extremely large, it is useful to simplify these models while maintaining their main characteristics.

Ignoring time delays inherent in transcription and translation the system can be modelled with a set of coupled ordinary differential equations (ODEs). Assuming a steady state of this sytem, it is possible to derive a set of equations that describe the concentration of products as non-linear functions of combination of substrates. The resulting equations are a combination of multiplications and sums of sigmoidals. So instead of solving the steady-state approximation to ODEs explicitly, it is possible to model the system using the sigma-pi-formalism. This approach, which can be applied using the software package Netbuilder, simplifies the modelling task by avoiding the need for an explicit solution of the system of ODEe, but maintains the qualitative behaviour of the system of interacting components.

In addition to standard continuous *AND* and *OR* regulation mechanisms (ports) implemented in Netbuilder, which correspond to purly cooperative and inhibitory gene interactions, continuous *XOR* ports can be constructed. This allows to model mixed cooperative-inhibitory interactions, and increases the amount of non-linearity in the interaction patterns. The parameters corresponding to these interactions can be chosen at random. Besides the intrinsic difficulty in obtaining 'realistic' parameters, it can be

stressed that the objective is not to mimic some particular experimental signal, but rather to generate signals that are typical of a given topology. The difficulty in learning the networks from the generated data in this way can be increased by adding some (additive) observational noise, for which different signal-to-noise ratios (SNR) can be used. For further details on the package Netbuilder and the sigma-pi-formalism see [52, 53].

Netbuilder is a very flexible software tool which offers a lot of different options for generating semi-realistic gene expression data, so that due to space limitations only the functional relationships which were actually used for generating data for the comparative evaluation study (see Section 5), are presented here:

The realisations of domain nodes which have no parent variables are simply sampled from independent uniform distributions over the interval [0,1]. The realisations of domain nodes X having parent nodes P_i, whereby in analogy to the definition for Bayesian networks each node from which an edge points to X is a parent node of X, depend on the realisations p_i of these parent nodes P_i as well as on an additive noise variable ϵ having a Gaussian distribution with mean zero and variance σ^2. Thereby in most cases three different noise levels were distinguished: weak noise ($\sigma = 0.01$), medium noise ($\sigma = 0.1$), and strong noise ($\sigma = 0.3$). Using the following simple auxiliary function:

$$f(x) = \left\{ \begin{array}{ll} 0, & x < 0 \\ x, & x \in [0,1] \\ 1, & x > 1 \end{array} \right\}$$

there is the following functional relationship between the parent nodes P_i and X:

If there is only one parent node P with realisation p, by default Netbuilder sets the corresponding realisation x of X simply to: $x = f(\frac{p}{p+1} + \epsilon)$. For parent sets of higher cardinalities Netbuilder was in most cases configured, so that there are *OR*-regulation

ports. For two parents P_1 and P_2 being realised as p_1 and p_2, such an *OR*-port leads to the following functional relationship:

$$x = f(OR(p_1, p_2) + \epsilon) = f\left(\frac{p_1}{p_1+1} + \frac{p_2}{p_2+1} \times \left(1 - \frac{p_1}{p_1+1}\right) + \epsilon\right).$$

For more than two parents the realisation x of X is obtained by successive usage of such *OR*-ports. For example if X has three parents P_i with realisations p_i (i=1,2,3), then the realisation x can be computed as follows: $x = f(OR(OR(p_1, p_2), p_3) + \epsilon)$ what in turn leads to:

$$x = f(p_1 + p_2 + p_3 - p_1p_2 - p_1p_3 - p_2p_3 + p_1p_2p_3 + \epsilon).$$

Without giving the formula explicitly for three parent variables P_i with realisations p_i (i=1,...,4) the realisation x is given by $x = f(OR(OR(OR(p_1, p_2), p_3), p_4) + \epsilon)$, etc.

Moreover, it was decided to generate interventional data by setting the realisations x of an intervened node X (independently of all other domain nodes) either to $x = f(1 + \epsilon_I)$ if the intervention is an activation (up-regulation), or to $x = f(\epsilon_I)$ if the intervention is an inhibition (down-regulation). For intervened observations the noise variable ϵ_I was set to a $N(0, 0.01^2)$ Gaussian distribution. That is the variance of this noise variable ϵ_I is independent of the variances (standard deviations) choosen for the noise variables ϵ of the other domain variables (σ=0.01, σ=0.1, and σ=0.3).

Alternatively, *AND* ports can be used in Netbuilder. *AND* ports are defined as follows:

$$x = f(AND(p_1, p_2) + \epsilon) = f\left(\frac{p_1}{p_1+1} \times \frac{p_2}{p_2+1} + \epsilon\right).$$

And in analogy to the *OR* ports described above, for more than two parents the realisation x of X can then be obtained by successive usage of such *AND*-ports. For

example if X has three parents P_i with realisations p_i (i=1,2,3), then the realisation x using AND ports can be computed as follows: $x = f(AND(AND(p_1, p_2), p_3) + \epsilon)$ what leads to:

$$x = f(\frac{p_1}{p_1+1} \times \frac{p_2}{p_2+1} \times \frac{p_3}{p_3+1} + \epsilon).$$

Furthermore, for target nodes having exactly two parent nodes, AND and OR regulation ports can be combined to obtain so called XOR regulation ports. For a variable X having two parent nodes P_1 and P_2 with realisations p_1 and p_2 its realisation x given such an XOR port can be computed as follows:

$$x = f(XOR(p_1, p_2) + \epsilon) = f(AND\,(OR(1 - p_1, p_2), OR(p_1, 1 - p_2)) + \epsilon)$$

Table 3.5 gives an overview to which realisations the three different types of Netbuilder regulation ports lead. Thereby it is assumed that the target node has two parent nodes P_1 and P_2 with different realisations p_1 and p_2. It can be seen that especially the XOR regulation port yields a non-linear relationship between the nodes P_1 and P_2 and their target node.

P_1 and P_2	$OR(P_1, P_2)$	$AND(P_1, P_2)$	$XOR(P_1, P_2)$
$p_1 = 0$ and $p_2 = 0$	0	0	0.25
$p_1 = 0$ and $p_2 = 1$	0.5	0	0
$p_1 = 1$ and $p_2 = 0$	0.5	0	0
$p_1 = 1$ and $p_2 = 1$	0.75	1	0.25

Table 3.5.: Realisations of a child node given different values of its two parent nodes P_1 and P_2 for the three regulation ports implemented in the Netbuilder software. Normally, some noise ϵ is added to these realisations.

4. Modelling a gene regulatory network

In this chapter the utility of the Bayesian network methodology presented in Section 3.5 for modelling gene regulatory networks is demonstrated. Within the scope of a cooperation with the 'Bioinformatics - Genomic Group' of the company Boehringer Ingelheim Pharma GmbH Co. KG the gene expression measurements of the mRNA levels of 200 genes in healthy human kidney cells were made available for analysing them with a Bayesian network approach. Due to a foregoing data exploration these genes appeared to be the most relevant ones for the pathogenesis of the human kidney cell carcinoma.

4.1. Data description and background

Originally, the 'Bioinformatics - Genomic Group' of the company Boehringer Ingelheim Pharma GmbH Co. KG measured the expression levels of 22,283 genes in 60 healthy and 15 carcinoma-diseased human kidney cells. These human kidney cells were taken from the kidney tissues of 75 different human individuals, whereby 60 individuals had no kidney disease while 15 individuals suffered from a kidney-cell-carcinoma. The purpose of their data collection was to identify the most significantly differentially expressed genes in healthy and carcinoma-diseased cells as well as to identify the interacting genes.
Consulting the results of a precedent analysis by [28], the 200 genes which appeared to be the most significantly differentially expressed ones in healthy and carcinoma-diseased cells, were selected for a further analysis. The objective of interest of this continuative analysis was to identify the interactions between these 200 genes under healthy conditions. Consequently, an independent gene expression profile sample of size 60 taken from

healthy probands was available for the analysis. For each sample profile the measurements of the expression levels of those 200 genes, which due to the precedent analysis are supposed to play a key role in the pathogenesis of the kidney cell carcinoma, were available for modelling a Bayesian network. There were no missing values in the observational data set consisting of 200 rows and 60 columns, one row for each gene and one column for each measurement.

4.2. Data preparation

According to the orders of the company that had collected the data, it was decided to analyse the data set using the discrete Bayesian network model in combination with Structure-MCMC sampling scheme. Thus the data set had to be discretised first. For the discretisation the Information bottleneck algorithm was used. More precisely, in a first step standard quantile discretisation was used to obtain 20 different discrete expression levels per gene. (Each level of each gene containing exactly three observations apiece.) Subsequently, the quantile discretised data set was used to initialise the Information bottleneck algorithm. Through the application of this information preserving algorithm the data were finally discretised to have three different discrete levels for each gene: under-expressed (-1), normally expressed (0), and over-expressed (+1).

4.3. Implementation and parameter settings

For reducing the computational costs of a Structure-MCMC Bayesian network simulation on a network domain with 200 variables, the (maximal) fan-in was set equal to three. Furthermore it was decided to consider reversals of non-compelled edges as invalid, to speed up the convergence a little bit. The total prior precision of the discrete multinomial Bayesian network model was set to one what renders the distribution of the prior parameters uninformative, because this choice can be interpreted as a ratio of one pseudo-count to sixty real counts (observations). The graph prior was set to an uniform

distribution over DAGs. To assert convergence of the simulations, it was decided to perform three independent Strcuture-MCMC runs over the domain, each run with a burn-in length of 20 million simulations. Afterwards, the next 80 million DAGs of each MCMC run were sampled. Since no biological prior-knowledge was available, it was decided to initialise the first run with an empty DAG without any edges, while the other two MCMC runs were initialised with two randomly selected DAGs. Using the self-written Matlab software, some weeks of computation time was needed for performing these three MCMC simulations. For each independent MCMC run trace-plot diagnostics along it gave an indication for convergence. But plotting the relation-feature confidences of different runs against each other as well as computing correlation coefficients revealed that the MCMC runs had led into different regions of the posterior-probabilities, such as local maxima. The Pearson correlation coefficients for the confidences of the Markov-relation-features can be found in Table 4.1. Although these coefficients are insufficiently low, it could be seen from discrete frequency-tables, that at least a certain convergence was given. Especially about 150 Markov-relation-features obtained high confidences in all three MCMC runs. So, it was decided to use the means of the Markov- and Order-relation-feature confidences, that were estimated for the three independent runs, to extract sub-networks. Using the means of all three independent runs, it could be ensured that in the end only those relation-features obtained a high confidence which were attached importance in all three 'regions' of the posterior probability.

	1st run	2nd run	3rd run
1st run	1.000	0.623	0.618
2nd run	0.623	1.000	0.656
3rd run	0.618	0.656	1.000

Table 4.1.: Correlation coefficients between confidences of Markov-releation-features for the three independent MCMC runs

4.4. Results

Using the means of the confidences of the three independent Structure-MCMC runs the algorithm of [14] was used (see A. Appendix I) to extract sub-graphs. The algorithm was initialised by several different triplets of nodes being pairwise in Markov-feature-relation of confidence higher than $t_M = 0.75$. Subsequently, that is during the algorithm, confidences lower than $t_F = 0.5$ were set to zero, so that exclusively undirected edges corresponding to a Markov-relation-feature with a confidence higher than 0.5 could be included into the sub-graphs. Afterwards a direction was given to some of the undirected edges using the following heuristic rule. Each undirected edge 'X—Y' in an extracted sub-graph is supposed to point from node X to node Y if and only if the corresponding Order-relation-feature confidence $F_{\triangleright}(X, Y)$ exceeds 75 percent of the Markov-relation-feature confidence $F_M(X, Y) = F_M(Y, X)$. In the end, 14 sub-graphs could be extracted which can be found in [21], whereby in the technical report the pseudo-names $X_1, ..., X_{200}$ had to be used instead of the real gene names, because the data set was made available for a confidential analysis only. Within this doctoral thesis only one example using the real gene names can be given (see Figure 4.1). The edge between the genes 'VRK2' and 'KIAA0779' has a Markov-relation-feature confidence of 0.6731 only. All other edges between the genes correspond to Markov-relation-features with confidences higher than 0.75. A direction was given to those edge-connections where a high Order-relation-feature confidence was given as explained above.

4.5. Conclusions

Although it is possible to interpret such sub-graphs from a statistical point of view, there is no possibility to confirm the extracted hypotheses about the regulatory mechanisms statistically. This is due to the fact that there is a difference between statistical and biological explanations. A low statistical confidence for a feature does not mean that it does not exist, but rather that the data set does not support it. On the other hand,

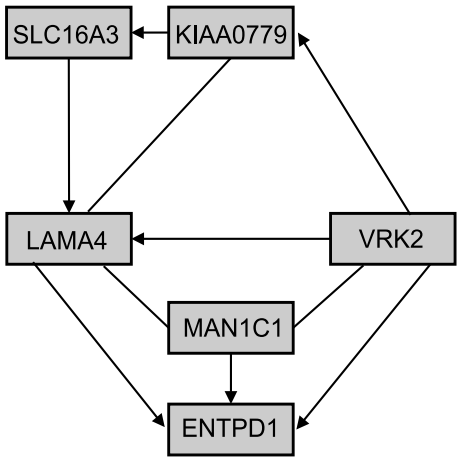


Figure 4.1.: Example of an extracted sub-network from the kidney cell expression data

a feature of high confidence might be a false positive one, that is a feature supported by the data by chance without having any biological reason. But nonetheless it is assumable - at least if the sub-graphs are biologically plausible, that is if there is no biological contradiction against the extracted mechanisms - that such findings provide some useful references for biologists, as they can focus their attention on them. The outputed regulatory mechanisms can be seen as extracted hypotheses which may be confirmed either by traditional molecular biology experiments or by the collection of further expression data for the involved genes in the sub-graphs.

4.6. Discussion

Due to time limitations of the project it was impossible to perform further Structure-MCMC runs on the kidney cell expression data. So, the project had to be finished, although the question, whether already a sufficient degree of convergence to the posterior probability was given, could not be answered adequately. Running Order-MCMC instead of Structure-MCMC simulations was impossible, as the latter one had been too time consuming for a domain with $n = 200$ nodes. Discrete multinomial Bayesian networks were prefered to continuous Gaussian Bayesian networks because the gene expression

measurements in the data set seemed to be very noisy. As it is known in Statistics that data discretisation yields a certain robustness against erroneous measurements, the discrete multinomial model was thought to be more useful. The application of other network models, such as Gaussian graphical models and Relevanve networks, was not wanted by the 'Bioinformatics - Genomic Group' of the company Boehringer Ingelheim Pharma GmbH Co. KG.

The stochastic details and the results of this gene expression data analysis were presented as a talk at the workshop 'Complex stochastic systems in Biology and Medicine' in Munich on 7.10.2004 and as a poster presentation at the 'Workshop on Statistics in Genomics and Proteomics' in Lisbon on 6.10.2005.

5. Comparative evaluation

5.1. Motivation of research

For inferring the architecture of biochemical pathways and regulatory networks from high-throughput postgenomic data various reverse engineering methods have been proposed in the literature. The most important machine learning methods among them have been described in detail in Chapter 3. But although it is important to understand their relative merits and shortcomings, no satisfactory cross-method comparisons between these different machine learning approaches can be found in the literature. Most of the evaluation studies that have been performed to assess the accuracy of reverse engineering, such as [54], [44] or [26], have investigated one particular inference method only.

In order to adress this shortcoming, an extensive evaluation study was carried out by [39]. The author compared Gaussian graphical models (GGMs) and Bayesian networks (BNs) on synthetic data generated from networks with random structures and different gene regulation mechanisms, where the latter differed with respect to the cooperative or competitive interactions between transcription factors regulating the same gene.

The comparative study presented in this Chapter is motivated by and based on the ideas of [39], but improves this earlier work in some important aspects, so that more understanding for this problem is established. In detail, thought has been given

to the following seven aspects to strengthen and upgrade the explanatory power of such a cross-method comparison:

1. **Realistic network architecture**

   Instead of considering random network structures, the structure of the well-investigated cytometric network is used as the true network, so that a topology which is biologically realistic and relevant is taken for granted. It has been decided to use the cytometric network, as for this signalling network real biological expression data are freely available and the real causal relationships are known from biological experiments (see [41]), so that a gold-standard network topology for the data is given.

2. **Methodical improvement (BNs)**

   The learning algorithm for Bayesian networks (BNs) has been improved. In order to capture uncertainty inherent in learning from sparse and noisy data, directed acyclic graphs (DAGs) have been sampled from the posterior distribution with Markov Chain Monte Carlo (MCMC) simulations. Such MCMC approaches are methodologically much more consistent than the optimization scheme applied in [39]. Especially, all four combinations of sampling scheme (Structure-MCMC and Order-MCMC) and Bayesian network model (discrete multinomial and continuous Gaussian) have been distinguished during the screening phase of the evaluation study.

3. **Methodical improvement (GGMs)**

   The inference for Gaussian graphical models (GGMs) has also been improved. The approach adopted by [39] is based on the PC algorithm of [45] only. In the present study, more recent algorithms for stabilizing the estimate of the inverse of the covariance matrix have been used, what is important due to intrinsic noise in postgenomic data. More precisely, it has been distinguished between three different bagging estimators as well as a novel shrinkage based estimator during screening.

4. **Inclusion of Relevance networks**

   A further reverse engineering method has been included in the study: the approach of Relevance networks proposed by [6], whereby it has always been distinguished between both Relevance networks based on pairwise mutual information scores and Relevance networks based on Pearson correlations during the screening phase.

5. **Inclusion of real biological data**

   Not only synthetic data sets, generated using more or less realistic data generators, have been included, but also real biological data sets have been used. Such real data sets could be used for the cross-method comparison, as real data for the cytometric network are freely available and the cytometric network archtitecture is sufficiently known from lots of molecular biological experiments.

6. **Inclusion of interventional data**

   The reverse engineering methods have not only been compared on observational but also on interventional data. Thereby especially real interventional data sets, which are usually rarely available, could be used for the cross-method comparison.

7. **A detailed comparative evaluation of the two scoring metric for Bayesian networks**

   In addition the two different scoring metrics BGe and BDe for Bayesian networks were cross-compared on different test data sets with different degrees of non-linearity.

5.2. The cytometric network

The *cytometric network* is a biologically well-known signalling network which describes the intracellular relationships between different molecules involved in *signal transduction*, that is the transmission of signals within living cells. So, the cytometric network describes a cascade of cellular protein-signalling. From a biological point of view,

special enzymes (*protein kinases*) modify other target proteins (*substrates*) by adding phosphate groups to them (*phosphorylation*) what usually leads to a functional change of the targets, so that further chemical reactions follow in the signalling cascade. As protein kinases are known to regulate the majority of cellular pathways as well as many aspects that control cell growth, disregulated kinase activity can lead to diseases, such as cancer.

Node	**Name**	**Phosphorylated protein or phospholipid**
X_1	RAF	Raf phosphorylated at position S259
X_2	ERK	*MAPKs* Erk1 and Erk2 (extracellular signal-regulated kinases) phosphorylated at T202 and Y204
X_3	P38	*MAPKs* p38 isoforms phosphorylated at T180 and Y182
X_4	JNK	Stress-activated protein kinases phosporylated at T183 and Y185
X_5	AKT	Protein kinase **B** (PKB) phosphorylated at S473
X_6	MEK	Mek1 and Mek2 phosphorylated at S217 and S221
X_7	PKA	Phosphorylation of protein kinase **A** substrates
X_8	PLC	Phosphorylation of phospholipase C-γ (PLC_γ) on Y783
X_9	PKC	Phosphorylation of protein kinase **C** on S660
X_{10}	PIP2	Phosphatidylinositol 4,5-bisphosphate (PIP_2)
X_{11}	PIP3	Phosphatidylinositol 3,4,5-triphosphate (PIP_3)

Table 5.1.: The meaning of the abbreviations in the **cytometric signalling network** shown in Figure 5.1. Mitogen-Activated-Protein-Kinase is abbreviated by *MAPKs*.

Measurements of the expression levels of $n = 11$ different phosporylated proteins and phospholipid components of the cytometric network were made in thousands of human immune system cells, and the conventionally accepted interactions between these 11 molecules in human immune system cells (as well as in almost all mammalian cells) are shown in Figure 5.1. Thereby some intermediating molecules were omitted from

the graphical presentation, as no measurements for them were available. Furthermore, some indirect interactions, that is interactions mediated through molecules not shown in the graph, are represented as if they were direct interactions. All these interactions in Figure 5.1 are biologically accepted signalling molecule interactions, reported in the biological literature. See [41] for a literature review. Consequently, the directed acyclic graph (DAG) which can be derived from the graphical representation, can be regarded as a gold-standard graph topology for the data. The names of the components of the cytometric network can be found in Table 5.1

In addition to about 1200 pure observational measurements, that is observations made under general experimental conditions, the $n = 11$ molecules in the signalling cascade were also measured after 9 different molecular interventions. To this end the $n = 11$ components in the cascade were also profiled 15 minutes after 9 different stimulations of the network. For each of these molecular interventions more than 600 measurements were made, whereby an effect on the molecules in the cascade could be observed for 6 of these perturbations only. As from these useful 6 interventions is known that they predominantely lead to an activation or inhibition of only one single molecule in the cascade, they can be considered as ideal interventions.

While the pure observational real cytometric measurements could be analysed without any further data preprocessing, it turned out that the interventional real measurements could not be used without preprocessing. Because not rarely, there was a clear discrepancy between expected and observed concentrations for intervened nodes, e.g. some inhibitions had not led to low concentrations while some activations had not led to high concentrations. The missing changes in concentrations are not surprising, as most of the experimental interventions affected the activity of its target instead of its concentration. Correspondingly, for some intervened nodes the measured concentrations do not reflect the strength of the true activity of the corresponding node, as for the pure observational measurements. Therefore, it was decided to replace in each real

interventional cytometric data set the values of the activated (inhibited) nodes by the maximal (minimal) concentration of that node measured under observational conditions (general perturbation of the system). Afterwards, quantile-normalisation was used to normalise each real interventional data set. That is for each of the 11 variables (proteins) its m realisations were replaced by quantiles of the standard normal distribution $N(0,1)$. More precisely, for each of the 11 variables (proteins) its j-th highest realisation was replaced by the $\left(\frac{j}{m}\right)$-quantile of the standard normal distribution, whereby the ranks of identical realisations were averaged. Identical realisation always occurred in the interventional real data sets, because as described above, in a first preprocessing step for each interventional realisation the value of the intervened node was replaced by the lowest (if inhibited) or highest (if activated) value of the values which were observed for the intervened node under general perturbation in that data set.

A brief summary of the effects of the six molecular interventions on the measured molecules activities can be found in Table 5.2. The three molecular interventions having no observable effect on the cascade, were completely discarded from the analysis. More details on the probe preparations, the exact experimental conditions as well as more information about the stimulatory agents can be found in [41].

Reagent	**Effect**
AKT-inhibitor	inhibits AKT
G06976	inhibits PKC
Psitectorigenin	inhibits PIP2
U0126	inhibits MEK
Phorbol Myristate acetate	activates PKC
8-bromo Adenosine 3‘,5‘-cyclic Monophosphate	activates PKA

Table 5.2.: Effects of the ideal interventions on the cytometric network.

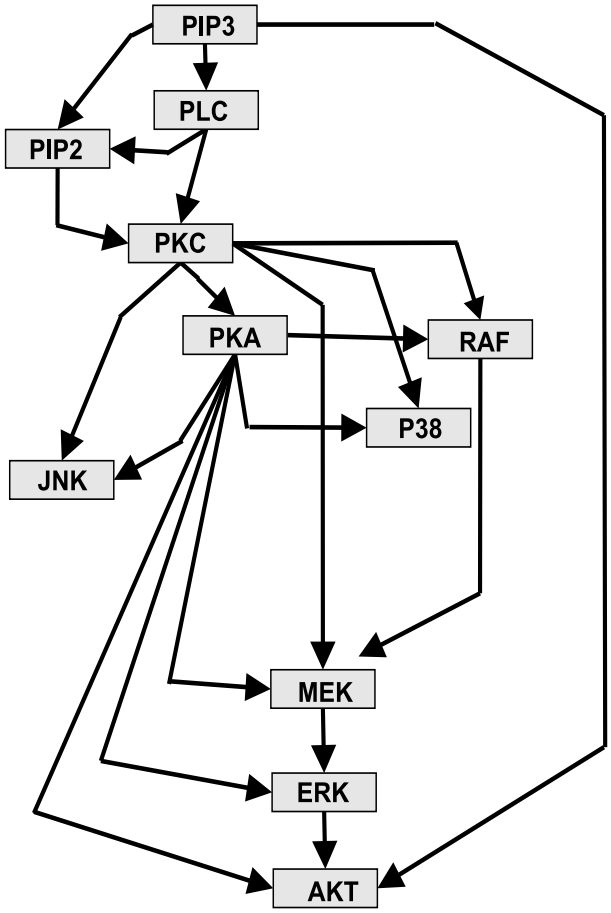


Figure 5.1.: Architecture of the cytometric signalling network
More information about the nodes can be found in Table 5.1. From a mathematical point of view, the cytometric network is a directed acyclic graph (DAG) with 11 nodes and 20 directed edges between its nodes.

5.3. A concrete example

This section gives some insight into the strategy of Bayesian network learning via Markov Chain Monte Carlo (MCMC) simulations. The corresponding statistical theory can be found in Section 3.5. Using a concrete data example, it is demonstrated in detail how convergence can be monitored, and how the final result can be evaluated when the true network topology is known. Furthermore, it is shown that the applied stochastic model (BDe or BGe) as well as the applied sampling scheme influence the result. To this end, a pure observational data set with $n = 1000$ observations and without any non-linear regulation was generated using the Bayesian network generator presented in Section 3.7.2, whereby the true graph was set to the cytometric network with $n = 11$ nodes. Afterwards the data set was analysed using all four combinations of stochastic model (BGe and BDe) and MCMC sampling scheme (Structure-MCMC and Order-MCMC). For each of these combinations some independent MCMC runs were accomplished.

5.3.1. Convergence monitoring

To obtain a first impression of the convergence and mixing of a MCMC simulation, trace-plot performance monitors can be used. Although such trace-plot diagnostics can give some useful indications for convergence, they are no sufficient citeria for it. In trace plots characteristical parameters of equidistant outputs, that is stated of the generated Markov Chain outputed at equidistant iteration steps, are plotted along the run. When a Markov Chain outputs directed acyclic graphs (DAGs), it is often useful to consider trace-plots of their scores and their total number of directed edges. Additionally, trace plots of the acceptance ratios are often considered. A trace plot of the acceptance ratios uses the ratio between the number of accepted and rejected proposed candidate DAGs of the run as characteristical parameter of the MCMC run itself. Thereby the ratios are computed from the frequencies of acceptances and rejections that occurred until the current iteration step. For instance, four different trace plots for the first Order-MCMC run on the data set, whereby the discrete multinomial model (BDe) was used, are shown

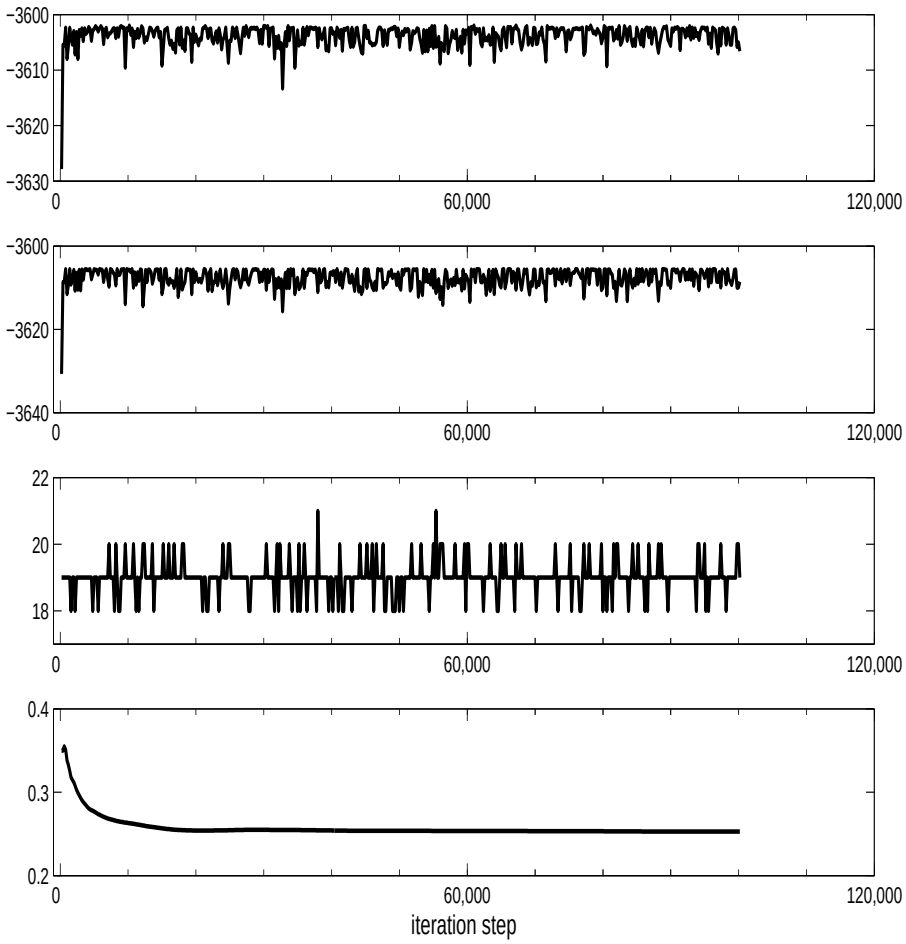


Figure 5.2.: Four trace plots for the first discrete multinomial Order-MCMC run, whereby each 200-th ordering was sampled, and altogether 100,000 iterations were accomplished. On the y-axis are plotted top down:

(1) the logarithmic likelihoods of the sampled node orderings,

(2) the logarithmic likelihoods of the DAGs sampled from these orderings,

(3) the number of edges of the sampled DAGs,

(4) and the acceptance ratios of the MCMC run.

in Figure 5.2.

The trace-plots in Figure 5.2 indicate, that convergence of the Order-MCMC run may be given. Already after a small number of iterations, the likelihoods of the sampled orderings as well as the likelihoods of the DAGs sampled from these orderings reach a 'plateau', that is all succeeding likelihoods are of comparable size. The number of edges of the DAGs sampled along the run fluctuates around 19 right from the start, and the acceptance-ratios seem to converge too. So, as there is no more change in all these monitored characteristical parameters long before the end of the run, it can be concluded that there is no trend being in contradiction to the convergence of the corresponding BDe-order run. But although this gives reason to believe that the generated Markov Chain has sampled from the true posterior probability over orderings, it might be that the run got stuck in a local maximum in the space of orderings, that is became trapped in a local region of orderings with high posterior probabilities. Possibly, Order-MCMC runs with other orderings as initialisations reach stationary behavior on different regions in the space of node orderings. Consequently, it is necessary to perform further Order-MCMC runs on the same data set using alternative initialisations. If and only if all these runs converge to the same region of the state-space, it can be concluded that convergence is actually reached. To assert, whether this is the case for different independent MCMC runs, it is useful to look at scatter plots of the confidences of pairwise relation-features (see Subsection 3.5.4). If the confidences are the same for all independent runs, these runs have outputed DAGs from the same region of the posterior distribution, and it is assumable that the runs have sampled from the true posterior probabilities. Otherwise, that is if the confidences differ systematically, there is a clear contradiction against the hypothesis that all runs have converged to the same stationary distribution, so that the MCMC results of all these runs should be discarded. Since from a theoretical point of view convergence will be reached after 'enough' iterations, the most likely reason for unsufficient convergence is that the run was stopped after too few MCMC iteration steps. Scatter plots for all four combinations of stochastic Bayesian network model and MCMC sampling scheme can be found in Figure 5.3. Each time the confidences of the *undirected*

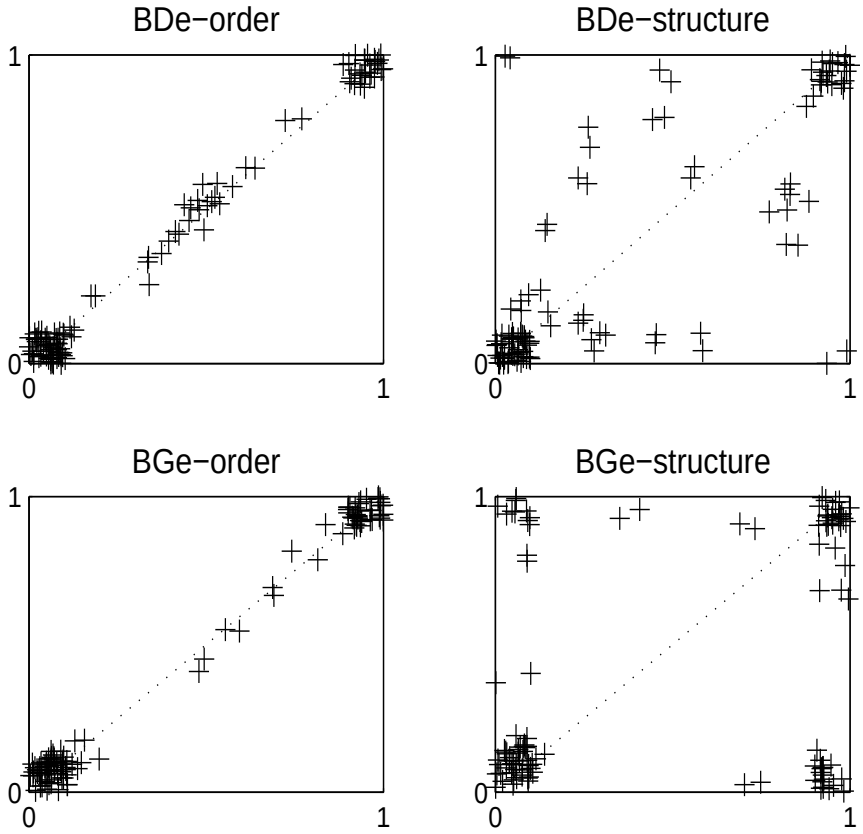


Figure 5.3.: Scatter plots for all four combinations of stochastic Bayesian network model and MCMC sampling scheme. Each time the confidences of the undirected edge relation-features of two independent MCMC runs were plotted against each other.
(1) top left: Order-MCMC using the discrete multinomial model (BDe)
(2) top right: Structure-MCMC using the discrete multinomial model (BDe)
(3) botton left: Order-MCMC using the continuous Gaussian model (BGe)
(4) bottom right: Structure-MCMC using the continuous Gaussian model (BGe)

For each Order-MCMC run 100,000 MCMC iterations were accomplished, whereby after a burn-in period of length 20,000 each 200-th DAG was sampled. For the Structure-MCMC 1,000,000 MCMC iterations were accomplished, whereby after a burn-in period of 200,000 each 2,000-th DAG was sampled.

edge relation-features estimated from the outputed DAGs of two independent MCMC runs were plotted against each other. Tt can be clearly seen from these plots that there is a strong level of convergence for the Order-MCMC runs, while the Structure-MCMC runs have not converged yet. This trend could be confirmed by looking at scatter plots of further independent MCMC runs. The Order-MCMC sampling scheme converges after a very small number of iterations, while even 1 million iterations are too few for convergence of the Structure-MCMC sampling scheme on this data set. The most likely explanation for the insufficient degree of convergence is that $n = 1000$ observations lead to a distribution of the true posterior distribution which has lots of 'peaks', that is local maxima. So, Structure-MCMC gets often trapped, and it takes much time until these peaks can be left by single edge operations. To produce a solid argument for this speculation, it is useful to have a look at the trace plots of the likelihoods for the corresponding MCMC runs. In Figure 5.4 the outputed likelihoods for both Structure-MCMC runs using the discrete multinomial Bayesian network model can be found, and it can be clearly seen that the likelihoods of these runs have become comparable not before the 610,000-th MCMC iteration. So, the Structure-MCMC runs (especially the second run), were stooped too early and had to be continued (see caption of Figure 5.4 for details).

When the confidences of the undirected edge relation-features obtained from the Order-MCMC runs using the continuous Gaussian (BGe) and the discrete multinomial (BDe) Bayesian network model are plotted against each other (see Figure 5.5), it can be seen that the different Bayesian network models BDe and BGe lead to different results for the same data set.

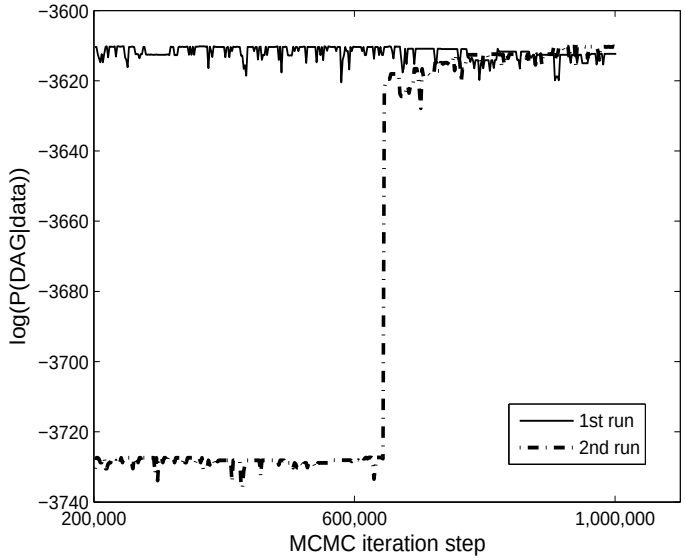


Figure 5.4.: Trace plots of the logarithmic likelihoods for both Structure-MCMC runs using the discrete multinomial Bayesian network model (BDe). For clarity, the iteration steps belonging to the burn-in period have been omitted. It clearly seems that the 2nd run, which was initialised by an empty DAG without any edges, got trapped somewhere till the 610,000-th iteration. Then suddenly, the run seems to leave this region, and the likelihoods become comparable to the likelihoods of the first run, which was initialised by a Greedy-Search optimized DAG. Consequently, it can be assumed that the second Structure-MCMC run has reached stationarity not before the 610,000-th iteration. So, at least the second MCMC run was stopped too early.

It was decided to continue both Structure-MCMC runs, and to extend the burn-in period to the last iteration accomplished so far. Afterwards, for both runs 400 new DAGs were sampled out of the next 800,000 iterations, and it could be seen from trace plots as well as scatter plots that the convergence level had clearly improved.

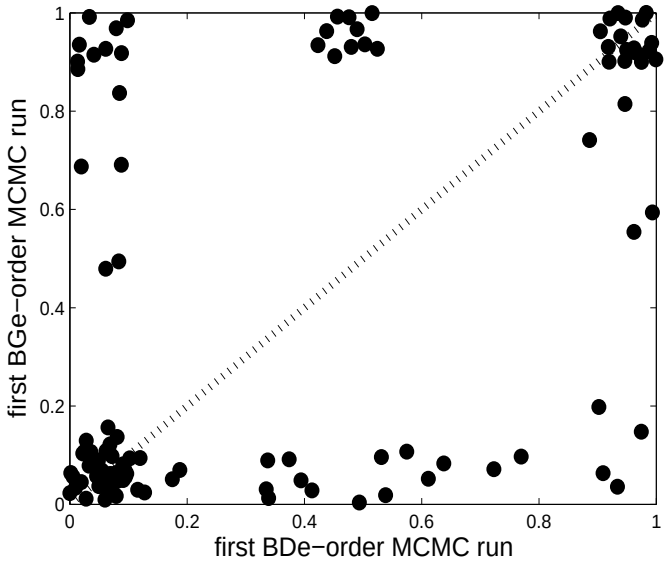


Figure 5.5.: Scatter plot of confidences of the undirected edge relation-features of the discrete multinomial Bayesian network model (BDe) *versus* the continuous Gaussian Bayesian network model (BGe). For both Bayesian network models the confidences were estimated from the outputs of the first Order-MCMC run.

5.3.2. Evaluation of performance

At the end of the last subsection it could be seen that both Bayesian network models lead to different results for the data set generated with the Gaussian Bayesian network generator. The goodness of performance can be evaluated using ROC curves and AUROC values, as the true network topology from which the data set was generated is known (see Section 3.6). For predicting the set of undirected edges of the true graph, that is its skeleton, the undirected edge relation-features can be used. The corresponding ROC curves for the first Order-MCMC runs can be found in Figure 5.6. From these ROC curves can be clearly seen that both Bayesian network models assert the highest confidences predominantly to the true edges, as there is an abrupt ascent in both ROC curves on the left-most of the plot. It even seems that the discrete BDe model performs a little bit better. But then for false discovery rates higher than 0.05 the continuous Gaussian model (BGe) becomes clearly superior to the multinomial model (BDe). For example, for a false discovery rate (FDR) of size 0.1, the discrete multinomial models (BDe) yields a sensitivity (TPR) of approximately 0.4 only, while the continuous Gaussian models (BGe) already reaches a sensitivity higher than 0.8. Some AUROC values for different thresholds can be found in Table 5.3. The entries of this table reveal the same trend already seen in the corresponding ROC curves. Only for very small false discovery rates the continuous Gaussian model is a little inferior to the discrete multinomial model. For higher false discovery rates the continuous Gaussian model becomes clearly superior.

This section was used to demonstrate how to assert convergence of MCMC runs, and how to evaluate the performance of different stochastic Bayesian network models in detail. In the following sections no more mention of these details is made. However, for all further MCMC runs excactly this strategy was used to assert, whether a sufficient degree of convergence was given, and to evalute the goodness of performance. If an insufficient degree of convergence was observed, the results were discared, and the corresponding runs were repeated with a longer burn-in period. Although this happend rarely, on those seldom occassions the strategy was to set the burn-in period of the new

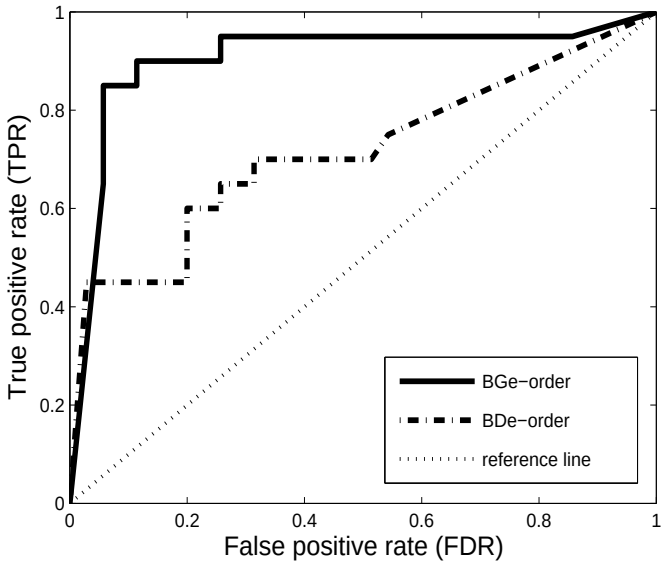


Figure 5.6.: ROC curves for the discrete multinomial (BDe) and the continuous Gaussian (BGe) Bayesian network models, computed from the outputed DAGs of the first Order-MCMC runs. For the prediction of the undirected edges of the true network, that is its skeleton, the estimated undirected edge relation-features were used.

runs equal to the total number of iterations of the discarded runs. Afterwards, exactly the same sampling scheme as before was used. Surprisingly, this simple strategy was effectual, that is always led to a sufficient degree of convergence of the new MCMC runs.

		AUROC_ϵ			
Model	Run	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 1.00$
BDe-order	1	0.00070	0.01469	0.03714	0.72143
	2	0.00079	0.01607	0.03857	0.71571
BGe-order	1	0.00057	0.01422	0.05500	0.90500
	2	0.00061	0.01531	0.05643	0.91000
reference	—	0.00005	0.00125	0.00500	0.5000

Table 5.3.: AUROC_ϵ values for different thresholds ϵ of the false discovery rate (FDR).
The true graph of the cytometric network consists of 11 nodes and 20 undirected edges. So, there are 20 true positive (TP) undirected edges and 35 true negative (TN) edges among all 55 possible undirected edges. Consequently, the different thresholds for the false discovery rate (FDR): 0.01, 0.05, 0.10, and 1.00 correspond to 0.35, 1.75, 3.5 and 35 false negative (FN) undirected edges.

5.4. Comparative evaluation study - Parameter settings

This section briefly describes the preparation of the data as well as some parameter settings for the comparative evaluation study.

All data discretisations were accomplished using the Information bottleneck algorithm of [22]. Thereby in a first discretisation step, a simple quantile discretisation procedure was used to obtain 50 (if N=1000 observations), 20 (if N=100 observations), or 5 (if N=10 observations) discrete levels as initialisation of the Information bottleneck algorithm. Afterwards these discrete levels were reduced to three ('-1', '0', and '+1') by applying the Information bottleneck algorithm. The three final levels can be easily interpreted as up-regulation ('+1'), down-regulation ('-1'), and no difference from the baseline ('0').
For the Bayesian network approaches self-implemented Matlab functions were used. Thereby two different scoring metrics had to be implemented: the BDe model and the BGe model. The BDe score is based on a multinomial distribution with a Dirichlet prior, whose hyperparameters have to satisfy certain constraints to ensure the likelihood-

equivalence of the score; see Subsection 3.5.2.1. These hyperparameters can be interpreted as pseudo-counts and their exact values depend on the size of a global hyperparameter α which was set equal to 1; this renders the prior distribution over the parameters very uninformative. The BGe score is based on a linear Gaussian distribution with a normal-Wishart prior. Again, the hyperparameters can be interpreted as pseudocounts from a prior network. To make the prior distribution over these parameters as uninformative as possible too, it was assumed that each domain node is stochastically independent and standard Gaussian distributed. That is, the prior network was set to the empty one (in which all nodes are unconnected). Furthermore the equivalent sample sizes were set to the smallest possible values subject to the constraint that the covariance matrix is non-singular.
As prior over graphs the distribution which is uniform over parent cardinalities subjet to a fan-in restriction of three was used.

To ensure convergence of the Markov Chain Monte Carlo (MCMC) runs for Bayesian network learning, each test data set was analysed using two independent MCMC runs. For Order-MCMC both independent runs were initialised by random node orderings. For Structure-MCMC the first run was initialied by a graph without any edges, and the second run was initialised by a graph found by greedy Hill Climbing. The burn-in lengths of all Structure-MCMC runs were set to 200 thousand. Afterwards 800 thousand Structure-MCMC simulations were performed, whereby each 2000-th graph was sampled leading to a graph sample of size 400 for each Structure-MCMC run. The burn-in lengths for Order-MCMC were set to 20 thousand, and afterwards 80 thousand MCMC-simulation were performed, whereby from each 200-th node-ordering a graph was sampled, also leading to a graph sample of size 400. Consequently, all Bayesian network MCMC learning results are based on 800 graphs sampled from two independent Structure-MCMC or Order-MCMC runs.
The computations for the Gaussian graphical models were carried out with the software provided by [42].

5.5. Comparative evaluation study - Screening

The first step of the comparative evaluation study can be seen as a kind of screening experiment which had been used to reduce the number of different machine learning reverse engineering methods whose performances subsequently were compared in more detail (see Section 5.6 and Section 5.7). During screening it was distinguished between all 10 different learning methods described in Chapter 3. A short summary of these methods is given in Table 5.4.

In this screening phase exclusively observational (non-interventional) test data sets were used and the undirected edges evaluation scheme (UGE) presented in Section 3.6 was used for evaluating and comparing the performances. Test data sets were generated by sampling from the real cytometric data and by generating linear observational data with the Bayesian network data Generator (see Section 3.7.1). From both sources 5 data sets of size $N = 100$ as well as 5 data sets of size $N = 10$ were generated, so that in the end 20 data sets were available for the screening experiment. While $N = 100$ represents an usual sample size for such expression data, $N = 10$ was used to include the case where the number of observations N is lower than the number of network nodes $n = 11$. Tables 5.5 and 5.6 give the empirical means and standard deviations of the AUROC_1 and $\text{AUROC}_{0.1}$ scores obtained for the test data sets, and what follows is an interpretation of these results, whereby p-values of two-sided one sample t-tests are used as descriptive measures for substantiating the differences (findings). Thereby it is important to mention that these p-values can *not* be interpreted in the sense of confirmative statistical tests, as no correction for multiple testing was applied. That is neither the overall error rate nor the false discovery rate was controlled. So, the t-test p-values were simply used to describe the pairwise differences with meaningful statistical characteristics.

Looking at the Bayesian network (BN) results only, it can be seen that there is not

much difference between the two sampling schemes Structure-MCMC and Order-MCMC. For the continuous BN model (BGe) as well as for the discrete BN model (BDe) with both sampling schemes in all cases approximately the same mean AUROC scores are obtained. The lowest p-value of size 0.1399 is given for the data sets from the Gaussian generator for $N = 10$ and the AUROC_1 score. This finding indicates that theBayesian network sampling scheme has no substantial influence on the output. Therefore and as Order-MCMC is computational less expensive than Structure-MCMC for the cytometric domain with $n = 11$ variables only, it was decided to restrict on the Order-MCMC sampling scheme for Bayesian network learning during the more detailed comparisons in the following sections. Furthermore it can be seen from the results of the Order-MCMC sampling scheme that the continuous BN-model (BGe) seems to outperform the discrete BN-model (BDe) on these test data sets. For the synthetic Gaussian data for all four combinations of sample size N and AUROC_ϵ criterium the corresponding t-test p-values lay between 0.0035 and 0.0141. This result is not surprising as there are exclusively linear relationships in the Gaussian data, so that the disadvantage that the discretisation incurs an information loss, can not be compensated by the modelling flexibility of the discrete BN model (BDe). For the real cytometric expression data, for which it is not known whether there is non-linear regulation or not, there is only one low p-value of size 0.0205 for the case $N = 10$ and the $\text{AUROC}_{0.1}$ criterion. As it strongly depends on the strength of non-linear regulation in the data, it was decided to compare the performance of BGe-order and BDe-order on data sets with different degrees of non-linear-regulation in more detail in Section 5.7.

Comparing the performance of Relevance networks based on the Pearson correlation (REL-PC) and the performance of Relevance networks based on the pairwise mutual information score (REL-MI) the same trend can be observed. That is REL-PC outperforms REL-MI on the Gaussian data having no non-linear regulation (p-values 0.0002, 0.0083, 0.0588, and 0.1220) as well as on the real data when the $\text{AUROC}_{0.1}$ criterion is used (p-values 0.0803 for $N = 100$ and 0.0152 for $N = 10$). Only for the AUROC_1 criterion REL-MI performs slightly better.

Method	Model	Training scheme
REL-MI	Mutual information Relevance network	Direct computation of pairwise association structures
REL-PC	Pearson correlation Relevance network	Direct computation of pairwise association structures
GGM-1	Gaussian graphical model	Observed partial correlation
GGM-2	Gaussian graphical model	Partial bagged correlation
GGM-3	Gaussian graphical model	Bagged partial correlation
GGM-4	Gaussian graphical model	Shrinkage based estimator
BN-BGe-struct	Bayesian network with BGe score	Structure MCMC
BN-BGe-struct	Bayesian network with BGe score	Order MCMC
BN-BDe-struct	Bayesian network with BDe score	Structure MCMC
BN-BDe-struct	Bayesian network with BDe score	Order MCMC

Table 5.4.: **Methods.** This table represents an overview of the machine learning methods compared during the screening phase of the evaluation study.

The results for the four different Gaussian graphical model (GGM) learning approaches are less systematic. But the shrinkage based estimator of the partial correlation matrix (GGM-4) performs clearly better than the other three GGM estimators on the Gaussian data with $N = 10$ when the AUROC_1 criterion is used (p-values 0.0081 (GGM-1), 0.0097 (GGM-2), and 0.0059 (GGM-3)) and in the end yields the highest mean AUROC score in 5 of eight cases. Only for the Gaussian data with $N = 100$ (GGM-2) and the cytometric data with $N = 10$ when the $\text{AUROC}_{0.1}$ citerion is used (GGM-3) another estimator leads to the highest average AUROC score. Thereby the lowest t-test p-value is higher than 0.06. Although these results do not show a clear superiority of the shrinkage based estimator (GGM-4), it was decided to take the results of a comparative study which had been carried out beforehand by the inventors of these learning methods for Gaussian graphical models (see [42],[43]) as well as profitable discussions and communications with these authors into consideration, which altogether point out that the shrinkage based estimator for GGM models is superior to the other three estimators. Beyond this justification it was decided to compare the different GGM estimators on further observational data sets generated with the

	$N = 100$				$N = 10$			
	AUROC_1		$\text{AUROC}_{0.1}$		AUROC_1		$\text{AUROC}_{0.1}$	
	mean	std-dev	mean	std-dev	mean	std-dev	mean	std-dev
BGe-order	0.8848	0.0543	0.0612	0.0157	0.7909	0.0488	0.0364	0.0092
BGe-struct	0.8631	0.0480	0.0573	0.0154	0.7712	0.0600	0.0342	0.0105
BDe-order	0.7060	0.0694	0.0262	0.0171	0.6274	0.0994	0.0139	0.0099
BDe-struct	0.7009	0.0630	0.0232	0.0120	0.6354	0.0791	0.0111	0.0079
REL-MI	0.6439	0.1159	0.0207	0.0122	0.6280	0.0737	0.0136	0.0090
REL-PC	0.6809	0.0816	0.0286	0.0123	0.7123	0.0646	0.0267	0.0114
GGM-1	0.9154	0.0364	0.0719	0.0090	0.5857	0.0616	0.0100	0.0076
GGM-2	0.9154	0.0374	0.0731	0.0105	0.6426	0.0905	0.0199	0.0115
GGM-3	0.9117	0.0347	0.0717	0.0095	0.5769	0.0826	0.0086	0.0045
GGM-4	0.8814	0.0373	0.0504	0.0153	0.7657	0.0627	0.0296	0.0095

Table 5.5.: Results for the observational Gaussian data (sample sizes $N = 100$ and $N = 10$)

Netbuilder generator (see Subsection 3.7.2). The results of this additional study affirm the truth of the latter statement to a certain degree and can be found in D. Appendix IV.

At the end of the screening phase it was decided to perform the following two more detailed comparisons. In a first step it seems to be useful to cross-compare all three model classes, that is the continuous Bayesian network model (BN-BGe), the Gaussian graphical model (GGM), and the Relevance network model based on Pearson correlation coefficients (REL-PC), in more detail. Thereby with respect to the results obtained during the screening phase, the following decisons were made: Firstly, it is obviously sufficient to consider one MCMC sampling scheme for the Bayesian network approach only. With regard to lower computational costs the Order-MCMC sampling scheme was choosen. Secondly, in the context of Gaussian graphical models it is effectual to apply the shrinkage based estimator for the partial correlation matrix only, because the latter one yield the best results during the screening phase, and was never much worse than the other estimators based on bagging. Thirdly, including the discrete Bayesian network model (BN-BDe) or the Relevance network based on pairwise mutual information scores

	$N = 100$				$N = 10$			
	AUROC_1		$\text{AUROC}_{0.1}$		AUROC_1		$\text{AUROC}_{0.1}$	
	mean	std-dev	mean	std-dev	mean	std-dev	mean	std-dev
BGe-order	0.6904	0.0376	0.0379	0.0108	0.5636	0.0373	0.0176	0.0062
BGe-struct	0.6780	0.0349	0.0374	0.0112	0.5664	0.0470	0.0176	0.0064
BDe-order	0.6620	0.0410	0.0348	0.0052	0.5452	0.0611	0.0074	0.0024
BDe-struct	0.6537	0.0555	0.0339	0.0057	0.5457	0.0685	0.0087	0.0033
REL-MI	0.6729	0.0561	0.0303	0.0070	0.5674	0.0604	0.0081	0.0036
REL-PC	0.6680	0.0546	0.0393	0.0094	0.5449	0.0769	0.0190	0.0042
GGM-1	0.6663	0.0705	0.0351	0.0092	0.5271	0.0845	0.0073	0.0039
GGM-2	0.6706	0.0713	0.0360	0.0102	0.5351	0.0988	0.0103	0.0051
GGM-3	0.6611	0.0725	0.0356	0.0103	0.6080	0.1040	0.0087	0.0075
GGM-4	0.6854	0.0542	0.0393	0.0093	0.5663	0.0506	0.0177	0.0059

Table 5.6.: Results for observational real cytometric data (sample sizes $N = 100$ and $N = 10$)

(REL-MI) does not make so much sense in such a cross-comparison, as these models can benefit exclusively if there are non-linear regulatory mechanisms in the data, that is mechanisms that can not be learnt by the former models. So supposably, it simply depends on the degree of non-linearity in the data, whether the latter models outperform the former ones or not. But to explore, whether this speculation is correct, it is useful to compare the performance of the two different stochastic models for Bayesian networks (BN-BGe and BN-BDe) in a second step on data sets with different degrees of non-linearity.

Method	Model	Training scheme
RN	Relevance network	Pearson Correlation.
GGM	Gaussian graphical model	Shrinkage based estimator
BN	Bayesian network	Order MCMC with BGe score

Table 5.7.: Overview of the three machine learning methods tested in this first more detailed cross-method comparison.

5.6. Detailed comparison between Bayesian networks, Gaussian graphical models, and Relevance networks

In this section the performances of the three model classes: Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN) are cross-compared and evaluated in more detail using lots of different test data sets. An overview of the applied training schemes can be found in Table 5.7.

Test data sets were sampled from the freely available real cytometric expression data, and synthetically generated with the Netbuilder software tool as well as the self-implemented Gaussian data generator. Details about these data generators and especially about the applied parameter settings can be found in Section 3.7. Thereby (unless otherwise noted) Netbuilder data were generated using OR ports, as OR regulation ports lead to almost linear relationships between the domain variables.

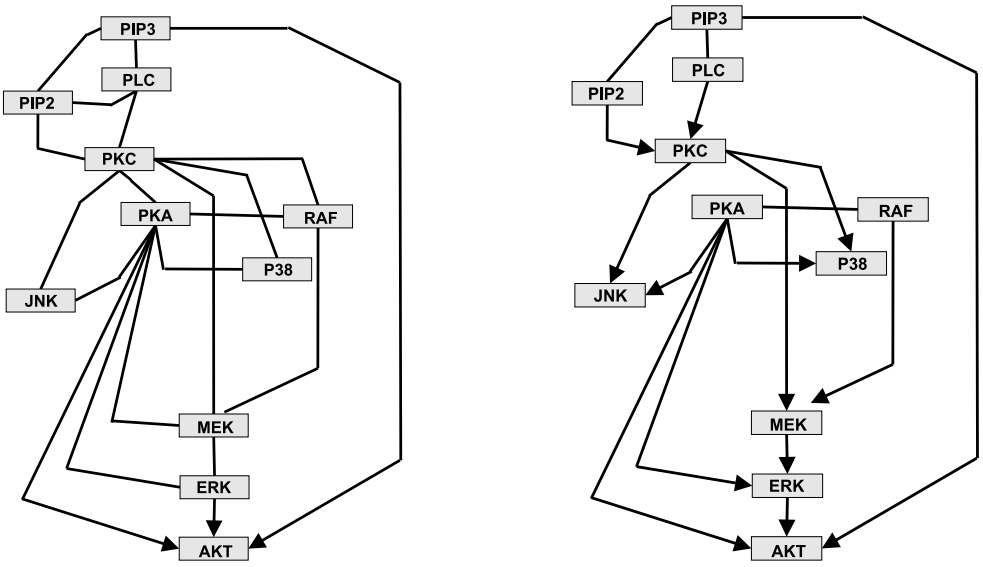
It is important to mention that it was decided to normalise each test data set, so that each domain variable (node) has empirical mean zero and empirical variance one, directly after generating it. Such a normalisation was included to avoid that there is a systematic difference in the (empirical) variances between variables having no parent nodes and variables having lots of parent nodes. During the screening experiments it could be seen that both data generators had led to such a systematical difference. For example, for the Gaussian data generator holds that variables with lots of parent nodes tend to have a higher variance than variables without parents.

At the beginning exclusively the gold-standard graph topology of the cytometric network topology was used as true graph for the synthetic data. But since this gold-standard

cytometric graph topology contains few v-structure only, four of its twenty directed edges were deleted at a later point of the analysis to obtain a graph with more v-structures. Including a graph topology with more v-structures is important for the cross-method comparison, as the CPDAG representation of such a directed acyclic graph with many v-structures contains more directed edges, so that the Bayesian network approach can benefit with regard to the directed graph evaluation scheme (DGE). For the remainder of this chapter the true gold standard cytometric graph topology is referred to as DAG_O and the modified graph topology is referred to as DAG_V. The CPDAG-representations of these two graphs are shown in Figure 5.7. It can be seen that the CPDAG of the original cytowmetric graph DAG_O contains seventeen undirected and only three directed edges, while the modified graph DAG_V contains three undirected and thirteen directed edges.

In a first step, for each of the three test data sources (available real expression data and the two data generators) five observational and five interventional data sets with $N = 100$ observations per set were generated. The interventional data sets were composed by $N_1 = 16$ pure observational data points and by $N_i = 14$ data points ($i = 1, ..., 6$) for each of the six different ideal interventions (see Table 5.2 in Section 5.2). The standard deviation of the noise variables in Netbuilder was set to the medium level ($\sigma = 0.1$) and exclusively OR regulation ports were used. The observational Gaussian data were generated as described in Subsection 3.7.2.

For all three methods under comparison the thirty test data sets were analysed, and the corresponding AUROC_1 values for both figures of merit (UGE and DGE) were computed. Figure 5.8 shows for each data source a coloured scatter plot of these AUROC_1 values. In these scatter plots the AUROC_1 scores of the Relevance network approach (RN) are plotted against the corresponding AUROC_1 scores of the other two methods BN (in red) and GGM (in blue). Each scatter plot contains the AUROC_1 values for all four combinations of data type (observational and interventional) and figure of merit (UGE and DGE). The combinations are characterised by different symbols (see caption of Figure 5.8).



(a) CPDAG(DAG_O) (b) CPDAG(DAG_V)

Figure 5.7.: CPDAG-representations of both directed acyclic graphs. The modified cytometric graph DAG_V was obtained by deleting the following four directed edges: (1) PKC→RAF, (2) PKA→MEK, (3) PLC→PIP2, (4) PKC→PKA of the original gold-standard cytometric graph DAG_O shown in Figure 5.1. Comparing the two panels, it can be seen that the four edge deletions have led to an immense increase in the number of directed edges in the CPDAG-representation.

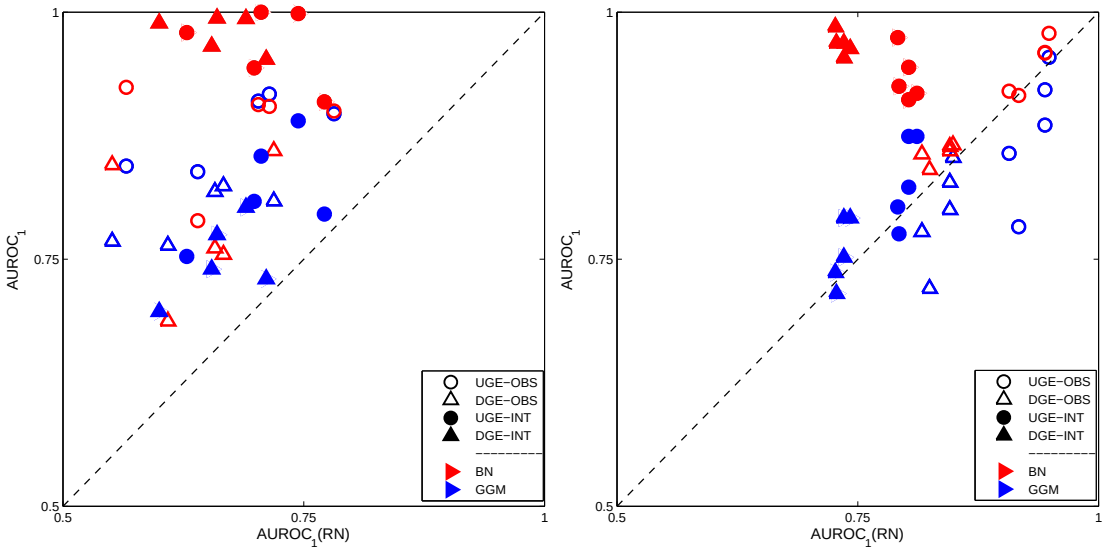
Scatter plot (a) refers to the Gaussian data, and it can be seen that Bayesian networks (BN) as well as Gaussian graphical models (GGM) outperform the Relevance network method (RN), as all points are located above the diagonal line. For the pure observational data sets BNs and GGMs achieve approximately the same overall performance in terms of the AUROC_1 scores. Only for the interventional data sets Bayesian networks substantially outperform GGMs.

Scatter plot (b) refers to the Netbuilder data, and it can be seen that all red symbols are located above the diagonal line as well as above the corresponding blue ones, so that BNs obviously outperform the other two methods. Furthermore, it seems that the RNs perform better than the GGMs for the pure observational data, as the empty symbols in blue are located below the diagonal line.

For the real cytometric expression data (c) there does not seem to be any difference between the method‘s performances for the observational data. All empty symbols are located around the diagonal line. Only for the interventional data BNs ouperform GGMs which in turn outperform RNs.

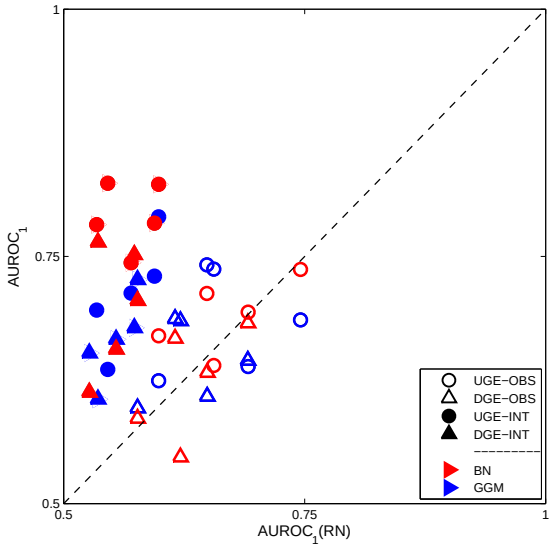
Tables containing one-sample t-test p-values which reflect these findings numerically can be found in E. Appendix V.

Panel (a) of Figure 5.8 raises the question, why there is not so much difference between the Gaussian graphical models (GGMs) and the Bayesian networks (BNs) for pure observational Gaussian distributed data sets. To address this issue it was decided to generate further five pure observational data sets with the Gaussian data generator for each of two different sample sizes N=10 and N=1000. Scatter plots of the AUROC_1 values for all three sample sizes can be found in Figure 5.9. From the scatter plots can be seen that the less sophisticated Relevance network approach (RN) becomes outperformed by the other two methods, since all points are located above the diagonal line. Thereby it seems that the superiority of the more sopisticated methods raises with the sample size. But anyway more interesting is the comparion between BNs and



(a) GAUSSIAN DATA

(b) NETBUILDER DATA ($\sigma = 0.1$)



(c) REAL DATA

Figure 5.8.: Scatter plots of AUROC_1 values: RN versus GGM (in blue) and RN versus BN (in red). Empty symbols represent observational data sets and filled symbols represent interventional data sets. The DGE figures of merit that take the edge directions into consideration are represented by triangles, while the UGE figures of merit that discard the edge directions are represented by circles.

GGMs.

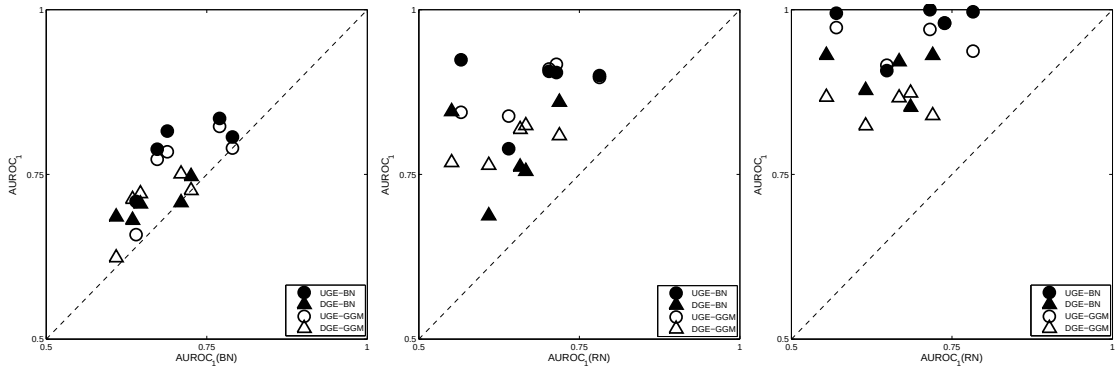
Comparing the locations of the filled and the empty symbols in panels (a)-(c) reveals that there is no big difference between the performance of Bayesian networks and Gaussian graphical models. It seems that Bayesian networks (BN) are slightly superior to GGMs for sample size N=10 when the UGE figure of merit is used, while for N=100 there does not seem to be a difference. And for sample size N=1000 Bayesian networks (BNs) outperform GGMs only in terms of the DGE figure of merit.

Tables containing one-sample t-test p-values which reflect these findings numerically can be found in F. Appendix VI.

To see whether Bayesian networks (BNs) show a better performance than Gaussian graphical models (GGMs) at least in the left, usually biologically more interesting, area of the ROC curves, where the amount of false positive (FP) extracted edges is low, exactly the same analysis was repeated using the $\text{AUROC}_{0.1}$ scores instead of AUROC_1 scores. The corresponding scatter plots as well as a table with t-test p-values can be found in G. Appendix VII. But the results of this analysis are consistent with the findings already reported above. Only for N=10 the inferiority of Relevance networks disappears when $\text{AUROC}_{0.1}$ scores are computed.

In the end it seems that Bayesian networks (BNs) even for pure observational Gaussian distributed data often perform at least slightly better than Gaussian graphical models, but it does not seem that these differences are significant.

Another reason for GGMs and BNs performing equally well for the pure observational Gaussian distributed test data, may be the fact that the cytometric graph topology has few v-structures only. Therefore for all six combinations of parameter settings five observational data sets were generated using the alternative graph topology DAG_V as true network. This analysis actually led to a very clear result. For sample sizes N=100 and N=1000 generated from the modified graph topology DAG_V having lots of v-structures, Bayesian networks (BNs) outperform Gaussian graphical models (GGMs) abundantly clear. Although due to space limitations not all scatter plots and tables can

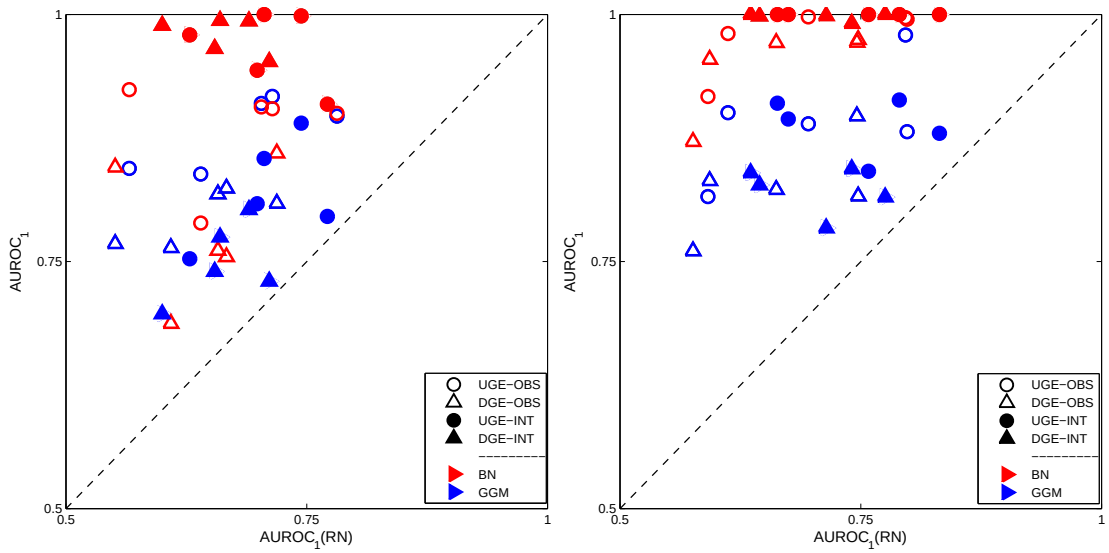


(a) N=10 (b) N=100 (c) N=1000

Figure 5.9.: Scatter plots of AUROC_1 values: RN versus GGM (empty symbols) and RN versus BN (filled symbols). Exclusively observational data sets were generated with the Gaussian data generator. Thereby three different sample sizes N were used. See text for further information. The DGE figures of merit that take the edge directions into consideration are represented by triangles, while the UGE figures of merit that discard the edge directions are represented by circles

be given in this thesis, the scatter plots for both graph topologies DAG_O and DAG_V using the original parameter setting (N=100 and $\sigma = 0.1$) are given in Figure 5.10. Once again from panel (a) can be seen that there is no difference in performance for the observational data sets using DAG_O. But when DAG_V is used instead, Gaussian graphical models are outperformed by Bayesian networks for both figures of merit and both data types (observational and interventional). As before, there are tables containing one-sample t-test p-values available which confirm these findings numerically (see H. Appendix VIII).

It can be summarised that Bayesian networks outperform Gaussian graphical models on Gaussian distributed data sets especially if there are either interventions or v-structures in the true network architecture. Otherwise there is no clear trend, and both methods seem to perform equally well. Except for the small sample sizes (N=10) and the DGE



(a) DAG$_O$ and N=100

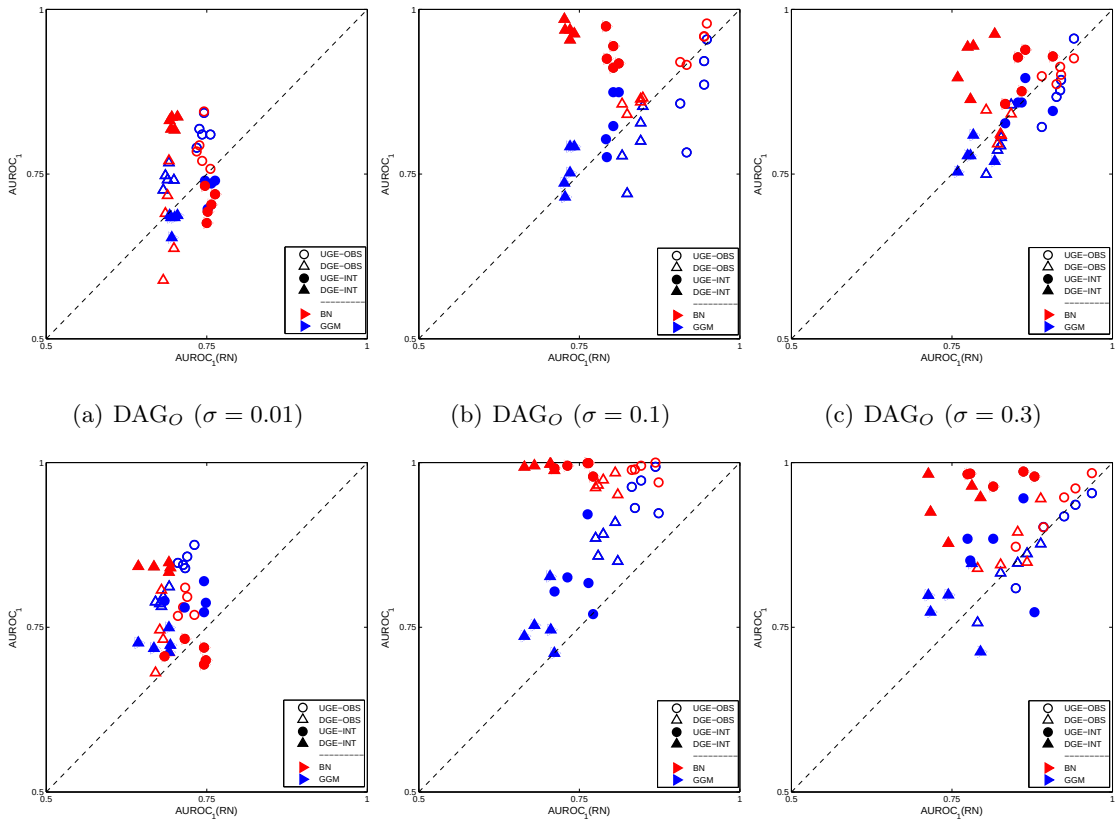
(b) DAG$_V$ and N=100

Figure 5.10.: Scatter plots of AUROC$_1$ values: RN versus GGM (in blue) and RN versus BN (in red). All data sets were generated with the Gaussian data generator. Thereby two different graph topologies DAG$_O$ and DAG$_V$ were used. As usual: Empty symbols represent observational data sets and filled symbols represent interventional data sets. The DGE figures of merit that take the edge directions into consideration are represented by triangles, while the UGE figures of merit that discard the edge directions are represented by circles

figure of merit the learning performance of Relevance networks on Gaussian distributed data is clearly inferior to the performance of the other two more sophisticated methods: Bayesian networks and Gaussian graphical models. The latter proposition is independent of the graph topology and does not depend on whether there are interventions either. But the problem associated with this finding is that Gaussian distributed data are not biologically realistic, so that the result is only of theoretical meaning. The Netbuilder tool is a software with which more biologically realistic expression data can be generated, so that it is useful to compare the performances on such semi-realistic data in more detail.

To this end for all six combinations of three different noise levels: weak ($\sigma = 0.01$), medium ($\sigma = 0.1$), and strong ($\sigma = 0.3$) noise, and the two different graph topologies: DAG_O and DAG_V 5 observational as well as 5 interventional data sets were generated using the Netbuilder software tool. Once again exclusively OR regulation ports were used. Overlaid coloured scatter plots of the outputed AUROC_1 scores: RN versus GGM (in blue) and RN versus BN (in red) are given in Figure 5.11.

From these scatter plots (panels (a) to (f) in Figure 5.11) can be seen that the more realistic (non-Gaussian) synthetic Netbuilder data have led to non-systematic AUROC_1 scores. As the scatter plot in panel (b) was already discussed above, it can be used as a starting point for further interpretations. Looking at panel (e), that is the same noise level σ but another network topology DAG_V, reveals that the inclusion of v-structures is clearly for the benefit of BNs and GGMs while the less sophisticated RNs which can not distinguish between direct and indirect interactions become inferior: all points are above the diagonal line, so for both types of data and both figures of merit RNs are outperfomed. Furthermore it can be seen that BNs are also superior to GGMs, as the red points lay above the corresponding blue ones. Only if the UGE figure of merit is used for observational data (red and blue empty circles), this superiority is not strongly pronounced. From the panels on the right ((c) and (f)) can be seen that a strong noise ($\sigma = 0.3$) leads to a deterioration of these trends. Exclusively for the interventional data the Bayesian networks (Bns) are clearly superior to both other methods (red filled symbols). The scatter plots on the left (panels (a) and (d)) reveal a strange relationship.



(a) DAG_O ($\sigma = 0.01$) (b) DAG_O ($\sigma = 0.1$) (c) DAG_O ($\sigma = 0.3$)

(d) DAG_V ($\sigma = 0.01$) (e) DAG_V ($\sigma = 0.1$) (f) DAG_V ($\sigma = 0.3$)

Figure 5.11.: Scatter plots of AUROC_1 values: RN versus GGM (in blue) and RN versus BN (in red). All data sets were generated with Netbuilder using OR regulation ports. Thereby three different noise levels σ as well as two different graph topologies DAG_O and DAG_V were used. See text for further information. As before: Empty symbols represent observational data sets and filled symbols represent interventional data sets. The DGE figures of merit that take the edge directions into consideration are represented by triangles, while the UGE figures of merit that discard the edge directions are represented by circles

On the one hand, if the DGE figure of merit is used Bayesian networks are superior to both other methods for the interventional data as usual. But on the other hand, if the UGE figure of merit is used for this low noise level $\sigma = 0.01$ Bayesian networks (BNs) perform worse for interventional than for observational data, and especially BNs are outperformed by Relevance networks for DAG_O (see panel (a)) and by Gaussian Graphical models for DAG_V (see panel (d)). Tables that contain one-sample t-test p-values for these Netbuilder data sets which confirm these findings numerically can be found in I. Appendix IX).

Especially, comparing the locations of the red symbols in panels (a) and (d), that is their y-coordiantes, with the corresponding locations in panel (b) and panel (e), it can be seen that Bayesian networks perform worse on Netbuilder with low dynamical noise ($\sigma = 0.01$) than on Netbuilder data with a higher dynamical noise ($\sigma = 0.1$). And although it is certainly less pronounced, in principle the same trend can be seen for the Gaussian Graphical Models (blue symbols) and Relevance networks. (For the Relevance networks one has to look at the x-coordinates of the symbols to see that.) These findings raise two questions. Firstly, why do all three methods under comparison perform worse for the low noise level ($\sigma = 0.01$) than for the medium ($\sigma = 0.1$) or even the strong ($\sigma = 0.3$) noise? And secondly, it is not clear why interventions are not for the benefit of Bayesian networks (BNs) in terms of the UGE figure of merit when there is only weak noise in the data.

An educated guess is that the non-linear functional OR(.)-relationships between connected nodes in Netbuilder automatically lead to some indirect associations between unconnected nodes which are sometimes stronger than the true associations. As an example a small network with $n = 3$ nodes only can be considered in which one node P has two child nodes C_1 and C_2, that is there are two edges $P \to C_i$ (i=1,2) in the true network. In this case there is no direct interaction between the two child nodes. But as C_1 and C_2 have a common parent node P, there is an indirect relationship between them. When generating Netbuilder data from this simple network *without* adding any noise, the realisations of both child nodes C_i are deterministic non-linear functions of

the parent node P. That is for each realisation $p \in [0,1]$ of P both children get the realisation $c_i = \frac{p}{1+p}$ (i=1,2). So, this also implies a deterministic and actually linear functional relationship between the two child nodes, because it holds $c_1 = c_2$. Consequently, as the methods under comparison model linear relationships in the data, an edge between the child nodes is stronger supported by the data than the two true edges. The deterministic relationship between the child nodes can be weakend by the addition of noise, which renders the values of each child node more similar to that of P than that of the other child. Taking this theoretical consideration into account, it is clear that the noise forms the basis for reverse engineering the true network topology from data. For example, adding independent Gaussian distributed noise variables $\epsilon_i \sim N(0, \sigma^2)$ as usually done in Netbuilder yields: $C_i = \frac{P}{1+P} + \epsilon_i$, so that the linear relationship between C_1 and C_2 becomes diluted. It holds: $C_1 = C_2 + \epsilon_\star$ with $\epsilon_\star = (\epsilon_2 - \epsilon_1)) \sim N(0, 2\sigma^2)$. On the other hand, between P and C_i there is the relationship: $C_i = \frac{P}{1+P} + \epsilon_i$. So, although the deterministic part of the relationship between the two child nodes is linear, the additional noise $\epsilon_\star$ has twice as much variance. As a consequence it can be concluded that for small noise levels σ the association between the two children C_1 and C_2 is stronger than the association between P and the child nodes C_i (i=1,2). On the other hand it is clear that too high noise levels σ not only destroy the indirect interaction between C_1 and C_2, but also hides the two regular relationship between $P \rightarrow C_1$ and $P \rightarrow C_2$, so that learning the true network topology is not possible either.

To verify this theoretical claim empirically, it was decided to generate and analyse some test data sets from simple network topologies using different noise levels. In a first step for each of 14 different noise levels $\sigma \in \{0, 0.01, 0.02, 0.03, 0.06, 0.1, 0.2, 0.3, 0.6, 1, 2, 3, 6, 10\}$ 25 Gaussian distributed data sets of size $n = 100$ were generated for two different graphs topologies $G(P_i, C)$ and $G(P, C_i)$ both consisting of 4 nodes only. In $G(P_i, C)$ three parent nodes P_i have a common child C with the dependence structure: $C = 1/3 \cdot (P_1 + P_2 + P_3) + N(0, \sigma^2)$, so that each parent node P_i has exactly the same influence on the common child C. Thereby the realisations for the three parent nodes have been sampled from independent standard Gaussian distributions $P_i \sim N(0, 1)$. For

$G(P, C_i)$ an 'opposite' dependence structure was choosen, that is $C_i = 1/3 \cdot P + N(0, \sigma^2)$, so that three child nodes C_i have a common parent P. Each child node obtains the same signal from node P. The realisations of node P were sampled from a standard Gaussian distribution $P \sim N(0, 1)$. With regard to the CPDAG representations in the context of Bayesian network methodology (see Subsection 3.5.1) all three edges of $G(P_i, C)$ are directed while the three edges in G_{P,C_i} are undirected.

Trace plots of the AUROC_1 means with error bars representing the AUROC_1 standard deviations for the 14 different noise levels σ are shown in Figure 5.12. From the UGE trace-plots for topology $G(P_i, C)$ (top, left) can be seen that that all three methods perform equally well, and that the mean AUROC_1 scores decrease for high noise levels only. As the three parent variables are stochatically independent, low noise levels σ do not cause any problems. That is even when there is no noise in the data ($\sigma = 0$) there is no indirect association between unconnected nodes. From the corresponding trace plot of the DGE figure of merit (bottom, left) can be seen that Bayesian networks are superior to the other two methods. This is due to the fact that all three edges in the CPDAG-representation of $G(P_i, C)$ are directed, that is the CPDAG is is identical to the graph itself, so that all three edge directions can be learnt by Bayesian networks. For the second graph topology $G(P, C_i)$ in which the strength of indirect associations between the child nodes C_i depends on the noise level σ, the curves show another progression. When there is no noise in the data ($\sigma = 0$) the true network can not be learnt by Relevance networks and Bayesian networks. Both methods reach an AUROC_1 score mean about 0.5 corresponding to the expected AUROC value of a random predictor. The reason for that is that there is a deterministic linear relationship between each pair of domain variables, so that all possible edges are equally supported by the data. However, Gaussian graphical models (GGMs) reach the maximum AUROC_1 score mean of size 1, indicating that the true network can be learnt although there are such linear relationships. As it is clear that the theoretical partial correlations are the same for all possible edges, this can be an effect of shrinkage only. Obviously the shrinkage approach enables GGMs to distinguish between the true and the non-present edges. Actually, the

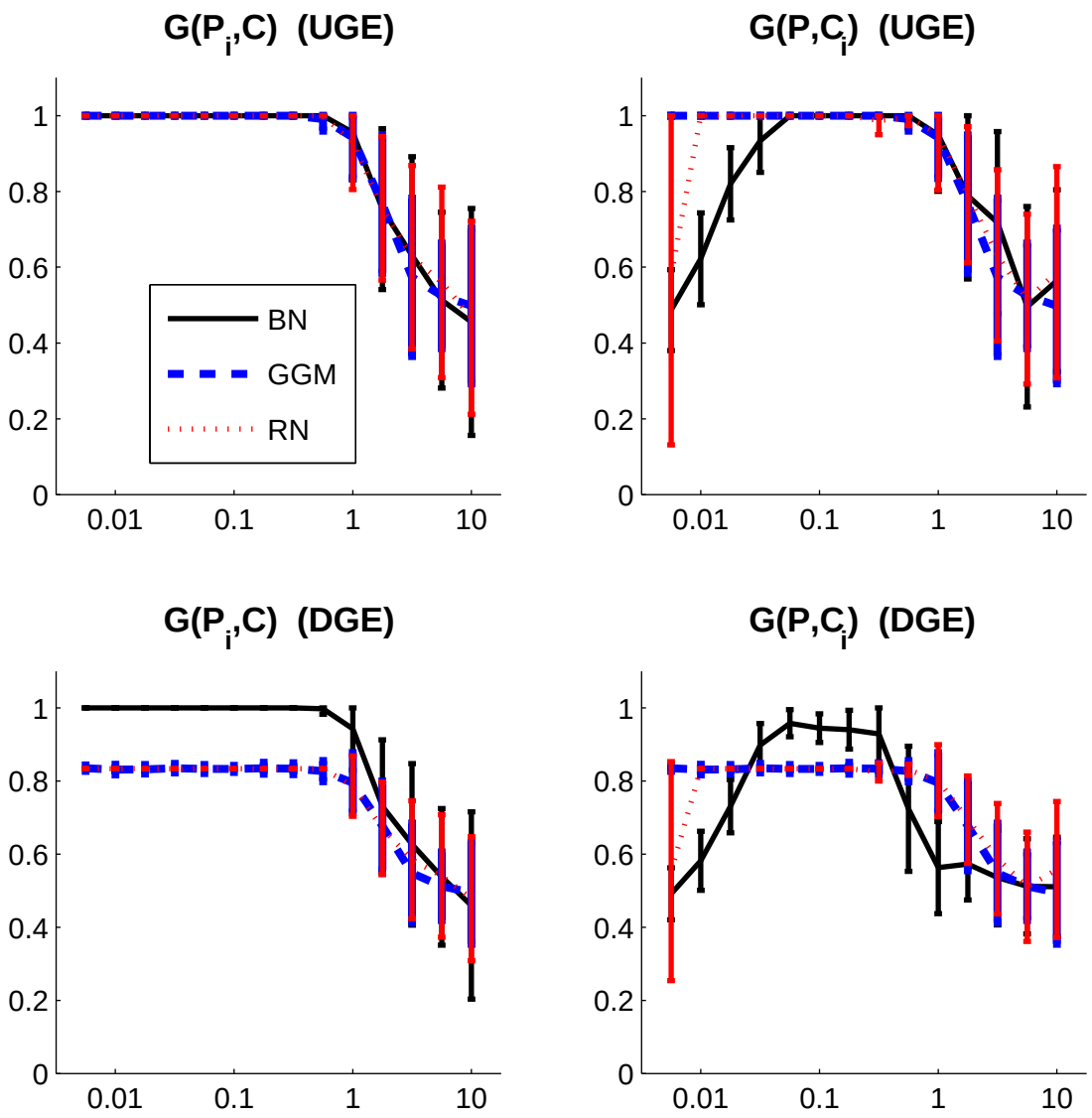


Figure 5.12.: Trace plots of the AUROC_1 means with error bars representing the AUROC_1 standard deviations for the three methods under comparison: Bayesian networks (black line), Gaussian graphical models (blue line), and Relevance networks (red line). For each of 14 different noise levels (σ) 25 Gaussian-distributed test data sets were generated using two simple graph topologies with 4 nodes each. In the first topology $G(P_i, C)$ (left panels) three parent nodes P_i have a common child node C with: $C = 1/3 \cdot (P_1 + P_2 + P_3) + N(0, \sigma^2)$, and in the second topology $G(P, C_i)$ (right panels) one parent node P has three child nodes C_i with: $C_i = 1/3 \cdot P + N(0, \sigma^2)$. For both networks and all three methods the UGE (top) as well as the DGE (bottom) figures of merit were computed. See text for further information.

outputed partial correlations are only slightly different, so that this must be interpreted as a more or less pure artificial effect. But for slightly higher noise levels ($0 < \sigma \leq 0.06$) it can be seen that Gaussian graphical models (GGMs) as well as Relevance networks (RNs) perform perfectly, while Bayesian networks (BNs) perform worse. Obviously BNs have the biggest problem with these low noise levels. Finally, for higher noise levels ($\sigma > 0.06$) all three methods perform equally well in terms of the UGE figure of merit, whereby again the mean AUROC_1 scores decrease with increasing noise levels σ. The corresponding DGE figure of merit trace-plots (bottom, right) for topology $G(P, C_i)$ reveal that the BNs are outperformed by the other two methods for small ($\sigma < 0.03$) and high ($\sigma > 0.3$), while on the other hand BNs are superior for the noise levels in between. Thereby it is surpring that BNs can learn the edge directions, although all three edges in $G(P, C_i)$ are undirected in the CPDAG-representation. As this behaviour is strange it was decided to consider the outputed DAG samples of the Order-MCMC runs in more detail. And actually it could be seen that the high DGE scores must be interpreted as artificial effects too. Because not rarely the CPDAGs of the sampled DAGs have the following constellation of edges: $C_k — P \rightarrow C_i \leftarrow C_j$, that is a v-structure with two directed edges from P and another child node C_j converging on C_i as well as an undirected edge between the parent node P and the third child node C_k. Replacing the three child nodes C_i, C_j, and C_k for all six possible permutations of them, the three correct directed edges $P \rightarrow C_i$, $P \rightarrow C_j$, $P \rightarrow C_k$ appear twice while all other directed edges appear only once. As a consequence, the three true directed edges get often slightly higher posterior probabilities than all other (directed) edges, what in turn leads to high AUROC scores. Neglecting the findings which are probably due to artificial effects only, it can be summarised that low noise levels are to the disadvantage of the Bayesian network (BN) approach. Obviously especially BNs are sensitive to indirected associations. This may be due to the fact that Bayesian networks - in contrast to Relevance networks and Gaussian graphical models - can consult more than one edge for explaining the realisations of a network node. That is, for a target node X which is directly associated with a node Y as well as indirectly associated with another node Z, both nodes Y and

Z can be used to explain the realisations of the target X. In the context of Relevance networks there is no adjustment for such indirect associations either, but at least it can be expected that the true association leads to a stronger correlation than the indirect one. Only for Gaussian graphical models, which are based on partial correlations, there is adjustment for all other domain variables, so that such indirect association become weaker than for the other two methods. From a theoretical point of view the same adjustment is given for Bayesian networks, as it is sufficient to select the nodes which are directly associated with the target node X as parent nodes of X. But practically is seems that the adjustement in the context of BNs is less effective than the adjustment reached by computing partial correlations, when the indirect association is only slightly weaker than the direct association, or when the indirectly associated node Y can be used in addition to the directly associated node Z to explain the target node X. Especially the second case could be observed for small noise levels σ and the network topology $G(P, C_i)$. Most of the outputed DAGs had the following type of edge constellation C_k— $P \rightarrow C_i \leftarrow C_j$, so that obviously an indirectly associated node C_j was used in addition to the directly associated node P to explain the realisation of the target node C_i.

To demonstrate that small noise levels σ can cause problems for all three methods under comparison, it was decided to continue with the little network analysis. More precisely, the deterministic linear functional relationships between the connected nodes in both graph topologies (see above) were transformed using the hyperbolic tangent function to obtain weaker linear associations between the child and parent nodes. From this analysis (see J. Appendix X) could be seen that all three methods can not learn the true (direct) relationships in graph topology $G(P, C_i)$ when such a non-linear transformation in combination with a low noise level σ is given.

These findings obtained from the little network diagnostic show that there are some problems associated with small noise levels. Especially for the synthetic Netbuilder data, low noise levels σ lead to strong indirect associations between unconnected nodes which have the same parent set. And not rarely these indirect associations are even stronger than the true direct associations represented by edges (i.e. direct causal relationships).

Therefore, these findings give an answer to the question, why all three methods under comparison performed worst for the Netbuilder data with low noise level in Figure 5.11. The bad performance is simply due to the fact that there are lots of strong indirect linear associations in Netbuilder data when a low noise level σ is used. Furthermore, the little network diagnostic reveals that especially Bayesian networks (BNs) tend to extract additional false edges from these indirect linear associations.
But it is still unclear why Bayesian networks (BNs) can not benefit from the interventions for such small noise levels when the UGE figure of merit is used.

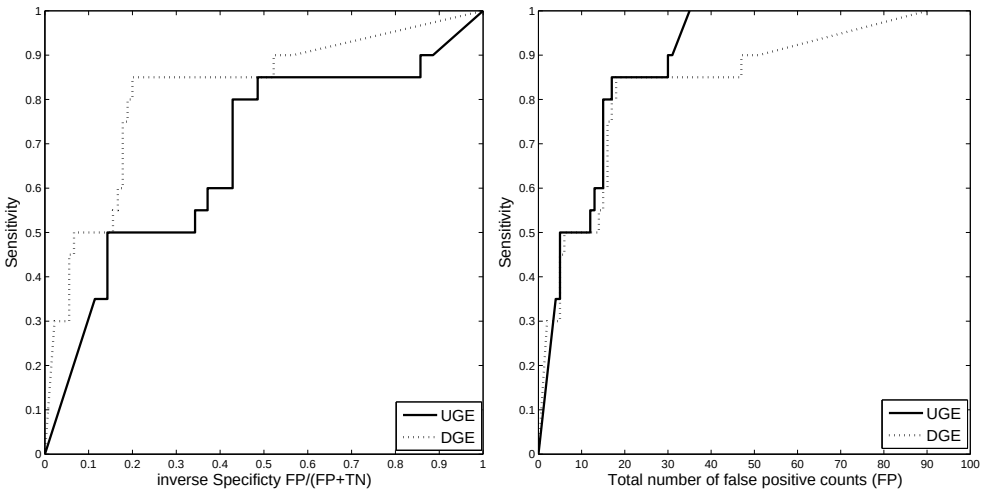
First of all, since learning the correct edge directions of the true network is much more difficult than learning its skeleton, that is the set of edge connections ignoring the edge directions, it is not surprising that Bayesian networks (BNs) reach higher UGE figure of merit AUROC_1 scores than DGE figure of merit AUROC_1 scores on pure observational data. But intuitionally it is not clear why quite the contrary happens for the interventional data. Theoretically, a possible explanation is that for the interventional data almost all extracted edges obtain a concrete edge direction whereby especially the true edges are extracted with their correct edge directions. In this case DGE and UGE reach approximately the same number of true positive (TP) counts for the same number of false positive (FP) counts. Because each true directed edge finding increments the number of TP counts for UGE as well as for DGE by 1, while each FP edge finding increments the number of false positives correspondingly. The same number of true positive (TP) counts yields the same sensitivity for UGE and DGE, as for both figures of merit there is the same number of true edges in the true network, either considered as directed or as undirected ones. But there is a difference in specificity, because the same number of false positive (FP) counts does not correspond to the same (inverse) specificity for UGE and DGE. Since the number of false directed and false undirected edges differs between UGE and DGE, the same number of false positive counts usually leads to a higher specificity for DGE than for UGE. For example, in the original cytometric network there are 20 true undirected and 35 false undirected edges. So, 10 true positive (TP) edge findings and 10 false positive (FP) edge findings yield a sensitivity of size 0.5

and a specificity of 0.71 in terms of UGE. In terms of DGE these counts (TP:10 and FP:10) yield the same sensitivity (0.5), but the corresponding DGE specificity is about 0.89, as there are 90 false directed edges. Consequently, whenever there is the trend that true edge connections are outputed with their correct edge directions (or otherwise not outputed at all), it follows that the DGE figure of merit reaches higher AUROC scores than UGE.

To find out whether this is the real reason for much higher DGE AUROC scores than UGE AUROC scores for the interventional Netbuilder data with low noise level, it is useful to additionally look at curves in which the sensitivities are plotted against the total numbers of true positive (TP) counts instead of only looking at the usual ROC curves in which the sensitivities are plotted against the inverse specificities. As an example the ROC curve as well as the alternative curve for the first interventional Netbuilder data set with low noise level (σ=0.01) are shown in Figure 5.13.

From panel (a) can be seen that the DGE and the UGE figure of merit ROC curves and so the AUROC_1 scores differ a lot. But from the alternative curve in panel (b) can be clearly seen that Bayesian networks reach for both figures of merit (UGE and DGE) approximately the same sensitivities (TP rates) for the same number of false positive (FP) counts. For all other interventional Netbuilder data sets the same progressions of these two curves could be observed too. So, the better DGE learning performance in terms of AUROC scores is indeed (mainly) due to the differences in the corresponding (inverse) specificities.

Another question which has not been answered yet, is why for the low noise level σ=0.01 Bayesian networks reach higher UGE AUROC score on pure observational Netbuilder data than on interventional Netbuilder data, while the opposite trend is given for the DGE figure of merit (see panels (a) and (d) in Figure 5.11). Although the reasons for a better DGE than UGE AUROC performance on interventional data could already be found, it is not apparent why interventions are for the disadvantage of Bayesian networks in terms of UGE figure of merit AUROC scores.



(a) inverse Specificity versus Sensitivity (b) total number of FPs versus Sensitivity

Figure 5.13.: ROC curve and curve of sensitivity against total number of false positive (FP) edges for the first interventional Netbuilder data set from DAG_O using the low noise level σ=0.01. The solid line corresponds to the UGE figure of merit and the dotted line corresponds to the DGE figure of merit. The DGE learning performance is better than the UGE learning performance in terms of the usual ROC curves (panel (a)), although the alternative curve reveals that the same sensitivities can be reached for the same numbers of false positive edges.

As it may be that this trend occured by chance only, it is useful to test in a first step whether this trend can be observed for further Netbuilder data sets too. To this end for each of six different noise levels $\sigma \in \{0, 0.02, 0.04, ..., 0.1\}$ further five observational as well as five interventional Netbuilder data sets with OR ports from the modified cytometric network topology were generated and analysed with Bayesian network Order-MCMC approaches. Scatter plots of the mean AUROC_1 scores can be found in Figure 5.14. The curves in the trace plot show the same trend already observed before. For small noise levels σ the Bayesian network approach yields better results in terms of the UGE figure of merit for pure observational than for interventional data. Only when a higher noise levels is given ($\sigma > 0.6$) the performance on the interventional data set becomes superior. On the other hand, in terms of the DGE figure of merit, interventions are always for the benefit of Bayesian networks.

To find possible explanations for that it is useful to compare the mean posterior probabilities of the 55 undirected edges, e.g. visually by scatter plots. In Figure 5.15 such a scatter plot is shown for the originally generated data sets from the modified cytometric network topology with low noise level. The mean posterior probabilities of the 55 possible undirected edges over the five observational data sets are plotted against the corresponding means over the five interventional data sets. It can be seen that some posterior probabilities differ a lot. Especially, when looking at the edges whose posterior probabilties are about 1 for the observational data, it can be seen that one true positive edge obtains a lower posterior probability (about 0.8) for the interventional data sets. This undirected edge represents the connection $PIP3 \rightarrow PIP2$. On the other hand, there are two false positive undirected edges whose posterior probabilities are about 1 for the interventional data and much lower for the observational data. These two undirected edges represent connections: $PIP2$—$PLCg$ and AKT—$PLCg$ not given in the true network topology. As both network variables $PIP2$ and AKT were inhibited, it seems that this curious finding is a consequence of inhibitions in a domain with weak noises. But diagnostics on smaller networks using low noise levels in combination with inhibitions did not lead to comparable trends, so that a concrete explanation could not

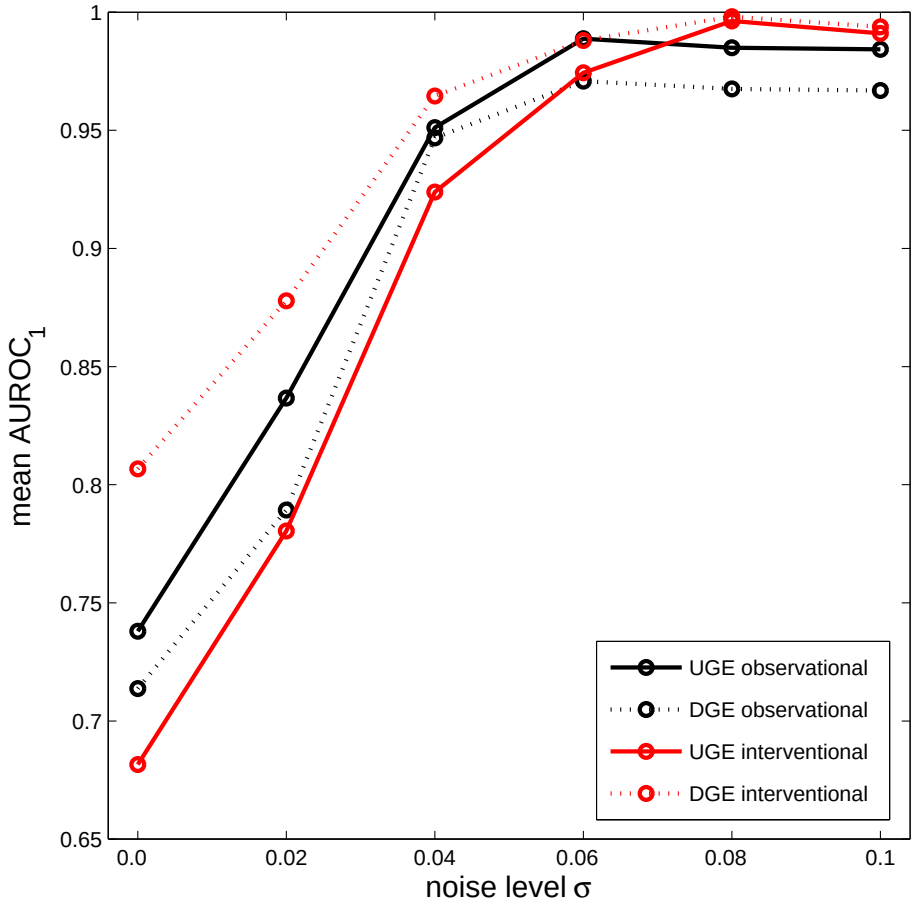


Figure 5.14.: Trace plots of the $AUROC_1$ means obtained for interventional and observational Netbuilder data with OR ports and different noise levels. For each of six different noise levels σ five observational as well as five interventional data sets were generated from the modified cytometric network topology. Afterwards the $AUROC_1$ means for both figures of merit (UGE and DGE) were computed and plotted. The trace plots for the UGE figure of merit are represented by solid lines, while the DGE trace plots are represented by dotted lines. The colours indicate the data set type: observational (black) and interventional (red).

be found.

Another theoretical explanation is that the parameter priors for Bayesian network models with the Gaussian BGe scoring metric for interventional data can not be specified as adequately as for pure observational data. For example, for pure observational it does not depend on whether a variable is seen as a child or parent node: in both cases especially its empirical parameters (mean, variance) are the same. But for interventional data the local score of each node has to be computed from the observations where this node was not intervened. That is, when an intervened node is scored some realisations of the network have to be excluded, so that the empirical mean vector and the empirical covariance matrix differ with respect to the node whose local score is computed. For each network variable this means that its empirical mean as well as its empirical variance differs with respect to the node whose local score is computed.

For this comparative evaluation study all test data sets were normalised and the parameter priors for the BGe scoring metric were selected, so that they can be interpreted as a Bayesian prior network in which all domain variables are independently standard Gaussian distributed. For normalised observational data sets this means that for each network variable the empirical mean (variance) and the prior mean (variance) are equal. So, there is a little information in the prior. A critical discussion about this combination of parameter prior and data normalisation can be found in Subsection 3.5.2.2. From this point of view, in the context of normalised interventional data the mean and variance prior parameters contain more information, when all network realisations are used for computing a local score, that is when the local score of a non-intervened node is computed. In this case the empirical means and variances are equal to the prior parameters. But on the other hand, when some network realisations have to be excluded, that is when the local score of an intervened node is computed, the empirical characteristics of all network variables change slightly, so that there is a little discrepancy between the two prior parameters and their corresponding empirical characteristics.
Usually, when setting the equivalent sample sizes to their minimums (see Subsec-

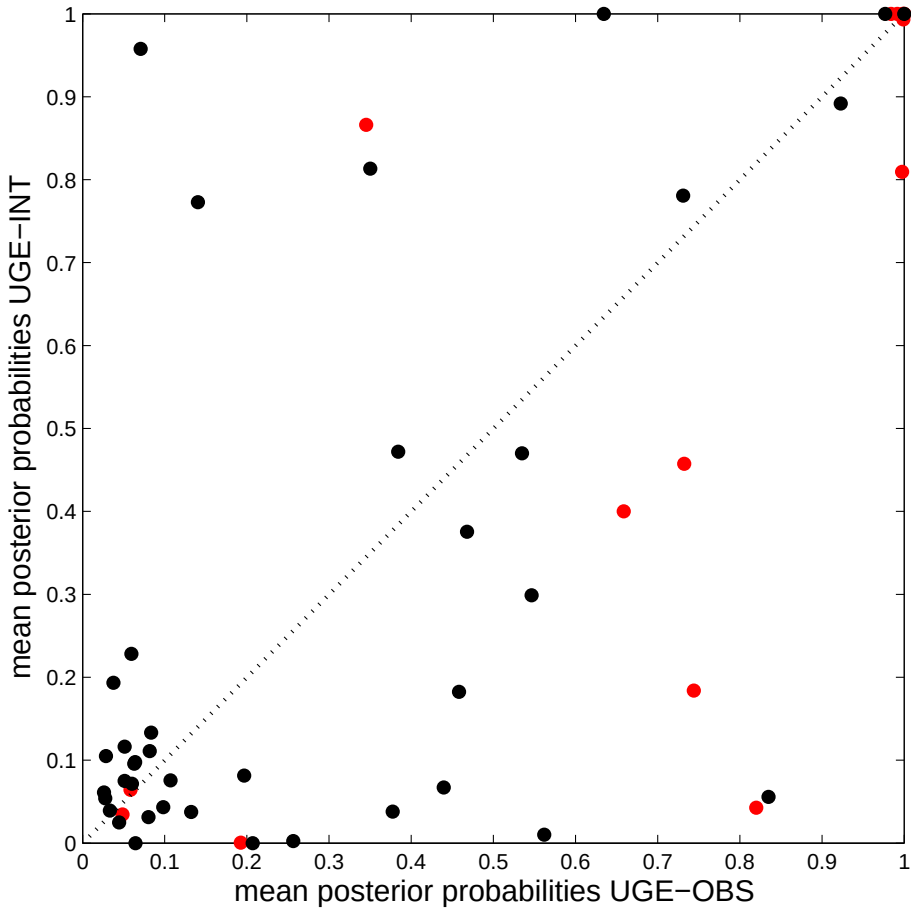
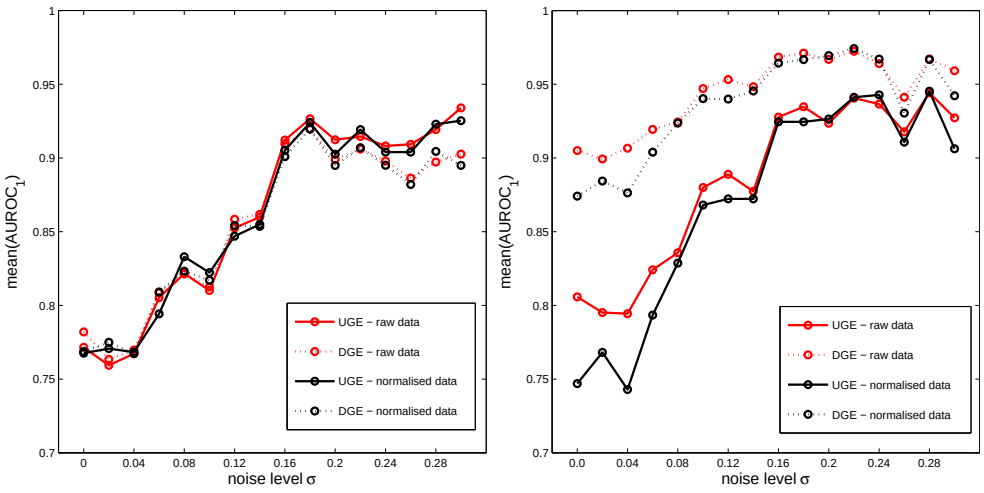


Figure 5.15.: Sactter plot of the mean posterior probabilities (undirected edges - UGE) for the Netbuilder data with low noise level from the modified cytometric network topology. The posterior probability means of all 55 possible undirected edges over the five observational data sets (x-axis) are plotted against the corresponsing 55 means over the five interventional data sets (y-axis). Points corresponding to one of the 16 true edges are plotted in red colour.

tion 3.5.2.2) the prior has only little effect on the local scores. But for a domain with lots of indirect associations, being approximately equally strong as the true direct associations, e.g. Netbuilder data with low noise levels, probably already such a litte effect can have some appreciable influence. Consequently, it may be that the better Bayesian network performance on pure observational Netbuilder data with low noise levels in terms of the UGE figure of merit is due to the more adequat prior parameter settings. To check this, it was decided to analyse further observational and interventional Netbuilder data sets. To this end five observational as well as five interventional data sets were generated from the modified cytometric network topology for each of sixteen different noise levels between 0 and 0.3. These 160 new Netbuilder test data sets were then analysed without normalising them as well as after having normalised them as usual. Thereby in both cases exactly the same prior Bayesian network, that is a network whose nodes are independently standard Gaussian distributed, was used. Figure 5.16 shows the results of this extensive analysis in terms of mean AUROC_1 trace plots. It can be clearly seen from panel (a) that the data normalisation has not effect on the learning performance for pure observational data. That is the true network topology can be learnt equally well with the continuous Gaussian scoring metric BGe. So, it does not depend on whether the raw data or the normalised data are used, although the assumption of a prior network of independently standard Gaussian distributed domain variables is more informative for the normalised data than for the raw data. Furthermore it can be seen that both figures of merit output approximately the same AUROC_1 scores. This is not surprising, because the modified cytometric graph topology DAG_V was used, in whose CPDAG most of the edges are directed, so that their directions can be learnt by Bayesian networks. For the interventional Netbuilder data (see panel (b)) the DGE figure of merit AUROCs are higher than the UGE figure of merit, and especially it can be seen that analysing the raw interventional data leads to higher AUROC_1 means for both figures of merit when the noise level is low ($\sigma < 0.06$). Only for higher noise levels σ the data normalisation has not effect on the learning performance. So, it can be concluded that the data normalisation weakens the learning performance for interventional data when

the noise level is low, what in turn means that the prior network can have a strong effect on the learning performance when the noise level is low. From a theoretical point of view the prior network of independently standard Gaussian distributed variables is more informative for the normalised data than for the raw data. But practically it seems that the small discrepancy between the prior information for scoring non-intervend and intervened nodes when analysing the normalised data strengthens the indirect associations which are given for Netbuilder data with small noise levels. And since the indirect associations are approximately equally strong as the direct associations (edges) this yields misleading results. On the other hand, when analysing the raw data, so that there is nearly no information in the prior network, the effect of the prior network seems to be negligible. Especially when comparing the corresponding curves in panel (a) and panel (b), it can be seen that (in contrast to the trend observed for normalised Netbuilder data) the performance in terms of the UGE figure of merit is not better for the pure observational data than for the interventional data when analysing the raw data. So, it can be concluded that the bad learning performance of Bayesian networks on interventional Netbuilder data with low noise level σ is most likely due to an interplay between the little influence of the prior network and the indirect associations being approximately equally strong as the true associations in the Netbuilder data with low noise level.
On the one hand, this finding points out that the prior network of the BGe scoring metric can have a strong effect on the learning performance when there are lots of indirect associations between the variables, so that it must be always set with extreme caution. On the other hand, as such weak noises will be rarely given for real expression data, the bad learning performance can be considered as an artificial phenomenon in the context of this cross-comparative evaluation study.

Bearing in mind all those findings, it seems to be useful to have a second look at the performances of all three learning methods on all different kinds of data sets generated from the original (DAG_O) and the modified (DAG_V) cytometric graph topology. But instead of looking at the AUROC_1 scores, which summarise the learning performance over all reachable combinations of sensitivity and (inverse) specificity by integrating ROC



(a) observational Netbuilder data (b) interventional Netbuilder data

Figure 5.16.: Trace plots of mean AUROC$_1$ scores. For sixteen different noise levels σ five observational as well as five interventional data sets were generated with the Netbuilder tool using OR regulation ports. All 160 data sets were then analysed without normalising them (raw data), and after having normalised them as usual (normalised data). The left panel (a) shows AUROC trace plots for the observational data, and in panel (b) the mean AUROC scores for the interventional data are plotted. The solid lines corresponds to the UGE figure of merit AUROC trace plots, and the dotted lines represent the DGE figure of merit mean AUROCs. The raw data analysis AUROC trace plots are represented in red, and the normlised data analyis AUROC trace plots are represented in black.

	Graph topology	
	DAG$_O$	**DAG$_V$**
UGE	0.86	0.87
DGE	0.94	0.90

Table 5.8.: Specificities corresponding to five false positive edges

curves, alternatively the number of true positive (TP) extracted edges, obtained when accepting 5 false positive (FP) edges, can be compared. This criterion is based on the selection of a threshold on the association scores of all edges, so that a specific network prediction is obtained. Each edge exceeding the threshold is outputed as a predicted edge of the network, and can be either a true edge of the real network (true positive (TP) edge finding) or a non-present edge of the real network (false positive (FP) edge finding). Correspondingly, if a certain edge is not extracted, this can be either a true negative (TN) edge finding, if this edge is not present in the true network, or a false negative (FN) edge finding, if the edge is present in the true network.

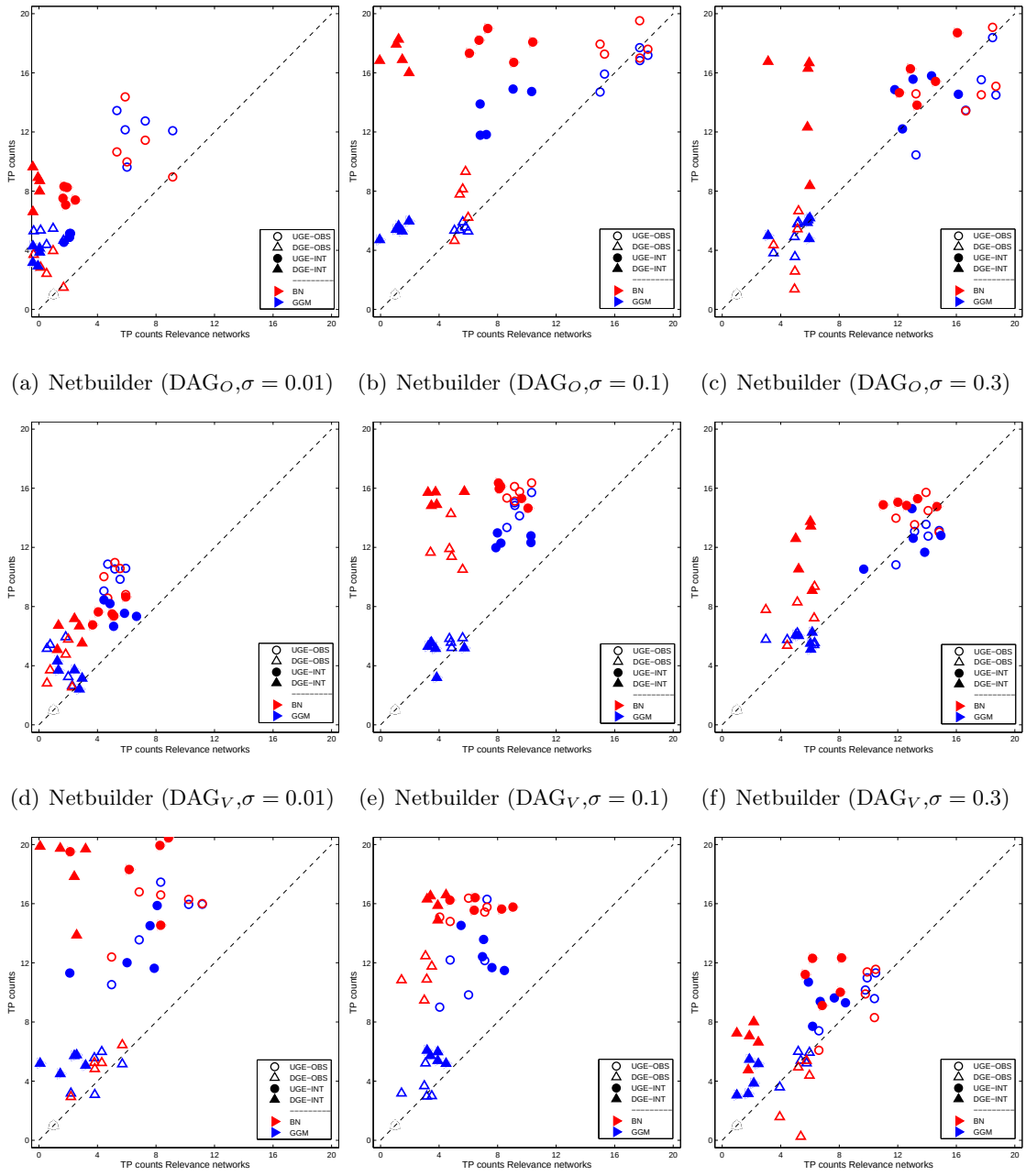
With regard to a cross-method comparison, the same threshold on the association scores can not be used for all three methods, because such an approach not only leads to different true positive (TP) counts but also to different false positive (FP) counts for the three methods, so that neither the TP counts (sensitivities) nor the FP counts (inverse specificities) can be adequately cross-compared. But choosing three different thresholds t_{BN}, t_{GGM}, and t_{RN}, so that each method outputs exactly 5 false positive edges, it can be guaranteed to compare the number of extracted true positive (TP) edges at the same specificity (i.e. the same number of false positive (FP) edges). The value 5 for the number of false positive edges is arbitrarily selected, but corresponds to practically relevant specificities, because in practical (biological) applications one is particularly interested in the performance of reverse engineering methods for low numbers (rates) of false positive (FP) counts.

Table 5.6 gives an overview to which specificities five false positive (FP) edge findings correspond for all four combinations of figure of merit (UGE and DGE) and graph topology (DAG$_O$ and DAG$_V$).

The specificities differ since there are 20 edges in DAG_O and 16 edges in DAG_V, what in turn means that there are 35 undirected (90 directed) true negative edges in DAG_O and 39 undirected (94 directed) true negative edges in DAG_V. If different numbers of true positive (TP) counts can be reached for the same number of false positive (FP) counts 5, it was decided to average over the lowest and the highest compatible TP count. Fortunately, it never happend that the predefined false positive (FP) rate of size 5 was skipped due to false and true positive edge findings with identical association scores (ties), so that no interpolations were necessary.

Scatter plots of these TP counts can be found in Figure 5.17. But when considering these plots, it is important to bear in mind that each combination of figure of merit (UGE and DGE) and network topology (DAG_O and DAG_V) corresponds to a different specificity (see Table 5.6). Therefore neither the TP-scores for UGE and DGE in each panel nor the TP-scores for the two different graph topologies can be directly compared. I.e. each panel has to be considered on its own, and it is not valid to cross-compare the locations of the two different symbols (circles and triangles). P-values of one sample t-tests can be found in K. Appendix XI.

For most of the test data set types these new scatter plot diagnostic results are similar (or at least comparable) to the corresponding $AUROC_1$ scatter plots. That is as before, Bayesian networks (BNs) are superior to both other methods when there are interventional data, and otherwise there is not much difference between GGMs and BNs. Only if the topology DAG_V which has lots of v-structures is used BNs even perform better than GGMs for pure observational data in terms of the DGE figure of merit. That is Bayesian networks can learn some edge directions then. On the other hand, the Relevance network (RN) approach is outperformed by both other methods (BNs and GGMs) except for the real expression data and the Netbuilder data with high noise level ($\sigma = 0.03$). Some possible explanations for that were already discussed above. But there is a clear difference for the Netbuilder data with low noise level (see panels (a) and (d) in Figure 5.17). When looking at the filled circles (red and blue), which correspond to scores obtained for



(a) Netbuilder (DAG_O,$\sigma = 0.01$) (b) Netbuilder (DAG_O,$\sigma = 0.1$) (c) Netbuilder (DAG_O,$\sigma = 0.3$)

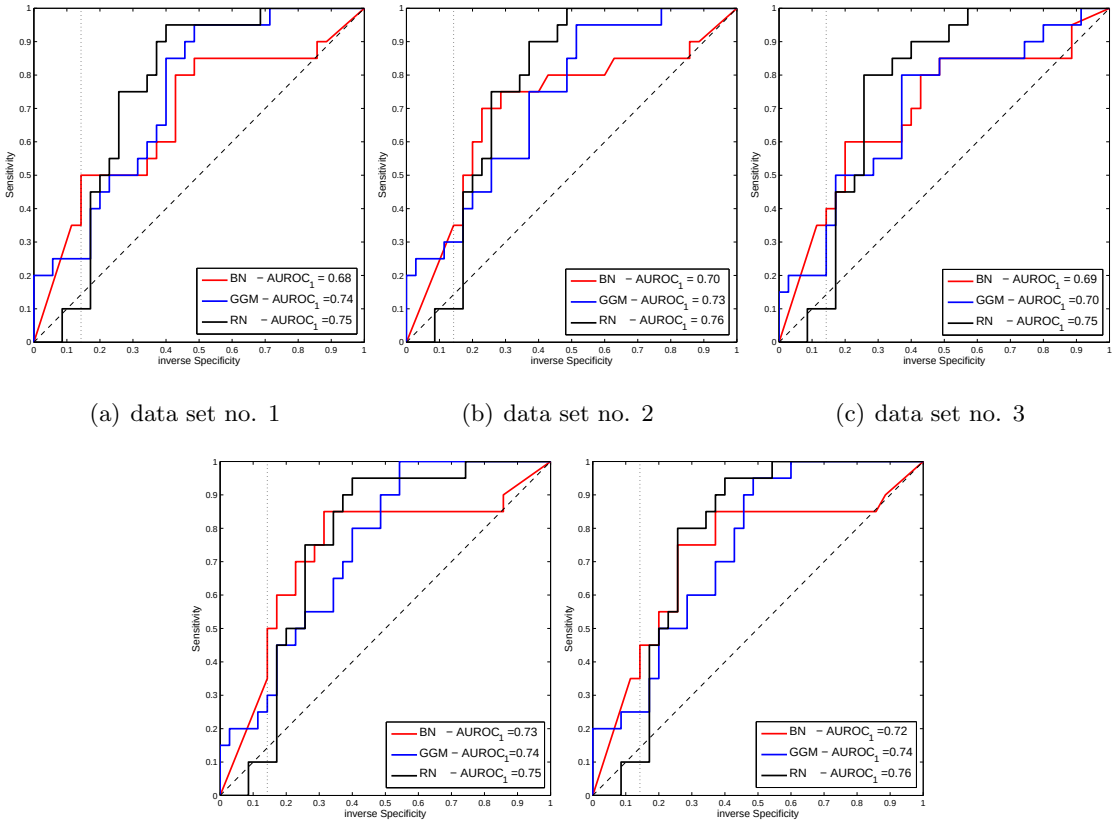
(d) Netbuilder (DAG_V,$\sigma = 0.01$) (e) Netbuilder (DAG_V,$\sigma = 0.1$) (f) Netbuilder (DAG_V,$\sigma = 0.3$)

(g) Gaussian (DAG_O) (h) Gaussian (DAG_V) (i) Real data (DAG_O)

Figure 5.17.: Scatter plot of the true positive (TP) counts obtained when accepting five false positive counts (FP=5). RN versus GGM (in blue) and RN versus BN (in red). Empty symbols represent observational data and filled symbols represent interventional data. The UGE (DGE) figures of merit are represented by circles (triangles). In each plot, to visualise overlaid points uniformly distributed random numbers between -0.5 and 0.5 were added to the coordinates of all points.

interventional data using the UGE figure of merit, it can be seen that the performance of GGMs (panel (a)) and BNs (panels (a) and (b)) has improved in terms of this new TP-score. (In contrast: in terms of the AUROC_1 scores especially Bayesian networks performed surprisingly bad on interventional data (UGE).) To find an explanation for this discrepancy it is useful to have a look at the corresponding ROC curves.

Figure 5.18 provides the UGE figure of merit ROC curves for each of the 5 interventional Netbuilder data sets with low noise level. It can be seen that the progression of the three different ROC curves is almost the same for all 5 data sets. Although Relevance networks yield the highest AUROC_1 scores, that is the biggest areas under the ROC curves, especially for low inverse specificities (i.e. high specificities) there is a directly opposed trend. From each panel can be seen that for very low inverse specificities GGMs perform better than BNs, which in turn perform better than RNs. And then there is always a region (slightly increased inverse specificities) where holds that BNs perform better than GGMs, which in turn perform better than RNs. The interpretation is straightforward: Obviously holds that RNs always learn some false positive edges first, but accepting higher inverse specificities RNs learn all true positive edges somewhen. GGMs learn some true positive edges first, but then extract more false positive edges for higher inverse specificities, that is GGMs have more difficulties in finding the remaining true positive edges. On the other hand, Bayesian networks extract many edges simultaneously at the beginning, and this mixtures of true positive and false positive edges leads to sensitivities which are higher than the sensitivities of the other two methods reached for the same (inverse) specificity. But it seems that some of the true positive edges are never found by Bayesian networks, so that BNs especially for high inverse specificities show a very bad learning performance, what in the end is the reason for the low AUROC_1 scores. This finding especially demonstrates that it is difficult to specify an adequate (fair) threshold ϵ on the inverse specificities in ROC curves for computing AUROC_ϵ values instead of AUROC_1 values (see Section 3.6). As it can be seen from the panels in Figure 5.18 different thresholds would have led to completely different AUROC_ϵ relations between the three methods.



(a) data set no. 1 (b) data set no. 2 (c) data set no. 3

(d) data set no. 4 (e) data set no. 5

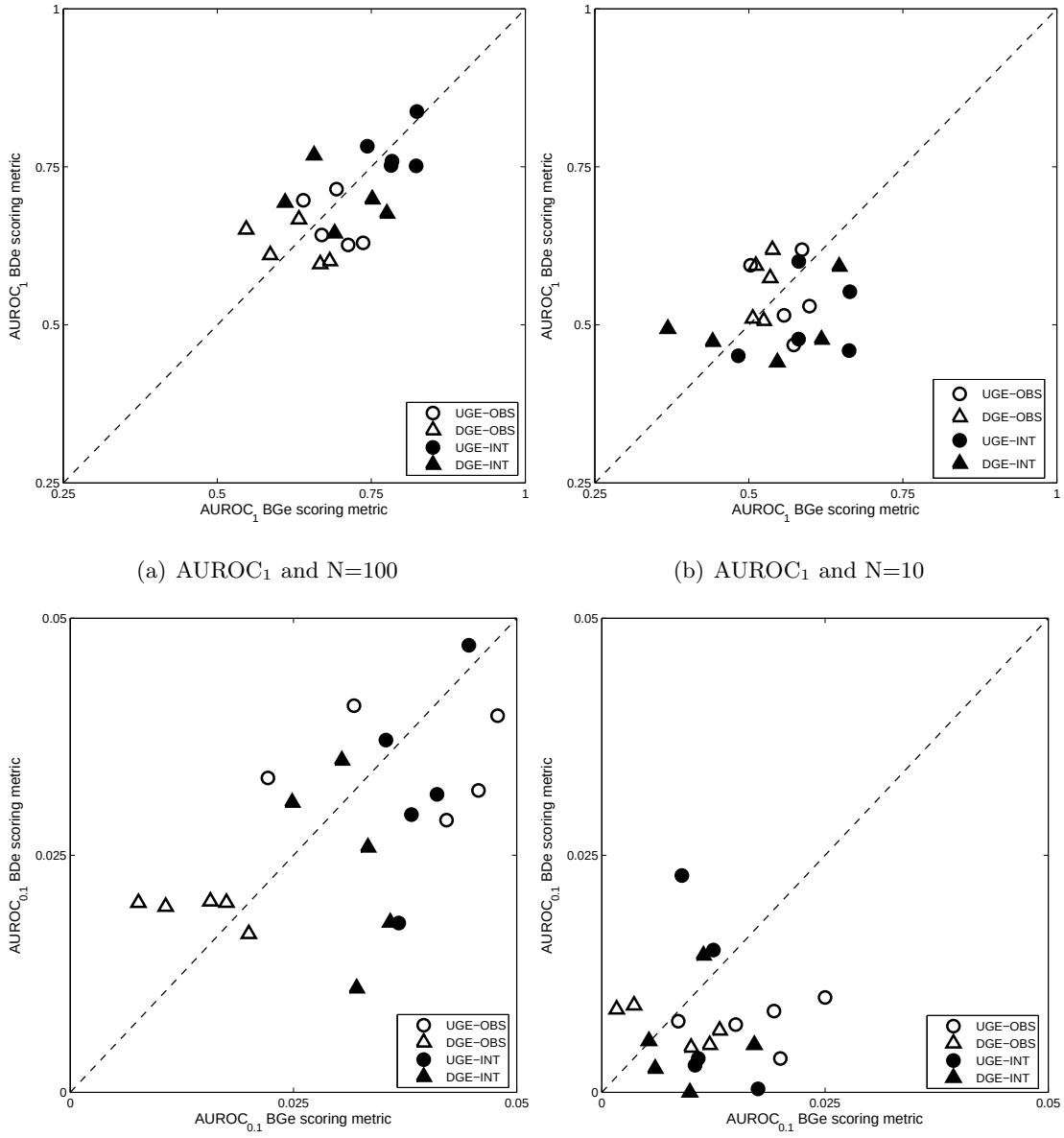
Figure 5.18.: ROC curves (using UGE figure of merit) for all 5 interventional (OR port) Netbuilder data sets generated from the original cytometric graph topology DAG_O with low noise level $\sigma = 0.01$. The vertical black line in each plot corresponds to an inverse specificity of $\frac{30}{35} \approx 0.14$, what in this case (UGE and DAG_O) corresponds to 5 false positive (FP) counts. The BN curves are red, the GGM curves are blue, and the RN curves are black. The diagonal dashed lines correspond to random predictors. Although RNs reach the best performance in terms of $AUROC_1$ scores, it can be clearly seen that the more sophisticated methods (GGMs and BNs) perform clearly better for high specificities (=low inverse specificities). Moreover, form the progressions of the ROC curves can be seen that BNs extract many edges simultaneously at the beginning (both true and false positives), while RNs each time extract some false edges first.

A concise summary of all the results of this first detailed cross-metod comparison can be found in Section 5.8.

5.7. Detailed comparison between BGe-order and BDe-order

In this section the performance of the two different stochastic models for Bayesian networks (BNs) are cross-compared on different test data sets. As described in detail in Section 3.5.2 there are two different stochastic models, also called scoring metrics, for Bayesian networks: the continuous Gaussian model (BGe) and the discrete multinomial model (BDe). The continuous Gaussian scoring metric (BGe) models the (continuous) data as realisations of multivariate Gaussian distributions, and thereby the BGe model uses a normal-Wishart distribution as parameter prior (see Subsection 3.5.2.2 for further details). The disadvantage is that only linear relationships in the data can be modelled. On the other hand, the BDe scoring metric models the data as realisations of multinomial distributions, whereby a Dirichlet distribution is used as parameter prior (see Subsection 3.5.2.1 for further details). Consequently, as multinomial distributions can deal with discrete observations only, it is necessary to discretise the data. But then the BDe score is a very flexible modelling tool, which even allows to model non-linear relationships between the variables. So, there is a certain trade off between the information loss incurred through data discretisation and the modelling flexibility.
If a user wants to use the Bayesian network (BN) methodology for learning regulatory networks he has to decide for one of these two Bayesian network models, before he can start learning the network from the data. From the literature it can be seen that researchers usually decide more or less arbitrarily for one of these two BN models. Usually, either the researchers hold the view that there may be non-linear regulation in the data, so that they decide for the BDe scoring metric, or they want to avoid the information loss incurred through data discretisation, so that they use the BGe scoring metric.

In [41] for example the BDe scoring metric was used to learn the cytometric network from an extensive interventional data set, but there is no comment on why the BDe scoring metric was used. However, the analysis of pure observational and interventional sub data sets sets of size N=100 and size N=10 of this freely available extensive real cytometric expression data set with both scoring metrics (BDe and BGe), using Order-MCMC for



(a) AUROC_1 and N=100

(b) AUROC_1 and N=10

(c) $\mathrm{AUROC}_{0.1}$ and N=100

(d) $\mathrm{AUROC}_{0.1}$ and N=10

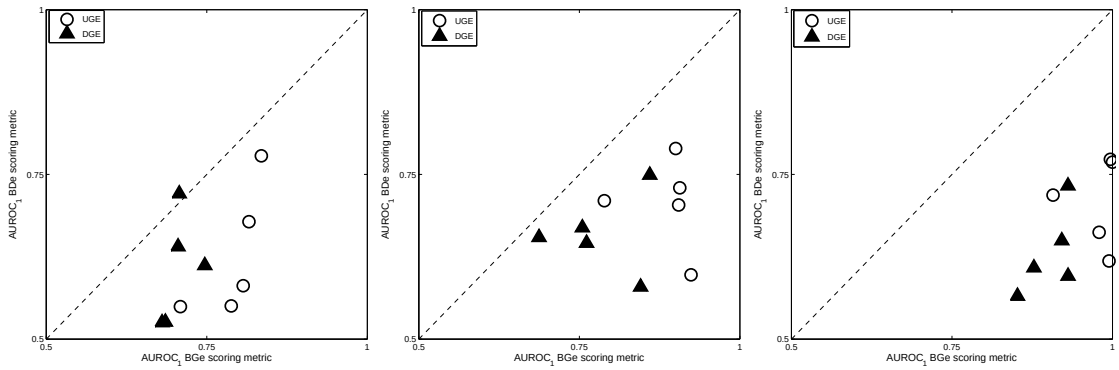
Figure 5.19.: Scatter plots of Bayesian network AUROC_ϵ values: BGe (x-axis) versus BDe (y-axis). All test data sets were sampled from the available real cytometric expression data. Two different sample sizes (N=10 and N=100) and two different ϵ values (0.1 and 1) were used. Empty symbols represent the AUROC scores for pure observational data and filled symbols represent the AUROC score for interventional data. As usual, the DGE figures of merit that take the edge directions into consideration are represented by triangles, while the UGE figures of merit that completely discard the edge directions are represented by circles

learning, reveals that none of this two scoring metrics performs significantly better than the other one. In Figure 5.19 scatter plots of AUROC_1 and $\text{AUROC}_{0.1}$ scores obtained with the UGE and DGE figure of merit can be found. For each combination of data set type (pure observational and interventional) and sample size (N=100 and N=10) five data subsets were sampled, and the AUROC_1 and $\text{AUROC}_{0.1}$ scores were plotted against each other. For the gold standard cytometric network topology see Figure 5.1 in Section 5.2.

From all four panels can be seen that the symbols are located around the diagonal, so that there is no clear trend for one of these two different scoring metrics. Unfortunately, the true regulatory mechanisms of the cytometric network are not known, so that the strength of non-linear regulation can not be appraised. Consequently, it can be concluded only that there is no trend in favour of one scoring metric for these real expression data. In L. Appendix XII there is a table with p-values which confirm these findings numerically (see Table L.1).

To obtain more meaningful cross-comparion results, it is necessary to generate test data sets with synthetic network generators, so that the degree of non-linearity is known. For example, Figure 5.20 shows AUROC_1 scatter plots for synthetic observational data generated from the original cytometric network topology using the Gaussian data generator (see Section 3.7.1). The degree of non-linearity was set to zero and three different sample sizes N=10, N=100, and N=1000 were used. From the three panels (a)-(c) can be seen that independently of the sample size there is a clear trend in favour of the continuous BGe model. For p-values see Table L.2 in L. Appendix XII.

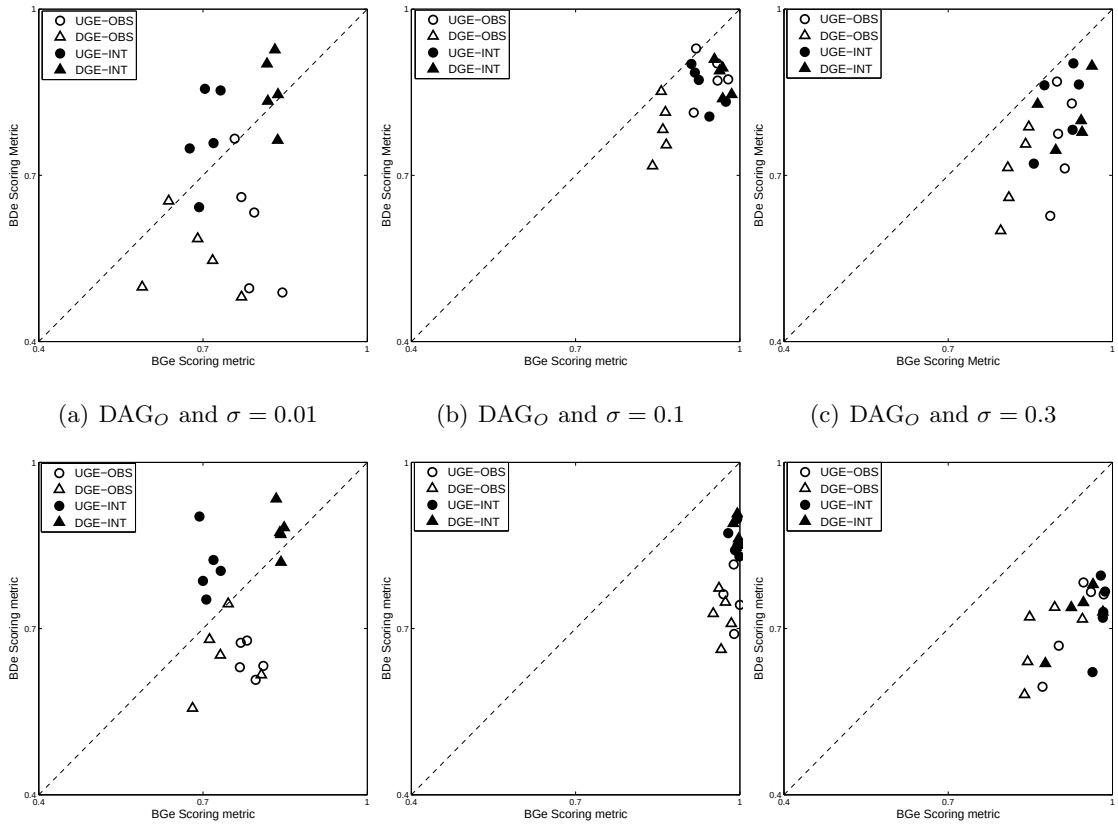
As the modelling flexibility of the BDe model is not needed for learning the true graph in this case of Gaussian distributed data, the results of this analysis demonstrate the information loss incurred through data discretisation. Although, theoretically the BDe model can also model the true relationships, and so can learn the true graph topology, the AUROC values obtained with the BDe metric are substantially lower than the BGe model AUROCs. Especially, it is surprising that even for the high sample size $N = 1000$ the BDe model is not capable of learning the true network topology as well as the BGe



(a) N=10 (b) N=100 (c) N=1000

Figure 5.20.: Scatter plots of Bayesian network AUROC_1 values: BGe (x-axis) versus BDe (y-axis). All observational test data sets were generated with the Gaussian data generator without including any non-linearity (p=0). But three different sample sizes N=10, N=100, and N=1000 were used. Empty circles represent the AUROC_1 scores for the UGE figure of merit, and the DGE figure of merit AUROC_1 scores are represented by filled triangles.

model. In the end, it can be concluded that the BGe scoring metric is inferior to the BDe scoring metric for Gaussian distributed data sets. But as Gaussian distributions of expression data are not biologically realistic, it is important to look at more realistic data generated with the Netbuilder data generator. The Netbuilder test data sets already used in the previous Section 5.6 were generated using OR-ports. So, there is only a slight degree of non-linearity in the Netbuilder data sets, as the effects of parent nodes on their child nodes are nearly additive. Figure 5.21 shows AUROC_1 scatter plots for all these Netbuilder data sets. As expected, in analogy to the AUROC scores observed for the Gaussian test data sets there is also a trend in favour of the BGe scoring metric. But an interesting deviation from this trend is given for the interventional data sets and the low noise level $\sigma = 0.01$ (see panels (a) and (d)). For interventional Netbuilder data with low noise level σ, it seems that the multinomial BDe scoring metric performs better than the Gaussian BGe scoring metric. The bad performance of the Gaussian BGe scoring metric on interventional Netbuilder data with low noise level σ was already investigated in more detail in Section 5.6.



(a) DAG_O and $\sigma = 0.01$ (b) DAG_O and $\sigma = 0.1$ (c) DAG_O and $\sigma = 0.3$

(d) DAG_V and $\sigma = 0.01$ (e) DAG_V and $\sigma = 0.1$ (f) DAG_V and $\sigma = 0.3$

Figure 5.21.: Scatter plots of Bayesian network $AUROC_\epsilon$ values: BGe (x-axis) versus BDe (y-axis). All test data sets of size 100 were generated with the Netbuilder software tool using OR regulation ports. Thereby two different graph topologies, that is the original (DAG_O) and the modified (DAG_V) cytometric graph topology, as well as three different noise levels σ were used. Empty symbols represent the $AUROC_1$ scores for pure observational data, and filled symbols represent the $AUROC_1$ scores for interventional data. Furthermore, the DGE figure of merit AUROCs are represented by triangles, while the UGE figure of merit AUROC values are represented by circles. Table L.3 and Table L.4 in L. Appendix XII provide the corresponding t-test p-values

Conjecturally the bad BGe performance is a consequence of an adversarial interplay between lots of strong indirect associations between the domain variables and the specified prior network. The prior network of independently standard Gaussian distributed domain variables in combination with a data normalisation seems to strengthen these indirect associations, and so weakens the learning performance of the BGe scoring metric drastically. Since this phenomenon can be considered as an artificial one, it is not reasonable to conclude that the multinomial BDe scoring metric for which the data normalisation has no effect is a better scoring metric for interventional Netbuilder data with low noise level. Especially, it can be seen from panel (a) and panel (d) that the BDe scoring metric is inferior to the BGe scoring metric on the corresponding observational Netbuilder data sets with low noise level σ.

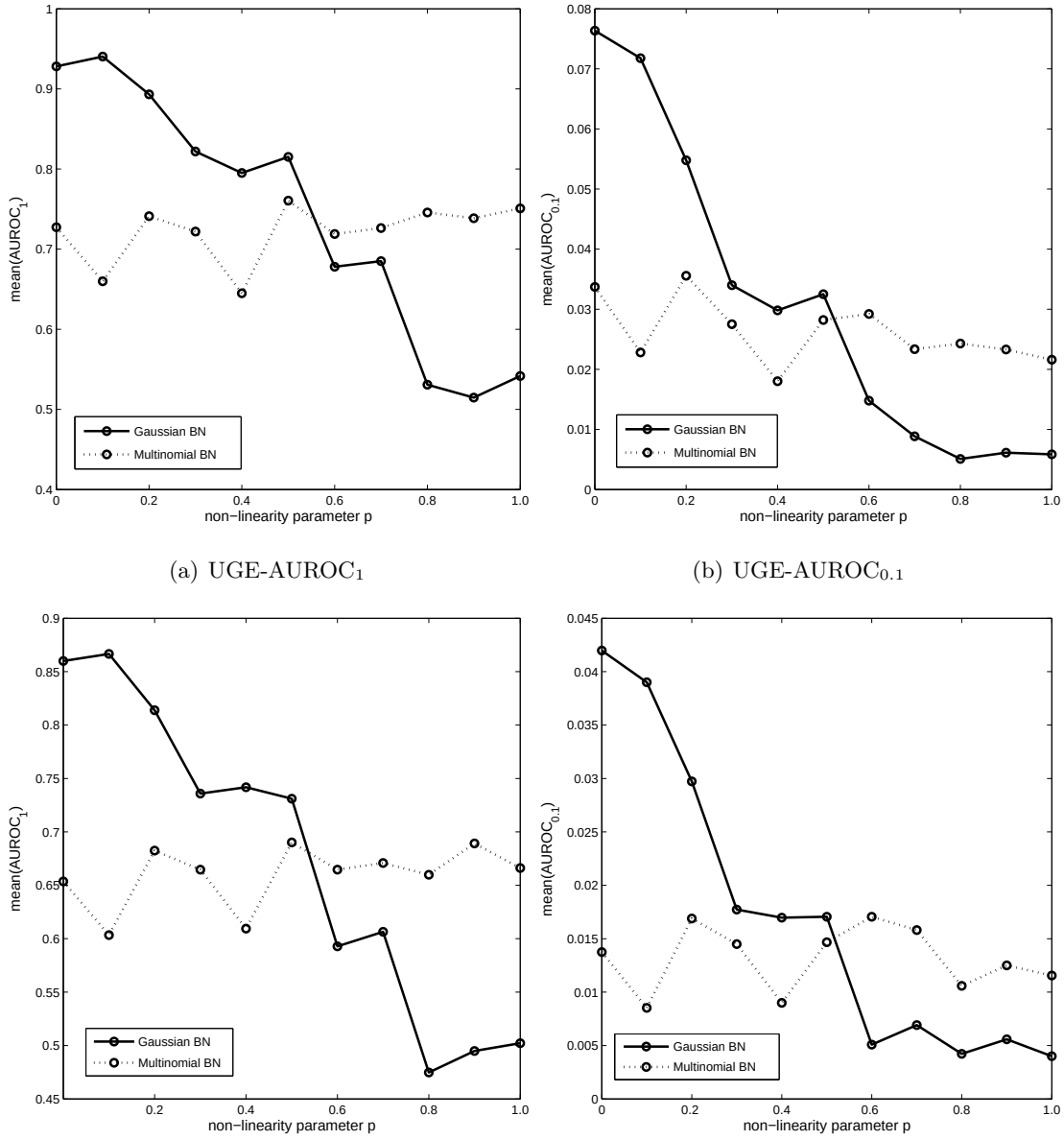
Although there is a little non-linearity in the Netbuilder data generated with OR regulation ports, except for the interventional data with low noise level, the BDe scoring metric performs sytematically worse than the BGe scoring metric. And as mentioned above the exception is presumably due to an artificial phenomenon. Consequently, it seems that there is need for a higher degree of non-linearity in the data to obtain better results with the BDe model than with the BGe model. For the Netbuilder with OR ports (see Figure 5.21) there are also tables with p-values available in L. Appendix XII.

A first possibility to check this, is to generate for each of 11 different non-linearity parameters p five observational data sets with N=100 observations each with the Gaussian data generator. Looking at the corresponding model equations in Section 3.7.1, it can be seen that the degree of non-linearity increases with $p \in [0, 1]$. Especially, for p=0 there is *no* non-linear regulation, and for $p = 1$ there is exclusively non-linear regulation. Trace plots of the mean AUROC_1 and $\text{AUROC}_{0.1}$ scores obtained with the two scoring metrics BDe and BGE for different degrees of non-linearity (p) using both figures of merit are shown in Figure 5.22. From these trace plots can be seen that the BDe model performance does not depend on the degree of non-linearity. The UGE mean AUROC_1 values fluctuate around 0.75, and the UGE mean $\text{AUROC}_{0.1}$ values fluctuate around 0.025. So, the mean AUROC scores do not depend on the parameter of non-linearity p. But

from the trace plots of the BGe scoring metric can be seen that the performance of this BN model strongly depends on the degree of non-linearity p. While the true cytometric graph topology can be learnt almost perfectly if there is little non-linear regulation ($p<0.2$), the mean AUROC scores are decreasing in p. And especially, for non-linearity parameters p higher than 0.5 the performance of the BGe scoring metric becomes worse than the performance of the BDe scoring metric. For $p\geq 0.8$ nearly nothing can be learnt (UGE and DGE mean $AUROC_1$ around 0.5). Nearly the same progressions of the mean AUROC curves can be observed for interventional data generated with the Gaussian data generator (not shown in this thesis).

As a little additional analysis these data sets were used to visualise to which degree Bayesian networks can benefit from the intervention information (see M. Appendix XIII).

Another alternative possibility to check the effect of non-linear regulation is to generate Netbuilder data with the alternative regulation ports AND and XOR. Thereby especially the XOR regulation port yields a high degree of non-linear regulation. But since the data transformations $x \rightarrow \frac{x}{x+1}$ usually applied in Netbuilder weaken the effect of non-linearity, it was decided to omit these transformations when generating new Netbuilder data sets with different regulation ports. Omitting those transformations the ports are simply designed as follows: $\mathrm{AND}(x,y)=x \cdot y$, $\mathrm{OR}(x,y)=x + y \cdot (1 - x)$, $\mathrm{XOR}(x,y) = (1-x+x\cdot y)\cdot(1-y+x\cdot y)$. In the modified cytometric network topology there are six nodes having exactly two parents. These six ports were set to OR, AND, and XOR regulation, while for the node having three parent nodes the OR port was left unchanged. For each of these three Netbuilder networks having the same topology DAG_V but different regulatory mechanisms, i.e. regulation ports, some test data sets were generated. More precisely for each of these networks for three different noise levels $\sigma = 0.01$, $\sigma = 0.1$ and $\sigma = 0.3$ as usual five observational as well as five interventional test data sets were generated and afterwards analysed with both Bayesian network scoring metrics using the usual Order-MCMC approaches. Scatter plots of the $AUROC_1$ scores can be found in Figure 5.23. Table L.5, Table L.6, and Table L.7 in L. Appendix XII provide the corresponding t-test p-values.



(a) UGE-AUROC$_1$

(b) UGE-AUROC$_{0.1}$

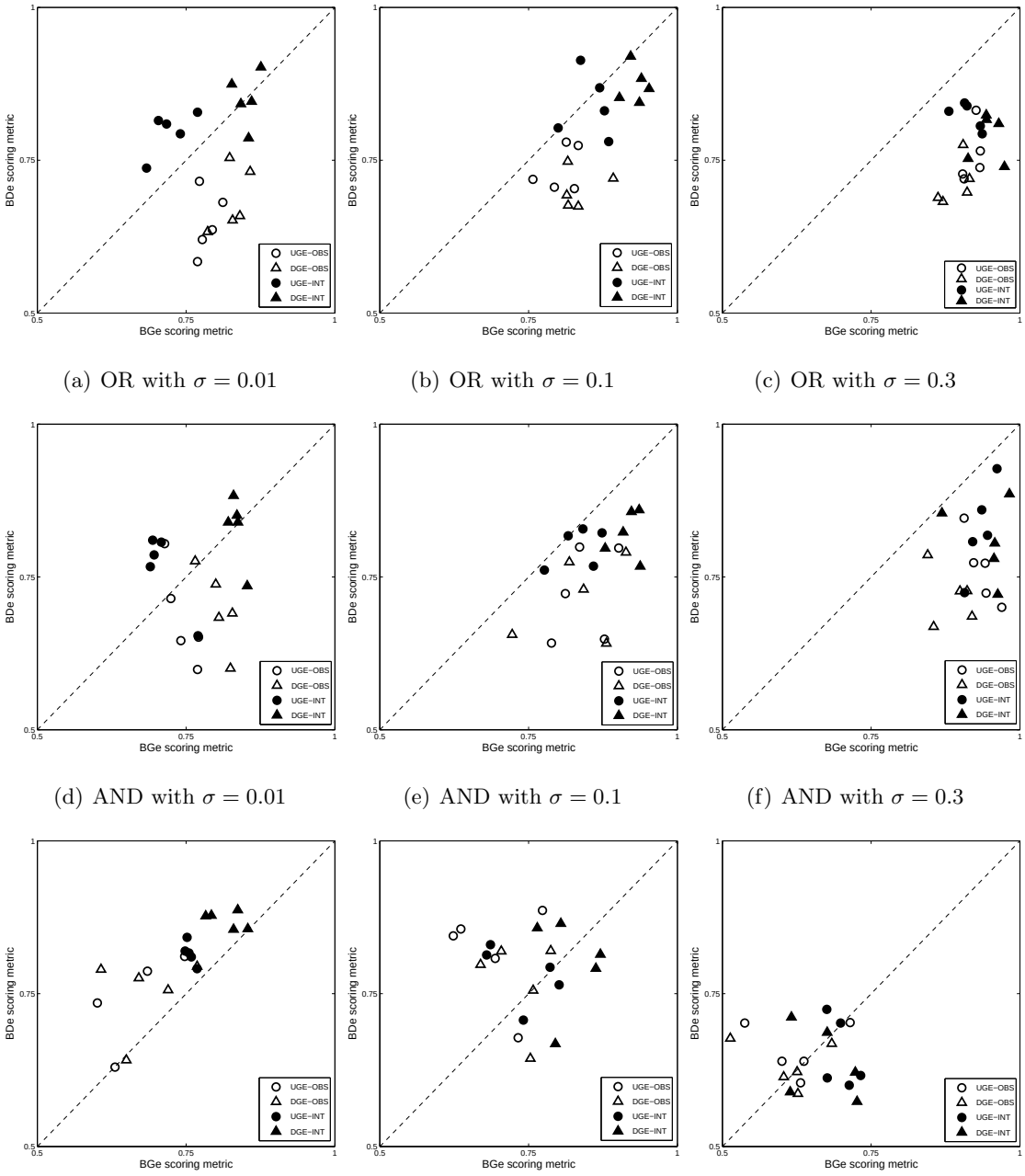
(c) DGE-AUROC$_1$

(d) DGE-AUROC$_{0.1}$

Figure 5.22.: AUROC trace plots illustrating the effect of non-linear regulation in the data. For each of 11 different non-linearity parameters (p=0,0.1,...,1.0) five observational data sets with N=100 observations were generated with the Gaussian data generator. The solid lines correspond to the mean AUROC scores obtained by a Bayesian network Order-MCMC approach using the continuous Gaussian scoring metric (BGe). The dotted lines correspond to the mean AUROC scores obtained by a Bayesian network Order-MCMC approach using the discrete multinomial scoring metric (BDe).

From panels (a)-(c) in Figure 5.23 can be seen that omitting the transformations does not yield a change in the trends for the OR-regulation port Netbuilder data sets. And even replacing six OR regulation ports by AND ports does not lead to different trends (see panels (d)-(f)). Except for the presumably artificial phenomenon which is observable for the interventional Netbuilder data with the low noise level, the BGe scoring metric performs better than the BDe scoring metric. Only when replacing the six OR ports for XOR regulation ports, the trend in favour of the BGe scoring metric disappears. It even seems (see panels (g)-(i)) that the BDe scoring metric is often slightly superior to the BGe scoring metric when XOR regulation ports are given. So for example, the BDe metric performs slightly better on the pure observational Netbuilder data with XOR ports for all three noise levels σ. Nevertheless although the presence of XOR regulation ports yields a high degree of non-linear regulation, the BGe scoring metric is only slightly inferior to the more flexible BDe scoring metric which is therotically capable of learning such non-linear XOR regulation ports.

All these findings which will be summarised and discussed concisely in Section 5.8 reveal that the information loss incurred through data discretisation is profoundly, so that the modelling flexibility of the discrete multinomial BDe scoring metric for Bayesian networks yields better results than the BGe scoring metric only if there is a very high degree of non-linearity in the data. Especially, it seems that the continuous Gaussian BGe scoring metric is capable of learning network topologies even if there is a slight degree of non-linear regulation.



(a) OR with $\sigma = 0.01$ (b) OR with $\sigma = 0.1$ (c) OR with $\sigma = 0.3$

(d) AND with $\sigma = 0.01$ (e) AND with $\sigma = 0.1$ (f) AND with $\sigma = 0.3$

(g) XOR with $\sigma = 0.01$ (h) XOR with $\sigma = 0.1$ (i) XOR with $\sigma = 0.3$

Figure 5.23.: Scatter plots of AUROC_1 scores: BGe scoring metric (x-axis) versus BDe scoring metric (y-axis) for each of nine different combinations of regulatory mechanisms (OR, AND, XOR) and noise level σ using the modified cytometric network topology. For the interpretation of the symbols see Figure 5.21.

5.8. Summary of the results

In this section the results of the comparative evaluation study are summarised. During the screening experiments the performance of ten different combinations of stochastic models and learning algorithms were compared on real expression from the cytometric networks as well as on synthetic data sampled from the same network topology. From the results in terms of AUROC scores it could be seen that neither the different learning algorithms for Gaussian graphical models (bagging and shrinkage) nor the two sampling schemes Structure-MCMC and Order-MCMC lead to substantial different results, so that it could be concluded that it is not necessary to distinguish between these algorithms (sampling schemes) in the following more detailed comparisons. In the context of Bayesian networks it was decided to focus on the Order-MCMC sampling scheme, while the shrinkage based estimator for the covariance matrix was selected for the Gaussian graphical models. Furthermore, as only Relevance networks based on pairwise mutual information scores and Bayesian networks with the discrete multinomial scoring metric (BDe) can model non-linear relationships, so that it presumably only depends on the degree of non-linearity in the data, whether these two models perform better than the others or not, it was decided to restrict on Bayesian networks with the Gaussian scoring metric (BGe), Gaussian graphical models, and Relevance networks based on correlation coefficients for a first more detailed comparison. These three reverse engineering models can learn exclusively linear relationships between domain variables, so that it makes sense to cross-compare their performances predominantely on data sets from domains where the relationships between the variables are not deviating too much from linearity. Afterwards a second more detailed comparison was accomplished in which the performance of the two different Bayesian network models BGe and BDe were cross-compared on data sets generated from networks with different degrees of non-linear regulation.

The first cross-comparative evaluation study, in which the three model classes: Bayesian networks, Gaussian graphical models, and Relevance networks were cross-compared, revealed that it does not necessarily depend on the sophistication of a reverse engineering

model, whether it performs well or not. The supposition that with respect to the amount of the statistical theory, the Bayesian networks must be superior to Gaussian graphical models which in turn must be superior to the simple Relevance networks could not be confirmed in general in the practical applications.

Rather it seems that the matematical and computational expenses associated with Bayesian network approaches lead to a benefit under certain conditions only. Firstly, it could be clearly seen that Bayesian networks clearly outperformed the other two model classes on interventional data sets except for some special cases. The exceptions were observed for Netbuilder data with very low noise levels. But using diagnostics of small networks and an extensive analysis of further Netbuilder data sets it could be revealed that this is presumably an effect of an adversarial interplay between the strong indirect associations in such Netbuilder networks with low noise level and an inadequately specified network prior for the BGe scoring metric. Although the superiority of Bayesian networks in learning network topologies from interventional data is not surprising, because from the three models under comparison only Bayesian networks can deal with and so benefit from interventions, it could at least be confirmed that Bayesian networks indeed clearly become superior then.

Secondly, it seems that Bayesian networks can learn edge directions and so causal relationships between the domain variables when there is a true network topology with special characteristics, that is lots of v-structures. This trend was especially observed for the Gaussian distributed test data sets, but could also be seen for the Netbuilder data with OR ports, except for the low noise level case already mentioned above. On the one hand, these findings can be used to justify the higher expenses of an inference with Bayesian networks even for pure observational data sets, but on the other hand the performance of Gaussian graphical models was often approximately equally well, so that especially with regard to domains with very high numbers of network variables it may make sense to avoid the superexponentially increasing computational costs associated with Bayesian network inference. For example the analysis of the real kidney cell gene expression data set with 200 variables (genes) presented in Chapter 4 took some weeks

of computation time. Alternatively, the data set could have been analysed within some minutes using Gaussian graphical models or Relevance networks.

In addition to these considerations, it must be kept in mind that real biological networks are not rarely based on feed back loops (cycles of directed edges), so that learning edge orientations (causalities) does not make sense at all. From data of such graph topologies with many cycles only the associations, that is undirected edges between variables, can be learnt, so that a computational expensive Bayesian network approach which is based on directed edges may be even extremely misleading.

Comparing the performances of Gaussian graphical models and Relevance networks, it can be seen that Gaussian graphical models are only sometimes superior to Relevance networks in terms of AUROC_1 scores, and it can be concluded that computing partial correlations instead of normal correlations, and so to distinguish between direct and indirect associations, does not always yield advantages. Only for the Gaussian distributed data the Gaussian graphical models perform clearly better than the Relevance networks. An exception is given for the sparse Gaussian data sets with 10 observations only. But this exception is obviously due to the fact that the partial correlations between eleven domain variables can not be learnt from ten observations only. For the Netbuilder data only the following systematic trend can be seen. Gaussian graphical models perform better than Relevance networks for the modified cytometric network topology with lots of v-structures when there is a low or medium noise level; while for the high noise level this superiority is given for the interventional data sets only. On the other hand, for the Netbuilder data from the original cytometric network topology there is less difference between Relevance networks and Gaussian graphical models, and it seems that it depends on whether there are interventional or observational data whether Gaussian graphical models or Relevance networks perform slightly better.
But especially when comparing the number of true positive edge findings which can be obtained when accepting five false positive edge findings, that is a practically more interesting figure of merit, a superiority of Gaussian graphical models to Relevance networks can be seen. While in some cases Gaussian graphical models are clearly superior to

Gaussian graphical models, in the other cases either both methods yield equally good results, or Gaussian graphical models are only slightly inferior. From this point of view the comparative evaluation study in which Bayesian networks with the BGe scoring metric, Gaussian graphical models and Relevance networks were cross-compared has revealed that a computational expensive Bayesian network inference is advisable especially if there are interventional data. For pure observational data it depends on the network topology whether a computational expensive Bayesian network is worthwile. So, especially, for domains with lots of variables where Bayesian network learning is hardly or not an option due to the immense computational costs, it is not unfavourable to use a Gaussian graphical model learning approach instead. Furthermore the results of the study show that Gaussian graphical models are preferable to Relevance network approaches.

In the second more detailed cross-model comparison (see Section 5.7) the two scoring metrics, that is the two different stochastic models, for Bayesian networks were cross-compared on data sets with different degrees of non-linear regulation. The results of this analysis have revealed that it is advisable to discretise continuous expression data only if actually a very high degree of non-linear regulation is given. If there is only a slight degree of non-linear regulation the modelling flexibility of the discrete multinomial BDe Bayesian network model does not compensate the information loss incurred through the necessary data discretisation. Especially, it could be seen that the continuous Gaussian BGe model is capable of learning network topologies even if the dependencies between the variables are deviating a little bit from linear relationships. So, it can be concluded that the application of the BDe scoring metric should be used for domains for which the data (the realisations of the variables) are measured at a discrete level a priori, while for continuous data a discretisation, and so the use of the BDe scoring metric should be avoided except for domains for which a high degree of non-linear regulation can be expected.

6. Discussion and outlook to future work

In this doctoral thesis different reverse engineering machine learning methods which have been proposed in the literature to infer the architecture of biochemical pathways and regulatory networks from high-throughput postgenomic data were cross-compared to understand their relative merits and shortcomings. Such cross-method evaluation studies are important, because all these learning methods are used in the field of systems biology to analyse expression data. But although it is known that all these methods are based on different criteria for quantifying the associations between the domain variables, and so lead to different network topology predictions for the same domain, that is extract different relationships from the same expression data set, researchers have to decide for one of these methods without being able to include the results of empirical studies in their decisions. Consequently, in most of the publications to date either no reasons for the decision in favour of the applied reverse enginneering method are given, or rather questionable justifications based on uncomfirmed theoretical suppositions are given. So for example, users of Relevance networks often argue that more sophisticated methods tend to overfit the data while Relevance networks approaches are based on simple and well-known association scores like Pearson correlation coefficients which are easy to interpret and 'catch' the most important information in the data. In contrast, users which infer expression data with Gaussian graphical models usually argue that is is useful to distinguish between indirect and direct associations, so that the usage of partial correlation coefficients renders Gaussian graphical model approaches superior to Relevance network approaches. While on the other hand they argue that it is not useful to model expression data with Bayesian networks, because it may be misleading

to extract causal relationships, although it is well known in systems biology that the real regulatory mechanisms lead to some time delays between causes and responses as well as not rarely include feed back loops, so that causal relationships can not be extracted from gene expression data when the network variables are not measured over time. Users of Bayesian networks usually do not agree in this point, but pronounce that it is possible to learn causal relationships with Bayesian networks. Especially, the experience of the author of this thesis is that biologists usually would like to have some causal relationships extracted from their data, so that if they are not familiar with the properties of the different reverse engineering methods difficulties, and so do not know about the difficulty of extracting directed edges from non-time dependent data, often tend to prefer a Bayesian network analysis of their data. But as soon as they are informed about these problems, they become as perplexed as the experts for these learning methods, so that it is difficult to reach a decision. And then, even if it is decided in favour of a Bayesian network inference of the data, the next problem arises: Which stochastic model (scoring metric) should be used? From a theoretical point of view it is difficult to say whether it is better to use the continuous Gaussian Bayesian network model (BGe) which can model linear relationships only, but avoids the information loss implied through data discretisation, or whether it is better to analyse discretised data with the much more flexible dicrete multinomial Bayesian network model (BDe).
Consequently, theoretical considerations can be used to discuss the relative merits and shortcomings of the different learning methods only, but do not help to make a decision as long as there are no empirical studies available which quantify these advantages and disadvantages. Only if the results of such cross-method comparisons can be taken into consideration, it is possible to objectively reach a decision in favour of a machine learning method. The theoretical suppositions can be replaced by well-grounded empirically confirmed arguments then.

It is clear that a single study, like the one presented in this thesis, can shed only some light onto that problem. So the presented research is certainly not exhaustive, and

must be seen as a first important step towards uncovering the relative advantages and disadvantages of the different machine learning models empirically.

The biggest problem is to find adequate test data sets for such cross-method comparions. Apriori neither true network topologies nor true regulatory mechanisms are known, and using a synthetic data set generated from a random network topology with some arbitrarily specified relationships between the domain variables is absolutely worthless, because the merits and shortcomings of the methods, that is their performances in learning, strongly depend on the network topology as well as on the regulatory mechanisms that generated the data. In other words it does not matter how well a machine learning method performs in learning when the true network domain is not biologically realistic.

On the one hand, there are lots of regulatory networks described in the biological literature whose topolgies have been extracted from lots of independent traditional molecular biological experiments. But although in these cases composing all information yields the true network topolgies, neither are there data (freely) available then, nor is anything known about the true regulatory mechanisms. On the other hand, real expression data sets are often collected for discovering the unknown network topology behind its variables, so that, altough the regulatory mechanisms which produced the data are real, they can not be used as test data sets, because the true network topology is not known. Consquently, it is impossible to extract the true regulatory mechanisms (since the true network topology is not known) either.

Therefore, it is nearly impossible to find test data sets for which the true network topology is known and the regulatory mechanisms are realistic. An invaluable exception is the freely available cytometric data set which was collected to confirm the network topology of the cytometric signalling pathway which had been composed from the results of lots of independent traditional molecular biological experiments before. So, the true gold standard cytometric network is known and in addition there are realistic

proteomic expression data available. Accordingly, it was decided to use the cytometric network topology to establish the basis for a cross-method comparsion. But as it is never sure to which degree such a true gold standard network topolgy derived from lots of different publications by biologists is reliable, additionaly synthetic data sets were generated and used for the comparative study. And in fact, in a recent publication (see [11]) it was pointed out that there may be a feedback-loop in the cytometric network topology which was not included in the assumed gold standard network topology of [41] (see Figure 5.1). More precisely, [11] report evidence for a feedback loop from ERK back to RAF : $ERK \rightleftharpoons RAF$. If [11] are right, and such a feedback loop really exists in the cytometric network, it may have led to some bias of the results obtained for the real expression data in term of the directed edge evaluation (DGE), as there is a little discrepancy between the true gold-standard network which was assumed in this thesis and the real cytometric network topology. As already mentioned above, especially Bayesian networks are intrinsically restricted to the modelling of directed acyclic graph topologies (without any loops). Consequently, if there is an ambiguity about the direction of the edge between ERK and RAF, this may have worsen the Bayesian network performance on the real cytometric expression data in terms of the DGE evaluation a little bit.

However, in the end such non-consistent publications show that the true gold standard cytometric network topology which was assumed in this thesis may be wrong, and so illustrates the importance of a combined evaluation based on real and synthetic expression data. In any evaluation study based solely on real biological expression data, there may be a discrepancy between the assumed gold standard and the true molecular biological network topology, so that a bias of the results can never be precluded.

Observational as well as interventional synthetic test data sets were generated with a self-implemented Gaussian data generator which produces Gaussian distributed data from a given network topology as well as from the Netbuilder software tool. While from a molecular biological point of view Gaussian distributed data are surely not realistic,

data produced by the Netbuilder tool are seen as much more realistic by biologists.

In the end lots of hypotheses about the merits and shortcomings of the different reverse engineering methods were generated, and not only scatter and trace plots were used to visualise the performances of the different machine learning methods in a concise way, but also t-test p-values were computed as descriptive characteristics for quantifying their differences in the learning performances.

Theoretically, it would have been possible to generate new test data sets with the two data generators using the same parameter settings, and to use the new data to verify (or falsify) the generated hypotheses by confirmative statistical tests. But this was not in the scope of the research presented in this tesis, because the validity of a confirmative statistical test result pointing out that a particular learning method is significantly superior to the other methods would be restricted to the particular kind of test data set, and so can not be generalised.

What does it mean that a method performs best on a data set generated with the Netbuilder software given a particular network topology and parameter setting? It is already questionable whether such a claim can be confirmed when just varying the noise level or the number of observations in the data sets. For that reason the author of this thesis holds the view that it is more important to report some trends for as many as possible different kinds of data sets instead of concentrating on some particular cases without any general validity.

As a next step of research it would be useful not only to vary the parameters, such as the number of observations and the noise level, but also to compare the performances of the machine learning methods on data generated from alternative network topologies, e.g. topologies having much more domain variables. Even if no real expression data are available for these network topologies, it will be interesting to see, whether the same trends observed for the cytometric topology can be seen for more extensive network topologies too. Especially, the computational costs will become an important issue when a network topology with much more domain nodes is considered. While

the computational costs of Relevance networks and Gaussian graphical models stay low, the costs of Bayesian networks increase exponentially with the number of domain variables, so that it will be necessary to apply some heuristic approximations to save some computational time. Furthermore, the modelling of time dependencies, such as time delays between causes and responses as well as the inclusion of feedback loops, that is cycles of directed edges between the domain variables, or selfloops, that is nodes activating or inhibiting themselves, would be further interesing issues.

Theoretically, it can *not* be expected that the reverse engineering methods which were cross-compared in this doctoral thesis can learn networks from time dependent data, since they have been originally developed and proposed to learn networks from non-time dependent expression data, and so can neither model nor deal with time delays and so on. However, this shortcoming has not been a drawback yet, as molecular biological experiments used to produce exclusively non-time-dependent high-throughput data. But the development and availability of modern biotechnologies has just enabled biologists to gather time dependent expression data. Due to the experimental costs and efforts this has not become common practise yet, so that even nowadays non time-dependent data are much more often collected than time dependent expression data. But it can be expected that such time-dependent expression data will be more often available in the next years. Time-dependent expression data can be either a single long time series which reports the realisations (expressions) of all domain variables at lots of points in time, or a set of short time series reporting the realisations of a domain measured in some different individual experimental units (cells) at some points in time. For example when a single long time series of length T is given then the realisations of the domain variables at different (usually equidistant) points of time t are known, and it is possible to model time-dependent interactions, that is the realisation of the variables at point t can be modelled using the realisations at the previous point of time t-1. The two model classes Bayesian networks and Relevance networks, can be easily transferred thus to graphical models for equidistant spaced time-dependent data. Instead of Gaussian

graphical models a novel approach has been proposed which can be used to extract undirected edges from a set of (not necessarily equidistant) short time series using partial dynamical correlations.

Bayesian networks for time-dependent data are referred to as dynamic Bayesian networks. In dynamic Bayesian networks each (directed) edge has a non-ambiguous meaning, as the same dependence relation can not be described by other edges (especially not by the oppositely orientated edge). Moreover, there is no more acyclicity constraint. Consequently, as the immense computational costs, which occur when Bayesian networks are used to model network domains with many nodes, are implied through the acyclicity constraints, dynamic Bayesian networks overcome the major bottleneck of Bayesian network inference; that is they are computationally much more efficient than Bayesian networks for non-time dependent data. More precisely, in the context of dynamic Bayesian networks an edge pointing from node X to node Y means that the realisation y_t of Y at time point t depends on the realisation x_{t-1} of node X at the previous point of time t-1. That is the local score of Y given its parent node X has to be computed from the realisations: (x_{t-1},y_t) for t=2,....,T, while the oppositely orientated edge pointing from Y to X describes the opposite relationship, namely Y_t depends on X_{t-1}, and accordingly has to be scored with the realisations (y_{t-1},x_t). Consequently, there is no more need for a CPDAG representation in dynamic Bayesian networks, as all edges are automatically compelled (non-reversible), and there are no equivalent graphs describing the same set of independence relations. Especially even cyclic graphs are allowed and can be modelled with dynamic Bayesian networks, e.g. self loops, that is edges pointing from a node X to itself. The local score of a self-loop from X to X can be computed from the realisations (x_{t-1},x_t) for t=2,....,T. For further details on dynamic Bayesian networks see [16].

Correspondingly, in the context of Relevance networks the association score between two domain variables X and Y can be computed from the realisations (x_{t-1},y_t) or (y_{t-1},x_t) for t=2,...,T, whereby the association between the former realisations corresponds to an edge

pointing from X to Y, and the association between the latter realisations corresponds to an edge pointing from Y and X. Furthermore for each domain variable X the strength of a self loop can be computed from the realisations (x_{t-1},x_t) for t=2,...,T. So, when the Pearson correlation coefficient is used for measuring the strength of association, the scores of such feedback loops are given by auto-correlations of order one. For further considerations and some efficient algorithms see [46].

Gaussian graphical models approaches can not be adapted straightforwardly, but a completely novel approach has been proposed recently in the literature which treats the realisations of a domain over time as a vector autoregressive (VAR) process, and so allows to transfer the Gaussian grahical model framework from non time-dependent data to analyse time-dependent data. More precisely, the approach is based on the notion of dynamical correlations between curves (trajectories), and computes the partial dynamical correlations between trajectories as association score. Consequently, since (partial) dynamical correlations measure the similarity between trajectories the idea behind this new approach is different from the Bayesian network and Relevance network approaches presented above. That is, instead of explaining the realisations at time point t by the realisations given for the previous time point t-1, loosely speaking, the idea of this novel approach is to specify which pairs of trajectories conditional on all other trajectories are either mostly on the same side of their time average function (positive partial dynamical correlation) or mostly on the other side of their time average function. Thereby as usal, the partial dynamical correlation between two curves is their dynamical correlation conditional on all other curves, and can be computed from the dynamical correlation matrix. Although a set of short time series is needed for this novel approach, and undirected edges can be extracted only, the advantage is that it is also applicable to irregularily spaced (non-equidistant) points of time t. See [36] for further details.

Since dynamic Bayesian networks as well as Relevance networks for time-dependent data are based on the same scores (scoring metrics and association measures) the comparative evaluation study presented in this doctoral thesis indirectly offers some first valuable clues to a cross-method comparion of these learning methods on time-dependent expres-

sion data. Especially in Section 5.7 the two scoring metrics BDe and BGe for Bayesian networks have been cross-compared, and these two competing scoring metrics are also available for dynamic Bayesian networks. It can be expected that in analogy to the results obtained for the non time-dependent data the multinomial BDe scoring metric is inferior to the Gaussian BGe scoring metric if the relationships between the variables depend linearily on the realisations at previous time points, while the BDe scoring metric is superior if and only if the true relationships between the domain variables strongly deviate from linearity.

A. Appendix I

This first appendix describes how to extract sub-networks, i.e. sub-structures of interacting genes, from the confidences of Markov-relation-features in Bayesian network methodology. A sub-network is a graph on a small subset of the variables (nodes) in the domain whose edges encode pairwise Markov-relation-features between these variables. The aim is to identify sub-networks of high statistical confidence, that is sub-networks containing many pairwise Markov-relation-features with high confidence. The approach presented in this appendix was developed by [14].

Within this appendix it is assumed that there are n variables in a domain, and $C(X_i, X_j)$ is the estimated confidence of the Markov-relation-feature between variable X_i and variable X_j, whereby $i, j \in \{1, ..., n\}$ and $i \neq j$. The key assumption of the approach is that the confidences of the pairwise Markov-relation-features are stochastically independent and identical distributed. This means that for all pairs of nodes (X_i, X_j) holds: $P(C(X_i, X_j) \geq c) = g(c)$. Considering a subset U of the domain variables with cardinality k, whereby without loss of generality within this appendix it is assumed that the set U is equal to the set $\{X_1, ..., X_k\}$, there are $K = \frac{k \cdot (k-1)}{2}$ Markov-relation-features between the variables in U. $G(U)$ is the sub-graph over the sub-domain U containing all the undirected edges with a Markov-relation-feature confidence not less than a specified threshold value t_M. Thereby, due to the fact that Markov-realtion-features are symmetric, i.e. $C(X_i, X_j) = C(X_j, X_i)$, undirected edges can be considered only. The probability $P(G(U); c_1, ..., cq)$ that the undirected sub-graph $G(U)$ contains at least q undirected edges with confidences $e_1, ..., e_q$ higher than $c_1, ..., c_q > t_M$ is then bounded by:

$$P(G(U); c_1, ..., cq) = P(e_{i_1} \geq c_1, ..., e_{i_q} \geq c_q | \{i_1, ..., i_q\} \subset \{1, ..., k\}) \leq \binom{K}{q} \cdot \prod_{i=1}^{q} g(c_i)$$

where $K = \frac{k \cdot (k-1)}{2}$ is the number of possible undirected edges in sub-graph $G(U)$, from which q can be selected randomly. The product $\prod_{i=1}^{q} g(c_i)$ provides the probability that for a fixed set of edges $e_1, ..., e_q$ holds $\{e_1 \geq c_1, ..., e_q \geq c_q\}$.
Thus the expected number of such sub-graphs of size k can be bounded by:

$$B(k; c_1, ..., c_q) = \binom{n}{k} \cdot \binom{K}{q} \cdot \prod_{i=1}^{q} g(c_i)$$

This bound can be used to 'score' sub-graphs, because it measures the frequency (probability) that a variable subset of size k contains at least q Markov-realtion-features with confidences higher than $c_1, ..., c_q$. It remains the question how to compute the distribution of the confidences: $g(.)$. Simple estimators for the probabilities $g(c_i)$ are given by the following fractions:

$$\widehat{g(c_i)} = \frac{|M[m_i]|}{n \cdot (n-1)/2}$$

where $|M[m_i]|$ is the number of estimated Markov-relation-features with confidences higher than c_i. Consequently, for a sub-graph $G(U_0)$ of size k containing exactly q_0 undirected edges with Markov-relation-features of confidences $m_1, ..., m_{q_0} \geq t_M$ holds:

$$B(k; c_1, ..., c_q) = \binom{n}{k} \cdot \binom{K}{q_0} \cdot \prod_{i=1}^{q_0} \frac{|M[m_i]|}{n \cdot (n-1)/2}$$

Thereby small scores mean that the corresponding sub-graph $G(U_0)$ contains many Markov-realtion-features of high confidence, that is, edge-feature-constellations that are rare for sub-graphs of size k. In other words, small scores indicate that there is a remarkable high concentration of Markov-realtion-featutres of high confidences in $G(U_0)$, because the probability for such an event is low.

A. Appendix I

To avoid impractical computations, [14] recommend to use the following procedure for finding such high-scoring sub-graphs:

- Initialisation:

  The search starts with a triple-set of nodes $U = \{X_i, X_j, X_k\}$ with pairwise Markov-relation-features of high confidences: $C_1 = C(X_i, X_j) \geq t_M$, $C_2 = C(X_i, X_k) \geq t_M$, and $C_3 = C(X_j, X_k) \geq t_M$, where $t_M \in [0, 1]$ is a specified threshold.

  The corresponding score $B(G(U); C_1, C_2, C_3)$ is then given by:

$$\binom{n}{3} \cdot \binom{3}{3} \cdot \prod_{i=1}^{3} \frac{|M[C_i]|}{n \cdot (n-1)/2}$$

- Iteration step:

  At each iteration step either a node is added or removed from the current set U, attempting to improve the score as much as possible. More precisely, the following new sets are build:

  $U_{add,l} = U \cup \{X_l\}$ for every node $X_l \notin U$ and

  $U_{rem,l} = U \setminus \{X_l\}$ for every node $X_l \in U$.

  For these n sets the corresponding sub-graphs $G_1, ..., G_n$, containing exclusively (undirected) edges with Markov-realtion-features of confidences higher than a second specified threshodl $t_F \in [0, 1]$ can be formed and scored. If the score of the set U_i with the lowest score is lower than the score of the current set U, U is substituted by U_i. Otherwise, the iteration-process stops.

It is recommended to run the search-agorithm several times taking different subsets U of size k as initialisations to obtain more than one sub-graph. Subsequently, it is possible to extract (partially) directed sub-graphs form the undirected edges in U by directing edges with high confidence in their orientation. For instance, if there are two nodes X and Y in the outputed sub-graph U with an Order-relation-feature from X to Y with

a confidence that is much higher than the confidence of the oppositely directed Order-relation-feature (from Y to X), it can be assumed that X causes Y in the domain. Although [14] recommend proceeding in this way, no strict decision rules can be given. The authors in [14] remark that in addition it is useful to take biological knowledge into consideration.

B. Appendix II

Within this second appendix a very brief summary of some important concepts of *chemical kinetics* is given. It is important to understand these biophysical concepts when trying to model the dynamic process of protein production within cells. The reaction between a substrate S and an enzyme E to form a product P via an activated complex ES can be mathematically described as follows:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \overset{k_2}{\rightarrow} E + P$$

where k_1 and k_{-1} are the *association* and *dissociation rates* for the enzyme-substrate compley ES, and k_2 is the *association rate* for the product P. Denote by [.] the concentration of a chemical compound. From the theory of chemical kinetics the following relationship is known:

$$\frac{d[ES]}{dt} = k_1[E][S] - k_{-1}[ES] - k_2[ES]$$

where t denotes the time (see [2] for further details). In equilibrium, the concentration of the active complex ES is constant, and hence the time derivative must be zero, that is:

$$\frac{d[ES]}{dt} = 0.$$

Applying the *law of conservation of mass*:

$$[E]_0 = [ES] + [E],$$

where the subscript 0 denotes the initial mass concentration at time $t = 0$, yields:

$$[ES] = \frac{k_1[E_0][S]}{k_{-1} + k_2 + k_1[S]} = 0.$$

This leads to the following expression for the reaction rate $u = k_2[ES]$, which was first derived by [33] and [3]:

$$u = k_2[ES] = \frac{k_2 k_1[E_0][S]}{k_{-1} + k_2 + k_1[S]} = \frac{k_2[E_0][S]}{\frac{k_{-1}+k_2}{k_1} + [S]} = \frac{V[S]}{K_M + [S]} \tag{B.1}$$

Here, $K_M = \frac{k_{-1}+k_2}{k_1}$ is called the *Michaelis-Menten constant*, and V is the *limiting rate constant*.

These equations of substrate-enzyme reaction kinetics can be used to model the binding of transcription factors to cis-regulatory elements in the promoter region upstream of a gene. In this approach, the substrate S corresponds to a gene, the enzyme E to a transcription factor, the activated complex ES to a gene with a transcription factor bound to its promoter, and the product P to mRNA transcribed. Interpreted this way, (B.1) describes the transcription of a gene induced by the binding of a single transcription factor to its promoter. More complex scenarios, where several transcription factors cooperate or inhibit each other, can be modelled with the same approach, albeit leading to more complex equations. In particular, the following equation originally proposed by [25] is a generalisation of (B.1) to allow cooperativity between the transcription factors:

$$u = \frac{V[S]^h}{K_{0.5}^h + [S]^h}$$

Thereby the constant h is called the *Hill coefficient*, which describes how the binding of one transcription factor to a cis-regulatory region affects the co-binding of other transcription factors.

C. Appendix III

In this third appendix some details about the properties of a self-written software library that deals with Bayesian network learning via Markov Chain Monte Carlo (MCMC) simulations (see Section 3.5) are described. While the computations for the Relevance networks (see Section 3.3) and Gaussian Graphical models (see Section 3.4) can be carried out with softwares provided by [6], [42], and [43], there are only few softwares freely available which are capable of learning Bayesian networks from data. Especially not any of the Bayesian networks softwares for free satisfies the required needs. Either there is no possibility of learning Bayesian networks with MCMC simulations, or the software includes MCMC learning but is too inefficient for dealing with domains containing lots of variables; a necessary attribute when the goal is to analyse gene expression data and to search for interacting genes. Consequently, for the Bayesian networks all approaches had to be implemented. Using the software package *Matlab* by the Mathworks company which provides a tool for coding mathematical programs in an interpreter computer language, the approaches were implemented in about 200 Matlab-subroutines. This section briefly describes some details about this self-implemented software library. Thereby it focuses on the implementation of Structure-MCMC as this MCMC sampling scheme is the more complicated one.

First of all, the user has to decide which stochastic model (BDe- or BGe-metric) and which learning scheme (Structure-MCMC or Order-MCMC) he wants to use. For each of the four combinations different functions are available. They all take a collection of observed data (i.e. gene expression data) as input and perfom the desired MCMC

simulations using the desired stochastic model. The user has to specify some optional arguments only, before running the simulations. Especially, a fan-in restriction has to be inputed and the prior over graphs as well as the prior precision parameters have to be specfied. Furthermore the user has to decide with which directed acyclic graph (DAG) or node ordering the MCMC simulation is initialised. For instance for Structure-MCMC, a random DAG, an empty DAG, and a DAG outputed from a simple Greedy-Search-procedure can be selected. At this stage it is not necessary to determine the length of the burn-in-time and/or the length of the sampling time, because the implementation has the following feature: after a certain number of MCMC iterations (also an optional argument) the program automatically stores all important data results and informs the user about the number of iterations so far accomplished, i.e. the current size of the sample. Especially, due to the fact that the current DAG or node ordering as well as the inputed parameters are stored then, it is possible to resume the MCMC simulation at a latter point of time, for example if the operating system hangs up during the computations, or if it is necessary to accomplish the simulation in serval steps, for example over nights, where the operating computer is not needed. Additionaly, the user can have a look at the current results and decide whether the simulation shall be continued or not. Furthermore, as it is often not useful to store every realisation of the Markov Chain, the user can decide which realisations will be stored. Usually it is sufficient to store every 1,000-th or 10,000-th realisation only, so that unnecessary processing time for storing can be avoided.

For the representation of DAGs within the software so called *adjacency-matrices* are used. The adjacency-matrix I_G of a DAG G over a domain with n nodes $X_1, ..., X_n$ is a $n \times n$-matrix, in which the entry (i, j) is either set to one if the DAG contains an directed edge pointing from node X_i to node X_j, or is set to zero otherwise. For example, if we set $X_1 = A$, $X_2 = B$, $X_3 = C$, $X_4 = D$, and $X_5 = E$ is set, the adjacency-matrix of the DAG presented in Figure 3.1 in Section 3.5 is given by:

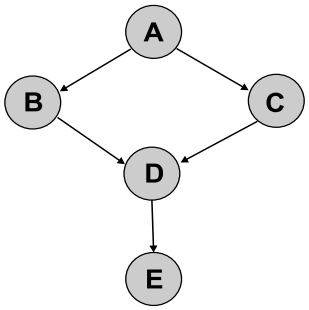


Figure C.1.: Example of a Bayesian network (DAG) with 5 nodes used in Section 3.5

$$I_G = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Furthermore the so called *ancestor-matrix* A_G of each DAG is computed. The $i \times j$ entry of the ancestor-matrix is either set to one if the DAG G contains a path of directed edges leading from node X_j to node X_i, or is set to zero otherwise. The computation of the ancestor matrix is useful as this matrix can be used to determine the neighbourhood of a DAG as shown below if Structure-MCMC is used. The matrices I_G and A_G are in the following relationship: The $i \times j$ entry of A_G is one if the $i \times j$ entry of $A_G^{\diamond}$ is positive, and it is zero if the corresponding entry of $A_G^{\diamond}$ is zero, whereby the matrix $A_G^{\diamond}$ is defined as follows:

$$A_G^{\diamond} = \left(I_G + I_G^2 + ... + I_G^{n-1}\right)^T$$

So, the ancestor-matrix of the DAG presented in Figure 3.1 in Section 3.5 is given by:

$$A_G = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Using the adjacency-matrix I_G and the ancestor matrix A_G it is relatively simple to determine all neighbour DAGs of the given DAG G, that is the collection of all DAGs that can be reached from G by a single edge deletion, addition, or reversal without introducing any directed cycles. For instance, in Figure 3.1 it is not allowed to add the edge '$D \to A$' because this would lead to the following two directed cycles: '$A \to B \to D \to A$' and '$A \to C \to D \to A$'. In general, the following rules can be used (see [20]):

Consider a DAG G over the domain $\{X_1, ..., X_n\}$ with adjacency matrix I_G and ancestor-matrix A_G. Then the following rules can be used to decide which single-edge operations lead to a neigbour graph, and which are invalid due to the acyclicity-constraint.

- Edge-Deletions: All possible edge deletions are always valid, because the removal of a directed edge can not introduce any directed cycle. Consequently, an edge '$X_i \to X_j$' is removable if and only if this edge is present in G, that is if $I_G(i, j) = 1$ holds. to this end, the implemented software searches for all adjacency-matrix entries being equal to 1.

- Edge-Additions: Edge additions are not always valid, and there are two problems that my occur. Firstly, the edge may be already present, so that it can not be added. And secondly, the addition of an edge may lead to one or more directed cycles. Consequently, the following condition for valid addition operations can be used: The addition of the directed edge '$X_i \to X_j$' is valid if and only if $I_G(i, j) = 0$ and $A_G(i, j) = 0$. In the implementation the following matrix M_G is computed

$$M_G = 1_{n,n} - I_G - E_n - A_G$$

where $1_{n,n}$ and E_n represent $n \times n$ matrices with the following properties: $1_{n,n}(i,j) = 1$ for all pairs (i,j) and $E_n = 1$ if $i \neq j$, and 0 otherwise.
Subsequenty, all non-zero entries of the matrix M_G are determined, as it is valid (with respect to the acyclicity-constraint) to add these edges not present in G. More precisely, it is valid to add the edge '$X_i \rightarrow X_j$' if and only if $M_G(i,j) > 0$.

- Edge-Reversals: The reversal of a directed edge '$X_i \rightarrow X_j$' is a two step move. Firstly, the edge is removed from G, and then the oppositely directed edge '$X_i \leftarrow X_j$' is added to G. Thereby the first step poses no problem as edge removals are always valid. But the second step can introduce direcetd cycles since it is an edge addition. Compared with edge addition operations the problem is that the j-th row of the ancestor-matrix A_G may indicate ancestors of X_j, which are exclusively inherited by the edge from node X_i to X_j which is removed within the first removal step. Therefore, it is necessary to determine all ancestors of node X_i through every parent-node X_k of X_j excluding X_i itself as a parent of X_j. This leads to the rule, that the reversal of the edge '$X_i \rightarrow X_j$' is valid if and only if the following two conditions hold: $I_G(i,j) = 1$ and $A_G(k,j) = 0$ for all $k \in \{1,...,n\} \setminus \{i\}$ with $I_G(k,i) = 1$.
  In the implementation, more generally, the matrix $R_G = I_G - (I_G^T \cdot A_G)^T$ is computed. All positive entries $R_G(i,j) > 0$ correspond to valid edge reversal operations from '$X_i \rightarrow X_j$' to '$X_i \leftarrow X_j$'.

If a fan-in restriction f is specified, a further condition must be satisfied for edge additions and edge reversals. Adding a directed edge '$X_i \rightarrow X_j$' increments the cardinality of the parent-set of node X_j. Therefore, it is valid only if: $I_G(1,j) + ... + I_G(n,i) < f$. Consequently, edge-additions and edge-reversals not satisfying this additional condition must be excluded too, as they are invalid with respect to the fan-in restriction.

The exclusion of non-compelled edge reversals from the valid edge reveral operations is more difficult to implement. If the user decides for the exclusion of non-compelled edge-reversals, the software checks for all valid edge-reversals from '$X_i \rightarrow X_j$' to '$X_i \leftarrow X_j$',

whether these edges between X_i and X_j are compelled or not. The edge '$X_i \rightarrow X_j$' is non-compelled if and only if the following condition holds:

$$pa(X_j) = pa(X_i) \cup \{X_i\}$$

That is, the edge '$X_i \rightarrow X_j$' is non-compelled in G if and only if the nodes X_i and X_j have the same parent-set after the removal of the edge from X_i to X_j. Thereby in turn the following equivalence relation holds:

$$pa(X_j) = pa(X_i) \cup \{X_i\} \Leftrightarrow I_G(k,j) = I_G(j,k) \text{ for all } k \in \{1,...,n\} \setminus \{i\}$$

So, if the user decides to exclude the reversal of non-compelled edges in Structure-MCMC, for each edge '$X_i \rightarrow X_j$' with $R_G(i,j) > 0$ is checked, whether the latter condition is satisfied. If it is not, the reversal of the edge '$X_i \rightarrow X_j$' is considered as an invalid one.

Adding, removing, or reversing an edge '$X_i \rightarrow X_j$' in G yields a new DAG $G^\star$. While the adjacency-matrix $I_{G^\star}$ of this new DAG can be simply obtained from I_G by setting $I_G(i,j) = 0$ if the edge is removed, $I_G(i,j) = 1$ if the edge is added, and $I_G(i,j) = 0$ as well as $I_G(j,i) = 1$ if the edge is reversed, it is more difficult to compute the ancestor-matrix $A_{G^\star}$ of the new graph. As the straightforward way, that is computing the matrix $A^{\diamond}_{G^\star} = \left(I_{G^\star} + I^2_{G^\star} + ... + I^{n-1}_{G^\star}\right)^T$, is computational expensive, [20] recommend to use *update-rules* for obtaining the new ancestor matrix. For each of the three edge operations they present an update rule that can be used to derive the ancestor-matrix of the neighbour DAG $G^\star$ from I_G and A_G. These update-rules are implemented in our software, and lead to a remarkable reduction of the computational costs of Structure-MCMC simulations.

D. Appendix IV

In this fourth appendix some results are presented that confirm that the shrinkage based Gaussian graphical model estimator (GGM-4) is superior to the other three bagging-based estimators for GGMs. In addition to the test data sets used in Section 5.5, some further test data sets were generated with the Netbuilder data generator (see Section 3.7.2). Thereby it was distinguished not only between the two sample sizes $N = 10$ and $N = 100$ and the two AUROC criteria AUROC_1 and $\text{AUROC}_{0.1}$, but also between two different 'experimental' noise levels in terms of two different signal-to-noise ratios. More precisely, to each variable X_i of the cytometric domain an additional noise variable E_i was added after having generated the data sets with the Netbuilder software. The realisations e_i of these noise variables E_i were sampled from a standard Gaussian distribution $N(0, 1)$ and mutiplied by the factor $\frac{1}{\tau} \times S_{x_i}$, whereby S_{x_i} is the empirical standard deviation of the realisations x_i of domain variable X_i outputed from Netbuilder. Consequently, the factor τ can be interpreted as the empirical signal-to-noise ratio (SNR): $\tau = \frac{S_{x_i}}{S_{e_i}}$. The signal-to-noise ratio was set to 1 and to 10, so that very noisy data ($\tau = 1$)as well as non-noisy ($\tau = 10$) data sets were included. As the noise was added after having generated the data, it is denoted 'observational' noise. The dynamic noise included during generating the Netbuilder was set to the low noise level $\sigma = 0.01$. For all four combinations of sample size N and signal-to-noise ratio τ five observational data sets were generated, and the mean AUROC scores for all four GGM estimators using both AUROC criteria can be found in Table D.1.

Only in three of eight cases there is another GGM estimator leading to a slightly higher mean AUROC value than the shrinkage based estimator (GGM-4). Consequently, these

SNR	N	AUROC$_\epsilon$	GGM-1	GGM-2	GGM-3	GGM-4
1	100	1	0.5663	0.5691	0.5660	**0.5726**
1	100	0.1	0.0084	0.0084	**0.0086**	0.0081
1	10	1	0.4906	0.4526	0.4760	**0.4917**
1	10	0.1	**0.0083**	0.0054	0.0037	0.0064
10	100	1	0.6889	0.6917	0.6857	**0.7711**
10	100	0.1	0.0274	**0.0274**	0.0274	0.0270
10	10	1	0.5037	0.5657	0.4674	**0.7300**
10	10	0.1	0.0084	0.0107	0.0056	**0.0210**

Table D.1.: Results for the observational Netbuilder data

results strengthen the superior position of the shrinkage-based estimator among all four different estimators of the partial correlation coefficients.

E. Appendix V

This fiftth appendix provides tables which summarise the AUROC_1 results that were obtained for data sets generated from three different sources: real gene expression measurments, the Netbuilder generator tool, and the Gaussian data generator.

For all three kinds of test data sets a table with the means $\mu[\mathrm{AUROC}_1]$, standard deviations $\sigma(\mathrm{AUROC}_1)$, and some p-values p(.) of two-sided one-sample Student t-tests are given.

The t-tests were used to test, whether the AUROC_1 means are different or not. More precisely, for each pair of methods M_i and M_j the null hypothesis H_0: $\mu[\mathrm{AUROC}_1(M_i)] = \mu[\mathrm{AUROC}_1(M_j)]$ was tested against the corresponding alternative H_1: $\mu[\mathrm{AUROC}_1(M_i)] \neq \mu[\mathrm{AUROC}_1(M_j)]$. Thereby no correction for multiple statistical testing was applied, so that these p-values must be considered with caution. Although they can be seen as meaningful descriptive statistics indicating, whether there may be a difference, they can *not* be used to confirm H_1 statistically.

All tables have the same structure. After a row indicating the figure of merit (UGE and DGE) and the data set type (pure observational and interventional), there is one row for each of the three methods under comparison (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN)) which contains the statistics mentioned above. The last three columns p(BN), p(GGM), and p(RN) provide the t-test p-values, whereby the abbreviations in brackets indicate against which other method was tested.

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.8848	0.0543	-	0.8815	0.0079
GGM	0.8814	0.0373	0.8815	-	0.0015
RN	0.6809	0.0816	0.0079	0.0015	-
DGE - Observational					
BN	0.7817	0.0711	-	0.6704	0.0239
GGM	0.7967	0.0286	0.6704	-	0.0015
RN	0.6407	0.0635	0.0239	0.0015	-
UGE - Interventional					
BN	0.9661	0.0391	-	0.0024	0.0018
GGM	0.8203	0.0532	0.0024	-	0.0082
RN	0.7097	0.0541	0.0018	0.0082	-
DGE - Interventional					
BN	0.9796	0.0187	-	0.0002	0.0002
GGM	0.7488	0.0409	0.0002	-	0.0081
RN	0.6631	0.0421	0.0002	0.0081	-

Table E.1.: AUROC$_1$ COMPARISON - Gaussian data

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9564	0.0273	-	0.0247	0.0469
GGM	0.8803	0.0656	0.0247	-	0.0909
RN	0.9323	0.0188	0.0469	0.0909	-
DGE - Observational					
BN	0.8572	0.0100	-	0.0288	0.0116
GGM	0.7957	0.0508	0.0288	-	0.0891
RN	0.8362	0.0146	0.0116	0.0891	-
UGE - Interventional					
BN	0.9346	0.0254	-	0.0188	0.0006
GGM	0.8300	0.0438	0.0188	-	0.1466
RN	0.8003	0.0082	0.0006	0.1466	-
DGE - Interventional					
BN	0.9678	0.0114	-	0.0004	0.0000
GGM	0.7574	0.0339	0.0004	-	0.1359
RN	0.7336	0.0064	0.0000	0.1359	-

Table E.2.: AUROC$_1$ COMPARISON - Netbuilder data with $\sigma = 0.1$

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.6904	0.0376	-	0.8754	0.2957
GGM	0.6854	0.0542	0.8754	-	0.6175
RN	0.6680	0.0546	0.2957	0.6175	-
DGE - Observational					
BN	0.6231	0.0564	-	0.5316	0.7276
GGM	0.6443	0.0419	0.5316	-	0.6139
RN	0.6307	0.0425	0.7276	0.6139	-
UGE - Interventional					
BN	0.7912	0.0335	-	0.0552	0.0003
GGM	0.7129	0.0559	0.0552	-	0.0010
RN	0.5686	0.0286	0.0003	0.0010	-
DGE - Interventional					
BN	0.6969	0.0676	-	0.4802	0.0076
GGM	0.6656	0.0437	0.4802	-	0.0010
RN	0.5533	0.0222	0.0076	0.0010	-

Table E.3.: AUROC$_1$ COMPARISON - Real cytometric expression data

F. Appendix VI

This sixth appendix provides tables which summarise the AUROC_1 results that were obtained for the observational data sets generated with the Gaussian data generator. For three different sample sizes (N=10, N=100, and N=1000) the means $\mu[\text{AUROC}_1]$, standard deviations $\sigma(\text{AUROC}_1)$, and some p-values p(.) of two-sided one-sample Student t-tests are given.

The t-tests were used to test, whether the AUROC_1 means are different or not. More precisely, for each pair of methods M_i and M_j the null hypothesis H_0: $\mu[\text{AUROC}_1(M_i)] = \mu[\text{AUROC}_1(M_j)]$ was tested against the corresponding alternative H_1: $\mu[\text{AUROC}_1(M_i)] \neq \mu[\text{AUROC}_1(M_j)]$. Thereby no correction for multiple statistical testing was applied, so that these p-values must be considered with caution. Although they can be seen as meaningful descriptive statistics indicating, whether there may be a difference or not, they can *not* be used to confirm H_1 statistically.

In the tables there are rows indicating the figure of merit (UGE and DGE) as well as the current sample size N, and then there is one row for each of the three methods under comparison (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN)) which contains the statistics mentioned above. The last three columns p(BN), p(GGM), and p(RN) provide the t-test p-values, whereby the abbreviations in brackets indicate against which other method was tested.

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE and N = 10					
BN	0.7909	0.0488	-	0.0238	0.0164
GGM	0.7657	0.0627	0.0238	-	0.0560
RN	0.7123	0.0646	0.0164	0.0560	-
DGE and N = 10					
BN	0.7051	0.0262	-	0.9407	0.0459
GGM	0.7066	0.0486	0.9407	-	0.0557
RN	0.6651	0.0503	0.0459	0.0557	-
UGE and N = 100					
BN	0.8848	0.0543	-	0.8815	0.0079
GGM	0.8814	0.0373	0.8815	-	0.0015
RN	0.6809	0.0816	0.0079	0.0015	-
DGE and N = 100					
BN	0.7817	0.0711	-	0.6704	0.0239
GGM	0.7967	0.0286	0.6704	-	0.0015
RN	0.6407	0.0635	0.0239	0.0015	-
UGE and N = 1000					
BN	0.9756	0.0389	-	0.1638	0.0015
GGM	0.9551	0.0275	0.1638	-	0.0027
RN	0.6911	0.0833	0.0015	0.0027	-
DGE and N = 1000					
BN	0.9025	0.0357	-	0.0613	0.0019
GGM	0.8541	0.0214	0.0613	-	0.0027
RN	0.6487	0.0648	0.0019	0.0027	-

Table F.1.: AUROC$_1$ COMPARISON - observational Gaussian data with different N.

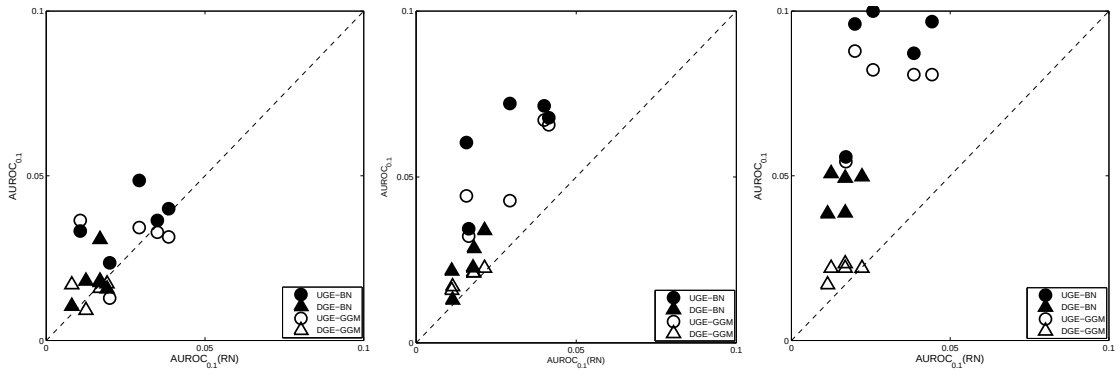
G. Appendix VII

In this seventh appendix scatter plots of $\mathrm{AUROC}_{0.1}$ for the Gaussian distributed observational data with sample size N=10, N=100, and N=1000 can be found. The corresponding statistics, that is means, standard deviations, and t-test p-values can be found in Table G.1. Especially from panel (a) in Figure G.1 as well as from the corresponding p-values in the table can be seen that except for the small sample size (N=10) the results are comparable to the results obtained with the AUROC_1 score.

In the table there are rows indicating the figure of merit (UGE and DGE) as well as the current sample size N, and then there is one row for each of the three methods under comparison (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN)) which contains the empirical statistics mentioned above. The last three columns p(BN), p(GGM), and p(RN) provide the t-test p-values, whereby the abbreviations in brackets indicate against which other method was tested.

Method	$\mu[\text{AUROC}_{0.1}]$	$\sigma(\text{AUROC}_{0.1})$	p(BN)	p(GGM)	p(RN)
UGE and N = 10					
BN	0.0364	0.0092	-	0.0896	0.1061
GGM	0.0296	0.0095	0.0896	-	0.6655
RN	0.0267	0.0114	0.1061	0.6655	-
DGE and N = 10					
BN	0.0185	0.0075	-	0.4278	0.2608
GGM	0.0154	0.0036	0.4278	-	0.7555
RN	0.0147	0.0044	0.2608	0.7555	-
UGE and N = 100					
BN	0.0612	0.0157	-	0.1108	0.0029
GGM	0.0504	0.0153	0.1108	-	0.0020
RN	0.0286	0.0123	0.0029	0.0020	-
DGE and N = 100					
BN	0.0239	0.0079	-	0.1691	0.0178
GGM	0.0195	0.0029	0.1691	-	0.0096
RN	0.0159	0.0045	0.0178	0.0096	-
UGE and N = 1000					
BN	0.0872	0.0182	-	0.0309	0.0014
GGM	0.0771	0.0131	0.0309	-	0.0014
RN	0.0291	0.0118	0.0014	0.0014	-
DGE and N = 1000					
BN	0.0454	0.0062	-	0.0004	0.0004
GGM	0.0212	0.0025	0.0004	-	0.0295
RN	0.0160	0.0043	0.0004	0.0295	-

Table G.1.: $\text{AUROC}_{0.1}$ COMPARISON - Gaussian data with different N



(a) N=10 (b) N=100 (c) N=1000

Figure G.1.: Scatter plots of $AUROC_{0.1}$ values: RN versus GGM (empty symbols) and RN versus BN (filled symbols). Exclusively observational data sets were generated with the Gaussian data generator. Thereby three different sample sizes N were used. See text for further information. The DGE figures of merit that take the edge directions into consideration are represented by triangles, while the UGE figures of merit that discard the edge directions are represented by circles

H. Appendix VIII

As former appendices this eighth appendix provides tables which summarise some AUROC scores. This appendix provides tables for test data sets generated with the Gaussian data generator using sample size N=100 for the two different graph topologies DAG_O and DAG_V. As before, tables with the means $\mu[\mathrm{AUROC}_1]$, standard deviations $\sigma(\mathrm{AUROC}_1)$, and some p-values p(.) of two-sided one-sample Student t-tests are given.

As usual, the t-tests were used to test, whether the AUROC_1 means are different or not, that is for each pair of methods M_i and M_j the hypothesis H_0: $\mu[\mathrm{AUROC}_1(M_i)] = \mu[\mathrm{AUROC}_1(M_j)]$ was tested against its alternative. But because no correction for multiple statistical testing was applied, the p-values can be seen as meaningful descriptive statistics only which simply indicate, whether there may be a difference. The tables are arranged as all tables of this kind, e.g. see Appendix 5.

Method	$\mu[\text{AUROC}_1])$	$\sigma(\text{AUROC}_1)$	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.8848	0.0543	-	0.8815	0.0079
GGM	0.8814	0.0373	0.8815	-	0.0015
RN	0.6809	0.0816	0.0079	0.0015	-
DGE - Observational					
BN	0.7817	0.0711	-	0.6704	0.0239
GGM	0.7967	0.0286	0.6704	-	0.0015
RN	0.6407	0.0635	0.0239	0.0015	-
UGE - Interventional					
BN	0.9661	0.0391	-	0.0024	0.0018
GGM	0.8203	0.0532	0.0024	-	0.0082
RN	0.7097	0.0541	0.0018	0.0082	-
DGE - Interventional					
BN	0.9796	0.0187	-	0.0002	0.0002
GGM	0.7488	0.0409	0.0002	-	0.0081
RN	0.6631	0.0421	0.0002	0.0081	-

Table H.1.: AUROC_1 COMPARISON - Gaussian data from graph topology DAG_O with N=100

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9775	0.0345	-	0.0087	0.0013
GGM	0.8933	0.0583	0.0087	-	0.0043
RN	0.6987	0.0981	0.0013	0.0043	-
DGE - Observational					
BN	0.9487	0.0440	-	0.0012	0.0004
GGM	0.8257	0.0487	0.0012	-	0.0043
RN	0.6649	0.0814	0.0004	0.0043	-
UGE - Interventional					
BN	1.000	0.0000	-	0.0010	0.0014
GGM	0.8878	0.0293	0.0010	-	0.0199
RN	0.7436	0.0730	0.0014	0.0199	-
DGE - Interventional					
BN	0.9976	0.0038	-	0.0001	0.0004
GGM	0.8220	0.0001	0.0001	-	0.0196
RN	0.7021	0.0004	0.0004	0.0196	-

Table H.2.: AUROC$_1$ COMPARISON - Gaussian data from graph topology DAG$_V$ with N=100

I. Appendix IX

This ninth appendix provides tables which summarise the AUROC_1 results that were obtained for data sets generated with the Nebuilder software tool. For all six combinations of network topology (DAG_O and DAG_V) and noise level σ a table with the means $\mu[\mathrm{AUROC}_1]$, standard deviations $\sigma(\mathrm{AUROC}_1)$, and some p-values p(.) of two-sided one-sample Student t-tests are given. The t-tests were used to test, whether the AUROC_1 means are different or not. More precisely, for each pair of methods M_i and M_j the null hypothesis

$$\mathrm{H}_0\text{: } \mu[\mathrm{AUROC}_1(M_i)] = \mu[\mathrm{AUROC}_1(M_j)]$$

was tested against the corresponding alternative

$$\mathrm{H}_1\text{: } \mu[\mathrm{AUROC}_1(M_i)] \neq \mu[\mathrm{AUROC}_1(M_j)].$$

Thereby no correction for multiple statistical testing was applied, so that these p-values must be considered with caution. Although they can be seen as meaningful descriptive statistics indicating, whether there may be a difference, they can *not* be used to confirm H_1 statistically.

All six tables have the same structure. After a row indicating the figure of merit (UGE and DGE) and the data set type (pure observational and interventional), there is one row for each of the three methods under comparison (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN)) which contains the statistics mentioned above. The last three columns p(BN), p(GGM), and p(RN) provide the t-test

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7901	0.0336	-	0.0764	0.0444
GGM	0.8143	0.0191	0.0764	-	0.0009
RN	0.7434	0.0081	0.0444	0.0009	-
DGE - Observational					
BN	0.6808	0.0703	-	0.0669	0.7977
GGM	0.7446	0.0150	0.0669	-	0.0010
RN	0.6893	0.0063	0.7977	0.0010	-
UGE - Interventional					
BN	0.7047	0.0221	-	0.0675	0.0076
GGM	0.7297	0.0183	0.0675	-	0.0410
RN	0.7537	0.0063	0.0076	0.0410	-
DGE - Interventional					
BN	0.8280	0.0097	-	0.0001	0.0000
GGM	0.6793	0.0144	0.0001	-	0.0468
RN	0.6973	0.0049	0.0000	0.0468	-

Table I.1.: AUROC$_1$ COMPARISON - Netbuilder data from topology DAG$_O$ with $\sigma = 0.01$

p-values, whereby the abbreviations in brackets indicate against which other method was tested.

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9564	0.0273	-	0.0247	0.0469
GGM	0.8803	0.0656	0.0247	-	0.0909
RN	0.9323	0.0188	0.0469	0.0909	-
DGE - Observational					
BN	0.8572	0.0100	-	0.0288	0.0116
GGM	0.7957	0.0508	0.0288	-	0.0891
RN	0.8362	0.0146	0.0116	0.0891	-
UGE - Interventional					
BN	0.9346	0.0254	-	0.0188	0.0006
GGM	0.8300	0.0438	0.0188	-	0.1466
RN	0.8003	0.0082	0.0006	0.1466	-
DGE - Interventional					
BN	0.9678	0.0114	-	0.0004	0.0000
GGM	0.7574	0.0339	0.0004	-	0.1359
RN	0.7336	0.0064	0.0000	0.1359	-

Table I.2.: AUROC$_1$ COMPARISON - Netbuilder data from topology DAG$_O$ with $\sigma = 0.1$

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9049	0.0150	-	0.2776	0.1310
GGM	0.8829	0.0486	0.2776	-	0.0750
RN	0.9163	0.0179	0.1310	0.0750	-
DGE - Observational					
BN	0.8208	0.0223	-	0.3024	0.8234
GGM	0.7979	0.0381	0.3024	-	0.0782
RN	0.8238	0.0139	0.8234	0.0782	-
UGE - Interventional					
BN	0.9053	0.0367	-	0.0168	0.0329
GGM	0.8571	0.0251	0.0168	-	0.7139
RN	0.8631	0.0273	0.0329	0.7139	-
DGE - Interventional					
BN	0.9219	0.0408	-	0.0013	0.0007
GGM	0.7776	0.0230	0.0013	-	0.7051
RN	0.7824	0.0212	0.0007	0.7051	-

Table I.3.: AUROC$_1$ COMPARISON - Netbuilder data from topology DAG$_O$ with $\sigma = 0.3$

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7845	0.0184	-	0.0055	0.0018
GGM	0.8529	0.0139	0.0055	-	0.0000
RN	0.7170	0.0094	0.0018	0.0000	-
DGE - Observational					
BN	0.7354	0.0467	-	0.0748	0.0558
GGM	0.7927	0.0117	0.0748	-	0.0000
RN	0.6801	0.0078	0.0558	0.0000	-
UGE - Interventional					
BN	0.7102	0.0156	-	0.0008	0.3208
GGM	0.7900	0.0180	0.0008	-	0.0110
RN	0.7280	0.0279	0.3208	0.0110	-
DGE - Interventional					
BN	0.8413	0.0052	-	0.0000	0.0001
GGM	0.7258	0.0143	0.0000	-	0.0115
RN	0.6773	0.0217	0.0001	0.0115	-

Table I.4.: AUROC$_1$ COMPARISON - Netbuilder data from topology DAG$_V$ with $\sigma = 0.01$

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9887	0.0114	-	0.0259	0.0002
GGM	0.9567	0.0294	0.0259	-	0.0024
RN	0.8513	0.0188	0.0002	0.0024	-
DGE - Observational					
BN	0.9674	0.0124	-	0.0002	0.0000
GGM	0.8788	0.0244	0.0002	-	0.0025
RN	0.7915	0.0156	0.0000	0.0025	-
UGE - Interventional					
BN	0.9927	0.0085	-	0.0019	0.0000
GGM	0.8277	0.0565	0.0019	-	0.0395
RN	0.7483	0.0257	0.0000	0.0395	-
DGE - Interventional					
BN	0.9944	0.0040	-	0.0002	0.0000
GGM	0.7547	0.0436	0.0002	-	0.0390
RN	0.6931	0.0200	0.0000	0.0390	-

Table I.5.: AUROC$_1$ COMPARISON - Netbuilder data from topology DAG$_V$ with $\sigma = 0.1$

Method	μ[AUROC$_1$]	σ(AUROC$_1$)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9332	0.0454	-	0.0437	0.0020
GGM	0.9038	0.0562	0.0437	-	0.2289
RN	0.9154	0.0460	0.0020	0.2289	-
DGE - Observational					
BN	0.8745	0.0452	-	0.0888	0.0931
GGM	0.8350	0.0466	0.0888	-	0.2135
RN	0.8447	0.0381	0.0931	0.2135	-
UGE - Interventional					
BN	0.9788	0.0090	-	0.0163	0.0018
GGM	0.8677	0.0630	0.0163	-	0.2972
RN	0.8214	0.0474	0.0018	0.2972	-
DGE - Interventional					
BN	0.9393	0.0406	-	0.0047	0.0013
GGM	0.7861	0.0489	0.0047	-	0.2943
RN	0.7500	0.0368	0.0013	0.2943	-

Table I.6.: AUROC$_1$ COMPARISON - Netbuilder data from topology DAG$_V$ with $\sigma = 0.3$

J. Appendix X

This tenth appendix continues with the little network diagnostic of Section 5.5. For both graph topologies $G(P_i, C)$ and G_{C,P_i} the originally defined linear functional relationships, were transformed using the hyperbolic tangent function.

From Figure J.1 can be seen that the hyperbolic tangent transformation of the functional relationships in topology $G(P_i, C)$ in which the three parent nodes are stochastically independent does not cause any problems. Both trace plots on the left have the same shape like the corresponding plots obtained for $G(P_i, C)$ with linear functional relationships. But for the second network topology G_{C,P_i} the transformation makes learning much more difficult for all three methods under comparison. For low noise levels $\sigma < 0.1$ the mean AUROC_1 scores are lower than 0.5 indicating that false relationships (between the three child nodes C_i) are extracted from the data. This is due to the fact that the hyperbolic tangent transformation weakens the linear association between P and its three child nodes C_i, while the strength of the linear association between the three children C_1, C_2, and C_3 more or less stays the same.
For all different noise levels σ all three methods reach approximately the same AUROC means, that is there is no difference in performance between the three machine learning methods. For all three methods the highest AUROC means are obtained for medium noises, while for high as well as low noise levels the true edges can not be learnt. Either there are indirect associations between the child nodes which are stronger than the true direct associations (for small noise levels) or no association at all can be found in the data (for too high noise levels).

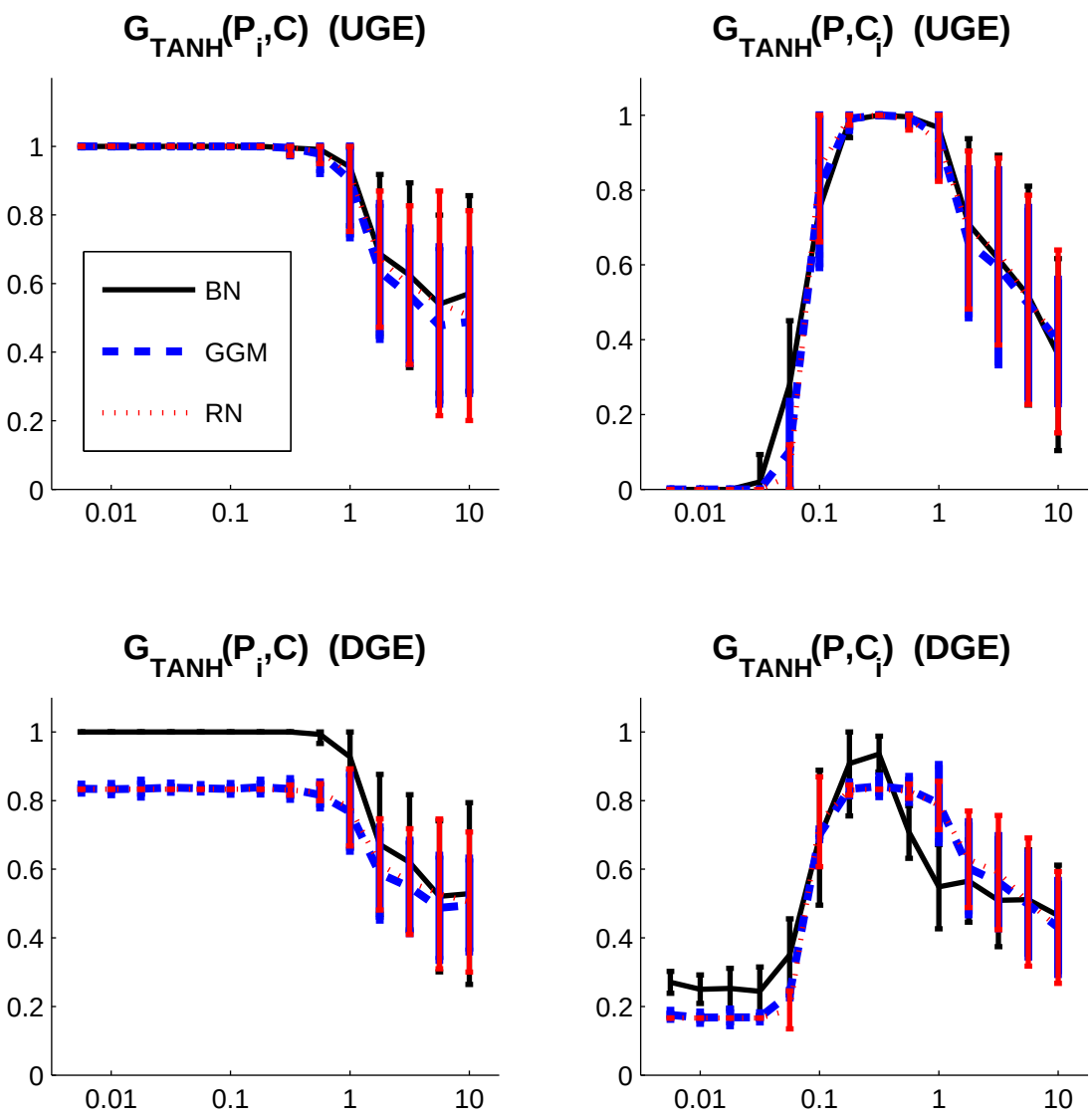


Figure J.1.: *Continuation of little network diagnostic.* Trace plots of the AUROC_1 means with error bars representing the AUROC_1 standard deviations for the three methods under comparison: Bayesian networks (black line), Gaussian graphical models (blue line), and Relevance networks (red line). For each of 14 different noise levels (σ) 25 test data sets were generated using two simple graph topologies with 4 nodes each. In contrast to the relationships considered before here the deterministic linear part was transformed by T(x) = tanh(3x), whereby tanh(.) is the hyperbolic tangent function. In the first topology $G(P_i, C)$ (left panels) where three parent nodes P_i have a common child node C this leads to the relationship: $C = tanh\,(P_1 + P_2 + P_3) + N(0, \sigma^2)$, and in the second topology G_{C,P_i} (right panels) where one parent node P has three child nodes C_i this transformation yields: $C_i = tanh\,(P) + N(0, \sigma^2)$. For both networks and all three methods the UGE (top) as well as the DGE (bottom) figures of merit were computed.

K. Appendix XI

This eleventh appendix provides nine tables which summarise and cross-compare the performances of the three machine learning methods under comparion in terms of the true positive (TP) counts obtained when accepting 5 false positive (FP) counts. As before in each table mutiple rows indicate the four combinations of figure of merit (UGE and DGE) and data set type (observational and interventional). For each of these four combinations and for each of the three methods (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance Networks (RN)) the mean μ[TP] and the the standard deviations σ(TP) of the five true positive (TP) counts obtained for 5 false positive (FP) counts can be found in the first columns. The last three columns provide one-sample t-test p-values p(.) for the hypthesis: H_0: $\mu[\mathrm{TP}(M_i)] = \mu[\mathrm{TP}(M_j)]$ against its two-sided alternative: H_1: $\mu[\mathrm{TP}(M_i)] \neq \mu[\mathrm{TP}(M_j)]$ given the combination indicated in the multiple row above. M_i and M_j represent the methods mentioned in the row and column. Low p-values p(.) indicate that there may be a significant difference in the number of true positive (TP) counts between these two methods for the particular combination of figure of merit and data set type. In these cases it can be seen from the entries in the mean score column μ[TP] which of the two methods performed better than the other one. In contrast to the $AUROC_1$ score cross-method comparison, this comparison focuses on a fixed inverse specificty point of the ROC curves. Since no correction for multiple testing was applied either, the p-values can be interpreted as descriptive measures only, and can not be used to confirm H_1 statistically.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.8	2.1	-	0.1662	0.0010
GGM	14.8	2.7	0.1662	-	0.0012
RN	8.1	2.5	0.0010	0.0012	-
DGE - Observational					
BN	4.9	1.5	-	0.6885	0.0042
GGM	4.7	1.1	0.6885	-	0.0705
RN	3.8	1.3	0.0042	0.0705	-
UGE - Interventional					
BN	18.5	2.4	-	0.0074	0.0028
GGM	13.2	2.0	0.0074	-	0.0011
RN	6.5	2.7	0.0028	0.0011	-
DGE - Interventional					
BN	18.4	2.6	-	0.0005	0.0005
GGM	5.2	0.7	0.0005	-	0.0036
RN	1.8	1.3	0.0005	0.0036	-

Table K.1.: TP COUNTS COMPARISON FOR THE GAUSSIAN DATA SETS USING DAG_O

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.6	0.5	-	0.0270	0.0000
GGM	11.8	2.7	0.0270	-	0.0024
RN	5.8	1.4	0.0000	0.0024	-
DGE - Observational					
BN	11.3	1.2	-	0.0000	0.0001
GGM	3.8	1.0	0.0000	-	0.1951
RN	3.0	0.6	0.0001	0.1951	-
UGE - Interventional					
BN	16.0	0.0	-	0.0025	0.0001
GGM	12.9	1.0	0.0025	-	0.0054
RN	7.1	1.3	0.0001	0.0054	-
DGE - Interventional					
BN	15.8	0.4	-	0.0000	0.0000
GGM	5.5	0.0	0.0000	-	0.0008
RN	3.7	0.4	0.0000	0.0008	-

Table K.2.: TP COUNTS COMPARISON FOR THE GAUSSIAN DATA SETS USING DAG_V

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	9.5	2.0	-	0.7489	0.7174
GGM	9.6	1.6	0.7489	-	0.3046
RN	9.3	1.6	0.7174	0.3046	-
DGE - Observational					
BN	3.3	2.3	-	0.1369	0.1369
GGM	5.1	0.9	0.1369	-	NaN
RN	5.1	0.9	0.1369	NaN	-
UGE - Interventional					
BN	11.1	1.3	-	0.0951	0.0099
GGM	9.6	1.1	0.0951	-	0.0204
RN	7.1	1.1	0.0099	0.0204	-
DGE - Interventional					
BN	6.9	1.1	-	0.0065	0.0009
GGM	4.1	1.1	0.0065	-	0.0093
RN	1.7	0.4	0.0009	0.0093	-

Table K.3.: TP COUNTS COMPARISON FOR THE REAL CYTOMETRIC EXPRESSION DATA

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	11.0	2.0	-	0.2577	0.0366
GGM	12.0	1.2	0.2577	-	0.0040
RN	6.9	1.4	0.0366	0.0040	-
DGE - Observational					
BN	2.8	1.3	-	0.0077	0.0890
GGM	5.1	0.7	0.0077	-	0.0016
RN	0.8	0.8	0.0890	0.0016	-
UGE - Interventional					
BN	7.9	0.7	-	0.0008	0.0000
GGM	5.2	0.3	0.0008	-	0.0000
RN	2.0	0.0	0.0000	0.0000	-
DGE - Interventional					
BN	8.4	1.2	-	0.0019	0.0001
GGM	3.7	0.4	0.0019	-	0.0001
RN	0.0	0.0	0.0001	0.0001	-

Table K.4.: TP COUNTS COMPARISON FOR THE NETBUILDER DATA SETS USING DAG_O AND THE LOW NOISE LEVEL ($\sigma = 0.01$)

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	18.1	1.1	-	0.0161	0.1216
GGM	16.5	1.1	0.0161	-	0.5291
RN	16.8	1.6	0.1216	0.5291	-
DGE - Observational					
BN	7.2	1.5	-	0.0673	0.0673
GGM	5.5	0.0	0.0673	-	NaN
RN	5.5	0.0	0.0673	NaN	-
UGE - Interventional					
BN	17.7	0.7	-	0.0046	0.0003
GGM	13.6	1.5	0.0046	-	0.0002
RN	8.0	1.7	0.0003	0.0002	-
DGE - Interventional					
BN	17.3	0.7	-	0.0000	0.0000
GGM	5.4	0.2	0.0000	-	0.0000
RN	1.2	0.7	0.0000	0.0000	-

Table K.5.: TP COUNTS COMPARISON FOR THE NETBUILDER DATA SETS USING DAG_O AND THE MEDIUM NOISE LEVEL ($\sigma = 0.1$)

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.5	1.7	-	0.4468	0.2756
GGM	14.8	2.9	0.4468	-	0.0213
RN	16.6	2.3	0.2756	0.0213	-
DGE - Observational					
BN	4.1	2.0	-	0.5158	0.3844
GGM	4.7	1.1	0.5158	-	0.3739
RN	5.1	0.9	0.3844	0.3739	-
UGE - Interventional					
BN	16.0	1.6	-	0.0890	0.0143
GGM	14.5	1.5	0.0890	-	0.3672
RN	13.6	1.5	0.0143	0.3672	-
DGE - Interventional					
BN	14.1	4.5	-	0.0052	0.0073
GGM	5.5	0.0	0.0052	-	0.3739
RN	5.0	1.1	0.0073	0.3739	-

Table K.6.: TP COUNTS CROSS-METHOD COMPARISON FOR THE NETBUILDER DATA SETS USING DAG_O AND THE HIGH NOISE LEVEL ($\sigma = 0.3$)

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	9.8	0.8	-	1.0000	0.0007
GGM	9.8	0.8	1.0000	-	0.0001
RN	5.2	0.3	0.0007	0.0001	-
DGE - Observational					
BN	3.9	0.7	-	0.3419	0.0019
GGM	4.5	0.9	0.3419	-	0.0020
RN	1.5	0.0	0.0019	0.0020	-
UGE - Interventional					
BN	7.2	0.4	-	0.4263	0.0102
GGM	7.5	0.4	0.4263	-	0.0078
RN	5.1	0.9	0.0102	0.0078	-
DGE - Interventional					
BN	6.6	0.4	-	0.0001	0.0000
GGM	3.4	0.2	0.0001	-	0.0002
RN	2.0	0.0	0.0000	0.0002	-

Table K.7.: TP COUNTS COMPARISON FOR THE NETBUILDER DATA SETS USING DAG_V AND THE LOW NOISE LEVEL ($\sigma = 0.01$)

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.6	0.4	-	0.0876	0.0000
GGM	14.7	1.0	0.0876	-	0.0001
RN	9.3	0.4	0.0000	0.0001	-
DGE - Observational					
BN	12.0	1.5	-	0.0006	0.0007
GGM	5.5	0.0	0.0006	-	0.1079
RN	4.8	0.8	0.0007	0.1079	-
UGE - Interventional					
BN	15.7	0.4	-	0.0002	0.0005
GGM	12.5	0.5	0.0002	-	0.0021
RN	8.9	1.0	0.0005	0.0021	-
DGE - Interventional					
BN	15.4	0.7	-	0.0000	0.0000
GGM	4.9	0.8	0.0000	-	0.1302
RN	3.8	1.0	0.0000	0.1302	-

Table K.8.: TP COUNTS COMPARISON FOR THE NETBUILDER DATA SETS USING DAG_V AND THE MEDIUM NOISE LEVEL ($\sigma = 0.1$)

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	14.2	1.1	-	0.0474	0.1087
GGM	13.2	1.0	0.0474	-	0.3375
RN	13.6	0.8	0.1087	0.3375	-
DGE - Observational					
BN	7.7	2.0	-	0.0714	0.0440
GGM	5.5	0.0	0.0714	-	0.2663
RN	5.0	0.9	0.0440	0.2663	-
UGE - Interventional					
BN	14.9	0.2	-	0.0093	0.0277
GGM	12.5	1.1	0.0093	-	0.5913
RN	12.8	1.4	0.0277	0.5913	-
DGE - Interventional					
BN	12.3	1.7	-	0.0009	0.0009
GGM	5.5	0	0.0009	-	NA
RN	5.5	0	0.0009	NA	-

Table K.9.: TP COUNTS COMPARISON FOR THE NETBUILDER DATA SETS USING DAG_V AND THE HIGH NOISE LEVEL ($\sigma = 0.01$)

L. Appendix XII

This twelfth appendix provides tables which summarise all AUROC scores that were presented in scatter plots in Section 5.7. For all different kinds of test data sets a table with the means $\mu[.]$ and standard deviations $\sigma(.)$ for both stochastic Bayesian network models BGe and BDe as well as some p-values p(.) of two-sided one-sample Student t-tests are given. One sample t-tests were used to test, whether the AUROC means are different or not. More precisely, the null hypothesis

$$\mathrm{H}_0\colon\ \mu[\mathrm{BGe}] = \mu[\mathrm{BDe}]$$

was tested against the corresponding alternative hypothesis

$$\mathrm{H}_1\colon\ \mu[\mathrm{BGe}] \neq \mu[\mathrm{BDe}].$$

Thereby no correction for multiple statistical testing was applied, so that these p-values must be considered with caution. Although they can be seen as meaningful descriptive statistics indicating, whether there may be a difference, they can *not* be used to confirm H_1 statistically.

All tables have the same structure. After multiple rows specifying parameters, such as the figure of merit (UGE and DGE), the sample size N, the noise level σ, or the network topology, there are rows which contain the means and standard deviations for both models: BGe (μ[BGe] and σ(BGe)) and BDe: (μ[BDe] and σ(BDe)). Finally, the last columns provide the p-values for the t-tests mentioned above.

design	N	μ[BGe]	σ(BGe)	μ[BDe]	σ(BDe)	p
UGE and $AUROC_1$						
observational	N=100	0.6904	0.0376	0.6620	0.0410	0.4125
interventional	N=100	0.7912	0.0335	0.7765	0.0364	0.4821
observational	N=10	0.5636	0.0373	0.5452	0.0611	0.6324
interventional	N=10	0.5943	0.0747	0.5081	0.0653	0.0857
DGE and $AUROC_1$						
observational	N=100	0.6231	0.0564	0.6251	0.0319	0.9564
interventional	N=100	0.6969	0.0676	0.6963	0.0454	0.9889
observational	N=10	0.5636	0.0373	0.5452	0.0611	0.6324
interventional	N=10	0.5943	0.0747	0.5081	0.0653	0.0857
UGE and $AUROC_{0.1}$						
observational	N=100	0.0379	0.0108	0.0348	0.0052	0.6007
interventional	N=100	0.0392	0.0037	0.0326	0.0107	0.1720
observational	N=10	0.0176	0.0062	0.0074	0.0024	0.0205
interventional	N=10	0.0120	0.0033	0.0089	0.0096	0.5857
DGE and $AUROC_{0.1}$						
observational	N=100	0.0143	0.0051	0.0193	0.0015	0.1375
interventional	N=100	0.0313	0.0041	0.0241	0.0097	0.2607
observational	N=10	0.0081	0.0052	0.0068	0.0021	0.7036
interventional	N=10	0.0099	0.0048	0.0055	0.0055	0.1977

Table L.1.: $AUROC_\epsilon$ COMPARISON between BGe and BDe - Real cytometric data

design	μ[BGe]	σ(BGe)	μ[BDe]	σ(BDe)	p
N=10 and $\sigma = 0.1$					
observational UGE	0.7909	0.0488	0.6274	0.0994	0.0108
observational DGE	0.7051	0.0262	0.6048	0.0826	0.0322
N=100 and $\sigma = 0.1$					
observational UGE	0.8848	0.0543	0.7060	0.0694	0.0019
observational DGE	0.7817	0.0711	0.6595	0.0608	0.0193
N=1000 and $\sigma = 0.1$					
observational UGE	0.9756	0.0389	0.7081	0.0672	0.0001
observational DGE	0.9025	0.0357	0.6303	0.0648	0.0000

Table L.2.: AUROC_1 COMPARISON between BGe and BDe - observational Gaussian data - DAG_O

design	μ[BGe]	σ(BGe)	μ[BDe]	σ(BDe)	p
DAG_O and $\sigma = 0.01$					
observational UGE	0.7901	0.0336	0.6089	0.1174	0.0106
observational DGE	0.6808	0.0703	0.5529	0.0698	0.0203
interventional UGE	0.7047	0.0221	0.7717	0.0883	0.1383
interventional DGE	0.8280	0.0097	0.8542	0.0636	0.3887
DAG_O and $\sigma = 0.1$					
observational UGE	0.9464	0.0273	0.8777	0.0431	0.0168
observational DGE	0.8572	0.0100	0.7840	0.0519	0.0148
interventional UGE	0.9346	0.0254	0.8595	0.0391	0.0070
interventional DGE	0.9678	0.0114	0.8753	0.0313	0.0003
DAG_O and $\sigma = 0.3$					
observational UGE	0.9049	0.0150	0.7626	0.0962	0.0114
observational DGE	0.8208	0.0223	0.7036	0.0751	0.0101
interventional UGE	0.9053	0.0367	0.8264	0.0733	0.0635
interventional DGE	0.9219	0.0408	0.8095	0.0577	0.0074

Table L.3.: AUROC_1 COMPARISON between BGe and BDe - Netbuilder data - DAG_O

design	μ[BGe]	σ(BGe)	μ[BDe]	σ(BDe)	p
DAG_V and $\sigma = 0.01$					
observational UGE	0.7845	0.0184	0.6446	0.0306	0.0000
observational DGE	0.7354	0.0467	0.6499	0.0704	0.0536
interventional UGE	0.7102	0.0156	0.8137	0.0560	0.0041
interventional DGE	0.8413	0.0052	0.8759	0.0406	0.0961
DAG_V and $\sigma = 0.1$					
observational UGE	0.9887	0.0114	0.7719	0.0620	0.0001
observational DGE	0.9674	0.0124	0.7238	0.0418	0.0000
interventional UGE	0.9927	0.0085	0.8595	0.0266	0.0000
interventional DGE	0.9944	0.0040	0.8711	0.0263	0.0000
DAG_V and $\sigma = 0.3$					
observational UGE	0.9332	0.0454	0.7150	0.0806	0.0007
observational DGE	0.8745	0.0452	0.6795	0.0668	0.0006
interventional UGE	0.9788	0.0090	0.7267	0.0662	0.0000
interventional DGE	0.9393	0.0406	0.7263	0.0534	0.0001

Table L.4.: AUROC_1 COMPARISON between BGe and BDe - Netbuilder data - DAG_V

design	μ[BGe]	σ(BGe)	μ[BDe]	σ(BDe)	p
DAG_V with OR ports and $\sigma = 0.01$					
observational UGE	0.7849	0.0178	0.6474	0.0516	0.0005
observational DGE	0.8271	0.0265	0.6858	0.0534	0.0007
interventional UGE	0.7228	0.0332	0.7966	0.0356	0.0094
interventional DGE	0.8519	0.0185	0.8502	0.0432	0.9365
DAG_V with OR ports and $\sigma = 0.1$					
observational UGE	0.8048	0.0307	0.7364	0.0374	0.0134
observational DGE	0.8344	0.0333	0.7024	0.0315	0.0002
interventional UGE	0.8537	0.0352	0.8393	0.0530	0.6256
interventional DGE	0.9305	0.0191	0.8735	0.0299	0.0071
DAG_V with OR ports and $\sigma = 0.3$					
observational UGE	0.9205	0.0144	0.7564	0.0455	0.0001
observational DGE	0.8928	0.0245	0.7127	0.0377	0.0000
interventional UGE	0.9138	0.0227	0.8224	0.0218	0.0002
interventional DGE	0.9477	0.0236	0.7884	0.0393	0.0001

Table L.5.: $AUROC_1$ COMPARISON between BGe and BDe - Netbuilder data with OR ports, whereby the transformations $x \rightarrow \frac{x}{x+1}$ were omitted

design	μ[BGe]	σ(BGe)	μ[BDe]	σ(BDe)	p
DAG_V with AND ports and $\sigma = 0.01$					
observational UGE	0.7849	0.0178	0.6474	0.0516	0.0005
observational DGE	0.8271	0.0265	0.6858	0.0534	0.0007
interventional UGE	0.7228	0.0332	0.7966	0.0356	0.0094
interventional DGE	0.8519	0.0185	0.8502	0.0432	0.9365
DAG_V with AND ports and $\sigma = 0.1$					
observational UGE	0.8048	0.0307	0.7364	0.0374	0.0134
observational DGE	0.8344	0.0333	0.7024	0.0315	0.0002
interventional UGE	0.8537	0.0352	0.8393	0.0530	0.6256
interventional DGE	0.9305	0.0191	0.8735	0.0299	0.0071
DAG_V with AND ports and $\sigma = 0.3$					
observational UGE	0.9205	0.0144	0.7564	0.0455	0.0001
observational DGE	0.8928	0.0245	0.7127	0.0377	0.0000
interventional UGE	0.9138	0.0227	0.8224	0.0218	0.0002
interventional DGE	0.9477	0.0236	0.7884	0.0393	0.0001

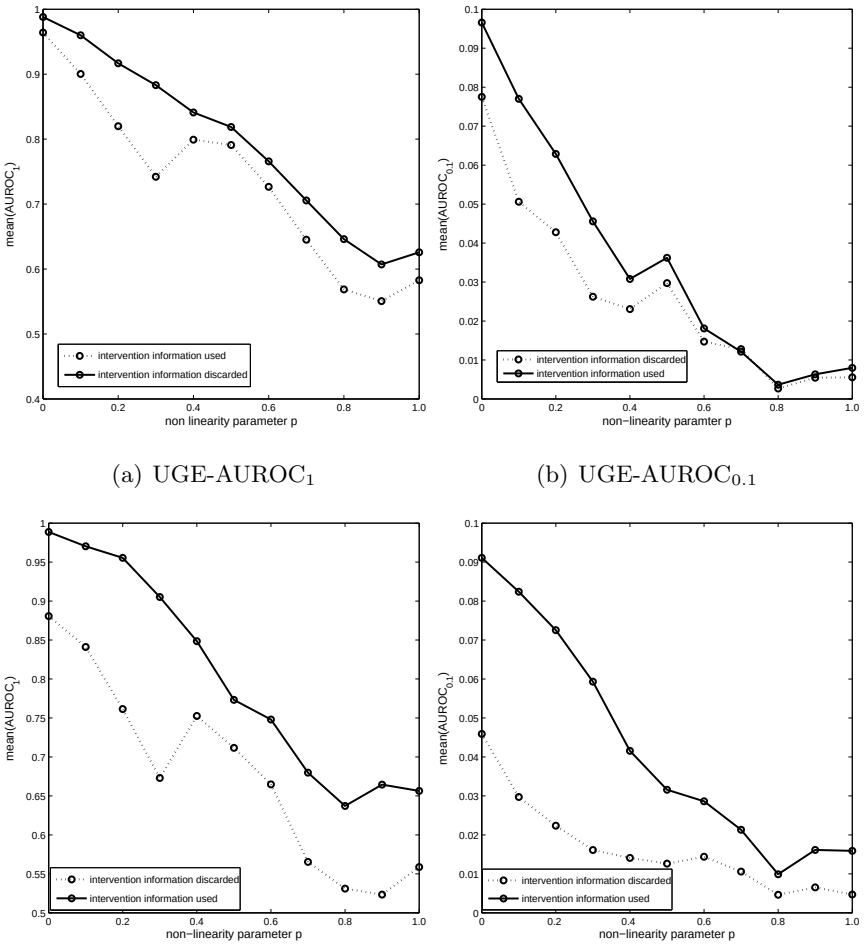
Table L.6.: $AUROC_1$ COMPARISON between BGe and BDe - Netbuilder data with AND ports, whereby the transformations $x \rightarrow \frac{x}{x+1}$ were omitted

design	μ[BGe]	σ(BGe)	μ[BDe]	σ(BDe)	p
DAG_V with XOR ports and $\sigma = 0.01$					
observational UGE	0.6327	0.0932	0.7393	0.0697	0.0749
observational DGE	0.6831	0.0627	0.7514	0.0633	0.1250
interventional UGE	0.7564	0.0077	0.8159	0.0185	0.0002
interventional DGE	0.8191	0.0302	0.8706	0.0143	0.0087
DAG_V with XOR ports and $\sigma = 0.1$					
observational UGE	0.6918	0.0633	0.8144	0.0813	0.0288
observational DGE	0.7342	0.0470	0.7673	0.0738	0.4230
interventional UGE	0.7388	0.0558	0.7816	0.0485	0.2317
interventional DGE	0.8195	0.0457	0.7991	0.0794	0.6317
DAG_V with XOR ports and $\sigma = 0.3$					
observational UGE	0.6242	0.0642	0.6575	0.0433	0.3641
observational DGE	0.6106	0.0620	0.6334	0.0382	0.5024
interventional UGE	0.6992	0.0244	0.6510	0.0576	0.1230
interventional DGE	0.6712	0.0550	0.6364	0.0603	0.3680

Table L.7.: $AUROC_1$ COMPARISON between BGe and BDe - Netbuilder data with XOR ports, whereby the transformations $x \rightarrow \frac{x}{x+1}$ were omitted

M. Appendix XIII

In this appendix the interventional data generated with the Gaussian data generator using the original cytometric network topology and different parameters of non-linearity are utilised to demonstrate that Bayesian networks especially benefit from the intervention information which corresponds to the interventional data. To this end, the mean AUROC scores of the standard interventional BGe scoring metric, which uses the intervention information as usual, were compared with an observational Bayesian network BGe scoring approach in which the intervention information was completely ignored. That is although the data sets are interventional the latter Bayesian network approach treats the interventional data as if they were pure observational data. From the trace plots of the AUROC_1 means in Figure M.1 can be seen that Bayesian networks benefit from the information for both figures of merit UGE and DGE. But it seems that especially the mean DGE figure of merit AUROCs decrease when the intervention information is discarded.



(a) UGE-AUROC$_1$ (b) UGE-AUROC$_{0.1}$

(c) DGE-AUROC$_1$ (d) DGE-AUROC$_{0.1}$

Figure M.1.: AUROC trace plots illustrating how much Bayesian networks can benefit from interventional data, that is the intervention information. For each of 11 different non-linearity parameters (p=0,0.1,...,1.0) five interventional data sets with N=100 observations were generated with the Gaussian data generator. These data sets were analysed using the BGe scoring metric. The solid lines correspond to the mean AUROC scores obtained by a Bayesian network Order-MCMC approach which used the intervention information as usual. The dotted lines correspond to the mean AUROC scores obtained by a Bayesian network Order-MCMC approach which treated the interventional data sets as if they were pure observational data.

Bibliography

[1] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J.D. Watson. *Molecular Biology of the Cell*. Garland Publishing, New York, 4th edition, 2002.

[2] P.W. Atkins. *Physical Chemistry*. Oxford University Press, Oxford, 3rd edition, 1986.

[3] G.E. Briggs and J.B.S. Haldane. A note on the kinetics of enzyme action. *Biochemical Journal*, 19:339–339, 1925.

[4] T.A. Brown. *Genetics: A Molecular Approach*. Chapman and Hall, London, 3rd edition, 1999.

[5] W.L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225, 1994.

[6] A.S. Butte and I.S. Kohane. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing*, 2000:418–429, 2000.

[7] A.S. Butte and I.S. Kohane. Relevance networks: A first step toward finding genetic regulatory networks within microarray data. In G. Parmigiani, E.S. Garett, R.A. Irizarry, and S.L. Zeger, editors, *The Analysis of Gene Expression Data*, pages 428–446. Springer, 2003.

[8] D.M. Chickering. A transformational characterization of equivalent Bayesian network structures. *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 11:87–98, 1995.

[9] D.M. Chickering. Learning equivalence classes of Bayesian network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

[10] G.F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[11] M.K. Dougherty, J. Muller, D.A. Ritt, M. Zhou, X.Z. Zhou, T.D. Copeland, T.P. Conrads, T.D. Veenstra, K.P. Lu, and D.K. Morrison. Regulation of raf-1 by direct feedback phosphorylation. *Molecular Cell*, 17:215–224, 2005.

[12] D.M. Edwards. *Introduction to Graphical Modelling*. Springer Verlag, New York, 2000.

[13] J.P. Egan. *Signal Detection Theory and ROC Analysis*. Academic Press, London, England, 1975.

[14] N. Friedman, G. Elidan, A. Regev, and D. Pe'er. Inferring sub-networks from perturbed expression profiles. *Bioinformatics*, 1:1–9, 2001.

[15] N. Friedman and D. Koller. Being Bayesian about network structure. *Machine Learning*, 50:95–126, 2003.

[16] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In Gregory F. Cooper and Serafín Moral, editors, *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 139–147, San Francisco, CA, 1998. Morgan Kaufmann Publishers.

[17] A. Gannoun, J. Saracco, W. Urfer, and G.E. Bonney. Nonparametric analysis of replicated microarray experiments. *Statistical Modelling*, 4:195–209, 2004.

[18] D. Geiger and D. Heckerman. Learning Gaussian networks. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 235–243, 1994.

[19] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, Suffolk, 1996. ISBN: 0-412-05551-1.

[20] P. Giudici and R. Castelo. Improving Markov chain Monte Carlo model search for data mining. *Machine Learning*, 50:127–158, 2003.

[21] M. Grzegorczyk and W. Urfer. Determination of interacting genes in kidney tissues using Bayesian networks. Technical report, Department of Statistics, University of Dortmund, Germany, 2004.

[22] A.J. Hartemink. *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks*. PhD thesis, MIT, 2001. http://citeseer.ist.psu.edu/hartemink01principled.html.

[23] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: Search methods and empirical results. *In Proceedings of the Fifth conference on Artificial Intelligence and Statistics*, 5:112–128, 1995.

[24] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:245–274, 1995.

[25] A.V. Hill. The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. *Journal of Physiology*, 40:4–7, 1910.

[26] D. Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19:2271–2282, 2003.

[27] F.V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, London, England, 1996.

[28] K. Jung, K. Quast, A. Gannoun, and W. Urfer. A renewed approach to the nonparametric analysis of replicated microarray experiments. *Biometrical Journal*, 48:245–254, 2006.

[29] O. Ledoit and M. Wolf. A well conditioned estimator for large-dimensional covariance matrices. *Journal of Mutivariate Analysis*, 88:365–411, 2004.

[30] R.J. Lipshutz, S.P.A. Fodor, T.R. Gingeras, and D.J. Lockhart. High density synthetic oligonucleotide arrays. *Nature Genetics*, 21:20–24, 1999.

[31] D.J. Lockhart, H. Dong, M.C. Byrne, M.T. Follettie, M.V. Gallo, M.S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E.L. Brown. Expression of monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, 1996.

[32] D. Madigan and J. York. Bayesian graphical models for discrete data. *International Statistical Review*, 63:215–232, 1995.

[33] L. Michaelis and M. Menten. Die Kinetik der Invertinwirkung. *Biochemische Zeitschrift*, 49:333–369, 1913.

[34] K.P. Murphy. An introduction to graphical models. Technical report, MIT Artificial Intelligence Laboratory, 2001. http://www.ai.mit.edu/∼murphyk/Papers/intro_gm.pdf.

[35] D.V. Nguyen, A.B. Arpat, N. Wang, and R.J. Carroll. Dna microarray experiments: Biological and technological aspects. *Biometrics*, 58:701–717, 2002.

[36] R. Opgen-Rhein and K. Strimmer. Inferring gene dependency networks from genomic longitudinal data: a functional data approach. *REVSTAT*, page to appear, 2006.

[37] W. Pan, J. Lin, and C. Le. A mixture model approach to detecting differentially expressed genes with microarray data. Technical report, Division of Biostatistics, University of Minnesota, University of Minnesota, U.S.A., 2001.

[38] M.S. Pepe, G. Longton, G.L. Anderson, and M. Schummer. Selecting differentially expressed genes from microarray experiments. *Biometrics*, 59:133–142, 2003.

[39] I.V. Pournara. *Reconstructing gene networks by passive and active Bayesian learning*. PhD thesis, Birbeck College, University of London, 2005.

[40] D.E. Rummelhart. *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 2nd edition, 1987.

[41] K. Sachs, O. Perez, D. Pe'er´, D.A. Lauffenburger, and G.P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308:523–529, 2005.

[42] J. Schäfer and K. Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.

[43] J. Schäfer and K. Strimmer. An shrinkage approach to large-scale covariance matrix estiamtion and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4, 2005b. (online journal - article 32).

[44] V.A. Smith, E.D. Jarvis, and A.J. Hartemink. Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics*, 18:S216–S224, 2002. (ISMB02 special issue).

[45] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer Verlag, New York, 2001.

[46] S. Swift, A. Tucker, and X. Liu. Evolutionary computation to search for strongly correlated variables in high-dimensional time-series. In D.J. Hand, J.N. Kok, and M.R. Berthold, editors, *The Proceedings of the Third Symposium on Intelligent Data Analysis (IDA-99)*, pages 51–61. Springer, 1999.

[47] J. Tian and J. Pearl. Active learning for structure in Bayesian networks. *Senventeenth International Joint Conference on Artificial Intelligence*, 17:863–869, 2001.

[48] O.G. Troyanskaya, M.E. Garber, P.O Brown, D. Botstein, and R.B. Altman. Nonparametric methods for identifying differentially expressed genes in microarray data. *Bioinformatics*, 18:1454–1461, 2003.

[49] T. Verma and J. Pearl. Equivalence and synthesis of causal models. *In: Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 6:220–227, 1990.

[50] L. Wernisch and I. Pournara. Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics*, 20:2934–2942, 2004.

[51] C. Yoo, V. Thorson, and G.F. Cooper. Discovery of causal relationships in a generegulation pathway from a mixture of experimental and observational DNA microarray data. *In Proceedings of PSB*, 7:498–509, 2002.

[52] C.H. Yuh, H. Bolouri, and E.H. Davidson. Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, 279:1896–1902, 1998.

[53] C.H. Yuh, H. Bolouri, and E.H. Davidson. Cis-regulatory logic in the endo16 gene: switching from a specification to a differentiation mode of control. *Development*, 128:617–629, 2001.

[54] D.E. Zak, F.J. Doyle, G.E. Gonye, and J.S. Schwaber. Simulation studies for the identification of genetic networks from cDNA array and regulatory activity data. *Proceedings of the Second International Conference on Systems Biology*, pages 231–238, 2001.