Université Libre de Bruxelles
Faculté des Sciences
Département d'Informatique

Causal Inference and Prior Integration in Bioinformatics using Information Theory

Catharina Olsen

Thèse présentée en vue de
l'obtention du grade académique
de Docteur en Sciences

Octobre 2013

This thesis has been written under the supervision of Prof. Gianluca Bontempi. The members of the jury are:

- Prof. Gianluca Bontempi (Université Libre de Bruxelles, Belgium)
- Prof. Pierre Geurts (Université de Liège, Belgium)
- Prof. Benjamin Haibe-Kains (Université de Montréal, Canada)
- Prof. Maarten Jansen (Université Libre de Bruxelles, Belgium)
- Prof. Tom Lenaerts (Université Libre de Bruxelles, Belgium)
- Dr. Patrick E. Meyer (Université Libre de Bruxelles, Belgium)
- Prof. John Quackenbush (Harvard University, USA)

Declaration

This thesis has been composed by the author herself and contains original work of her own execution. Some of the reported work has been done in collaboration with a number of co-authors whose contributions are acknowledged in the relevant sections.

To my family

Acknowledgements

One of the aspects of research I enjoy the most is the possibility to collaborate with many people from all over the world. In the following I would like to express my gratitude to those that have contributed in one way or another to the process of writing this thesis.

First and foremost I would like to thank my supervisor Gianluca: for offering me the possibility to write my thesis in MLG, for giving me the freedom to follow my research interests and finally for making sure that I could spend some of my time abroad. At this point, I would like to thank John for inviting me to visit his group for a research internship and even though he has a crazy schedule finding the time for several fruitful discussions. During this stay I enjoyed immensely the collaboration with Benjamin, his motivation, focus on his research and openness to collaborative work are to this day a great inspiration for me.

A great thanks also to Patrick who introduced me to the topic of network inference in the very beginning. Furthermore to all the other researchers I had the pleasure to collaborate with during the years of my thesis. Then I would like to thank several people from MLG who helped me improve different versions of this thesis: Yann-Aël, Miguel and Elisa.

Thanks to the members of my jury who kindly accepted to read and comment this work, namely Prof. Pierre Geurts (Université de Liège), Prof. Benjamin Haibe-Kains (Université de Montréal), Prof. Maarten Jansen (ULB), Prof. Tom Lenaerts (ULB), Dr. Patrick Meyer (ULB) and Prof. John Quackenbush (Harvard University).

Then I would like my friends and colleagues both from MLG and the Computer Science Department (former and current) for making every day at university such an enjoyable experience.

And finally, I wish to convey special thanks to my family and my friends for helping and supporting me in so many ways throughout these years.

Abstract

An important problem in bioinformatics is the reconstruction of gene regulatory networks from expression data. The analysis of genomic data stemming from high-throughput technologies such as microarray experiments or RNA-sequencing faces several difficulties. The first major issue is the high variable to sample ratio which is due to a number of factors: Firstly, a single experiment captures all genes while the number of experiments is restricted by the experiment's cost, time and patient cohort size. Secondly, these data sets typically exhibit high amounts of noise. In recent years many different network inference algorithms were developed, these include Gaussian graphical models, feature selection strategies based on the computation of pairwise interaction scores and finally techniques inferring Bayesian networks. Methods belonging to the first two classes were developed to deal with large variable to sample ratios but mainly infer undirected networks. Whereas more complex techniques such as those belonging to the third class were designed to infer directed networks but usually require data sets with fewer variables than typically collected in expression data.

Another important problem in bioinformatics is the question of how the inferred networks' quality can be evaluated. The current best practice is a two step procedure. In the first step, the highest scoring interactions are compared to known interactions stored in biological databases. The inferred networks passes this quality assessment if there is a large overlap with the known interactions. In this case, a second step is carried out in which unknown but high scoring and thus promising new interactions are validated 'by hand' via laboratory experiments. Unfortunately when integrating prior knowledge in the inference procedure, this validation procedure would be biased by using the same information in both the inference and the validation. Therefore, it would no longer allow an independent validation of the resulting network.

The main contribution of this thesis is a complete computational framework that uses experimental knock down data in a cross-validation scheme to both infer and validate directed networks. Its components are i) a method that integrates genomic data and prior knowledge to infer directed networks, ii) its implementation in an R/Bioconductor package and iii) a web application to retrieve prior knowledge from PubMed abstracts and biological databases. To infer directed networks from genomic data and prior knowledge, we propose a two step procedure: First, we adapt the pairwise feature selection strategy *mRMR* to integrate prior knowledge in order to obtain the network's skeleton. Then for the subsequent orientation phase of the algorithm, we extend a criterion based on interaction information to include prior knowledge. The implementation of this method is available both as part of the prior retrieval tool *Predictive Networks* and as a stand-alone R/Bioconductor package named *predictionet*. Furthermore, we propose a fully data-driven quantitative validation of such directed networks using experimental knock-down data: Firstly, we use statistical tests to identify the set of genes that were truly affected by the perturbation experiment. The rationale behind our validation strat-

egy is to then consider these affected genes as gold-standard for any inferred network in the sense that these genes should be inferred as part of the perturbed gene's childhood. Consequently, we can compute a performance score for an inferred network based on how many truly affected genes are inferred as part of the childhood of the perturbed gene. We complete this part of the thesis using experimental knock-down data from colorectal cancer cell lines to show the quality of the retrieved prior knowledge and the benefit that integrating prior knowledge has on the inferred networks' quality. Furthermore, we use human tumor data to show that the experiments carried out on cell lines can be translated to collected observational patient data.

One shortcoming of pairwise feature selection strategies is their sensitivity to small changes in the data sets which might lead to significant changes in the set of selected variables. To overcome this problem we propose an ensemble feature selection strategy extending the classic mRMR strategy to ultimately improve the robustness of the feature selection step in network inference methods. The standard orientation scheme based on interaction information only orients those triplets with negative scores. We propose an improved orientation scheme to orient a maximum number of edges by also exploiting triplets with positive scores. The entropy estimation step is the bottleneck of methods using information theoretic measures to compute dependencies between the variables. Many estimators are too computationally demanding to be applied to high variable low sample data sets. In the final part of this thesis we compare experimentally the influence of entropy estimation on the quality of the inferred networks.

Résumé

Dans cette thèse nous présentons des approches pour la résolution de deux problèmes importants en bioinformatique. Le premier problème concerne l'inférence de réseaux de gènes à partir de plusieurs sources. D'une part, en utilisant l'information se trouvant dans la littérature et d'autre part, en utilisant des données d'expression génique. Dans le domaine de l'inférence de réseaux, l'approche de modélisation générale utilise des graphes dans lesquels les nœuds correspondent à des gènes, et les arcs représentent la régulation entre les gènes. L'inférence des réseaux géniques à partir de données génétiques est difficile en raison du grand nombre de variables et du petit nombre d'échantillons de données. Pour inférer ces réseaux, en d'autres termes la dépendance entre les variables, nous adaptons des méthodes basées sur des mesures issues de la théorie de l'information, plus précisément l'information mutuelle et l'information d'interaction, pour qu'elles puissent combiner plusieurs sources de données.

Le deuxième problème que nous abordons dans cette thèse est la validation quantitative de ces réseaux. Dans l'état de l'art, les interactions inférées sont comparées avec les interactions connues. Parce que nous utilisons ces interactions dans le processus d'inférence des réseaux afin de réduire sa variabilité, elles ne peuvent être aussi utiliser pour une validation indépendante. Pour cette raison, nous utilisons des données du type knock-down qui permettent d'étudier l'effet direct de chaque gène sur l'ensemble des autres gènes. Pour ces expériences un nombre des gènes ont fait l'objet de perturbations par RNAi (en anglais : RNA interference), une technologie permettant de forcer l'expression d'un gène à zéro. Pour obtenir un score de performance, il suffit de comparer cette liste des gènes concernés par les expériences knock-down avec les effets de chaque gène dans le réseau inféré.

Cette thèse a mené au développement de plusieurs outils informatiques : i) Un outil qui permet de collecter l'information se trouvant dans la littérature implémenté par Entagen et ii) une librairie qui permet l'inférence des réseaux orientés à partir de cette information et des données génétiques, implémentée dans R/Bioconductor et intégrée dans le premier outil.

Contents

Chapter 1

Introduction

In this thesis we use machine learning techniques to design and assess a complete framework in which we i) infer directed networks combining genomic data and prior knowledge and ii) quantitatively validate these networks making use of experimental perturbation data. We apply these techniques to colorectal cancer data in order to better understand the underlying mechanisms governing this disease.

There have been many cornerstones on the way to the current massive generation of genomic data: the first isolation of deoxyribonucleic acid (DNA) in the late 19th century, the deciphering of its structure by Watson and Crick in the middle of the 20th century, the first sequencing of DNA some decades later and the first (almost) complete sequencing of a human genome in the early years of the current century. Nowadays, full genome sequencing is achieved using high-throughput techniques.

Amongst the high-throughput technologies developed in the last twenty years are *microarray experiments* and *RNA-sequencing*. They allow the measuring of gene expressions for tens of thousands of genes in a single experiment. From this data, researchers hope to gain a better understanding of various diseases with a major focus on cancer research.

In recent years researchers have come to the conclusion that diseases are not guided by a specific gene but rather by networks of genes [BGL11]. As these networks play such an important role in the development of a specific disease, they can help us reach a deeper understanding of the studied disease and thus facilitate the search for new targeted treatments. A consequence of understanding the importance of gene networks is that drugs are increasingly designed to not only intervene on a single gene but on multiple targets. Therefore, these networks are the key to identifying the set of genes that need to be targeted to most effectively treat the studied disease. Furthermore these networks allow the identification of other binding partners for these treatments and thus let us understand and possibly avoid the occurrence of unwanted side-effects [BGL11, TB07].

As promising as it is to infer such networks from expression data sets, these data sets are difficult to analyze with standard statistical techniques due to primarily their large variable to sample ratio – possibly tens of thousands of variables and only up to a few hundreds of samples. Furthermore, these data tend to contain high amounts of noise partially due to the inherent variability in gene expression and partially due to experimental noise [CWK+02, KEBC05].

In this thesis we develop tools to infer directed networks from genomic data. We base our inference algorithms on information theoretic measures such as mutual information and interaction information.

Our main contributions are i) an algorithm that overcomes the difficulties posed by these data sets via the integration of prior knowledge in the inference process and ii) a completely data-driven framework to validate these interactions focussing on colorectal cancer data sets. We developed our inference and validation framework in collaboration with the Computational Biology and Function Genomics Laboratory, Dana-Farber Cancer Institute, Harvard School of Public Health[1] and Entagen[2] as part of the EUREKA project *Predictive Network Refinement Through Perturbation*. The two principal components of this project and also of the resulting implementation were the retrieval of known gene-gene interactions from biological databases and PubMed abstracts and the subsequent integration of this knowledge in a network inference algorithm.

Our final contributions in this thesis include i) an ensemble version of the feature selection strategy *minimum redundancy maximum relevance* (mRMR) which allows to infer more robust networks; ii) an improved orientation algorithm based on interaction information which is able to orient more edges in a network than the standard approach and iii) an experimental study investigating the influence of entropy estimation techniques on the quality of inferred networks.

1.1 Colorectal cancer

Responsible for about 13% of all deaths worldwide in 2008[3], cancer is considered one of the leading causes of death. After lung and prostate cancer, *colorectal cancer* is the third most common cancer in the world with an estimated 1.24 million diagnosed people in 2008[4]. It is the fourth most common cause of cancer death worldwide[5].

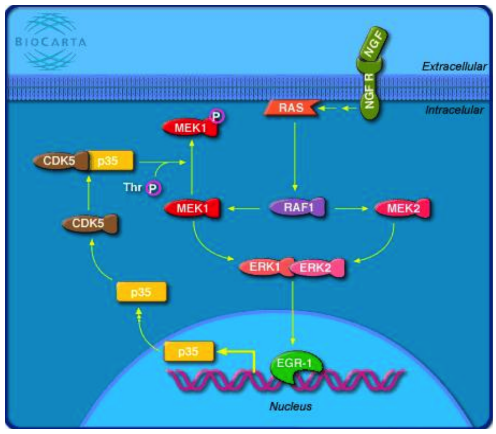
Originally, cancer that began in the tissues of the colon was called *colon cancer* and cancer that started in the rectum was called *rectal cancer* and the two types together

[1] http://compbio.dfci.harvard.edu/
[2] Entagen, Newburyport, MA, 01950, USA
[3] http://www.who.int/mediacentre/factsheets/fs297/en/
[4] http://globocan.iarc.fr/
[5] http://www.cancerresearchuk.org/cancer-info/cancerstats/world/colorectal-cancer-world/

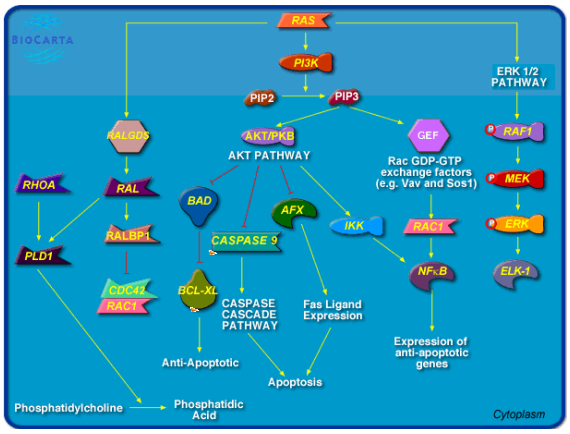
were named *colorectal cancer*[1]. Recent studies have shown that colon and rectal cancer are genetically the same cancer [Net12].

Partially due to increasing screenings for colorectal cancer, the number of mortalities has started to decline. In the USA for example, in 1975 approximately 60 new cases were diagnosed per 100000 people and the mortality rate was at approximately 28 deaths per 100000 people. In 2007 the incidence rate had decreased to 45 new cases and the mortality rate to 17 deaths per 100000 people[2]. Whilst early detection has been instrumental in reducing the number of mortalities, identifying key genes and their interactions will allow us to understand this disease's underlying mechanisms and thus to develop targeted treatments.

The RAS genes HRAS, KRAS and NRAS have been identified as important players in a variety of tumors including therein colorectal cancer [Bos89]. This makes the *RAS pathway* an ideal candidate for further research. We describe pathways and more generally *gene regulatory networks* in Section 1.2.2. In Figure 1.1 we present two versions of the RAS pathway. The first image (Figure 1.1(a)) displays the known gene interactions as



(a) RAS pathway as in BioCarta 2007



(b) RAS pathway as in BioCarta 2010

Figure 1.1: Evolution of knowledge about the RAS pathway between 2007 and 2010 available for download from BioCarta: both the involved genes and the interactions have changed greatly

of 2007 and the second (Figure 1.1(b)) those known as of 2010. The comparison of these figures shows the quick evolution of knowledge indicating a high amount of research focused on this pathway and thus implicitly also its importance.

We start the following section by explaining some basic biology concepts necessary to understand the experimental parts of this thesis. Then we describe the process of collecting

[1] http://www.cancer.gov/cancertopics/types/colon-and-rectal
[2] http://www.cancer.gov/cancertopics/factsheet/cancer-advances-in-focus/colorectal

expression data via microarray experiments and RNA-sequencing.

1.2 Expression data

The basis for all living organisms is their DNA which consists of nucleic acid/nucleotide sequences arranged in a double-helix structure. A gene corresponds to a specific sequence of the DNA. Three nucleotides, i.e. a codon, code for an amino acid. Certain codons do not correspond to an amino acid but instead mark the end of a gene.

Genes encode proteins which are the functional and structural units in the cells. The *central dogma of molecular biology* describes this encoding process. In the first step, the *transcription*, DNA is copied into messenger RNA (mRNA). In the second step, the *translation*, the information in the mRNA is used to synthesize proteins. Gene expression measurements quantify the amount of transcribed mRNA in a biological system. Until a few years ago, most of these measurements were collected through microarray experiments.

In a first step, the probes[1] are synthesized onto the array surface. These probes serve as binding sites for the complementary DNA (cDNA) extracted from the tissue of interest, see Figure 1.2. Each of these cDNAs is fluorescently labeled before hybridization. During

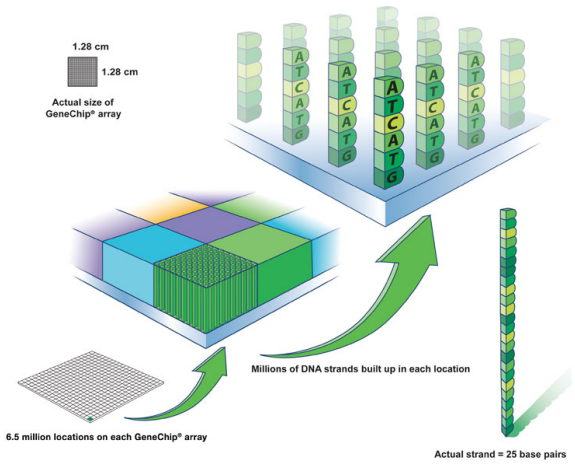
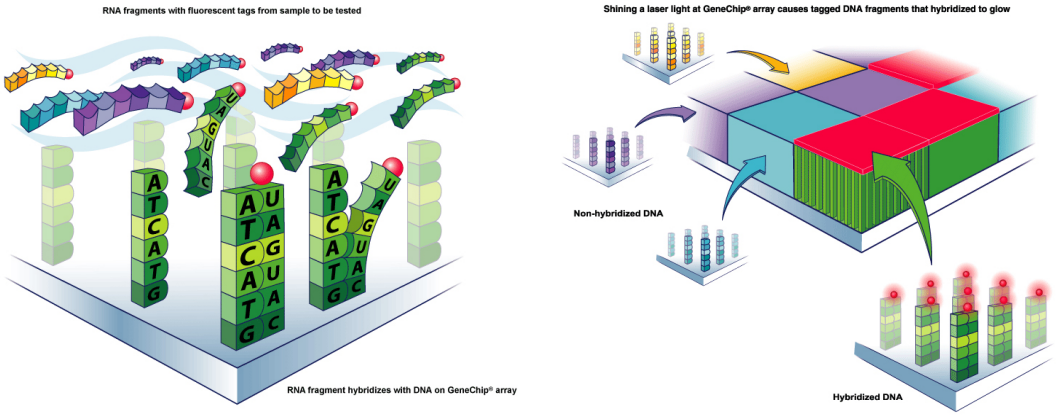


Figure 1.2: Preparation of microarray (Courtesy of Affymetrix)

hybridization (Figure 1.3(a)) the cDNA attaches itself to the probe strands. The greater the number of samples that binds to the probes present in a specific spot, the higher is the concentration of labels in this spot. During the scanning phase (Figure 1.3(b)), a laser excites the dye and the emission is measured. The higher the number of labeled

[1] In oglinucleotide microarrays such as those manufactured by Affymetrix: short sequences matching parts of the sequence of known or predicted gene coding regions.

probes bound to a spot on the array, the higher the intensity.



(a) Hybridization (Courtesy of Affymetrix) (b) Scanning (Courtesy of Affymetrix)

Figure 1.3: Microarray technology: hybridization and scanning phase

A more recently developed technique to generate gene expression data is RNA-sequencing. The workflow of this technology is outlined in Figure 1.4 (taken from [WLB10]). The main steps are the same for the different available platforms such as Roche 454, Illumina or Solid. Therefore the explanations here will only focus on the general idea rather than on a given platform's specific details which would be out of the scope of this thesis. In a first step, the sample mRNA is transformed into fragmented cDNA pieces either by a) RNA fragmentation followed by reverse transcription or by b) reverse transcription followed by cDNA fragmentation. In the second step, adaptors are added to the cDNA fragments. The fragments are then attached to the surface used by the sequencing platform of choice. As the sequencing machines cannot detect single fragments, these are amplified and then sequenced.

The reads are then mapped back to the reference genome assigning each read to an exon[1] or a gene using unspliced read alignment or spliced alignments. The expression data is generated based on the read counts from the sequencing step.

Both microarray technology and next-generation sequencing measure the expression values of a larger number of genes simultaneously for each experiment. However the number of samples is very low, due to several factors: 1) When studying a specific disease/problem the number of patients might be low. 2) The high price of an experiment still prohibits carrying out a higher number of experiments. As discussed before, microarray experiments lead to noisy data sets. Even though next-generation sequencing

[1]The subsequences of the primary transcript that are eliminated in order to form the mRNA do not actually code for anything and are called *introns*. The regions that will be used to build the gene product are called *exons*. [Dra03]

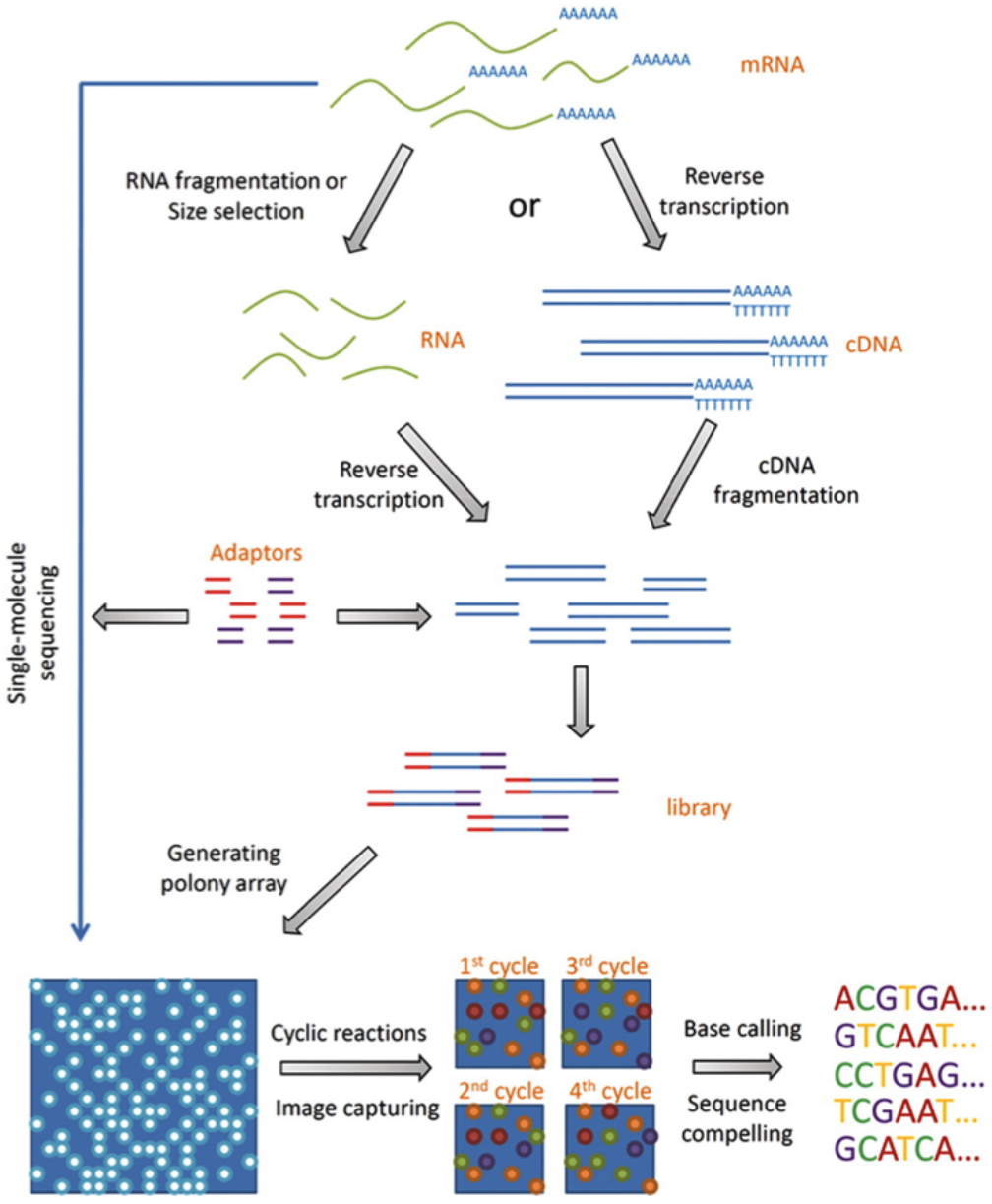


Figure 1.4: RNA-sequencing workflow, figure taken from [WLB10]

technology is supposed to deliver more accurate data, currently the cost of experiments with the same accuracy of microarrays is still very high [WLB10]. Further advantages of RNA-sequencing experiments are i) their ability to identify transcripts that have not been previously annotated and ii) their capability to quantify very low and very high expression values which is not possible in microarray experiments due to background noise interference and hybridization saturation, respectively [SKCR12].

1.2.1 Knock-down data

Whenever biologists try to understand which genes are responsible for other genes' expression or suppression they rely on *perturbation data*. Each sample in such a data set is generated by carrying out a *knock-down experiment* in which a single gene is silenced. This then allows us to study the influence this gene has on the remaining genes and thus to identify their causal relationships.

The silencing is achieved carrying out *RNA interference (RNAi)* [MM05] also known as *post transcriptional gene silencing*. The expression of the manipulated gene may not be completely removed and hence RNAi differs from knockout[1] experiments in which the removal of the targeted gene's expression is complete.

RNAi works by preventing the translation of messenger RNA (mRNA) which was produced during the transcription phase, taking advantage of the fact that without the mRNA the associated gene is basically shut down. After introducing double stranded RNA that will trigger the RNA interference, they will be cut into small fragments by a *dicer*. These fragments are then used to identify the mRNA that matches the fragments. The subsequent destruction is achieved by slicing the mRNA.

1.2.2 Transcriptional regulatory networks

The activities of cells depend on multiple hierarchical networks representing metabolic interactions, protein-protein interactions and gene interactions [BdlFM02]. These different networks and the interactions between them are presented in Figure 1.5. In this figure we observe that genes do not interact directly with each other but via proteins and metabolites.

However, expression data is only a collection of the genes' expression values. Therefore, making the simplifying assumptions that genes interact directly with other genes, these indirect gene-gene relationships are represented by so-called *gene regulatory networks*, also known as *transcriptional regulatory networks*. An example of such a network is the RAS pathway in Figure 1.1.

[1] Term for the generation of a mutant organism in which the function of a particular gene has been completely eliminated (`www.biology-online.org`)

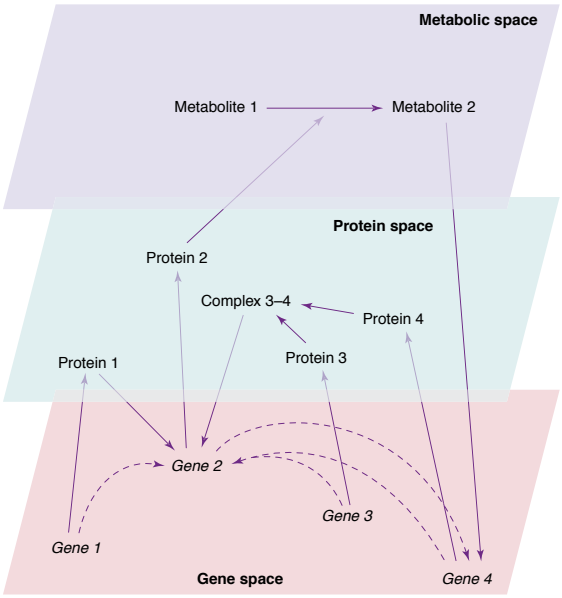


Figure 1.5: An example of a biochemical network. Molecular constituents (nodes of the network) are organized in three levels (spaces): mRNAs, proteins and metabolites. Solid arrows indicate interactions, the signs of which are not specified in the diagram. Figure and caption from [BdlFM02].

Thus, under the assumptions that i) genes directly affect each other and ii) gene expression is independent of protein and metabolite levels, one of the challenges in systems biology is how to efficiently learn these gene regulatory networks from expression data.

1.3 Machine learning context

Given the difficulties inherent to expression data such as a large variable to sample ratio and a high amount of noise, different machine learning techniques have been used to reconstruct gene regulatory networks from this data. In this section we present the necessary concepts used throughout the thesis.

1.3.1 Model selection [MR05]

Given expression data sets, our main focus lies on the modeling of dependencies between gene expression values. The question of determining how a *target* gene is regulated by other genes can be formulated as a *supervised learning problem*. Specifically, the target gene is the *target variable*, often denoted by Y, the remaining genes are the *input variables*, denoted by $\mathbf{X} = (X_1, \ldots, X_n)$, and the discovered relationship between input variables and target variable is the *model*

$$Y = f(\mathbf{X}) + \varepsilon, \tag{1.1}$$

with $f(\cdot) : \mathcal{X} \to \mathcal{Y}$, ε assumed to be independent of $\mathbf{X}$ and $\mathbb{E}(\varepsilon) = 0$.

This model can then be used to predict the outcome for new objects. A set of input/output data $\mathbf{D} = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_m, Y_m)\}$ is called a *training set* and a set of objects is called *test set*.

When characterizing supervised models according to the output type, the two main classes are *classification* and *regression*. The former maps the input space into a predefined finite set of classes (the classes are sometimes called *factors*) whereas the latter maps the input space into an integer or a real-valued domain. An example of an application for classification problems is patient classification in breast cancer. Using expression data to classify a patient's tumor, the appropriate treatment can then be selected thus ensuring that the patient has the best chance of survival [VtVDVdV+02]. A possible application of regression models is to use them in order to predict the expression of a gene given the model.

The problem of selecting the best model given the data is known as *model selection*. Suppose a learning algorithm returns the parameters θ of a model $h(\mathbf{X}, \theta)$. Different models have different levels of complexity and selecting the best model given the data typically requires minimizing the *mean squared error* (MSE). Let $\hat{\theta}$ be the estimate of an unknown parameter θ given such a model. Then its MSE is defined as

$$\mathrm{MSE}(\hat{\theta}) = \mathbb{E}\{(\hat{\theta} - \theta)^2\}. \tag{1.2}$$

It has been shown [Geu10] that two different quantities contribute to an estimator's MSE, its *bias*

$$bias(\hat{\theta}) = \theta - \mathbb{E}(\hat{\theta}) \tag{1.3}$$

and its *variance*

$$var(\hat{\theta}) = \mathbb{E}\{\hat{\theta}^2\} - (\mathbb{E}\{\hat{\theta}\})^2. \tag{1.4}$$

The MSE can be then decomposed as follows

$$\mathrm{MSE}(\hat{\theta}) = (bias(\hat{\theta}))^2 + var(\hat{\theta}). \tag{1.5}$$

Thus minimizing the MSE is equivalent to finding the optimal bias-variance trade-off (Figure 1.6).
A model with low complexity results in small variance of the predictions made but at the cost of higher bias, whereas a highly complex model will reduce the bias but increase the variance. This phenomenon is closely related to the problem of *overfitting*.

Overfitting occurs when the model chosen for the given data is too complex. This model will usually score very well on the training data whereas it will score badly on the test data.

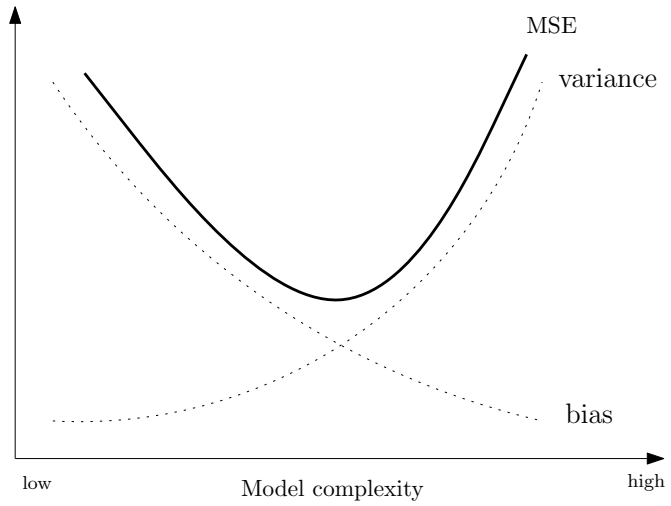


Figure 1.6: Relation between model complexity, mean squared error, bias and variance; figure from [Kon09].

The converse problem, *underfitting* occurs when the chosen model is not complex enough. In this case, the model will score badly on both the training data and on the test data.

1.3.2 Feature selection

In view of the high variable/low sample problem and the high amount of noise intrinsic to expression data, selecting relevant features has several advantages over models using all variables.

i. Easier interpretation of the model: the selection of a subset of variables enables us to better understand the underlying process.

ii. Higher generalizability of the model: using fewer variables to build the model makes it less prone to overfitting and thus enables it to make good predictions on new samples.

iii. Building a model only on a small subset of variables reduces the computational cost compared to that of building the model using all variables.

There are three main types of feature selection algorithms: *filter*, *wrapper* and *embedded* methods [GE03]. Filter methods select a subset of variables independently of the chosen predictor. Wrapper methods select a subset of variables based on their predictive power for the chosen predictor. Embedded methods select the variables in the process of the learning procedure specific to the given learning machine [GE03].

1.3.3 Reconstruction of gene regulatory networks

An important tool for the representation of gene regulatory networks are graphical models. In these models the variables correspond to genes and the edges correspond to the interactions between the genes. Depending on the model, the directed or undirected edges represent different types of dependencies. The differences between directed and undirected graphs will be discussed in Section 2.2.

A key problem in the reconstruction of gene regulatory networks is the large number of variables. Aside from the difference in the type of dependencies the methods also vary in their complexity which makes them better or less suited for the reconstruction task. These methods can be grouped as follows by decreasing complexity.

The highest computational complexity is attributed to methods inferring *Bayesian networks*. These are directed acyclic graphs together with a probability distribution. In these networks, the absence of an edge between two nodes corresponds to a conditional independence relationship between the associated variables. A detailed discussion is provided in Sections 3.2 and 3.3.1.

The next class of methods are *Gaussian graphical models*. The main assumption for these models is that the variables follow a multivariate normal distribution. In these models, a missing edge between two nodes corresponds to the two variables being independent conditionally on all remaining variables. These are typically undirected. We discuss methods to infer Gaussian graphical models in Section 3.1.

The last category of methods we present are feature selection techniques which have been been successfully applied to reverse engineering of regulatory networks in [BK00, MNea06, FHea07, MKLB07]. These inference techniques start by selecting a number of relevant features for each variable in the data set. The relevance criterion varies depending on the algorithm and is usually based on correlation or mutual information. For each selected feature an undirected edge is drawn in the network to the target variable, thus inferring a network variable by variable. These methods will be discussed in Section 3.3.2.1.

1.4 Causality

Whenever predictions involve the outcome of manipulations, standard feature selection techniques do not suffice. As they were designed to return good predictions they do not necessarily model the underlying mechanisms [GAE07] and may therefore include both causes and effects of the target variable. In fact, the most relevant variables are those present in the target variable's *Markov blanket* which includes the target variable's parents, children and spouses [TA03]. Therefore, selecting features based only on relevance with the target will not yield information concerning cause-effect relationships.

However, in certain prediction tasks this information is crucial, for example when designing targeted treatments for a disease under study. In other words, when predicting the outcome of a manipulation.

When discussing the inference of causal relationships between variables using only observational data, probably the most frequently cited statement is (for example in [GAC+08])

association is not causality.

As undoubtably as this statement is true, there are frameworks and methods that allow for causal inference using measures of association.

Throughout this thesis we will be interested in cause-effect relationships defined based on the notion of manipulation. We present in this section the basic assumptions and definitions needed to infer such cause-effect relationships from observational data [GC99, HR13].

Let $\mathbf{V}$ be the set of modeled variables and p an associated joint probability distribution. Each variable that is a member of the minimum set of variables whose manipulation will change the distribution of a target variable Y is said to *causally influence* Y. In other words this set is the minimum set of variables sufficient to change the distribution of Y.

A typical example of a causal effect occurs in epidemiological studies when comparing the outcome of a treatment versus that of withholding this treatment on a population. Whenever the two outcomes differ, the treatment is said to have a causal effect on the outcome. In real-world studies it is impossible to observe the outcome for a patient in both scenarios: having received treatment and not having received treatment. For each individual, only one of the two *counterfactual* outcomes is observed, while the other one remains unknown. The idea of *randomized experiments* is to ensure that these missing values occur randomly so that cause-effect relations can be inferred. However, in practice randomization is infrequently applied due to ethical and/or practical reasons.

Another typical approach to causal inference is the use of *controlled experiments* [HR13]. In this type of experiment the research subjects are divided into two groups, one receiving treatment and the control group which does not receive the treatment. The control group needs to be indistinguishable from the treatment group except for the variable whose effect is being studied. In observational studies, experimental manipulations are impossible but the controlled experiment can be imitated by using a part of the data set in which the controlled variable is constant. In probabilistic terms this is equivalent to considering dependencies conditioned on the controlled variable. In the following we present the usual assumptions for causal inference from observational data and furthermore how to use conditional dependencies to detect specific causal patterns.

A causal relationships of the type 'X causes Y' can be visualized using a directed graph $X \to Y$. This graphical representation allows the encoding of both causal relations and

dependence relations. A typical example for the intrinsic difference between the two relations is the 'smoking, risk of lung cancer and carrying a lighter' example. Common

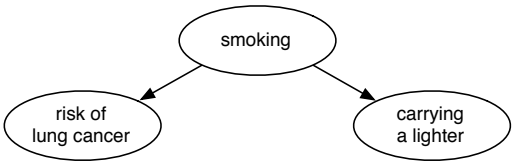


Figure 1.7: Causal relationship between smoking and carrying a lighter and lung cancer. Example taken from [HR13]

knowledge about the causal relationships can be represented by the directed graph in Figure 1.7: On the one hand, carrying a lighter has no causal effect on the risk of lung cancer. On the other hand, smoking does have a causal effect on both carrying a lighter and lung cancer and vice versa. However, there is an association between carrying a lighter and lung cancer: carrying a lighter increases the likelihood of smoking which in turn increasing the risk of developing lung cancer. This is one of the examples for which an existing dependence between variables does not imply causality between these variables.

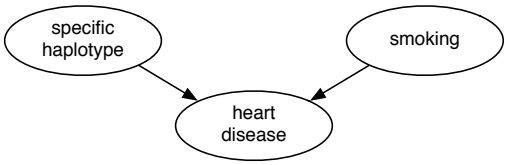


Figure 1.8: Causal relationship between a haplotype without causal effect on the risk of becoming a smoker but with a causal effect of the risk of heart disease. Example taken from [HR13]

The second part of this example concerns the causal influence of smoking and a specific haplotype on heart disease, see Figure 1.8. There is no causal effect between the specific haplotype and smoking and there is also no association between the two variables: having additional information on somebody's haplotype does not help to predict whether or not they are smoking, due to

$$p(\text{smoking}|\text{does not have haplotype}) = p(\text{smoking}|\text{has haplotype}). \tag{1.6}$$

In graph theory, the node such as 'heart disease' is known as *collider*. Colliders block the association along their path (Section 2.2.2) and form the pattern that algorithms based on independence tests try to identify (see Section 3.2.1)

The network is modeled by a directed acyclic graph in which the nodes correspond to the

variables and the arcs to the causal relationships. This is known as a Bayesian network, see Section 2.2.2 for a formal definition.

Two conditions guarantee the one-to-one mapping between the probability distribution p and the directed graph $\mathcal{G}$.

Causal Markov Condition The CMC is satisfied if and only if $\forall X \in \mathbf{V}$ it holds that according to p, X is independent of its non-effects given its direct causes.

Causal Faithfulness Condition Two variables are only probabilistically independent if their independence is due to the Markov condition.

Apart from the examples presented in Figures 1.7 and 1.8, there exist situations in which it is more difficult or even impossible to infer causality from observational data. These problems usually manifest through the simultaneous occurrences of 'no causal effect of the treatment on the outcome' and 'association between treatment and outcome'.

Common causes Two variables are *confounded* if they are probabilistically dependent due to one or more common causes. A *confounder* is a variable that can be used, possibly not by itself, to remove confounding. Therefore, if the confounders have been measured, conditioning on them will remove the statistical dependency that is due to confounding. In randomized experiments confounding should not exist as the fact of receiving the treatment is random. This random selection cannot be the cause of the outcome.

When learning from observational data, the assumption is made that enough variables are measured such that every common cause of any two (or more) variables is in $\mathbf{V}$. This condition is known as *causal sufficiency*.

Conditioning on common effect The problem of conditioning on a common effect is easiest understood using an example. Consider Figure 1.9, in which a treatment A is administered which has a causal influence on the development of a certain disease Y. Let $C \in \{0, 1\}$ represent 'death of the patient'. Then both A and Y are causes of C:

- Having disease Y will increase the risk of death and
- A could reduce the likelihood of death due to other reasons than those incorporated in Y.

When this study is now restricted to patients that survived it is conditioned on $C = 0$. This would result in an induced association between A and Y which is not due to a causal relation between them. This problem is known as *selection bias*.

This selection bias can be induced when a data set originally contains missing values and is then restricted to complete data samples.

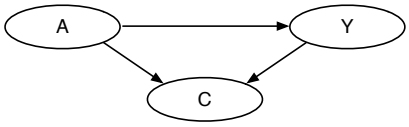


Figure 1.9: Conditioning on a common effect creating selection bias. Example taken from [HR13]

1.5 Causal inference

Methods that infer causal relations between variables are mainly those that infer Bayesian networks using independence tests: the so-called constraint-based techniques (Section 3.2.1). They mainly rely on identifying v-structures as described in Figure 1.8. This is due to the fact that triplets (X, Y, Z) in a v-structure $X \rightarrow Y \leftarrow Z$ correspond to a specific set of (in)dependences

$$X \not\perp\!\!\!\perp Z|Y, \tag{1.7}$$

$$X \perp\!\!\!\perp Z, \tag{1.8}$$

where $X \perp\!\!\!\perp Z$ denotes that X is independent of Z and $X \not\perp\!\!\!\perp Z|Y$ that X and Z are not conditionally independent given Y.

The remaining three possible configurations of triplets X, Y, Z with X and Z unconnected are depicted in Figure 1.10 and represent the independence relation

$$X \perp\!\!\!\perp Z|Y. \tag{1.9}$$

Therefore, algorithms can exploit this difference in dependence relations to unambiguously distinguish v-structures from the remaining possible configurations. However, the

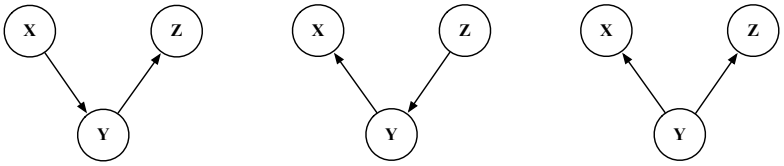


Figure 1.10: Directed graphs of three variables X, Y and Z describing the independence relationship $X \perp\!\!\!\perp Z|Y$.

original orientation methods proposed in [Pea00, SGS01] based on the identification of v-structures using dependency tests, require a number of statistical tests that is exponential in the number of variables. This high number of tests prohibits the application for any data set with more than a hundred variables. Improvements with regards to the number of independence tests have been achieved by focusing on Markov blankets

(Section 3.3.1). It has been shown that from observational data alone, only networks belonging to the same equivalence class as the true network can be learnt. All graphs in this class agree on the network's skeleton[1] and the v-structures. If all equivalent networks 'agree' on an edge $X \to Y$, it can be deduced that X is a cause of Y [FLNP00].

1.6 Prior integration

The difficult nature of expression data with only few samples compared to the large number of variables explains why the inclusion of prior knowledge in the inference process is deemed beneficial. The effective integration of prior knowledge in the modeling requires i) an automated retrieval of said prior knowledge, ii) the transformation of the prior knowledge in a format such that inference methods can make use of them and iii) an approach that can efficiently use it to improve the accuracy of the inferred network.

1.6.1 Tools for knowledge retrieval

The results of research conducted on biomedical data sets are available from different sources. Abstracts of published articles are available from PubMed, open access articles are featured on PubMed Central and experimentally validated interactions between genes have been stored in biological data bases such as Pathway Commons [CGD+11]. However when one is interested in the known interactions between sets of genes, it is difficult to manually search these sources due to the large quantity of information. Therefore, different tools [HV05, MRF+08, IRMM10, HKOD+12] have been implemented that can carry out automized searches and usually return the compiled set of known interactions.

The online service *information Hyperlinked Over Proteins* (iHOP) was designed such that the user could retrieve biomedical literature linked to a gene of interest [HV05]. However it does not allow automated retrieval for multiple genes of interest. This was the main motivation behind the development of the *Gene Interaction Miner* (GIM): to provide an interface in which a list of genes can be uploaded and all associated publications cab be retrieved [IRMM10]. The output of GIM are gene-gene interactions, supporting references and an output graph in which the edges' weights are the number of citations found.

The networks extracted in *GeneMANIA* consist of nodes representing genes or proteins and undirected edges representing co-functionality of the associated genes or proteins [MRF+08]. These networks are known as functional association networks. The weight of an edge corresponds to the confidence in the co-functionality as extracted from the given data source.

The aim of *GeneMANIA* is twofold: to build functional association networks using different data sources and subsequently to fuse these into one final network. Secondly, this

[1] Undirected network after removal of all edges' orientation

single functional association network is then used to predict gene functions. Furthermore, it has been implemented to carry out these computations in real-time on a web server.

After uploading a gene list and selecting the desired sources, the web server presents a network in which edges are color coded by source. Different exporting options are available for both the network graph, the connections themselves and the search parameters.

1.6.2 Using prior knowledge for the inference of gene regulatory networks

Until now only a few inference techniques integrated prior knowledge, mainly Bayesian networks and kernel methods presented in detail in Sections 3.2 and 3.5, respectively.

In order to make use of this edge-wise prior knowledge in the Bayesian network framework, it must first be transformed into a prior distribution over the set of directed acyclic graphs. We present these different techniques in Section 3.2.2.2. Once a prior distribution is available, different advances can be made in score-based learning to integrate this prior knowledge: either in the scoring function itself or in the search algorithm (Sections 3.2.2.1 and 3.2.2.1).

Another class of methods that combines data from different sources are kernel methods. Algorithms that belong to this class start by representing the genomic data, prior knowledge and other sources of information by kernels and then interpret these as undirected networks. We present different kernel-based techniques in Section 3.5.

1.7 Contributions' summary

1.7.1 Methodology

In this thesis we develop a comprehensive framework that addresses two important problems in bioinformatics. The first problem is the inference of directed networks from expression data and the challenges these data sets pose, i.e. the high variable to sample ratio and the high amount of noise. The second issue concerns the subsequent validation of the inferred networks. To the best of our knowledge there exists no purely data-driven quantitative evaluation in the literature for non-temporal data (more details on time-series data in Appendix B.5).

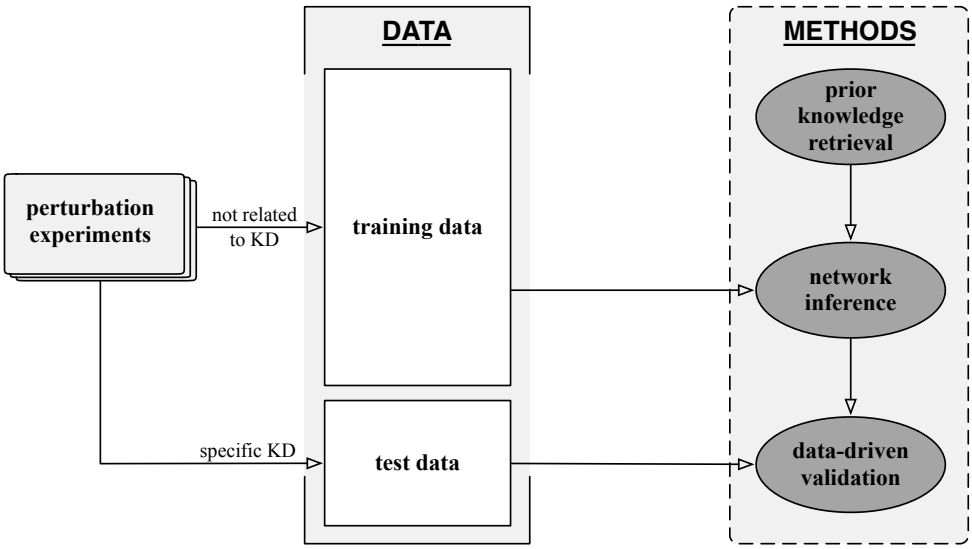


Figure 1.11: The comprehensive analysis framework, using *Predictive Networks* for prior knowledge retrieval, *predictionet* to infer directed networks and knock-down (KD) data in a cross-validation scheme for the data-driven validation.

Our framework addresses these problems as follows (see Figure 1.11).

Prior knowledge retrieval: We implement a web application named *Predictive Networks* that retrieves known gene-gene interactions from biological databases and PubMed abstracts using text-mining techniques [HKOD$^+$12].

Network inference: We base our network inference approach on feature selection techniques using information theoretic measures and adjust these rankings in a linear combination scheme with the prior knowledge [ODF$^+$13]. We implement this inference approach in an R/Bioconductor package named *predictionet* [HKOBQ12] and additionally make it available as part of *Predictive Networks*.

Data-driven validation: Using experimental knock-down data from colorectal cancer cell lines, we design a cross-validation scheme in which the samples related to a specific knock-down are used as test set and the remaining samples as training set. We use the training set together with retrieved prior knowledge to infer directed networks and the test set to evaluate the inferred network's quality [ODF+13].

The remaining contributions of this thesis involve extensions to the different techniques used to infer the directed networks.

Causal inference We present a novel heuristic based on negative and positive interaction information which takes into account an edge's neighborhood for its orientation. Furthermore, we present a fast way of computing the three-variate interaction information via bivariate quantities under the assumption of Gaussianity [OMB13].

Ensemble mRMR The standard mRMR feature selection procedure selects at each step the variable which has the highest score with the target variable. Due to the high variable to sample ratio, feature selection tends to be sensitive to small changes in the data set. In order to obtain a more robust solution, we infer at each step a set of models by taking not only the highest scoring feature into account but a predefined number of high scoring features. The result of this technique is a tree of mRMR models in which each branch corresponds to a single mRMR model. We implemented our ensemble mRMR approach in the R/Bioconductor package named *mRMRe* [DJPCO+13]. We evaluated the performance of our ensemble mRMR approach on different generated data sets both in terms of robustness of the selected features as well as in terms of the inferred networks' quality.

1.7.2 Software

Cran R package *predictionet* The functionalities of this package can be categorized as follows: i) Functions related to network inference using genomic data and prior knowledge: the user provides the data, prior knowledge and the weighting factor between the two sources. ii) Functions needed for validating the inferred networks: a cross-validation scheme which automatically computes the stability of each edge and the predictive ability for each variable. The inference part of the package has been integrated into the *Predictive Networks* web application.

OMbIT We implement the arc orientation algorithm we developed in R and will integrate it in the next release of the network inference package *MINET* as part of [OMB13]. It will extend *MINET* 's functionality to the inference of directed networks, including the fast estimation of the three-variate interaction-information.

1.7.3 Experimental findings

Causal discovery in colorectal cancer

In this case study (Chapter 5), we focus on a specific biomedical problem: the question of how genes in the RAS pathway interact in colorectal cancer. The experiments were carried out in the Computational Biology and Function Genomics Laboratory, Dana-Farber Cancer Institute, Harvard School of Public Health[1], generating a knock-down data set in which eight core genes of the RAS pathway, as described in BioCarta 2007, were knocked down one at a time. The perturbation experiments were performed in two colorectal cancer cell lines. We propose a set of novelties with the following goals: i) We establish the validity of the proposed purely data-driven quantitative validation framework using the described knock-down data; ii) We experimentally verify the quality of prior knowledge downloaded from *Predictive Networks*; iii) We experimentally prove that the quality of inferred networks when combining genomic data and prior knowledge is higher than that of the respective networks using only one of the sources; iv) We show that the knock-down experiments obtained using colorectal tumor cell lines can also be used for real colon tumor patients.

Determining the influence of entropy estimation on network inference

The estimation of mutual information is a difficult task for two reasons: i) The high number of variables requires a very efficient computation therefore not all estimators can be applied to expression data; ii) The low number of samples makes it very difficult to obtain a good estimate. Nonetheless, estimation of mutual information is an integral part of all network inference based on information theory. In the last section of the contribution chapter (Section 6.3) we study a number of state-of-the-art estimators with respect to their influence on network inference performance both on generated data sets and on biological data.

1.7.4 Publications

1.7.4.1 Used in thesis

Peer-reviewed journal

- [DJPCO$^+$13] Nicolas De Jay*, Simon Papillon-Cavanagh*, Catharina Olsen, Gianluca Bontempi & Benjamin Haibe-Kains. *mRMRe: an R package for parallelized mRMR ensemble feature selection.* Bioinformatics, 2013.
- [OMB13] Catharina Olsen, Patrick E. Meyer & Gianluca Bontempi. *A scalable heuristic to orient arcs in undirected networks.* preprint.
- [ODF$^+$13] Catharina Olsen, Amira Djebbari, Kathleen Fleming, Niall Prendergast, Renee Rubio, Frank Emmert-Streib, Gianluca Bontempi, Benjamin Haibe-Kains*

[1] http://compbio.dfci.harvard.edu/

& John Quackenbush*. *Inference of predictive gene networks from biomedical literature and gene expression data.* Submitted to Genomics, 2013.

- [HKOD+12] Benjamin Haibe-Kains, Catharina Olsen, Amira Djebbari, Gianluca Bontempi, Mick Correll, Christopher Bouton & John Quackenbush. *Predictive Networks: A Flexible, Open Source, Web Application for Integration and Analysis of Human Gene Networks.* Nucleic Acids Research, 2011.

- [OMB09a] Catharina Olsen, Patrick E. Meyer & Gianluca Bontempi. *On the Impact of Entropy Estimation on Transcriptional Regulatory Network Inference Based on Mutual Information.* EURASIP Journal on Bioinformatics and Systems Biology, 2009.

Software

- [HKOD+12] Benjamin Haibe-Kains, Catharina Olsen, Gianluca Bontempi & John Quackenbush. *predictionet: Inference for predictive networks designed for (but not limited to) genomic data*, 2012. R package version 1.1.5.

Book chapter

- [OHKQB13] Catharina Olsen, Benjamin Haibe-Kains, John Quackenbush & Gianluca Bontempi. *On the Integration of Prior Knowledge in the Inference of Regulatory Networks.* Accepted for publication, 2013.

- [MOB11a] Patrick E. Meyer, Catharina Olsen & Gianluca Bontempi. *Transcriptional Network Inference based on Information Theory.* In Applied Statistics for Network Biology: Methods in Systems Biology. Wiley, 2011.

Conferences and workshops

- [OMB09b] Catharina Olsen, Patrick E. Meyer & Gianluca Bontempi. *Poster: Inferring causal relationships using information theoretic measures.* In Proceedings of the 5th Benelux Bioinformatics Conference (BBC09), 2009.

- [OMB08] Catharina Olsen, Patrick E. Meyer & Gianluca Bontempi. *On the impact of entropy estimator in transcriptional regulatory network inference.* Proceedings of WCSB08, 2008.

1.7.4.2 Not used in thesis

Peer-reviewed journal

- [PCJH+13] Simon Papillon-Cavanagh, Nicolas De Jay, Nehme Hachem, Catharina Olsen, Gianluca Bontempi, Hugo Aerts, John Quackenbush & Benjamin Haibe-Kains. *Comparative Study and Validation of Genomic Predictors for Anticancer Drug Sensitivity.* Journal of the American Medical Informatics Association, 2013.

- [MOB11b] Abhilash Alexander Miranda, Catharina Olsen & Gianluca Bontempi. *Fourier spectral factor model for prediction of multidimensional signals.* Signal Processing, vol. 91, no. 9, pages 2172 – 2177, 2011.

1.8 Outline

This thesis is divided into five parts. First, we present the necessary definitions for modeling dependencies. This includes a short introduction to undirected and directed graphs, definitions and properties related to information theory and finally, different estimators for entropy and mutual information. In the second part, we present state-of-the-art methods for network inference, causal inference and prior integration. These include Gaussian graphical models, Bayesian networks, methods based on feature selection, ensemble techniques and kernel-based methods. The next three chapters are devoted to our contributions. In the first one we present our methodological contributions to the inference of directed networks from genomic data and prior knowledge together with a purely data-driven approach to network validation. The next chapter is dedicated to the experimental study applying the proposed methods to colon cancer data. In the final contribution chapter we present extensions to mRMR feature selection, causal inference and a study on the influence of entropy estimation has on the quality of inferred networks. Finally we draw our conclusions and present directions for future work.

Chapter 2

Preliminaries: graphical models and information theory

This chapter serves as foundation for the algorithms presented later in this thesis. We start with definitions and properties from probability theory. Then we introduce concepts from graph theory as well as their role in modeling probabilistic relations between variables. Afterwards we introduce different linear dependency measures and outline of the importance of information theory for modeling dependencies. Then we present different testing procedures important for constraint-based algorithms. We conclude this section with an overview of estimation techniques for the different information theoretic quantities.

2.1 Probabilistic relations

In this thesis, capital letters X, Y, Z will denote random variables, $x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}$ their realizations.

2.1.1 Discrete random variables

Let X and Y be two discrete random variables, following a probability distribution p. X and Y are **independent** if

$$p(X = x, Y = y) = p(X = x)p(Y = y), \qquad \forall x \in \mathcal{X}, y \in \mathcal{Y}.$$

The **conditional probability** is defined as

$$p(X = x|Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)}, \qquad \forall y : p(Y = y) > 0.$$

Consider three random variables X, Y and Z. X and Y are called **conditionally inde-**

pendent given Z, if

$$p(X = x, Y = y|Z = z) = p(X = x|Z = z)p(Y = y|Z = z),\ \forall z : P(Z = z) > 0.$$

Notation 2.1
Conditional independence between X and Y given Z will be denoted by $X \perp\!\!\!\perp Y|Z$.

The following property (known as *Bayes' theorem*) builds the basis for the Bayesian approach to probability theory. It relates the conditional and marginal probabilities

Property 2.2 *[Fel68]*
Let X and Y be two random variables with probability distribution p. Then, given that $p(X = x) > 0$ for all realizations x of X, the following equation holds

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)},$$

where $p(Y)$ is known as prior or marginal probability of Y, $p(Y|X)$ as posterior probability and $p(X)$ serves as normalization constant.

2.1.2 Continuous random variables

In the following, we state some of the analogous definitions and properties for continuous random variables.

Let X and Y be two continuous random variables. X and Y are **independent** if their joint density $f_{X,Y}$ factorizes into the product of their marginal densities f_X and f_Y:

$$f_{X,Y}(x, y) = f_X(x)f_Y(y).$$

The **conditional probability density function** of Y given X is defined as

$$f_{Y|X}(y, x) = \frac{f_{X,Y}(x, y)}{f_X(x)}, \qquad f_X(x) > 0.$$

Definition 2.3 *[Whi90]*
*X and Y are called **conditionally independent** given Z if*

$$f_{X,Y|Z}(x, y, z) = f_{X|Z}(x, z)f_{Y|Z}(y, z) \quad \forall z : f_Z(z) > 0.$$

2.2 Graphical representations of probabilistic relations

In this section we introduce concepts from graph theory necessary for the later work of this thesis. We discuss both undirected and directed representations of probabilistic independence relations together with two requirements to obtain a one-to-one relation between a probability distribution and an undirected/a directed graph, namely

the Markov condition and the faithfulness condition. In the final part, we discuss the conceptual difference between undirected and directed graphs.

2.2.1 Undirected graphs

An **undirected graph**, denoted by $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, is a pair which consists of a finite set $\mathbf{V}$ of *nodes* (also called *vertices*) and a finite set $\mathbf{E}$ of *edges* (also called *arcs*) between these nodes. Two vertices $A, B \in \mathbf{V}$ are **adjacent** if there is an edge between them. In Figure 2.1 the vertices A, D are adjacent as well as A, B and B, C and B, D.

Notation 2.4
The set of adjacent variables of a node X will be denoted by $\mathbf{adj}(X)$.

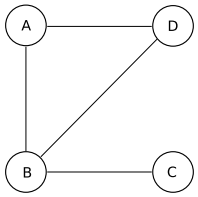


Figure 2.1: undirected graph

For three subsets $\mathbf{A}, \mathbf{B}$ and $\mathbf{S}$ of $\mathbf{V}$, the set $\mathbf{S}$ **separates** the two sets $\mathbf{A}$ and $\mathbf{B}$ if all paths from $\mathbf{A}$ to $\mathbf{B}$ intersect $\mathbf{S}$. In Figure 2.1, the node B separates $\{A, D\}$ and $\{C\}$.

A link between the separation of variables and probability theory is provided via the *Markov property* which given the graph allows to deduce conditional independencies between two variables from their adjacencies.

Property 2.5 *(Markov properties for undirected graphs)[Edw00]*

- ***Pairwise Markov property:*** *If two variables are not adjacent, then they are conditionally independent given the set of remaining variables.*
- ***Local Markov property:*** *Each variable X is conditionally independent of its non-neighbors given its neighbors*

$$\forall X \in \mathbf{V}: \quad X \perp\!\!\!\perp (\mathbf{V} \setminus \{X \cup \mathbf{adj}(X)\}) \,|\, \mathbf{adj}(X), \tag{2.1}$$

  where $\mathbf{adj}(X)$ *denotes the set of adjacent nodes of X, without X itself.*
- ***Global Markov property:*** *If two sets of variables $\mathbf{X}, \mathbf{Y}$ are separated by a third set of variables $\mathbf{Z}$ (all three being disjoint subsets of $\mathbf{V}$), then*

$$\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z}. \tag{2.2}$$

These properties are related as follows.

Property 2.6 *[Lau96]*
For any undirected graph $\mathcal{G}$ and the associated probability distribution on $\mathcal{X}$

$$\textit{Global Markov} \Rightarrow \textit{Local Markov} \Rightarrow \textit{Pairwise Markov.} \tag{2.3}$$

In general, these properties are not equivalent but it has been shown that equivalence holds if the probability density function is continuous and strictly positive [Lau96].

The Markov condition(s) provide(s) a tool to read probabilistic independence relations directly from the undirected graph. However, it does not guarantee that the inverse relation holds: the independence relations entailed in the graph are the only ones entailed in the corresponding probability distribution. A second property is needed to ensure this:

Definition 2.7 *[Pea00]*
*A probability distribution p is **faithful** to $\mathcal{G}$ if for all random variables X_i and X_j and sets $\mathbf{S} \subseteq \mathbf{V} \setminus \{X_i, X_j\}$ with $X_i \perp\!\!\!\perp X_j | \mathbf{S}$ it holds that $\mathbf{S}$ separates X_i and X_j.*

2.2.2 Directed graphs

A **directed graphs** will be denoted by $\mathcal{G} = (\mathbf{V}, \mathbf{E})$. Unlike the definition of undirected graphs, the set $\mathbf{E}$ of edges now contains ordered pairs of vertices. In the drawing of the graph the edges will be represented as arrows. Figure 2.2: there is an arrow from A to B, thus $(A, B) \in \mathbf{E}$.

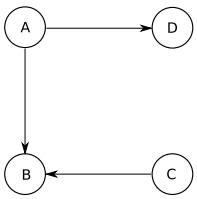


Figure 2.2: directed graph

If there exists an arrow between two vertices A and B, that is $A \to B$ or $B \to A$, the nodes are called **adjacent**. The set of adjacent variables of a variable X will again be denoted by $\mathbf{adj}(X)$.

A **directed path** from node V_1 to node V_k requires that $V_i \to V_{i+1}$ for all $i = 1, \ldots, k-1$. A path for which the first and the last node coincide is called **directed cycle**. In the following, the class of interest are directed graphs with no directed cycles. These are called **directed acyclic graphs (DAG)**.

If $V_1 \to V_2$, then V_1 is called a **parent** of V_2, and V_2 is called **child** of V_1. The set of parents of V_2 is denoted by $\mathbf{pa}(V_2)$, the set of children as $\mathbf{ch}(V_2)$. If there is a directed

path from V_1 to V_2 then V_1 is called **ancestor** of V_2, denoted by $an(V_2)$, and V_2 is called a **descendant** of V_1, denoted by $de(V_1)$. These definitions can be extended to sets of nodes: denoted by $\mathbf{an}(V_2)$ and $\mathbf{de}(V_1)$, respectively.

2.2.2.1 Bayesian networks

In this section the definition of Bayesian networks is provided via the association of a DAG $\mathcal{G}$ with a probability distribution such that a node is conditionally independent of all its non-descendants given its parents.

The first necessary property relates the absence of directed cycles to the existence of an ordering of the nodes in the graph.

Property 2.8 *[Edw00]*
The absence of any directed cycles is equivalent to the existence of an ordering of the nodes $\{V_1, \ldots, V_k\}$ such that $V_i \to V_j$ only when $i < j$. The numbering is not necessarily unique.

Assuming an ordering $V_1, \ldots, V_n$ for the given variables such that V_i is prior to V_{i+1} for $i = 1, \ldots, n-1$. Corresponding to the ordering, the joint density of $V_1, \ldots, V_n$ can be factorized as

$$\begin{aligned} & f_{V_1}(v_1) f_{V_2|V_1}(v_2, v_1) \ldots f_{V_n|V_{n-1},\ldots,V_1}(v_n, \ldots, v_1) \\ & = \prod_{V \in \mathbf{V}} f_{V|\mathbf{pa}(V)}(v|\mathbf{pa}(v)). \end{aligned} \tag{2.4}$$

Based on this factorization, a Bayesian network can be defined as follows.

Definition 2.9 *[CGK+01]*
*A **Bayesian network** $BN = (\mathcal{G}, p)$ consists of a directed acyclic graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ and the probability distribution p over all possible realizations. Each node $X \in \mathbf{V}$ represents a variable and each arc $E \in \mathbf{E}$ represents a probabilistic dependency between the associated nodes. Furthermore, all variables $X \in \mathbf{V}$ are probabilistically independent of all non-descendants of X given the parents of X, as described by equation (2.4).*

In order to construct a DAG, an arrow is added from V_i to V_j, where $i < j$, unless V_i and V_j are conditionally independent given all prior variables

$$\begin{aligned} & V_i \perp\!\!\!\perp V_j \mid (\{V_1, \ldots V_j\} \setminus \{V_i, V_j\}) \\ & = V_i \perp\!\!\!\perp V_j \mid \mathbf{an}(\{V_i, V_j\}). \end{aligned} \tag{2.5}$$

This behavior of DAGs corresponds to the pairwise Markov property for undirected graphs (this will be referred to as **Markov condition** for DAGs).

Definition 2.10 *[Pea00, SGS01, CGK+01]*
*If a DAG and a probability distribution satisfy equation (2.4), the graph is said to satisfy the **Markov condition**.*

Thus a Bayesian network satisfies the Markov condition by definition. Moreover, it is only a Bayesian network if no edge can be removed without also changing the implied dependencies [Mar03].

The other two Markov properties (local: equation (2.1) and global: equation (2.2)) cannot be related one-to-one to directed graphs. However, there exists a property which provides the connection between the graph-theoretic separation and conditional independence in directed graphs. This criterion is known as d-separation.

2.2.2.2 D-separation

A triplet of variables $X \rightarrow Y \leftarrow Z$, where X and Z are not connected is called **v-structure** and the center node Y is called **collider**.

In Figure 2.2, the node B is a collider and the three nodes A, B, C form a v-structure.

Let $\mathbf{S}_1$ and $\mathbf{S}_2$ be two sets of nodes connected by a set of paths. A set of nodes $\mathbf{S}_3$ **blocks** such a path between $\mathbf{S}_1$ and $\mathbf{S}_2$ if there is a node X on this path satisfying one of the following conditions [Pea00, Mar03]:

i. X is a non-collider and $X \in \mathbf{S}_3$, or

ii. X is a collider but $X \notin \mathbf{S}_3$ and $\mathbf{de}(X) \cap \mathbf{S}_3 = \emptyset$.

In Figure 2.3, all paths between the two sets $\mathbf{S}_1$ and $\mathbf{S}_2$ are blocked by the set $\mathbf{S}_3$. The first two paths in the graph correspond to the first criterion as they depict a chain and a fork, respectively. The third path corresponds to the second criterion where neither the collider nor its descendants belong to the separating set $\mathbf{S}_3$.

Definition 2.11 *[Edw00]*
*Let $\mathbf{S}_1, \mathbf{S}_2$ and $\mathbf{S}_3$ be sets of vertices in the graph. If $\mathbf{S}_3$ blocks all paths between $\mathbf{S}_1$ and $\mathbf{S}_2$, then $\mathbf{S}_3$ **d-separates** $\mathbf{S}_1$ and $\mathbf{S}_2$.*

Under the Markov condition, the d-separation criterion allows to read conditional independence relations off a directed graph. That is, if two variables X and Y are d-separated by a set $\mathbf{Z}$, then $X \perp\!\!\!\perp Y | \mathbf{Z}$ [Shi02].

Definition 2.12 *[Pea00]*
*The minimal set of nodes which d-separates node X from all other nodes is called **Markov blanket** of X.*

Notation 2.13
*The Markov blanket of X will be denoted by $\mathbf{MB}(X)$; in this context X is called **target variable**.*

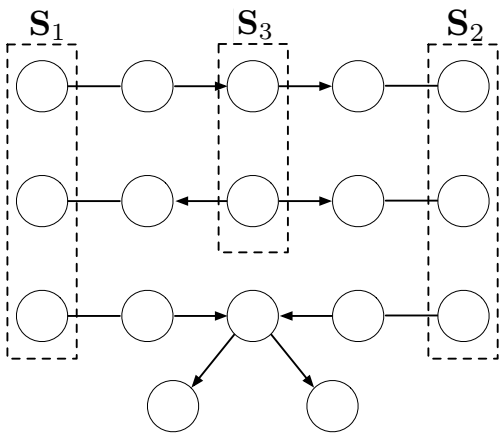


Figure 2.3: d-separation: variables in $\mathbf{S}_1$ are d-separated from variables in $\mathbf{S}_2$ by the set of variables in $\mathbf{S}_3$

2.2.2.3 Faithfulness

Given a graph, the Markov condition (Definition 2.10) determines a set of independence relations. As it is the case for undirected graphs, a probability distribution p on a graph satisfying the Markov condition may include other independence relations besides those entailed by the Markov condition applied to the graph [SGS01]. In Figure 2.4, W and Z might be independent even though the d-separation does not entail their independence. In order to guarantee that only those independencies entailed by the graph are also present in the given probability distribution, faithfulness has to be assumed.

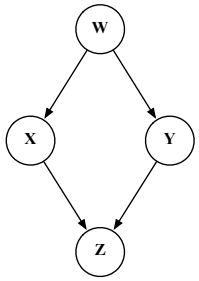


Figure 2.4: Example of an unfaithful directed graph.

Definition 2.14 *(Faithfulness) [SGS01, Pea88, CGK+01]*
If all and only the conditional independence relations true in the probability distribution p are entailed by the Markov condition applied to the graph $\mathcal{G}$, p and $\mathcal{G}$ are called ***faithful to one another****.*

It has been shown that many distributions (most Gaussian and most discrete distributions) for a given network structure are faithful [CGK+01].

In the framework of directed graphs, the term **skeleton** is used for the undirected graph after having removed directions from all edges. The **structure** of a directed graph is then the skeleton together with the directionality information.

Definition 2.15 *[FLNP00]*
*Whenever the independencies implied by a graph $\mathcal{G}$ are the exact same set of independencies encoded in a second graph $\mathcal{G}'$, the two graphs are **equivalent**.*

Theorem 2.16 *[PV91]*
*Two directed acyclic graphs are **equivalent** if and only if they have the same skeleton and the same v-structures.*

The following theorem provides a second characterization of faithfulness and is the basis of constraint-based algorithms (Sections 3.2.1 and 3.3.1).

Theorem 2.17 *[SGS01]*
A directed acyclic graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ and a probability distribution p are faithful to each other if and only if

i. for all vertices X and Y, X and Y are adjacent if and only if X and Y are dependent conditional on every set of vertices of $\mathcal{G}$ that does not include X or Y; and

ii. for all vertices X, Y and Z such that $X - Y - Z$ and $X \not\rightarrow Z$, $X \rightarrow Y \leftarrow Z$ if and only if X and Z are conditionally dependent on every set containing Y but not X or Z.

2.2.3 Undirected versus directed graphs

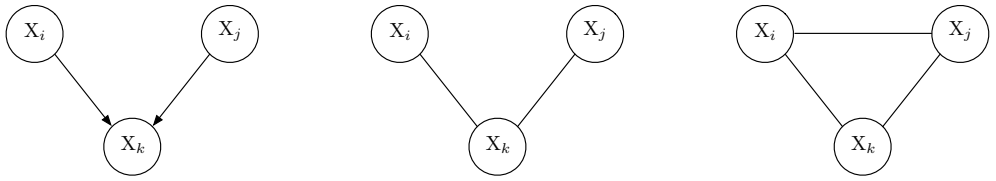
As described in the two previous sections, probability distributions can be represented either by undirected or by directed graphs. However, the meaning of an edge in probabilistic terms is not equivalent for the two modeling approaches [SGS01]. In an undirected graph $\mathcal{G}_u = (\mathbf{V}, \mathbf{E})$, an edge between two vertices V_i and V_j represents a probability distribution p if and only if

$$V_i \not\rightarrow V_j \iff X_i \perp\!\!\!\perp X_j | \mathbf{V} \setminus \{X_i, X_j\}, \tag{2.6}$$

where $V_i \not\rightarrow V_j$ denotes that there is no edge between the two vertices V_i and V_j.

On the other hand, the same probability distribution can be represented by a directed acyclic graph $\mathcal{G}_d$. Let $\mathcal{G}_s$ denote the the skeleton of $\mathcal{G}_d$, then

$$\mathcal{G}_s \subseteq \mathcal{G}_u, \tag{2.7}$$



(a) directed graph $\mathcal{G}_d$

(b) skeleton $\mathcal{G}_s$ of graph in Figure 2.5(a)

(c) undirected graph $\mathcal{G}_u$

Figure 2.5: Directed graph, its skeleton and the undirected graph.

with equality if and only if $\mathcal{G}_d$ does not contain any colliders [WL90].

It can be seen that colliders are the key to differentiating the underlying concepts between undirected and directed graphs. Let us consider only three variables X_i, X_j and X_k. Then, the directed graph $\mathcal{G}_d$ in Figure 2.5(a) represents $X_i \perp\!\!\!\perp X_j$ and the independence relation

$$X_i \not\perp\!\!\!\perp X_j | X_k. \tag{2.8}$$

Its skeleton $\mathcal{G}_s$ is depicted in Figure 2.5(b).

When interpreting the graph in Figure 2.5(b) as undirected independence graph, the corresponding independence relation is as follows

$$X_i \perp\!\!\!\perp X_j | X_k. \tag{2.9}$$

However, the independence relation described in equation (2.8) is represented by the undirected independence graph $\mathcal{G}_u$ in Figure 2.5(c). As described by the inclusion criterion, equation (2.7), the set of edges in the skeleton is a subset of those in the undirected independence graph.

2.3 Measures of linear dependency

Both undirected and directed graphs are tools to represent certain dependency relations between variables. In this section we present different measures for linear dependencies.

2.3.1 Correlation

Correlation is a parameter representing the strength and the direction of the linear relationship between two random variables. The **Pearson correlation** between two random variables X and Y is defined as

$$\rho_{XY} := \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)}{\sqrt{\mathbb{E}(X^2) - \mathbb{E}^2(X)}\sqrt{\mathbb{E}(Y^2) - \mathbb{E}^2(Y)}}, \tag{2.10}$$

where $cov(X, Y)$ denotes the **covariance** between X and Y and σ_X, σ_Y the standard deviations of X and Y, respectively.

Correlation takes values in the interval $[-1, 1]$, i.e. the larger the correlation between two variables is, in absolute terms, the stronger is the linear dependence between the two variables. If two variables are independent, the correlation between them is equal to zero. The inverse does not necessarily hold, i.e. in general zero correlation does not imply independence. More precisely, equivalence between zero correlation and independence holds when the variables are jointly normal distributed [Lau96].

A special case of the Pearson correlation is the **Spearman rank correlation** [KSOA99] for which the data are converted to rankings before calculating the coefficient. The Spearman rank correlation coefficient generalizes the Pearson correlation coefficient in the sense that it can detect not only linear relationships between the variables but any kind of monotone relation without making any assumptions about the distribution of the variables.

Given a set of m measurements of two genes X and Y, the Pearson correlation (equation (2.10)) can be estimated from the measurements x_i and y_i as follows. The covariance in equation (2.10) is estimated by

$$\frac{1}{m-1}\sum_{i=1}^{m}(x_i - \frac{1}{m}\sum_{i=1}^{m}x_i)(y_i - \frac{1}{m}\sum_{i=1}^{m}y_i)$$

while the standard deviation's estimator is given by

$$\hat{\sigma}_X = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \frac{1}{m}\sum_{i=1}^{m}x_i)^2,$$

and $\hat{\sigma}_Y$ analogous.

$$\hat{\rho}_{XY} = \frac{m\sum_{i=1}^{m}x_i y_i - \sum_{i=1}^{m}x_i \sum_{i=1}^{m}y_i}{\sqrt{m\sum_{i=1}^{m}x_i^2 - (\sum_{i=1}^{m}x_i)^2}\sqrt{m\sum_{i=1}^{m}y_i^2 - (\sum_{i=1}^{m}y_i)^2}} \tag{2.11}$$

The Spearman correlation can be calculated using equation (2.11) but replacing the terms x_i and y_i by their respective ranks.

2.3.2 Partial correlation

Given a set of variables $\mathbf{X} = (X_1, \ldots, X_n)$, the covariance matrix of $\mathbf{X}$ is defined as

$$\Sigma = \mathbb{E}\left((\mathbf{X} - \mathbb{E}(\mathbf{X}))(\mathbf{X} - \mathbb{E}(\mathbf{X}))^T\right).$$

The covariance between two variables X_i and X_j, $cov(X_i, X_j)$, is the ij-th element of the covariance matrix.

Definition 2.18 *[Lau96]*
*The **concentration matrix** is defined as the inverse covariance matrix*

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

Definition 2.19 *[Lau96]*
*The **partial correlation** between X_i and X_j conditional on $\mathbf{X}_K \subseteq \mathbf{X} \setminus \{X_i, X_j\}$ is given by*

$$\rho_{ij|K} = \frac{-\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}},$$

where Ω is the concentration matrix of the variables $\mathbf{X}_K \cup \{X_i, X_j\}$.

The partial correlation coefficient quantifies the correlation between two variables X_i and X_j conditional on a set of other variables [FBHM04]. When examining the correlation between two variables X_i and X_j, removing the influence of variable X_k results in the **partial correlation** [KSOA99]

$$\rho_{i,j|k} = \frac{\rho_{ij} - \rho_{ik}\rho_{jk}}{\sqrt{1-\rho_{ik}^2}\sqrt{1-\rho_{jk}^2}}. \tag{2.12}$$

This can be equivalently denoted for three variables X, Y and Z

$$\rho_{X,Y|Z} = \frac{\rho_{XY} - \rho_{XZ}\rho_{YZ}}{\sqrt{(1-\rho_{XZ}^2)(1-\rho_{YZ}^2)}}. \tag{2.13}$$

2.3.3 Partial correlation and linear regression

Linear regression is closely related to partial correlation. We present in this section the definitions and properties necessary to understand their relation.

Assuming linear dependencies, a target variable Y can be predicted by a linear combination of the input vector $\mathbf{X} = (X_1, \ldots, X_n)$ [HTF03]

$$Y = \beta_0 + \sum_{i=1}^{n} X_i\beta_i + \varepsilon, \tag{2.14}$$

where ε represents the noise or random error and is typically assumed to be independent of $\mathbf{X}$ and furthermore $\mathbb{E}(\varepsilon) = 0$. The coefficients $\beta_1, \ldots, \beta_n$ measure the influence of each of the inputs $X_1, \ldots, X_n$ on the target variable. The model is linear in the parameters.

Having m measurements, therefore an m-dimensional vector $\mathbf{X}_i$, the most popular method to estimate the parameters β_i is the **ordinary least squares** method, which aims to minimize the residual sum of squares [HTF03]

$$\mathrm{RSS}(\overline{\boldsymbol{\beta}}) = \sum_{i=1}^{m} \left(Y_i - \overline{\beta}_0 - \sum_{j=1}^{n} X_{ij}\overline{\beta}_j \right)^2 . \tag{2.15}$$

Let $\mathbf{X}$ be the $m \times (n+1)$ matrix where each row is an input vector with a 1 in the first position and $\mathbf{Y}$ the m-dimensional vector of outputs. Then equation (2.15) can be rewritten as

$$RSS(\overline{\boldsymbol{\beta}}) = (\mathbf{Y} - \mathbf{X}\overline{\boldsymbol{\beta}})^T(\mathbf{Y} - \mathbf{X}\overline{\boldsymbol{\beta}}). \tag{2.16}$$

Under the assuming that $\mathbf{X}$ has full column rank, the estimate of $\boldsymbol{\beta}$ can be obtained via

$$\hat{\boldsymbol{\beta}}_{LS} = \underset{\overline{\boldsymbol{\beta}}\in\mathbb{R}^{n+1}}{\operatorname{argmin}}\{(\mathbf{Y} - \mathbf{X}\overline{\boldsymbol{\beta}})^T(\mathbf{Y} - \mathbf{X}\overline{\boldsymbol{\beta}})\} \tag{2.17}$$

$$= \underset{\overline{\boldsymbol{\beta}}\in\mathbb{R}^{n+1}}{\operatorname{argmin}}\{\|\mathbf{Y} - \mathbf{X}\overline{\boldsymbol{\beta}}\|_2^2\} \tag{2.18}$$

$$= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{2.19}$$

Where the L_p-norm of a vector $\mathbf{X} = (X_1, \ldots, X_n)$ is defined as

$$\|\mathbf{X}\|_p = \left(|X_1|^p + \ldots + |X_n|^p\right)^{1/p}, \; p \geq 1.$$

The ordinary least squares estimate of the regression coefficient is closely linked to the partial correlation coefficients [CW93]. Let $\mathbf{X} = (X_1, \ldots, X_n)$ denote an n-dimensional random vector with zero mean. Regressing now each X_i in turn on the remaining variables, that is $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n$, leaves us with the task of estimating $\boldsymbol{\beta}^{(i)} = (\beta_1^{(i)}, \ldots, \beta_{i-1}^{(i)}, \beta_{i+1}^{(i)}, \beta_n^{(i)})$, $i = 1, \ldots, n$. Analogous to equations (2.15) and (2.16) and using the zero mean characteristic to null the intercept, the predictor is obtained via

$$\hat{\boldsymbol{\beta}}^{(i)} = \underset{\overline{\boldsymbol{\beta}}\in\mathbb{R}^{n-1}}{\operatorname{argmin}} \left(X_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} X_j\overline{\beta}_j^{(i)} \right)^2 , \quad i \in \{1, \ldots, n\}.$$

Then it can be derived that [CW93]

$$\hat{\beta}_j^{(i)} = \frac{-\omega_{ij}}{\omega_{ii}}, \tag{2.20}$$

where ω_{ij} is the ij-th element of the concentration matrix (Definition 2.18).

2.4 Conditional independence tests

Depending on the type of data, there might exist reliable tests for conditional independence:

- discrete/categorical data: χ^2 and G,
- continuous: no standard test,
- multivariate Gaussian variables: Fisher's z-transform of partial correlation.

2.4.1 χ^2-test for discrete data

In order to test whether a variable X is independent of variable Y given a third variable Z, this test calculates the probability of making an error when assuming that the two variables are dependent given the data.

Given two random variables X and Y with an associated probability measure p. The null hypothesis states that the outcomes of X and Y are statistically independent:

$$H_0 : p(X = x_i, Y = y_j) = p(X = x_i)p(Y = y_j), \quad \forall x_i, y_j. \tag{2.21}$$

Now, the χ^2 test compares the values of observed and expected frequencies and calculates the sum of normalized squares which is approximately χ^2 distributed [KK51]. In more detail, suppose that the distribution of X has been grouped in r categories and that of Y in c categories. The observations $O = (o_{ij})_{\substack{i\in 1,\dots,r \\ j\in 1,\dots,c}}$ can then be represented by a $r \times c$ contingency table (Table 2.1). The expected frequencies $\hat{\mathbb{E}}_{ij}$ are determined based on

	Y_1	$\cdots$	Y_c	
X_1	o_{11}	$\cdots$	o_{1c}	$o_{1\cdot}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$
X_r	o_{r1}	$\cdots$	o_{rc}	$o_{r\cdot}$
	$o_{\cdot 1}$	$\cdots$	$o_{\cdot c}$	o

Table 2.1: Contingency table.

the null hypothesis. Therefore, assuming independence between the two variables, the expected number of cases in each category is its probability times the total number of cases. Using the maximum likelihood estimator yields:

$$\begin{aligned} \hat{\mathbb{E}}_{ij} &= \hat{p}(X = x_i, Y = y_j) \cdot o \\ &= \hat{p}(X = x_i) \cdot \hat{p}(Y = y_i) \cdot o \\ &= \frac{o_{i\cdot}}{o} \cdot \frac{o_{\cdot j}}{o} \cdot o = \frac{o_{i\cdot} \cdot o_{\cdot j}}{o}. \end{aligned} \tag{2.22}$$

The test statistic is then obtained using these quantities

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{\left(o_{ij} - \hat{\mathbb{E}}_{ij}\right)^2}{\hat{\mathbb{E}}_{ij}} = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{\left(o_{ij} - \frac{o_{i\cdot}\cdot o_{\cdot j}}{o}\right)^2}{\frac{o_{i\cdot}\cdot o_{\cdot j}}{o}} \tag{2.23}$$

The probability that this value would have been reached even if the values were independent is given by

$$p = P\left(\chi^2_{(r-1)\times(c-1)} \geq \chi^2\right), \tag{2.24}$$

where r is the number of rows in the contingency table and c is the number of columns. If p is smaller than the chosen level of significance α, the hypothesis H_0 can be rejected and thus the variables are not independent.

When testing for conditional independence of X and Y given knowledge about a third variable Z, the hypothesis can be defined as follows

$$H_0 : p(X = x_i, Y = y_j | Z = z_k) = p(X = x_i | Z = z_k) p(Y = y_j | Z = z_k), \quad \forall x_i, \forall y_j, \forall z_k.$$

The χ^2 test statistic is computed analogously to before.

2.4.2 G-test for discrete data

In cases where $|o_{ij} - \hat{\mathbb{E}}_{ij}| > \hat{\mathbb{E}}_{ij}$ the approximation to the χ^2-distribution (equation (2.23)) can be improved by using the G-test [McD09]. The test statistic is defined as

$$G = 2\sum_{i=1}^{r}\sum_{j=1}^{c} o_{ij} \ln\left(\frac{o_{ij}}{\hat{\mathbb{E}}_{ij}}\right), \tag{2.25}$$

where $\hat{\mathbb{E}}_{ij}$ as defined in equation (2.22). G is approximately following a χ^2-distribution with $(r-1)\times(c-1)$ degrees of freedom, the same as for the χ^2-test statistic in equation (2.23).

2.4.3 Partial correlation test for multivariate Gaussian data

Assuming multivariate Gaussian variables X and Y, the null hypothesis for zero correlation is

$$H_0 : \hat{\rho}_{XY} = 0, \tag{2.26}$$

where $\hat{\rho}_{XY}$ is the sample correlation. The hypothesis is rejected with significance level α given m samples if

$$\sqrt{m-3} \underbrace{\left|\frac{1}{2}\ln\left(\frac{1+\hat{\rho}_{XY}}{1-\hat{\rho}_{XY}}\right)\right|}_{|\text{Fisher's z-transform}|} > \Phi^{-1}\left(1 - \frac{\alpha}{2}\right), \tag{2.27}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

The partial correlation $\rho_{XY|\mathbf{Z}}$ is zero if and only if X and Y are conditionally independent given $\mathbf{Z}$. In order to test for statistical independence, the null hypothesis is defined as

$$H_0 : \hat{\rho}_{XY|\mathbf{Z}} = 0, \tag{2.28}$$

where $\hat{\rho}_{XY|\mathbf{Z}}$ is the sample partial correlation. Simultaneously to the sample correlation, the hypothesis is rejected with significance level α given m samples if

$$\sqrt{m - |\mathbf{Z}| - 3} \left| \frac{1}{2} \ln \left(\frac{1 + \hat{\rho}_{XY|\mathbf{Z}}}{1 - \hat{\rho}_{XY|\mathbf{Z}}} \right) \right| > \Phi^{-1} \left(1 - \frac{\alpha}{2} \right). \tag{2.29}$$

2.5 Information theoretic background

In Chapter 3, we will introduce methods to infer networks from genomic data. Many of these methods rely on measures from information theory to compute dependencies between variables, namely entropy (Section 2.5.1), mutual information (Section 2.5.2) and interaction information (Section 2.5.3). The definitions and properties in this section are stated for discrete variables.

2.5.1 Entropy

Definition 2.20 *[CT90]*
Let X be a discrete random variable. The entropy of X is defined as

$$H(X) = -\sum_{x} p(x) \log p(x). \tag{2.30}$$

Considering now pairs of random variables.

Definition 2.21 *[CT90]*
Let X_1 and X_2 be two random variables. The joint entropy of X_1 and X_2 is defined as

$$H(X_1; X_2) = -\sum_{x_2} \sum_{x_1} p(x_1, x_2) \log p(x_1, x_2).$$

The conditional entropy of X_1 given X_2 is defined as

$$H(X_1|X_2) = -\sum_{x_2} \sum_{x_1} p(x_1, x_2) \log \frac{p(x_1, x_2)}{p(x_2)}, \; p(x_2) > 0.$$

These definitions can be extended to triplets of variables X_1, X_2 and X_3

$$H(X_1; X_2; X_3) = -\sum_{x_3} \sum_{x_2} \sum_{x_1} p(x_1, x_2, x_3) \log p(x_1, x_2, x_3) \tag{2.31}$$

and

$$H(X_1; X_2|X_3) = -\sum_{x_3}\sum_{x_2}\sum_{x_1} p(x_1, x_2, x_3) \log \frac{p(x_1, x_2, x_3)}{p(x_3)}. \tag{2.32}$$

The following two properties relate the conditional entropy with the joint entropy and formalize its relation with the entropy of one variable.

Property 2.22 *[CT90]*
Let X_1 and X_2 be two random variables. The conditional entropy is then defined as

$$H(X_1|X_2) = H(X_1; X_2) - H(X_2).$$

Property 2.23 *[CT90]*
Let X_1 and X_2 be two random variables. The knowledge of one variable reduces the entropy of the other with equality if and only if X_1 and X_2 are independent

$$H(X_1|X_2) \leq H(X_1).$$

2.5.2 Mutual information

Definition 2.24 *[CT90]*
Given two variables X_1 and X_2, the mutual information between them is defined as

$$I(X_1; X_2) = \sum_{x_2}\sum_{x_1} p(x_1, x_2) \log \frac{p(x_1, x_2)}{p(x_1)p(x_2)}$$

The mutual information $I(X_1; X_2) \geq 0$ with equality if and only if X_1 and X_2 are independent random variables.

Mutual information can be stated in terms of entropy

$$\begin{aligned} I(X_1; X_2) &= H(X_1) - H(X_1|H_2) \\ &= H(X_2) - H(X_2|X_1) \\ &= H(X_1) + H(X_2) - H(X_1; X_2). \end{aligned}$$

For sets of three variables X_1, X_2 and X_3 the mutual information is smallest for the pair which is conditionally independent of the third.

Property 2.25 *(Data processing inequality (DPI))[CT90]*
Let X_1, X_2, X_3 be three random variables. If X_1 and X_2 are conditionally independent given X_3, then

$$I(X_1; X_2) \leq \min\{I(X_1; X_3), I(X_2; X_3)\}.$$

Definition 2.26 *[CT90]*
Given three variables $X_1, X_2, X_3 \in \mathbf{X}$, the conditional mutual information of X_1 and X_2

given X_3 is defined as

$$I(X_1; X_2|X_3) = \sum_{x_3} \sum_{x_2} \sum_{x_1} p(x_1, x_2, x_3) \log \frac{p(x_3)p(x_1, x_2, x_3)}{p(x_1, x_3)p(x_2, x_3)}. \tag{2.33}$$

Expressing equation (2.33) in terms of entropy

$$I(X_1; X_2|X_3) = -H(X_3) + H(X_1; X_3) + H(X_2; X_3) - H(X_1; X_2; X_3).$$

And equation (2.33) can be furthermore equivalently expressed as

$$I(X_1; X_2|X_3) = \sum_{x_3} \sum_{x_2} \sum_{x_1} p(x_1, x_2, x_3) \log \frac{p(x_1, x_2|x_3)}{p(x_1|x_3)p(x_2|x_3)} \tag{2.34}$$

and in conditional entropies

$$I(X_1; X_2|X_3) = H(X_1|X_3) + H(X_2|X_3) - H(X_1; X_2|X_3).$$

It holds that $I(X_1; X_2|X_3) \geq 0$, but unlike entropy the conditional mutual information is neither always larger nor always less than the mutual information between X_1 and X_2.

Property 2.27 *(Chain rule mutual information)[CT90]*
Given a subset of variables $X, Y_1, \ldots, Y_n \subseteq \mathbf{X}$, the mutual information of X with the remaining variables $Y_1, \ldots, Y_n$ can be obtained using the following chain rule

$$I(X; Y_1, \ldots, Y_n) = I(X; Y_1) + \sum_{i=2}^{n} I(X; Y_i|Y_1, \ldots, Y_{i-1}).$$

Furthermore, it can be deduced that

$$I(X; Y_1, \ldots, Y_n) = I(X; Y_1, \ldots, Y_{n-1}) + I(X; Y_n|Y_1, \ldots, Y_{n-1}).$$

Using the chain rule for three variables X_1, X_2 and X_3, we obtain the following equations

$$\begin{aligned} I(X_1; X_2, X_3) &= I(X_1; X_2) + I(X_1; X_3|X_2) \\ &= I(X_1; X_3) + I(X_1; X_2|X_3). \end{aligned} \tag{2.35}$$

and together with the definition of conditional mutual information, Definition 2.26,

$$\begin{aligned} I(X_1; X_2, X_3) &= I(X_1; X_2) + I(X_1; X_3|X_2) \\ &= H(X_1) + H(X_2; X_3) - H(X_1; X_2; X_3). \end{aligned} \tag{2.36}$$

2.5.3 Interaction information

The concept of mutual information can be extended from considering pairs of variables to sets of variables. In this case, we will no longer investigate the information of one

variable with respect to the other but take also into consideration conditional information given the remaining variables.

Definition 2.28 *[McG54, Bel03]*
*The **interaction information** between n random variables $X_1, \ldots, X_n$ is defined as*

$$\mathcal{I}(X_1, \ldots, X_n) := \sum_{k=1}^{n} \sum_{\mathbf{X_S} \subset \{X_1, \ldots, X_n\} : |\mathbf{X_S}| = k} (-1)^{k+1} H(\mathbf{X_S}), \tag{2.37}$$

where $H(\mathbf{X_S})$ denotes the entropy of the set of k random variables in $\mathbf{X_S}$.

The most widely considered case is the interaction information of three variables X_1, X_2 and X_3. In this case, equation (2.37) can be simply stated as

$$\begin{aligned}&\mathcal{I}(X_1, X_2, X_3) = \\ &H(X_1) + H(X_2) + H(X_3) - H(X_1; X_2) - H(X_1; X_3) - H(X_2; X_3) + H(X_1; X_2; X_3).\end{aligned}$$

When considering three variables X_1, X_2 and X_3, the interaction information is symmetric in the three variables

$$\begin{aligned}\mathcal{I}(X_1, X_2, X_3) &= I(X_1; X_2) - I(X_1; X_2 | X_3) \\ &= I(X_1; X_3) - I(X_1; X_3 | X_2) \\ &= I(X_2; X_3) - I(X_2; X_3 | X_1).\end{aligned}$$

The interaction information $\mathcal{I}(X_1, X_2, X_3)$ can be equivalently expressed as [Ana07]

$$\mathcal{I}(X_1, X_2, X_3) = I(X_1; X_2) + I(X_1; X_3) - I(X_1; X_2, X_3).$$

Given the symmetry in the three variables, it can be deduced that

$$\begin{aligned}\mathcal{I}(X_1, X_2, X_3) &= I(X_1; X_2) + I(X_1; X_3) - I(X_1; X_2, X_3) \\ &= I(X_1; X_3) + I(X_2; X_3) - I(X_3; X_1, X_2) \\ &= I(X_1; X_2) + I(X_2; X_3) - I(X_2; X_1, X_3).\end{aligned}$$

2.6 Estimators

For any method that uses entropy and/or mutual information without knowing the underlying probability distribution, the desired quantities have to be estimated from the available data. Different approaches exist, which can be mainly categorized into plug-in techniques and direct estimation techniques. In the former approach, the probability distribution is estimated using the maximum likelihood estimator or one of its derivatives. This estimated probability distribution is then used in lieu of the actual probability distribution in the entropy definition, Definition 2.20. Whereas for direct estimation an

underlying normal distribution is assumed for which there exist closed form definitions of entropy and mutual information.

2.6.1 Plug-in methods

The general idea of plug-in methods is to estimate the probability distribution of the random variable based on the available data and to plug this estimate into equation (2.30). The simplest estimate of the probability function is the maximum likelihood estimator. In this approach the number of occurrences of each event is evaluated and then divided by the total number of events. The data is assumed to be discrete.

Definition 2.29 *[Pan03, Hau06]*
*Let the number of possible outcomes of the experiment be denoted by k ($= |\mathcal{X}|$). If the experiment is renewed m times, the outcome x_i will occur $\#(x_i)$ times ($m = \sum_{i=1}^{k} \#(x_i)$). The **maximum likelihood estimator** (also known as empirical estimator) of the probability distribution p is then defined as*

$$\hat{p}_i^{ML} := \hat{p}^{ML}(x_i) := \frac{\#(x_i)}{m}.$$

Property 2.30 *[Mil55]*
The asymptotic bias of the entropy maximum likelihood estimator $\hat{H}^{ML} := H(\hat{p}^{ML}) = -\sum \hat{p}_i^{ML} \log \hat{p}_i^{ML}$ amounts to

$$bias(\hat{H}^{ML}) = -\frac{k' - 1}{2m}, \tag{2.38}$$

where k' is the number of bins with non-zero probability.

The Miller-Madow entropy estimator has been proposed to counteract this bias [Pan03]. It is a combination of the maximum likelihood probability distribution estimator with a bias correction term, namely the asymptotic bias (equation (2.38)).

Definition 2.31 *[Pan03]*
*The **Miller-Madow entropy estimator** of entropy is defined as*

$$\hat{H}^{MM} := H(\hat{p}^{ML}) + \frac{k' - 1}{2m}.$$

Another possibility to reduce the estimation bias introduced by the maximum likelihood estimator is a shrinkage estimator [SS05b]. The basic idea is to combine two estimators to balance their flaws in order to reduce the mean squared error: one estimator exhibiting low bias and high variance (maximum likelihood) and a second estimator having high bias and no variance. This combination improves the performance compared to both taken alone [Hau06].

Definition 2.32 *[SS05b]*
Under the assumptions of Definition 2.29, the ***shrinkage estimator*** *of the probability distribution p is defined as*

$$\hat{p}_i^{shrink} = \lambda \frac{1}{k} + (1-\lambda)\frac{\#(x_i)}{m},$$

where $\lambda \in [0,1]$ *is the shrinkage parameter.*

The optimal shrinkage parameter with respect to the minimization of the mean squared error has been evaluated in [SS05b].

Property 2.33 *[SS05b]*
The optimal shrinkage parameter $\hat{\lambda}^*$*, minimizing the mean square error is*

$$\hat{\lambda}^* = \frac{k(m^2 - \sum_i \#(x_i)^2)}{(m-1)(m\sum_i \#(x_i)^2 - m^2)}.$$

2.6.2 Direct methods

In this section we present two different direct approaches to entropy estimation which do not rely on estimating the probability distribution first. For the first approach, the data is assumed to follow a Gaussian distribution. For the second, the entropy is estimated using a function of the underlying distribution.

2.6.2.1 Assuming Gaussianity

Let

$$f_X(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp^{(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))}.$$

be the density of a multivariate Gaussian variable X with mean μ and covariance matrix Σ.

The definitions of entropy and mutual information can be stated analoguously for continuous random variables (Appendix B.2).

The entropy of the multivariate Gaussian variable X is given by [CT90]

$$H(X) = \frac{1}{2}\ln\{(2\pi e)^n |\Sigma|\},$$

where $|\Sigma|$ is the determinant of the covariance matrix [Hay94]. The mutual information between the two variables X and Y is then

$$I(X;Y) = -\frac{1}{2}\ln(1-\rho_{XY}^2),$$

where ρ_{XY} denotes the Pearson's correlation between the two variables X and Y.

Furthermore, the interaction information can be expressed as follows

$$\begin{aligned} \mathcal{I}(X_1, X_2, X_3) &= I(X_1, X_2) - I(X_1, X_2|X_3) && (2.39) \\ &= -\frac{1}{2} \ln \frac{\left(1 - \rho^2_{X_1X_3}\right)\left(1 - \rho^2_{X_1X_2}\right)\left(1 - \rho^2_{X_2X_3}\right)}{1 + 2\rho_{X_1X_2}\rho_{X_1X_3}\rho_{X_2X_3} - \rho^2_{X_1X_2} - \rho^2_{X_1X_3} - \rho^2_{X_2X_3}} && (2.40) \end{aligned}$$

using the equality $I(X_1, X_2|X_3) = -\frac{1}{2} \ln \left[1 - \rho^2_{X_1X_2|X_3}\right]$, where

$$\rho_{X_1X_2|X_3} = \frac{\rho_{X_1X_2} - \rho_{X_1X_3}\rho_{X_2X_3}}{\sqrt{1 - \rho^2_{X_1X_3}}\sqrt{1 - \rho^2_{X_2X_3}}} \tag{2.41}$$

is the partial correlation of X_1 and X_2 given a third variable X_3. Thus, under the assumption of Gaussianity, the interaction information can be estimated from combinations of bivariate estimations.

2.6.2.2 Estimating the entropy as function of p

The second method tries to estimate a function F dependent of p. Let this function $F(p)$ be the entropy $H(p)$. Following the Bayesian approach to statistics and taking the prior probability to be Dirichlet with parameter a, the entropy can be estimated via

$$\begin{aligned} \hat{H}^{Dir} = \mathbb{E}(H) &= -\sum_{i=1}^{k} \mathbb{E}(p_i \log(p_i)) \\ &= \frac{1}{m + ka} \sum_{i=1}^{k} (y_i + a)(\psi(m + ka + 1) - \psi(y_i + a + 1)), \end{aligned}$$

where $\psi(z) = \frac{\mathrm{d} \ln \Gamma(z)}{\mathrm{d}z}$, the Digamma function [WNR+07].

Considering now $F(p) = p$, then the Dirichlet estimate with parameter a is given by $\frac{\#x_i + a}{m + ka}$. In the literature various values for the Dirichlet parameter a have been proposed

- $a = 0$ equals the maximum likelihood estimator,
- $a = \frac{1}{2}$ Jeffrey' prior [KT81],
- $a = \frac{1}{k}$ Schürmann-Grassberger estimator [SG02].

Another possible choice is the NSB prior [NSB02].

2.6.3 Estimation in case of continuous data

All estimators - except the estimator assuming Gaussianity - require discrete data. However in many applications the given data is continuous. The easiest way to apply the

estimation techniques to continuous data is to discretize the given data. However, this approach may lead to a loss of the information that exists in the data [MLB08]. Therefore, several approaches to entropy estimation have been introduced which directly use the continuous data. This section will first present the most common discretization techniques and then mention some estimators for continuous data.

2.6.3.1 Discretization methods

In order to apply the estimation techniques described in Sections 2.6.1 and 2.6.2 to continuous data, it must be discretized first. The two most widely used techniques split the variable's domain into equal width or the equal frequency subintervals [DKS95].

Equal width : this discretization method partitions the domain of X into $|\mathcal{X}|$ subintervals of equal size. As a consequence, the number of data points in each bin is likely to be different.

Equal frequency : this method divides the domain of $\mathcal{X}$ into $|\mathcal{X}|$ subintervals, each containing the same number of data points. It follows that subinterval sizes are typically different.

The term *subinterval* is also called a *bin*. The number of subintervals should be chosen so that all bins contain a significant number of samples. In [YW03] the authors propose to use $|\mathcal{X}| = \sqrt{m}$, where m is the total number of samples.

Another extension to classical binning techniques is the B-spline approach [DSSK04] in which each data point may be assigned to multiple bins. The final estimator is then calculated using weighting functions which are defined as B-spline functions with a chosen degree d. The degree $d = 1$ corresponds to the classical binning.

2.6.3.2 Estimation without discretization

Facing the possible loss of information that might occur by applying discretization methods to continuous data, several more complex estimators for continuous data have been developed.

- The kernel density estimation [MRL95, SKD$^+$02] tries to estimate the probability density function by additively assigning kernel functions onto each observation. The estimator depends on a smoothing parameter (also called window width) which has to be fixed. Several studies have been carried out to determine the optimal width under certain conditions, for example the optimal Gaussian bandwidth has been determined in [Sil86].
- Furthermore, a strategy of binless entropy estimation for continuous distribution in a Euclidean vector space has been presented in [Vic02].

Chapter 3

State-of-the-art

In this chapter we present state-of-the-art techniques to infer networks using graphical representations of dependencies. We roughly group the techniques in terms of how they model the dependencies between the variables: a) Gaussian graphical models (GGMs) and regression techniques, b) Bayesian network inference, c) feature selection techniques based on correlation and information theory, d) ensemble algorithms and e) kernel techniques. In particular, methods are distinguished by their ability to i) infer directed networks, ii) include prior knowledge and iii) whether an ensemble/bagging step is included. Figure 3.1 shows an overview of the different methods and the described classification.

We start this chapter by introducing Gaussian graphical models (Section 3.1) in which conditional independencies are represented by undirected edges in the network. A couple of algorithms have been developed to infer directed networks, *GeneNet* [ORS07], and to include prior knowledge, *Modified BoostiGraph* [ADH09].

The second class of algorithms infer Bayesian networks (Section 3.2). There are two main approaches to infer Bayesian networks: constraint-based and score-based. While the former set of techniques is typically data-driven, the latter allows the integration of prior knowledge. Hybrid algorithms are partly score-based and partly constraint-based and allow for prior integration in the score-based part of the algorithm. Bayesian networks are computationally very heavy and are only applicable when the data sets do not contain more than a few hundred variables [FLNP00].

The need for algorithms able to infer networks from data sets with thousands of variables led to the development of feature selection methods originally using correlation to select relevant variables and later measures based on information theory (Section 3.3). Most of these methods infer undirected networks by computing a relevance score for each edge based on mutual information. There are however a couple of extensions to also infer directed edges, namely *MI3* [LHW08] and *SRI* [WLW+09] using scores based on interaction information.

Feature selection methods for data sets with high numbers of variables and low numbers of samples tend to be unstable [HY10]. More precisely, the addition or removal of few samples may lead to significantly different networks. To improve feature selection techniques with respect to this problem, a few bootstrap/ensemble extensions to existing methods have been introduced. We present examples of such techniques in Section 3.4. We discuss *GENIE3* [HTIWG10], a tree based ensemble method and *Crowd Wisdom* [MCKf$^+$12] an ensemble of different network inference methods applied to the same data set with the goal of building a better consensus network.

Finally we review methods representing data with kernels. This representation allows to combine multiple data sources and thus improving the quality of the inferred networks.

Conventions

Whenever an algorithm states 'such that $X \perp\!\!\!\perp Y|\mathbf{S}$', it should be understood as: a statistical test is carried out and the null hypothesis of conditional dependence is rejected with a p-value smaller than a given threshold.

The 'data' considered in the presented algorithms, unless stated otherwise, is an $m \times n$ matrix in which the rows correspond to samples and the columns to the variables. For instance in the case of patient related data, each sample corresponds to one patient.

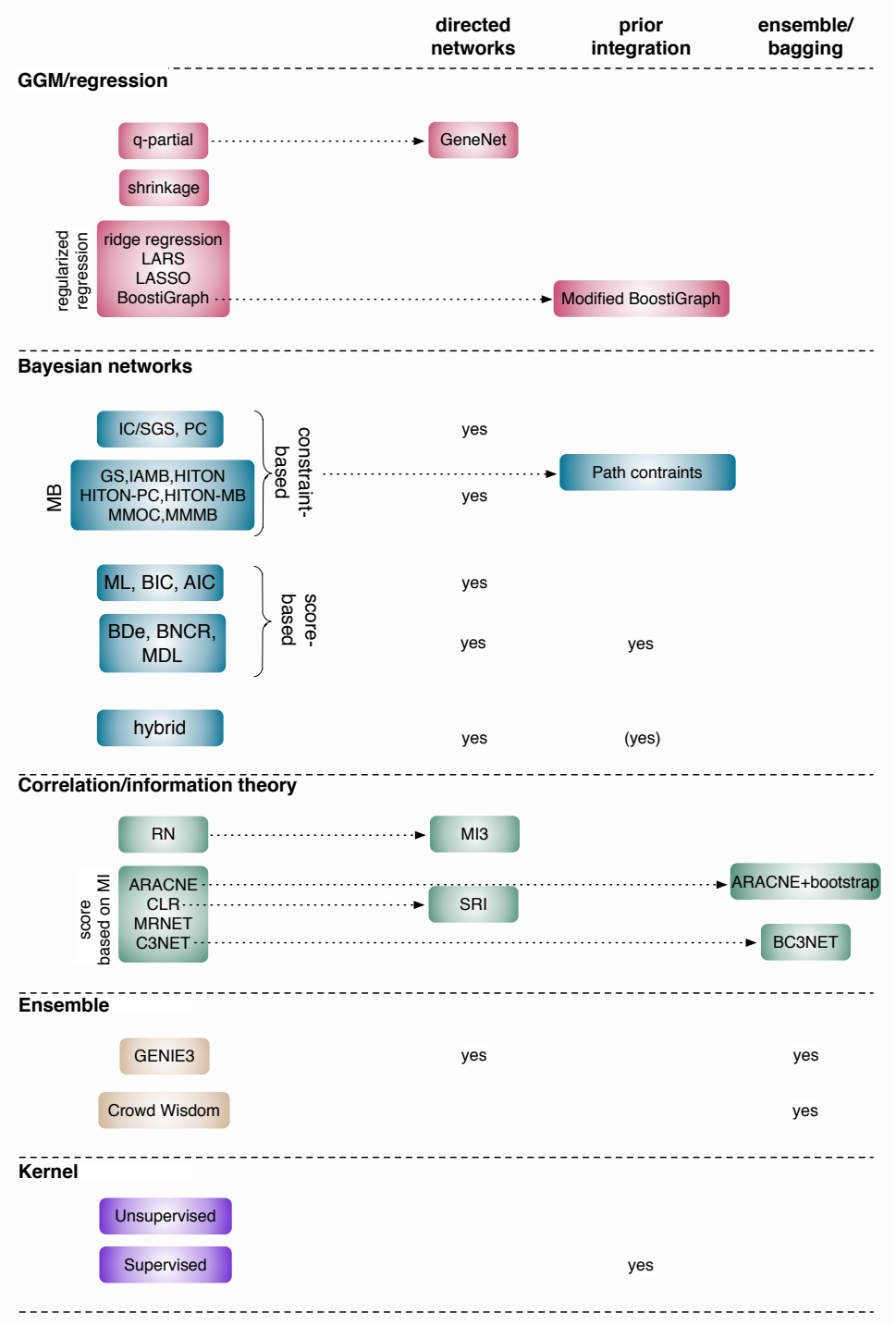


Figure 3.1: Overview of the methods presented in the state-of-the-art section. Whenever entry in **directed networks** and/or **prior integration** and/or **ensemble/bagging** is 'yes', it is an intrinsic property of the method, otherwise the extension's name is provided.

3.1 Gaussian graphical models

Assuming that n continuous variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ follow a multivariate normal distribution, a network can be derived by determining the set of conditional independencies [Lau96]. A missing edge between two variables X_i and X_j in these networks corresponds to the conditional independence relation

$$X_i \perp\!\!\!\perp X_j | (\mathbf{X} \setminus \{X_i, X_j\}), \tag{3.1}$$

that is X_i is independent of X_j given all other variables (Definition 2.3). Under the assumption that $\mathbf{X}$ follows a multivariate Gaussian distribution, equation (3.1) is equivalent to the partial correlation being zero. The general inference procedure can be summarized as follows. In a first step, the correlation matrix is estimated. In the second step, the partial correlation coefficients are estimated. This is equivalent to estimating the inverse covariance matrix, i.e. the concentration matrix (Definition 2.18). Finally it is tested whether an entry is different from zero. In this case an edge is added to the network [SS05b]. However, the sample concentration matrix requires the sample covariance matrix to be positive definite which only holds with probability one if and only if the number of variables is lower than the number of samples [Dyk70, SS05a, Kon09]. As this thesis is focused on data sets with a high variable/sample ratio, different techniques trying to overcome this problem are presented in the following:

- shrinkage techniques (Section 3.1.2.1),
- limited order partial correlation graphs (Section 3.1.2.2) and
- regularized regression (Section 3.1.2.4).

A few extensions exist to extend Gaussian graphical models to make possible i) the inference of directed networks and to ii) integrate prior knowledge. We will present these in Sections 3.1.3 and 3.1.4, respectively.

3.1.1 Basic definitions and properties

Definition 3.1 *[Lau96]*
Assuming that $\mathbf{X}$ *follows an n-variate normal distribution with mean* $\mu = \mathbb{E}(\mathbf{X})$ *and covariance matrix* $\Sigma = var(\mathbf{X})$. *Then the* ***Gaussian graphical model*** *for* $\mathbf{X}$ *is given by the undirected graph* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ *which obeys the undirected pairwise Markov property (Property 2.6):*

$$X_i \not\sim X_j \Rightarrow X_i \perp\!\!\!\perp X_j | (\mathbf{X} \setminus \{X_i, X_j\}). \tag{3.2}$$

A covariance matrix Σ for which the concentration matrix $\Omega = \Sigma^{-1}$ is well defined is called **regular** or **invertible**.

Property 3.2 *[Lau96]*
Assuming that $\mathbf{X}$ *follows an n-variate normal distribution with mean* $\mu = \mathbb{E}(\mathbf{X})$ *and*

regular covariance matrix $\Sigma = var(\mathbf{X})$*. Then, for* $X_i, X_j \in \mathbf{X}$ *with* $X_i \neq X_j$

$$X_i \perp\!\!\!\perp X_j | (\mathbf{X} \setminus \{X_i, X_j\}) \iff \omega_{ij} = 0,$$

where $\Omega = \Sigma^{-1} = \{\omega_{ij}\}_{X_i, X_j \in \mathbf{X}}$ *is the concentration matrix of the distribution.*

Using this property, the relation described in equation (3.2) between the graph $\mathcal{G}$ and the concentration matrix can be equivalently expressed as

$$X_i \not\perp\!\!\!\perp X_j \Rightarrow \omega_{ij} = 0, \qquad \forall X_i, X_j \in \mathbf{X}.$$

Thus the task of determining the graph $\mathcal{G}$ from the given data consists in identifying the zero-valued entries in the inverse covariance matrix, as these correspond to the conditional independencies among the variables [Whi90, Kon09]. Thus, the general approach for the reconstruction of a GGM is described in Algorithm 1 [SS05a].

Algorithm 1: Inferring a GGM

Input: data, vertex set $\mathbf{V}$, p-value
Output: undirected graph

1.1 Estimate the correlation matrix;
1.2 Estimate the partial correlation coefficients from this matrix;
1.3 Employ statistical tests (Section 2.4) to determine which entries are statistically significant different from zero using the user-defined p-value;
1.4 **foreach** *variable pair* X_i, X_j *in* $\mathbf{V}$ **do**
1.5 | **if** *partial correlation coefficient* $\omega_{ij} \neq 0$ **then**
1.6 | | Infer an edge $X_i - X_j$;
1.7 | **end**
1.8 **end**

Methods tackling the problematic step of estimating the partial correlation will be presented in the following.

3.1.2 Estimating the sample covariance matrix for large n, small m

Different techniques have been proposed to obtain an estimate of the covariance matrix for data sets with high number of variables and low number of samples. The first one presented in this section is the *shrinkage* technique which combines two estimators into an overall better estimator. The second possibility is to focus on lower order partial correlations (Section 3.1.2.2) and thus obtaining a graph which is a superset of the full order partial correlation graph. The last set of methods exploits the link between partial correlation and linear regression coefficients (Section 3.1.2.4).

3.1.2.1 Shrinkage estimation

The main idea behind shrinkage estimators is to combine two separate estimators of the desired quantity with opposing properties: one with high bias and low variance, say $\mathbf{T}$, and the second one with low bias but high variance $\mathbf{U}$. Subsequently the **linear shrinkage estimate** is then obtained by linearly combining $\mathbf{U}$ and $\mathbf{T}$ with a weighting factor λ

$$\lambda\hat{\mathbf{T}} + (1-\lambda)\hat{\mathbf{U}}, \quad \lambda \in [0,1]. \tag{3.3}$$

$\hat{\mathbf{U}}$ is known as **unrestricted estimate** and $\hat{\mathbf{T}}$ as **shrinkage target**. This combination of estimators often outperforms the individual estimators in terms of accuracy [SS05b].

In order to estimate the covariance matrix, the unbiased empirical covariance matrix

$$\hat{\mathbf{S}} = \frac{1}{m-1}\mathbf{X}^T\mathbf{X}$$

is used as unrestricted estimate $\hat{\mathbf{U}}$. The biased shrinkage target is defined as

$$\hat{\mathbf{t}}_{ij} = \begin{cases} \hat{\mathbf{s}}_{ii} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

By using a positive semi-definite matrix for the unrestricted part and a positive definite matrix as target, the convex combination (3.3) will always result in a positive definite matrix which is an intrinsic property of the true covariance matrix.

A key problem in shrinkage estimation is the question of how to fix the shrinkage parameter λ. A minimal mean squared error (MSE) can be guaranteed following [LW03] when deriving the optimal shrinkage parameter λ^* analytically [SS05b]. Using the optimal shrinkage parameter λ^*, the optimal shrinkage estimator of the covariance matrix can be computed

$$\hat{\Sigma}^* = \lambda^*\hat{\mathbf{T}} + (1-\lambda^*)\hat{\mathbf{S}}.$$

The optimal shrinkage estimator of the concentration matrix is then obtained by inverting the estimated covariance matrix

$$\hat{\Omega}^* = \left(\hat{\Sigma}^*\right)^{-1}$$

and thus allowing the computation of the shrinkage estimate of the partial correlations (Definition 2.19)

$$\hat{\rho}^*_{ij|K} = \frac{-\hat{\omega}^*_{ij}}{\sqrt{\hat{\omega}^*_{ii}\hat{\omega}^*_{jj}}},$$

where $K = \mathbf{X} \setminus \{X_i, X_j\}$.

It has been in shown in [KB09] that this optimal shrinkage intensity estimator is biased and a bootstrap approach was used to derive a bias-corrected estimation.

3.1.2.2 Limited-order partial correlations

Computing the correlation between two variables X_i and X_j conditioned on all other variables in the data set determines the full conditional relationship between them. These full conditional relationships can be estimated using low order partial correlation approximations.

Definition 3.3 *[CR06]*
*For an integer $0 \leq q \leq (n-2)$, the **q-partial graph**, denoted by $\mathcal{G}^{(q)}$, is defined as undirected graph, in which X_i and X_j are not connected in $\mathcal{G}^{(q)}$ if and only if there exists a subset $\mathbf{X}_K \subseteq \mathbf{X}$ with $|\mathbf{X}_K| \leq q$ and $X_i, X_j \notin \mathbf{X}_K$ such that*

$$X_i \perp\!\!\!\perp X_j | \mathbf{X}_K.$$

Special cases include the concentration graph $\mathcal{G}^{(n-2)}$ and the covariance graph $\mathcal{G}^{(0)}$.

The most basic approach is to consider only partial correlations with one conditioning variable. In [WZV+04, WB06] the partial correlation $\rho_{ij|k}$ for all triplets of variables X_i, X_j and X_k is computed. An edge between X_i and X_j is inferred if and only if

$$\rho_{ij} \neq 0 \quad \text{and} \quad \rho_{ij|k} \neq 0 \quad \forall X_k \in \mathbf{X} \setminus \{X_i, X_j\}.$$

It has been shown that for faithful graphical models (Definition 2.7) this graph contains at least all edges of the full conditional independence graph [WB06]

$$\mathcal{G} \subseteq \mathcal{G}^{(1)}.$$

One step further, that is partial correlations up to order two are used in [FBHM04] to remove indirect connections from an initial correlation network. In a first step, those edges with zero partial correlation conditioned on one variable are removed. And in a second step, those edges with zero partial correlation conditioned on pairs of variables are deleted from the network.

3.1.2.3 q-partial graphs

In the previous paragraph we presented methods based on q-order partial correlation with $q \leq 2$. These methods ultimately try to construct a network in which the edges represent full order conditional correlations. To support this reasoning, it has been shown in [CR06] that these methods become more useful with growing q. The authors show the order of inclusion for different orders of graphs $\mathcal{G}^{(q)}$ and $\mathcal{G}^{(r)}$ with $r \leq q$.

Property 3.4 *[CR06]*
Let $\mathcal{G}^{(q)}$ and $\mathcal{G}^{(r)}$ be the q-partial and the r-partial graph of $\mathbf{X}$, respectively. If $r \leq q$ then

$$\mathcal{G} \subseteq \mathcal{G}^{(q)} \subseteq \mathcal{G}^{(r)}.$$

In the low sample, high variable setup, it is not feasible to test all possible subsets for non-zero partial correlation. However, by applying a randomization procedure, the authors of [CR06] show that it is possible to remove some higher order partial correlation edges in the network and thus approaching $\mathcal{G}$ further than a lower order partial correlation graph could.

Inspired by the PC-algorithm (Algorithm 4, [SGS01]) and using Property 3.4, [KB08] proposed a nested procedure to obtain the q-partial graph with higher values for q.

3.1.2.4 Regularized regression

As presented in Section 2.3.3, there is a close connection between partial correlation and linear regression. In practice, equation (2.17) will be modified such that a regularization term on the β coefficients is added to the term that is being minimized. Choices for this penalty term are L_1 or L_2 norm or a combination thereof.

Ridge regression [HK70] uses the L_2 norm and its solutions are

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_2\},\ \lambda > 0 \tag{3.4}$$

$$= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2\},\ \lambda > 0 \tag{3.5}$$

$$= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y},\ \lambda > 0. \tag{3.6}$$

The problem with ridge regression is that in general no coefficient is equal to zero which makes any model different to interpret. The least absolute shrinkage and selection operator (**LASSO**) [Tib96] uses the L_1 norm to overcome this problem.

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1\},\ \lambda > 0$$

$$= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\},\ \lambda > 0.$$

It has been shown that the LASSO approach has difficulties dealing with highly correlated variables and will most probably only select one out of this group of correlated variables [ZH03]. Therefore, the authors proposed to combine both penalties for the

elastic net method [ZH03].

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}) + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2\},\ \lambda_1,\lambda_2 > 0 \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{\|\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2\},\ \lambda_1,\lambda_2 > 0.\end{aligned}$$

Contrary to the possibility of computing $\hat{\boldsymbol{\beta}}$ via a closed-form formula, see equation (3.6), both the lasso and the elastic net coefficients have to be computed numerically. This can be achieved via numerical analysis methods such as the least angle regression (**LARS**) [EHJT04].

Another possibility to determine the regression coefficients is the **BoostiGraph** algorithm (Algorithm 2), presented in [ADH09]. The general procedure in boosting algorithms is to start with a solution and to update it in each iteration [BB03]. This algorithm initializes the $\hat{\beta}_j^{(i)}$ with the product $X_j^T X_i$. In each boosting iteration, the maximal $|\hat{\beta}_j^{(i)}|$ is selected. This can be interpreted as adding the undirected edge with the highest score to the network.

Algorithm 2: Boosting lasso regression: BoostiGraph

Input: data, vertex set $\mathbf{V}$, number of boosting iterations T, step-length $0 < \eta \leq 1$
Output: $\hat{\beta}^{(1),T}, \ldots, \hat{\beta}^{(n),T}$

for $i \in 1, \ldots, n$ **do**
  Standardize each column vector of the data to have zero mean and unit standard deviation;
end
Set $t = 0$, $\hat{\beta}_j^{(i),t} = 0$, $X_i^* = X_i, \forall i, j$;
Fit univariate regressions: $\hat{\beta}_j^{(i)} = X_j^T X_i^*$, $\forall i \neq j$;
`/* Boosting iterations */`
while $t < T$ **do**
  Find best predictor: $\hat{i}, \hat{j} = \operatorname{argmax}_{i,j}\{|\hat{\beta}_j^{(i)}|\}$;
  Boosting update: $\hat{\beta}_{\hat{j}}^{(\hat{i}),t+1} = \hat{\beta}_{\hat{j}}^{(\hat{i}),t} + \eta\hat{\beta}_{\hat{j}}^{(\hat{i})}$ and $X_{\hat{i}}^* = X_{\hat{i}}^* - \eta X_{\hat{i}}\hat{\beta}_{\hat{j}}^{(\hat{i})}$;
  Update $\hat{\beta}_j^{(\hat{i})} = X_j^T X_{\hat{i}}^*$ for $j \neq \hat{i}$;
  $t = t + 1$;
end

3.1.3 Extension to directed networks

In the first part of this section, different methods have been presented to infer undirected networks for data sets with low numbers of samples and high numbers of variables. Now, an approach is presented which extends GGMs to infer directed networks [ORS07]. As starting point, a partial correlation network is constructed and subsequently converted into a partially directed network.

Making use of the direct link between the regression parameters $\beta_j^{(i)}$ and the partial correlation between X_i and X_j presented in Section 2.3.3, equation (2.20) and the definition of partial correlation, equation (2.19) (as a reminder: $\rho_{ij|K} = \frac{-\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}$ and $\beta_j^{(i)} = \frac{-\omega_{ij}}{\omega_{ii}}$)

$$\beta_j^{(i)} = \frac{-\omega_{ij}}{\omega_{ii}} = \rho_{ij|K}\sqrt{\frac{\omega_{jj}}{\omega_{ii}}}.$$

where $\mathbf{X}_K = \mathbf{X} \setminus \{X_i, X_j\}$. This can be now expanded using variances of X_i and X_j:

$$\sigma_{ii} := var(X_i) = cov(X_i, X_i)$$

and analogous σ_{jj}, such that

$$\begin{aligned}\beta_j^{(i)} &= \rho_{ij|K}\sqrt{\frac{\omega_{jj}\frac{\sigma_{jj}}{\sigma_{jj}}}{\omega_{ii}\frac{\sigma_{ii}}{\sigma_{ii}}}} \\ &= \rho_{ij|K}\sqrt{\frac{\frac{\omega_{jj}}{\sigma_{jj}}}{\frac{\omega_{ii}}{\sigma_{ii}}}}\sqrt{\frac{\sigma_{jj}}{\sigma_{ii}}}.\end{aligned} \tag{3.7}$$

The quotient $\frac{\omega_{ii}}{\sigma_{ii}}$ is known as standardized partial variance (SPV) of variable X_i. Equation (3.7) is used as basis for the inference algorithm in [ORS07]. Starting with adding edges whenever $\rho_{ij|K}$ is non-zero and in a second step the orientation of these edges is deduced for those undirected edges for which the logarithmic value of $\sqrt{\frac{\omega_{jj}}{\sigma_{jj}}/\frac{\omega_{ii}}{\sigma_{ii}}}$ is non-zero. The remaining edges remain undirected. Whenever $\log\left(\sqrt{\frac{\omega_{jj}}{\sigma_{jj}}/\frac{\omega_{ii}}{\sigma_{ii}}}\right)$ is non-zero, the edge is oriented from the variable with the larger SPV to the variable with the smaller SPV because the SPV is considered to be a measure of exogeneity.

This algorithm has been implemented in the CRAN/R package *GeneNet* [SORS09].

3.1.4 Including prior knowledge

The integration of prior knowledge with GGM inference has been introduced as extension to the boosting algorithm presented in Section 3.1.2.4 [ADH09]. Prior knowledge about edges that are present in the true network and those that are absent can be represented

by a weighted, symmetric adjacency matrix $\mathbf{P} = (p_{ij})_{i,j \in 1,\dots,n}$ such that

$$p_{ij} = \begin{cases} \in]0.5, 1] & \text{if } i \neq j \text{ and } X_i - X_j \text{ known} \\ \in [0, 0.5[& \text{if } i \neq j \text{ and } X_i \neq X_j \text{ known} \\ 0 & \text{if } i = j \\ 0.5 & \text{otherwise.} \end{cases}$$

The difference to the original BoostiGraph algorithm is the way the 'best' predictor $\hat{i}, \hat{j}$ is now selected taking the prior knowledge into account

$$\hat{i}, \hat{j} = \underset{ij}{\text{argmax}}\{\log(\text{score}_{ij})\}.$$

Where the score is defined as follows

$$\log(\text{score}_{ij}) = -\frac{1}{2}(X_i^* - X_j \hat{\beta}_j^{(i)})^T (X_i^* - X_j \hat{\beta}_j^{(i)}) + \log \frac{p_{ij}}{1 - p_{ij}}.$$

Once an edge is added its prior score is reset to 0.5.

3.2 Bayesian networks

A *Bayesian network* (Section 2.2.2.1) is a directed acyclic graph coding a bijective mapping between a graph an a probability distribution.

When learning Bayesian networks from data, two tasks need to be accomplished: learning the network structure and learning the probabilities [CGK+01, SPP+05]. It has been shown that the latter is trivial once the structure is known [CH92, CGK+01] therefore most of the published literature is focused on the former task.

In practice, there are two main approaches to learn the structure of a Bayesian network: constraint- and score-based. The constraint-based methods employ statistical tests (Section 2.4) to determine the structure of the Bayesian network (Section 3.2.1). In score-based methods, a score is computed for each network to evaluate how well the network matches the given data (Section 3.2.2). For this class of techniques different possibilities of integrating prior knowledge have been introduced in the literature. It was integrated as part of the scoring function and alternatively as part of the search heuristic. Hybrid algorithms combine ideas from both classes (Section 3.2.3).

3.2.1 Constraint-based

Constraint-based algorithms proceed in two steps, applying Theorem 2.17: they infer the network's skeleton by identifying separating sets for each pair of variables and subsequently orient the skeleton's edges based on the separating sets found in the first step. An edge between two variables X and Y is included in the skeleton whenever a dependency between the two associated nodes exists. In more detail, this means that whenever there exists no set $\mathbf{S}_{XY}$ conditioned on which two variables X and Y are independent, that is $X \perp\!\!\!\perp Y|\mathbf{S}_{XY}$, the edge $X - Y$ should be included in the network:

$$\nexists\, \mathbf{S}_{XY} \subseteq \mathbf{X} \setminus \{X, Y\} :\ X \perp\!\!\!\perp Y|\mathbf{S}_{XY} \implies X - Y. \tag{3.8}$$

Equivalently, no edge should be inferred between two variables X and Y, whenever such a set $\mathbf{S}_{XY}$ exists that would render X and Y conditionally independent

$$\exists\, \mathbf{S}_{XY} \subseteq \mathbf{X} \setminus \{X, Y\} :\ X \perp\!\!\!\perp Y|\mathbf{S}_{XY} \implies X \not- Y. \tag{3.9}$$

Equation (3.8) can be used in forward selection algorithms such that whenever no separating set is identified for a pair of variables an edge is added between them. Equation (3.9) can be used in backward elimination algorithms such that an edge is removed between a pair of variables whenever a separating set was found. The main problem with these constraint-based algorithms is the number of tests that have to be performed. As the number of dependency tests is exponential in the number of variables, these methods cannot be applied to data sets containing thousands of variables. There are heuristic approaches trying to reduce the number of tests by only considering specific subsets of

X as conditioning sets, such as growing subsets. Due to these efforts, constraint-based algorithms can be applied to data sets with higher number of variables, however it has been observed that they tend to be less robust than score-based methods [CH12]. More precisely, small changes in the input data can result in large changes in the resulting networks. This problem is related to the sequential nature of the procedure, that is later tests rely on earlier test results and thus are sensitive to the errors of previous tests [Mar03, KFGT07].

3.2.1.1 Learning the skeleton

In this section we present the first two algorithms using equations (3.8) and (3.9) to infer directed networks. These are the *Inductive Causation (IC)* and the *SGS* algorithms (the latter one was named after its authors Spirtes Glymour and Scheines), respectively. These two algorithms were conceived independently around the same time in [Pea00] and [SGS01].

The SGS algorithm (Algorithm 3) starts with a fully connected network and removes an edge between two variables X and Y if a separating set between the two variables exists. If such a separating set $\mathbf{S}_{XY}$ is identified, it is kept for the orientation phase. In the worst case all possible conditioning sets have to be checked, which results in an exponential number of tests with respect to the number of variables [SGS01].

Algorithm 3: Constraint-based: backward deletion

Input: data, vertex set $\mathbf{V}$
Output: undirected graph, separating sets $\mathbf{S}_{XY}$ for all pairs of variables X and Y for which such a separating set was identified

3.1 add all edges to the graph (undirected);
3.2 **foreach** *pair of variables* X *and* Y *in* $\mathbf{V}$ **do**
3.3 **if** *there exists a subset* $\mathbf{S}_{XY}$ *of* $\mathbf{V} \setminus \{X, Y\}$ *such that:* $X \perp\!\!\!\perp Y | \mathbf{S}_{XY}$ **then**
3.4 remove the undirected edge between X and Y;
3.5 save the identified set $\mathbf{S}_{XY}$ for the pair X, Y;
3.6 **end**
3.7 **end**

The IC algorithm proceeds in the opposite direction as it starts with an empty graph and adds an edge between two variables X and Y if no set exists to separates them, that is whenever there exists no set conditioned on which X and Y are independent [Pea00].

The number of conditional independence tests can be reduced by ordering the considered conditioning sets. In Algorithm 4, [SGS01], the authors start with conditioning sets of size zero and then continue by using sets of size one, etc. Then the complexity's upper bound is dependent of the largest degree, that is the highest number of edges connected to one node. Denoting the maximal degree of any node by d and the number

Algorithm 4: Constraint-based: backward deletion, growing separating sets

Input: data, vertex set $\mathbf{V}$

Output: undirected graph, separating sets $\mathbf{S}_{XY}$ for all pairs of variables X and Y for which such a separating set was identified

4.1 Start with a fully connected undirected graph $\mathcal{G}$ on the vertex set $\mathbf{V}$;
4.2 $k = 0$;
4.3 **repeat**
4.4 **repeat**
4.5 select an ordered pair of variables X, Y that are adjacent in $\mathcal{G}$ such that $|\mathbf{adj}(X) \setminus Y| \geq k$ and a subset $\mathbf{S} \subset \mathbf{adj}(X) \setminus Y$ with $|\mathbf{S}| = k$;
4.6 **if** X, Y *are d-separated given* $\mathbf{S}$ **then**
4.7 delete the edge $X - Y$ from $\mathcal{G}$;
4.8 record $\mathbf{S}$ in $\mathbf{S}_{XY}$;
4.9 **end**
4.10 **until** *all pairs of adjacent variables* X, Y *such that* $|\mathbf{adj}(X) \setminus Y| \geq k$ *and all subsets* $\mathbf{S} \subset \mathbf{adj}(X) \setminus Y$ *with* $|\mathbf{S}| = k$ *have been tested for d-separation*;
4.11 k=k+1;
4.12 **until** *for each pair of adjacent vertices* X, Y, $|\mathbf{adj}(X) \setminus Y| < k$;

of vertices by n, the upper bound of the number of conditional independence tests is given by [SGS01]

$$2\binom{n}{2}\sum_{i=0}^{d}\binom{n-1}{i} \leq \frac{n^2(n-1)^{d-1}}{(d-1)!}.$$

We observe in Figure 3.2 the increase of the number of conditional independence tests with growing n and d.

3.2.1.2 Learning the directions

Based on the learnt skeleton, directed networks can be obtained by using the identified separating sets (Algorithm 5). For each pair X, Y with a common neighbor Z, the chain $X - Z - Y$ can be oriented into the v-structure $X \to Z \leftarrow Y$ if $Z \notin \mathbf{S}_{XY}$ [VP91].

Furthermore, a set of rules has been identified as sufficient in order to maximize the number of oriented edges while at the same time avoiding the creation of new v-structures [Mee95, PE08].

R1 Orient $Y - Z$ into $Y \to Z$ whenever there is an arrow $X \to Y$ such that X and Z are nonadjacent (no new v-structures): Figure 3.3.

R2 Orient $X - Y$ into $X \to Y$ whenever there is a chain $X \to Z \to Y$ (avoid circles): Figure 3.4.

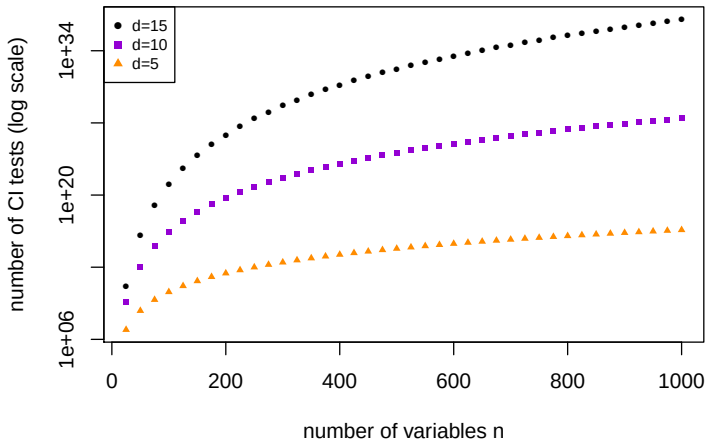


Figure 3.2: The upper bound of the number of conditional independence (CI) tests for Algorithm 4 as function of the number of variables n and the maximum degree of any node in the graph d.

R3 Orient $X - Y$ into $X \to Y$ whenever there are two chains $X - Z \to Y$ and $X - W \to Y$ such that Z and W are nonadjacent (avoid new v-structures and circles): Figure 3.5.

Figure 3.3: Rule R1: by inferring a chain, the algorithms avoids to create a new v-structure.

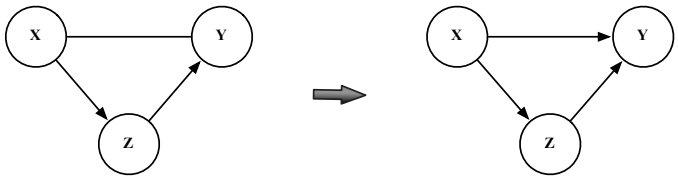


Figure 3.4: Rule R2: By inferring an edge that follows the same direction of an already existing chain, the algorithm avoids to create a cycle.

Computationally, the first step is the most expensive making these algorithms unsuitable for high number of variables [SGS01].

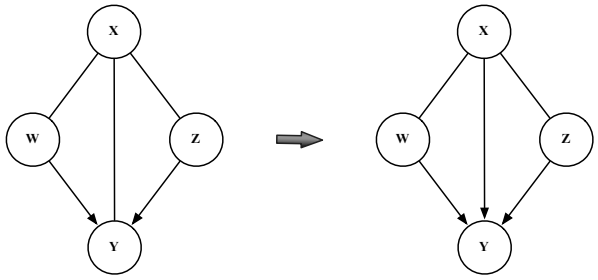


Figure 3.5: Rule R3: By inferring an edge into an existing collider node, the algorithms avoids creating a new v-structure or a cycle. If $Y \to X$ was inferred, $X \to Z$ would create a cycle and the alternative $X \leftarrow Z$ will create a new v-structure.

Algorithm 5: IC/SGS, PC algorithm: orientation phase

Input: data, undirected graph, separating sets $\mathbf{S}_{XY}$ for all pairs of variables X and Y for which such a separating set was identified

Output: partially directed graph

5.1 **foreach** *pair of non-adjacent variables* X *and* Y *with a common neighbor* Z **do**
5.2 | **if** $Z \notin \mathbf{S}_{XY}$ **then**
5.3 | | add arrowheads pointing at Z: $X \to Z \leftarrow Y$
5.4 | **end**
5.5 **end**
5.6 **while** *there exist edges which can be oriented* **do**
5.7 | orient edges according to rules R1,R2 and R3, subject to:
| | • the orientation should not create a new collider
| | • the orientation should not create a directed cycle
5.8 **end**

3.2.1.3 Including prior knowledge

Recently, a constraint-based algorithm to integrate prior knowledge has been developed. This algorithm takes advantage of prior knowledge by using it to constrain certain paths [BT12]. In more detail, starting from a partially directed acyclic graph, the prior knowledge about the presence $X \to Y$ or the absence of an edge $X \not\to Y$ is used to orient additional edges. Some examples of possible orientations are

- given $X - Y - Z$ and prior knowledge $X \to Z$ then the graph can be oriented to $X \to Y \to Z$,
- for the same undirected graph with prior knowledge $X \not\to Z$ yields the partially oriented $X \leftarrow Y - Z$.

3.2.2 Score-based

Score-based methods aim to select the network which optimizes a global fitting score from all possible networks. The problem of learning the network structure maximizing this score is NP-hard [CGH94]. Due to the very high number of possible network structures, evaluating the score for each of them is not feasible in practice. In general, heuristic search techniques are used to identify a good network. Examples of strategies are greedy hill-climbing, simulated annealing, forward-selection or backward-deletion [Hec95, FLNP00]. Due to the heuristic nature of the search algorithms, it cannot be guaranteed that the globally best network with respect to the chosen criterion will be found [KFGT07]. Instead of identifying the globally best network, a locally best network will be chosen. That is, the chosen heuristic will stop searching because there is no one change that will lead the search away from the locally best network. Possible choices of scoring functions include Maximum Likelihood (ML), Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC). Examples of scoring metrics that allow the integration of prior knowledge are the Minimum Description Length (MDL) [Ris86, Bou93], the BDe [HGC95] and the BNCR metric [ISG$^+$02].

A crucial characteristic of many scoring metrics is the *score equivalence*, that is equivalent networks (Definition 2.16) receive the same score [Chi95] and are thus undistinguishable using these scoring metrics.

An important requirement for scoring metrics is the *decomposability*, that is the possibility of computing the complete network's score based on local scores over all variables. Thus the contribution of each variable only depends on its value and its parents' values [FLNP00]. This property allows the application of heuristic algorithms as it is possible to update the current score by simply re-evaluating the local score for the variables that are affected by the change of that step in the algorithm.

3.2.2.1 Scoring functions

The **likelihood** L of a graph $\mathcal{G}$ and a set of conditional probabilities $\mathbf{p}$ after observing $\mathbf{D}$ is defined as

$$L(\mathcal{G}, \mathbf{p}, \mathbf{D}) := p(\mathbf{D}|\mathcal{G}, \mathbf{p}).$$

Maximizing the likelihood over all possible probabilities is known as **maximum likelihood metric**

$$s_{ML}(\mathcal{G}, \mathbf{D}) = \max_{\mathbf{p}} L(\mathcal{G}, \mathbf{p}, \mathbf{D}).$$

In general this metric takes higher values for structures with higher number of edges, moreover the highest ML score is usually assigned to the complete graph [Chi95] which can encode any probability distribution [JN07]. Penalizing the score for the size of the inferred network can be done using the **Akaike information criterion**

$$s_{AIC}(\mathcal{G}, \mathbf{D}) = \log s_{ML}(\mathcal{G}, \mathbf{D}) - |\mathcal{G}|.$$

Another strategy to penalize the network size is to include the data set size which leads to the **Bayesian information criterion**

$$s_{BIC}(\mathcal{G}, \mathbf{D}) = \log s_{ML}(\mathcal{G}, \mathbf{D}) - \frac{1}{2}|\mathcal{G}| \cdot \log m.$$

None of the three previously presented metrics includes prior knowledge to bias the score of the current network. However in the literature several Bayesian metrics have been defined that take prior knowledge $p(\mathcal{G})$ into account. A Bayesian metric is defined as a metric with the following form

$$s(\mathcal{G}, \mathbf{D}) = \log p(\mathcal{G}) + \log p(\mathbf{D}|\mathcal{G}) + c,$$

where $p(\mathcal{G})$ is the prior probability of the structure, $p(\mathbf{D}|\mathcal{G})$ the likelihood of the data that is the posterior probability of the data given the structure and c is a constant. Different metrics have been proposed, one often used is the MDL metric

$$s_{MDL}(\mathcal{G}, \mathbf{D}) = \log p(\mathcal{G}) + s_{BIC}(\mathcal{G}, \mathbf{D}).$$

Other metrics taking advantage of prior knowledge include the **BDe** metric [HGC95] and the **BNCR** metric [ISG+02].

In [Chi95] the score equivalence property for the BIC, AIC, MDL and BDe scores has been proven.

3.2.2.2 From prior knowledge to prior probability

The standard Bayesian approach can be summarized as follows [CS00, Mur01]:

$$\text{posterior(model|data)} \propto \text{prior(model)} \cdot \text{likelihood(model,data)}.$$

Prior knowledge can be the result of an expert specifying edges that are likely to be present in a network of a specific domain and edges which are not [HGC95]. Other sources of prior information are publicly available databases or article abstracts [GVVDM07]. However, it is less clear how to transform these known edge-wise interactions into prior distributions that can be used in the Bayesian framework because this requires a probability distribution over the entire structure. This transformation is considered one of the most difficult tasks in Bayesian learning [HGC95]. We will present different approaches in the following paragraphs: penalizing Bayesian networks which deviate from the prior [HGC95], completing partial knowledge [CS00] and using a Gibbs distribution [IHG+03, HW07].

Penalize [HGC95]
The prior knowledge can be transformed into a *prior network*, denoted by $\mathcal{G}_0 = (\mathbf{V}, \mathbf{E}_0)$, such that there is an edge between two variables X_i and X_j whenever there is a known relationship between X_i and X_j. In this approach, a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ that is similar

to the $\mathcal{G}_0$ will have a higher prior probability than a graph with less similarity. This is achieved by penalizing a network for each edge that is different from the prior network.

$$p(\mathcal{G}) = c\kappa^{\delta},$$

where c is a normalization constant, $0 < \kappa \leq 1$ the constant penalizing factor and δ is the number of arcs in which $\mathcal{G}$ differs from the prior network $\mathcal{G}_0$.

Completing partial knowledge [CS00]
This method consists of two steps. The first completes the prior knowledge by assigning probabilities to all edges for which no prior information exists while in the second the prior knowledge is coded as a prior probability over network structures.

In the first step, it is assumed that for each edge between variables X_i and X_j three options are possible: $X_i \rightarrow X_j$, $X_i \leftarrow X_j$ and $X_i \not- X_j$. The edges for which there exists no prior information will be updated using the following rules.

- If none of the three options has an associated prior probability, they will be equally probable
$$p(X_i \rightarrow X_j) = p(X_i \leftarrow X_j) = p(X_i \not- X_j) = \frac{1}{3}$$
- If one option has a probability of p, then the other two options will be assigned the probability $(1-p)/2$.
- If two options have probabilities p' and p'' respectively, the third option will have a probability of $1 - p' - p''$.

Subsequently, this prior over links is formalized into a prior over a Bayesian network structure using the concept of *oriented graphs*. Oriented graphs are a superset of directed acyclic graphs (DAGs) as they include graphs with cycles of size greater than two. The advantage of using an oriented graph is that its probability can be expressed as product over the probabilities of the existence of an edge between pairs of variables [CS00], as stated in equation (3.10). Assuming that the prior knowledge for one pair is independent from that of another pair, the prior over a structure is then the product of prior probabilities over all pairs of variables

$$p(\mathcal{G}) = \prod_{i,j \in 1,\ldots,n, i<j} p(X_i - X_j), \tag{3.10}$$

where $X_i - X_j \in \{X_i \rightarrow X_j, X_i \leftarrow X_j, X_i \not- X_j\}$ specifies the type of interaction between X_i and X_j in $\mathcal{G}$. The set of oriented graphs is larger than the set of DAGs, therefore the probability $p(\mathcal{G})$ is not yet a prior distribution over the set of Bayesian networks. In order to obtain such a distribution, the authors in [CS00] propose two methods to take these additional structures into account: a uniform correction

$$p_{uniform}(\mathcal{G}) = c_1 + p(\mathcal{G})$$

and a proportional correction

$$p_{proportional}(\mathcal{G}) = c_2 \cdot p(\mathcal{G}),$$

with c_1 and c_2 appropriately chosen normalization factors.

Gibbs distribution
In this framework, the prior knowledge is represented as a prior network and then a score is computed such that the lower the score the higher is the amount of prior knowledge. This score can then be used to compute a Gibbs distribution which forms the prior over the network structure. This score is known as the *energy function* [IHG+03]. The prior knowledge is transformed into an $n \times n$ matrix U such that $u_{ij} = \zeta_1$ if there is a known relationship from gene X_i to gene X_j. It is assigned a value ζ_2 if there is no known relationship, $0 < \zeta_1 < \zeta_2$. The energy of the prior network can then be computed as

$$E(\mathcal{G}) = \sum_{\{i,j\} \in \mathcal{G}} u_{ij}.$$

The prior probability $p(\mathcal{G})$ of a network $\mathcal{G}$ is then modeled by a Gibbs distribution

$$p(\mathcal{G}) = \frac{1}{Z} \exp\{-\zeta E(\mathcal{G})\},$$

where $\zeta > 0$ is a hyperparameter and $Z = \sum_{\mathcal{G} \in \mathcal{G}} \exp\{-\zeta E(\mathcal{G})\}$ a normalization factor. This method has been used in combination with the BNRC [ISG+02] criterion in a score based algorithm. A similar approach has been proposed in [HW07].

3.2.2.3 Search algorithms

In order to maximize the scoring metric, different heuristic search techniques have been proposed in the literature. Let the score be *decomposable*, that is the scoring function can be rewritten such that the contribution of each variable X_i is captured in a separate term of the sum

$$s(\mathcal{G}, \mathbf{D}) = \sum_i s_i(X_i, \mathbf{pa}(X_i), \mathbf{D}).$$

In this case, a local search procedure can be applied, modifying a single arc at each iteration by adding, removing or reversing it [FNP99]. Examples of these algorithms are **greedy hill-climbing** and **simulated annealing**. The first method selects at each step the local change that maximizes the gain of the scoring function [FLNP00]. The second technique starts by allowing a high number of alterations to the network in one iteration. Then the number of alterations is gradually reduced in order to converge to a good network [HGJY02].

Prior knowledge as part of the search algorithm
Two different techniques have been proposed to integrate prior knowledge into the search

algorithm. The first consists in using the prior knowledge to seed the network search [DQ08], such that the search starts out from a network in which each connection is an interaction that has been validated in previous research. On the other hand prior knowledge can restrict the search space [ALCD08] such that only those structures are tested in which known interactions are present whereas those structures without any known interactions are avoided.

3.2.3 Hybrid algorithms

There have been advances to combine score-based and constraint-based algorithms in order to, at least partially, overcome their respective shortcomings while at the same time taking advantage of their strengths. The former one suffers from the exponentially large number of possible networks to test whereas the latter offers no possibility to control error rates and depends on the reliability of the testing procedure [SS11]. In general the hybrid algorithms use a constraint-based approach, such as Algorithm 3, to generate an initial network. This network is then used as input for a score-based approach [FNP99, TBA06].

A slightly different approach was presented in [FNP99]. The authors propose a two step approach which starts by selecting a set of relevant candidate parents based on a pairwise score such as mutual information (Definition 2.24). In the second step, the network from the first step is used as input for a score-based algorithm, which is then identifying the oriented network which maximizes the given scoring function.

3.3 Feature selection methods

In the first part of this section we discussed algorithms which can be interpreted as extensions to the constraint-based class of Bayesian network inference. In order to develop algorithms which can deal with a higher number of variables than those presented in the previous section, the concept of Markov blankets is exploited in order to reduce the number of independence tests. We start this section by introducing the link between relevance and Markov blankets and continue with different techniques to infer a network structure. We conclude the first part with techniques that orient the obtained structure based on the previously identified Markov blankets.

The second half of this section is dedicated to methods based on information theoretic measures such as mutual information and interaction information. The former set of methods has been developed to infer undirected networks for possibly thousands of variables. The latter has been used to extend some of the algorithms inferring undirected graphs in order to infer directed networks. These are the MI3 [LHW08] and the synergistic regulation index (SRI) [WLW$^+$09].

3.3.1 Using Markov blankets

In this section, we explain the relation between Markov blankets and relevant features by providing the necessary definitions of a *relevant feature*, *Markov blankets* and the properties relating the two. Then we present algorithms taking advantage of the notion of Markov blankets for the inference of directed networks. These algorithms usually proceed in four steps (Figure 3.6).

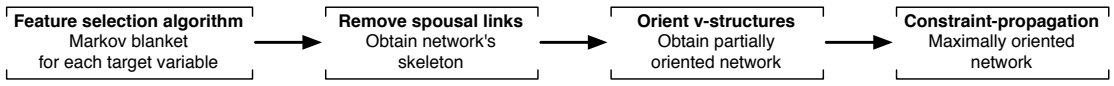


Figure 3.6: A directed network is inferred in four steps: i) the Markov blanket of each variable is identified using a feature selection algorithm, ii) the spousal links are removed in order to infer the skeleton from the Markov blankets, iii) the v-structures are identified to partially orient the skeleton and iv) a maximum number of edges not part of any v-structure are oriented using rules avoiding any cycles or additional v-structures.

i. Identifying the members of each variable's Markov blanket.

ii. Combining these Markov blankets into an undirected network.

iii. Orienting v-structures.

iv. Orient as many of remaining edges as possible avoiding new v-structures and cycles.

The general procedure is outlined in Figure 3.6 and roughly follows the description in [PE08].

3.3.1.1 Basic definitions and properties

Let $\mathbf{X}^{\setminus i}$ denote the set $\mathbf{X} = \{X_1, \ldots, X_n\}$ without the i-th variable.

Definition 3.5 *[KJ97]*

- *A feature X_i is **strongly Kohavi-John (KJ) relevant** to the target Y iff there exist some values x, y and $\mathbf{v}$ with $P(X_i = x, \mathbf{X}^{\setminus i} = \mathbf{v}) > 0$ such that:*

$$P(Y = y | X_i = x, \mathbf{X}^{\setminus i} = \mathbf{v}) \neq P(Y = y | \mathbf{X}^{\setminus i} = \mathbf{v})$$

- *A feature X_i is **weakly KJ relevant** to the target Y iff it is not strongly relevant and if there exist a subset of features $\mathbf{X}^{\setminus i}$ for which there exist some values x, y and $\mathbf{v}$ with $P(X_i = x, \mathbf{X}^{\setminus i} = \mathbf{v}) > 0$ such that:*

$$P(Y = y | X_i = x, \mathbf{X}^{\setminus i} = \mathbf{v}) \neq P(Y = y | \mathbf{X}^{\setminus i} = \mathbf{v})$$

- *A feature is **irrelevant** if it is not strongly or weakly relevant.*

A feature is either irrelevant, strongly relevant or weakly relevant.

Theorem 3.6 *[Pea88]*
In a Bayesian network satisfying the faithfulness and the Markov condition, the unique Markov blanket of a target variable contains the parents, children and spouses (other parents of the children) of the target variable (Figure 3.7).

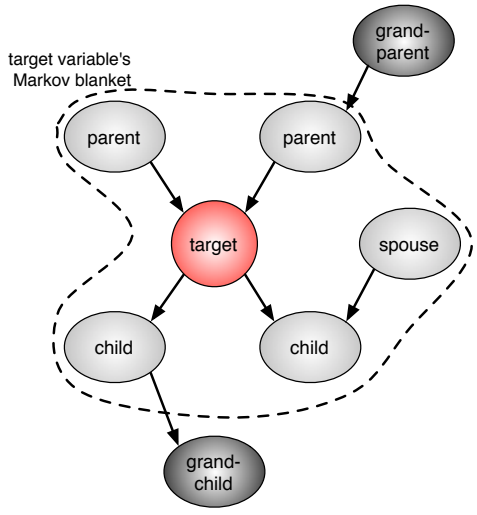


Figure 3.7: The members of the target variable's Markov blanket in light grey: these are its children, parents and spouses.

In the following theorem, the direct connection of the Markov blanket to the notion of strong relevance is stated.

Theorem 3.7 *[TA03]*
In a Bayesian network satisfying the faithfulness and the Markov condition, a variable is strongly relevant to the target variable if and only if the variable belongs to its Markov blanket.

In non-faithful distributions, there might be several Markov blankets for the target variable. In this case, the strongly relevant features are those included in the intersection of the Markov blankets [TA03].

The following theorem relates the Markov blanket to the notion of weak relevance.

Theorem 3.8 *[TA03]*
In a Bayesian network satisfying the faithfulness and the Markov condition, a variable X is weakly KJ relevant to the target variable if and only if it is not KJ strongly relevant and there is an undirected path from X to the target variable.

In Figure 3.7, the two nodes colored dark grey are weakly KJ relevant because they are not members of the target's Markov blanket and thus not strongly KJ relevant and because furthermore there exists an undirected path to the target for both variables, in case of the grandparent even an directed path.

3.3.1.2 Determining the undirected network

In order to infer the skeleton of a Bayesian network using the notion of Markov blankets, each variable plays the role of the target variable once. For each variable, the members of its Markov blanket are identified using conditional independence tests or a hybrid algorithm. Subsequently, the set of Markov blankets have to be combined into the network's skeleton.

3.3.1.2.1 Identifying the members of a Markov blanket

In the literature two families of algorithms have been introduced to identify the members of a variable's Markov blanket. The first family consists of those algorithms that use a two step procedure to first select a superset of the actual Markov blanket and then removes variables that do not belong to the Markov blanket, the *GS type*. In the following, we will present two examples: the Grow-Shrink (GS) algorithm and the Incremental Association Markov Blanket (IAMB) algorithm. The second family consists of *PC type* algorithms that start by identifying the set of parents and children of the target variable and then identify the spouses to complete the Markov blanket in a second step. We will present two examples: the HITON algorithm and Max-Min algorithm. See Table 3.2 for an overview on the algorithms presented in this section.

family	name	MB identification	
		first step	second step
GS	Grow-Shrink Algorithm 6	superset of MB (constraint)	shrink superset to MB (constraint)
	IAMB Algorithm 7	superset of MB (score)	shrink superset to MB (constraint)
PC	HITON	HITON-PC: Algorithm 8 (constraint)	HITON-MB/MMMB: Algorithm 10 (constraint)
	Max-Min	MMPC: Algorithm 9 (hybrid)	

Table 3.1: Overview algorithms for Markov blanket identification: algorithms belonging to the GS family first identify a superset of the target variable's Markov blanket and then remove the excess variables. Algorithms that belong to the PC family first identify the target variable's parents and children and in a second step the missing variables (the spouses).

GS type algorithms

The first algorithm designed to identify the members of a target variable's $X_T \in \mathbf{X}$ Markov blanket was the Grow-Shrink (GS) algorithm [MT99, Mar03], Algorithm 6. In the first step, the **growing**, variables are added to the candidate list as long as there exists a variable in $Y \in \mathbf{X} \setminus \{\mathbf{S} \cup X_T\}$ that is dependent on X_T given the current candidate list $\mathbf{S}$

$$\exists Y \in \mathbf{X} \setminus \{\mathbf{S} \cup X_T\} : \; Y \not\perp\!\!\!\perp X_T | \mathbf{S}. \tag{3.11}$$

The second step is devoted to the removal of those variables not actually belonging to the Markov blanket, the **shrinking**. Some variables that were added in an early stage of the growing step might no longer be dependent of X_T, thus all variables Y for which

$$Y \perp\!\!\!\perp X_T | \{\mathbf{S} \setminus Y\} \tag{3.12}$$

holds are removed from $\mathbf{S}$.

Algorithm 6: GS algorithm: calculation of $\mathbf{MB}(X_T)$

Input: data, vertex set $\mathbf{V}$, target variable $X_T \in \mathbf{V}$
Output: Markov blanket $\mathbf{MB}(X_T)$ of X_T

6.1 $\mathbf{S} = \emptyset$;
6.2 **while** $\exists Y \in \mathbf{V} \setminus \{\mathbf{S} \cup X_T\}$ *such that* $Y \not\perp\!\!\!\perp X_T | \mathbf{S}$ **do**
6.3 $\quad \mathbf{S} = \mathbf{S} \cup Y$; `/* Growing phase */`
6.4 **end**
6.5 **while** $\exists Y \in \mathbf{S}$ *such that* $Y \perp\!\!\!\perp X_T | \{\mathbf{S} \setminus Y\}$ **do**
6.6 $\quad \mathbf{S} = \mathbf{S} \setminus Y$; `/* Shrinking phase */`
6.7 **end**
6.8 $\mathbf{MB}(X_T) = \mathbf{S}$;

This algorithm needs $O(n)$ statistical tests (where n denotes the number of variables in

the data set) [Mar03].

The second algorithm that has been proposed in the literature to identify the elements of a variable's Markov blanket is the **Incremental Association Markov Blanket (IAMB)** algorithm [TAS03b], Algorithm 7. Instead of applying conditional independence tests in the forward selection step, the algorithm maximizes function measuring conditional independence in order to select variables. The set of selected variables will be denoted by $\mathbf{S}$ and is initialized as $\mathbf{S} = \emptyset$. In each iteration, the variable $Y \in \mathbf{X} \setminus \{\mathbf{S} \cup X_T\}$ maximizing a function $f(Y; X_T|\mathbf{S})$ will be added to $\mathbf{S}$ whenever

$$Y \not\perp\!\!\!\perp X_T|\mathbf{S}. \tag{3.13}$$

This function is usually defined as a measure of information, for example conditional mutual information (Definition 2.26). The second phase is the same as in the GS algorithm.

Algorithm 7: IAMB algorithm

Input: data, vertex set $\mathbf{V}$, target variable $X_T \in \mathbf{V}$
Output: Markov blanket $\mathbf{MB}(X_T)$ of X_T

7.1 $\mathbf{S} = \emptyset$;
7.2 **while** *S has changed ;* `/* Growing */`
7.3 **do**
7.4 | Find the feature Y in $\mathbf{V} \setminus \{\mathbf{S} \cup X_T\}$ maximizing $f(Y, X_T|\mathbf{S})$;
7.5 | **if** $Y \not\perp\!\!\!\perp X_T|\mathbf{S}$ **then**
7.6 | | $\mathbf{S} = \mathbf{S} \cup Y$;
7.7 | **end**
7.8 **end**
7.9 Remove from $\mathbf{S}$ all variables Y for which $Y \perp\!\!\!\perp X_T|\{\mathbf{S} \setminus Y\}$; `/* Shrinking */`
7.10 $\mathbf{MB}(X_T) = \mathbf{S}$;

This algorithm is a typical example of a hybrid approach where the score-based approach is used for the growing phase and the constraint-based for the shrinking phase.

Both algorithms, GS and IAMB, require the number of samples being exponential to the size of the Markov blanket [ATS+03].

PC type algorithms
The main idea behind this family of algorithms is to identify in the first step the target variable's parents and children (**PC**) and only in the second step the complete the Markov blanket consisting of the target variable's parents, children and spouses. The identification of the target variable's parents and children is usually achieved by first using an association scoring function to select a superset of **PC**. Then variables that become independent of the target given a subset of **PC** are removed.

HITON-PC (Algorithm 8) [ATS+03] adds variables to the set of selected variables $\mathbf{S}$ in a forward selection procedure maximizing an association score $f(\cdot,\cdot)$ between $X_i \in \mathbf{X} \setminus \{X_T \cup \mathbf{S}\}$ and the target X_T. This set is initialized as empty set and will in the end contain the parents and children of the target variable. The association score used by the authors is the negative p-values returned by a G test. They use the negative p-value because they are maximizing this function (line 8.4). The variable having the highest association with the target is then added to the set of selected variables $\mathbf{S}$. It is removed again if there exists a variable $X_j \in \mathbf{S}$ and a subset $\mathbf{S}' \subseteq \mathbf{S}$ such that

$$X_j \perp\!\!\!\perp X_T | \mathbf{S}' \tag{3.14}$$

because in this case X_j lies outside of the target variable's Markov blanket or is a spouse and thus can neither be its parent nor its child.

Algorithm 8: HITON-PC algorithm

Input: data, vertex set $\mathbf{V}$, target variable $X_T \in \mathbf{V}$
Output: Parents and children of X_T stored in $\mathbf{S}$

8.1 $\mathbf{S} = \emptyset$;
8.2 $\mathbf{C} = \mathbf{V}$;
8.3 **while** $\mathbf{C}$ *is not empty* **do**
8.4 Find variable $X_i \in \mathbf{C}$ with $X_i \notin \mathbf{S}$ maximizing $f(X_i, X_T)$;
8.5 Add X_i to $\mathbf{S}$;
8.6 **foreach** $X_j \in \mathbf{S}$ **do**
8.7 **if** $\exists \mathbf{S}' \subseteq \mathbf{S} \setminus X_j$ *such that* $X_j \perp\!\!\!\perp X_T | \mathbf{S}'$ **then**
8.8 $\mathbf{S} = \mathbf{S} \setminus X_j$;
8.9 **end**
8.10 **end**
8.11 Remove X_i from $\mathbf{C}$;
8.12 **end**

The **Max-Min Parents and Children(MMPC)** algorithm [TAS03a] (Algorithm 9) differs from HITON-PC in the selection phase as it uses a conditional association function $f(\cdot,\cdot|\cdot)$ to identify the set of parents and children stored in $\mathbf{S}$. First a conditioning set is identified for each variable $X_i \in \mathbf{X} \setminus X_T$ as the subset of the current set of parents and children $\mathbf{S}'_i \subseteq \mathbf{S}$ that minimizes the conditional association function $f(X_i, X_T|\mathbf{S}'_i)$. Then the variable $X_j \in \mathbf{X} \setminus \{X_T \cup \mathbf{S}\}$ that maximizes the conditional association with the target given the previously identified conditioning set $\mathbf{S}'_j$ is chosen. It is included in the current set of parents and children $\mathbf{S}$ if $X_j \not\perp\!\!\!\perp X_T | \mathbf{S}'_j$.

The false positives that were added to $\mathbf{S}$ in the growing phase are then removed in the backward deletion part of the algorithm. Whenever a subset $\mathbf{S}' \subseteq \mathbf{S}$ can be found that renders X_i independent of X_T given $\mathbf{S}'$ then X_i is removed from $\mathbf{S}$.

Algorithm 9: MMPC algorithm

Input: data, vertex set $\mathbf{V}$, target variable $X_T \in \mathbf{V}$
Output: Set $\mathbf{S}$ containing parents and children of X_T

9.1 $\mathbf{S} = \emptyset$;
/* forward */
9.2 **while** $\mathbf{S}$ *does not change anymore* **do**
9.3 **foreach** $X_i \in \mathbf{V} \setminus \{X_T\}$ **do**
9.4 find $\mathbf{S}'_i \subseteq \mathbf{S}$ minimizing $f(X_i, X_T|\mathbf{S}'_i)$;
9.5 **end**
9.6 $X_j \in \mathbf{V} \setminus \{X_T \cup \mathbf{S}\}$ maximizing $f(X_j, X_T|\mathbf{S}'_j)$;
9.7 **if** $X_j \not\perp\!\!\!\perp X_T|\mathbf{S}'_j$ **then**
9.8 $\mathbf{S} = \mathbf{S} \cup X_j$;
9.9 **end**
9.10 **end**
/* backward */
9.11 **foreach** $X_i \in \mathbf{S}$ **do**
9.12 **if** $\exists \mathbf{S}' \subseteq \mathbf{S}$ *such that* $X_i \perp\!\!\!\perp X_T|\mathbf{S}'$ **then**
9.13 $\mathbf{S} = \mathbf{S} \setminus X_i$;
9.14 **end**
9.15 **end**

Having identified the parents and children of X_T, the set of spouses is missing in order to obtain the full Markov blanket. A spouse of X_T is any other parent of its children. **HITON-MB** and **MMMB** (Algorithm 10) start by identifying all parents and children of the previously found set of parents and children (**PC**) by applying *HITON-PC* or *MMPC* to each of the variables in **PC**. This is a superset of the actual Markov blanket containing the parents, children and spouses but also grandchildren, grandparents and siblings. The latter three family members have to be removed in order to identify the Markov blanket. This is achieved by using a statistical test similar to the one used in the backward deletion in Algorithm 3 [SGS01].

3.3.1.2.2 Combining the Markov blankets into an undirected network

Having identified the variables belonging to the Markov blanket of each variable $X_T \in \mathbf{X}$, the network's skeleton can be obtained by combining the Markov blankets of each variable (Algorithm 11). Lines 11.9 to 11.11 apply the criterion described in equation (3.8):

$$\nexists\, \mathbf{S}_{XY} \subseteq \mathbf{X} \setminus \{X, Y\} : \; X \perp\!\!\!\perp Y|\mathbf{S}_{XY} \implies X - Y.$$

This step is necessary because the Markov blankets consist of parents, children and spouses. While parents and children are part of the target variable's adjacencies, spouses are not.

Algorithm 10: HITON-MB/MMMB algorithm

Input: data, vertex set $\mathbf{V}$, target variable $X_T \in \mathbf{V}$
Output: Markov blanket $\mathbf{MB}(X_T)$ of X_T

10.1 $\mathbf{PC}$ = HITON-PC(data,$\mathbf{V}$,X_T) or $\mathbf{PC}$ = MMPC(data,$\mathbf{V}$,X_T) ;
10.2 $\mathbf{PCPC}$ = parents and children of all variables in $\mathbf{PC}$ returned by HITON-PC or MMPC applied to each variable in $\mathbf{PC}$;
10.3 **foreach** $X_j \in \mathbf{PCPC} \setminus \mathbf{PC}$ `/* for each potential spouse */`
10.4 **do**
10.5 Find $\mathbf{S}'$ such that $X_j \perp\!\!\!\perp X_T | \mathbf{S}'$ `/* exists because otherwise there would be a link between` X_j `and` X_T `*/`
10.6 **foreach** $X_k \in \mathbf{PC}$ **do**
10.7 **if** $X_j \not\perp\!\!\!\perp X_T | \{X_k \cup \mathbf{S}'\}$ **then**
`/*` X_j `is a spouse of` X_T `*/`
10.8 Mark X_j for inclusion;
10.9 **end**
10.10 **end**
10.11 Remove X_j from $\mathbf{PCPC}$ unless it is marked;
10.12 **end**
10.13 $\mathbf{MB}(X_T) = \mathbf{PCPC}$

Therefore, for all variables $X \in \mathbf{X}$, Algorithm 11 starts by identifying the set of adjacent variables. For each X, all elements of the corresponding Markov blanket $Y \in \mathbf{MB}(X)$ are tested for conditional independence given all subsets of the Markov blanket together with X (more precisely, the set $\mathbf{T} = \min\{(\mathbf{MB}(X) \setminus Y), (\mathbf{MB}(Y) \setminus X)\}$ is considered). If X and Y are conditionally dependent given all subsets $\mathbf{S} \subseteq \mathbf{T}$, the variable Y is not a spouse and thus can be added to the adjacencies of X.

If $b = \max_X(|\mathbf{MB}(X)|)$ this algorithm carries out $O(nb2^b)$ conditional independence tests. In the worst case $b = O(n)$ which occurs when the set was produced by a dense original network [Mar03].

3.3.1.3 Combining the Markov blankets into a directed network

After having identified each variable's Markov blanket and subsequently the network's skeleton using the GS or IAMB algorithm, the directed graph has to be constructed. This is achieved using four steps: orienting v-structures using a conditional independence criterion, removing orientations of one edge in each cycle, redirecting these in the opposite direction and finally, if there are still undirected edges, orienting them in the direction of already existing paths between two adjacent variables, Algorithm 12 [Mar03].

An improvement to this two step procedure was proposed in [PE08] such that the non-inclusion of spousal links in Algorithm 11 (lines 11.9 to 11.11) and the v-structure ori-

Algorithm 11: GS algorithm: recovering the skeleton from the set of MBs

Input: data set, vertex set $\mathbf{V}$, $\mathbf{MB}(X)$ for all $X \in \mathbf{V}$
Output: set of adjacent variables $\mathbf{adj}(X)$ for all $X \in \mathbf{V}$

```
11.1  foreach X ∈ V do
11.2      adj(X) = ∅;
11.3      foreach Y ∈ MB(X) do
11.4          if |MB(X) \ Y| < |MB(Y) \ X| then
11.5              T = MB(X) \ Y;
11.6          else
11.7              T = MB(Y) \ X;
11.8          end
11.9          if X ⫫̸ Y|S for all S ⊆ T then
11.10             adj(X) = adj(X) ∪ Y;
11.11         end
11.12     end
11.13 end
```

entation in Algorithm 12 (lines 12.1 to 12.5) are carried out in one step. This is possible because identifying a set $\mathbf{S}$ that makes X and Y conditionally dependent implies that they are parents in a v-structure and any variable not part of $\mathbf{S}$ but part of their Markov blankets must be a collider as described by the d-separation criterion (Definition 2.11).

The orientation phase requires $O(nb^2 2^b)$ conditional independence tests (b denotes again the maximum number of elements in Markov blankets $\mathbf{MB}(X)$ over all variables $X \in \mathbf{X}$). The edge removal phase takes $O(l(n+l))$, where l is the total number of edges in the network. Reversing the l edges takes $O(l)$ and the last step $O(nb(n+l))$ [Mar03].

An alternative algorithm orienting the network returned by MMPC (Algorithm 9) is the score-based **Max-Min Hill-Climbing (MMHC)** algorithm [BTA04, TBA06]. It starts by identifying all variables' parents and children using the MMPC, Algorithm 9. This is then followed by a score-based orientation phase. The algorithm maximizes a score by trying to

- add edges $X_j \to X_i$ for which $X_j \in \mathbf{PC}_i$,
- reverse edges and
- delete edges.

Due to the combination of constraint-based and score-based parts, MMHC belongs to the class of hybrid algorithms.
Some of these methods have been implemented in the R package *bnlearn* [Scu10].

Algorithm 12: Orient edges based on Markov blankets

Input: data set, vertex set $\mathbf{V}$, Markov blanket for all $X \in \mathbf{V}$, set of adjacent variables $\mathbf{adj}(X)$ for each $X \in \mathbf{V}$

Output: directed network

`/* Orient v-structures */`

foreach $X \in \mathbf{V}$ *and* $Y \in \mathbf{adj}(X)$ **do**
  if $\exists Z \in \mathbf{adj}(X) \setminus \{\mathbf{adj}(Y) \cup Y\}$ *such that* $Y \not\perp\!\!\!\perp Z|\{\mathbf{S} \cup X\}, \forall \mathbf{S} \subseteq \mathbf{T}$, *where* $\mathbf{T}$ *is the smaller one of* $\mathbf{MB}(Y) \setminus \{X, Z\}$ *and* $\mathbf{MB}(Z) \setminus \{X, Y\}$ **then**
    Orient $Y \rightarrow X$;
  end
end

`/* Remove orientation in cycles */`

$\mathbf{R} = \emptyset$;
while *there exist cycles in the graph* **do**
  Compute the set of edges $\mathbf{C} = \{X \rightarrow Y$ such that $X \rightarrow Y$ is part of a cycle $\}$;
  Remove from the current graph the edge in $\mathbf{C}$ that is part of the greatest number of cycles and put it into $\mathbf{R}$
end

`/* Reorient edges */`

foreach *edge in* $\mathbf{R}$ **do**
  insert the edges in reverse order of removal, oriented in opposite direction
end

`/* Orient in direction of already existing paths */`

foreach $X \in \mathbf{V}$ *and* $Y \in \mathbf{adj}(X)$ *such that neither* $Y \rightarrow X$ *nor* $X \rightarrow Y$ **do**
  if *there exists a directed path from* X *to* Y **then**
    Orient $X \rightarrow Y$;
  end
end

Algorithm 13: MMHC algorithm

Input: data, vertex set $\mathbf{V}$

Output: highest scoring DAG

foreach $X_i \in \mathbf{V}$ **do**
  $\mathbf{PC}_i$ = MMPC(data,$\mathbf{V}$,X_i);
end
Start with empty network;
Perform greedy hill-climbing with operators *add edge, delete edge, reverse edge.* Only try *add edge* $X_j \rightarrow X_i$ if $X_j \in \mathbf{PC}_i$;

3.3.1.4 Complexity comparison

family	MB identification (for one variable X_T)		skeleton	directed
	parents & children	Markov blanket		
Grow-Shrink	/	$O(n)$	$O(nb2^b)$	$O(nb^2 2^b)$
IAMB	/	$O(n)$	/	/
HITON	$O(n^{(k+1)})$		/	/
Max-Min	$O(n^{(k+1)})$	$\lvert\mathbf{PC}(X_T)\rvert O(n^{(k+1)})$	/	/

Table 3.2: Worst case number of conditional independence tests for methods based on Markov blanket identification. Depending on the practical problem the algorithms might be faster in practice. Here, $b = \max_{X_T \in \mathbf{X}}(|\mathbf{MB}(X_T)|)$ which in the worst case yields $b = O(n)$. Furthermore, k is the maximum size of the conditioning set.

The class of algorithms based on the notion of Markov blankets is an improvement over the original constraint-based/hybrid algorithms in terms of runtime. However, they are far from applicable to data sets with thousands of variables and only few samples. This is due to the large number of conditional independence tests that needs to be conducted, the GS type algorithms have as limiting factor the worst case complexity of $O(nb^2 2^b)$ with $b = O(n)$. Whereas the PC type algorithms are limited by the size of the conditioning sets $O\left(n^{(k+1)}\right)$ where k is the size of the conditioning set. This is prohibitive but for very small sets (Table 3.2).

3.3.2 Inference using information theory

3.3.2.1 Undirected networks based on mutual information

The common denominator of the network inference methods we present in this section is the computation of mutual information (Definition 2.24) for all pairs of variables. The first and simplest method takes this mutual information matrix as input and returns an undirected network containing all edges for which the corresponding mutual information surpasses the chosen threshold [BK00] (Section 3.3.2.1.1). More recently developed methods take the mutual information matrix as input to compute a score for each variable pair by avoiding possible indirect interactions [MNea06] (Section 3.3.2.1.2) or by taking into account the specific mutual information distribution for each variable [FHea07] (Section 3.3.2.1.3) or by including the variables with the highest relevance with the target variable and at the same time the lowest redundancy with already selected variables [MKLB07] (Section 3.3.2.1.4).

3.3.2.1.1 Relevance networks

The original implementation of the *Relevance networks* algorithm computed pairwise correlations to infer a network. Pairs of variables would be connected in this network whenever their correlation surpassed a given threshold [BTS+00]. To infer also interactions for non-linear relationships between variables, this methods was extended by

computing pairwise mutual information for each pair of variables [BK00]. Then the network is inferred by keeping only those edges whose mutual information is higher than a given threshold. The complexity of this algorithm is determined by the calculation of the pairwise mutual information $O(n^2)$, where n represents the number of variables in the network [BK00, BTS$^+$00].

3.3.2.1.2 ARACNE

The *Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE)* can be interpreted as an extension to the *Relevance networks* algorithm [MNea06]. The first step is effectively the same as in *Relevance networks*: computing the pairwise mutual information values. However, ARACNE then tries to avoid inferring an edge between two variables that are in reality only linked via an indirect dependency. This is achieved by evaluating interconnected triplets of variables and removing the link with the smallest associated mutual information. This procedure is based on the data processing inequality (DPI, Property 2.25, [CT90]) which states that if two genes X_1 and X_2 interact through a third variable X_3 and there is no alternative path from X_1 to X_2, then

$$I(X_1, X_2) \leq \min \{I(X_1, X_3), I(X_3, X_2)\} . \tag{3.15}$$

The algorithmic complexity is $O(n^3)$, where n represents the number of genes in the network since all triplets of variables are considered [MNea06].

One extension to the standard ARACNE algorithm is the use of a bootstrap procedure to build more robust networks [MWL$^+$06]. For this approach a set of data sets with the same dimensions as the original data set is generated via resampling with replacement of the original samples. After applying ARACNE to each of these data sets the thus obtained networks are combined into a consensus network as follows: based on a permutation test, the most significant edges are identified and then added to the final network [MWL$^+$06].

3.3.2.1.3 CLR

Another extension to the *Relevance networks* algorithm is the *Context Likelihood of Relatedness (CLR)* method [FHea07]. The score for each pair of variables X_i, X_j is now based on the mutual information in combination with the empirical mean and standard deviation for each of the two variables

$$z_{ij} = \sqrt{z_i^2 + z_j^2} \tag{3.16}$$

with

$$z_i = \max \left(0, \frac{I(X_i, X_j) - \hat{\mu}_i}{\hat{\sigma}_i}\right) , \tag{3.17}$$

where $\hat{\mu}_i$ and $\hat{\sigma}_i$ correspond to the mean and the standard deviation of the mutual information values' $I(X_i, X_k)$, $k = 1, \ldots, n$, empirical distribution. As with relevance

networks, the complexity of the CLR algorithm is $O(n^2)$ resulting from the pairwise mutual information calculations.

3.3.2.1.4 MRNET

Yet another way to infer a network using the mutual information matrix as a starting point is to use the minimum redundancy, maximum relevance (mRMR) criterion [DP05, MKLB07]. Every variable in the data sets plays the target variable X_T once. The MRNET forward selection strategy starts by selecting the variable X_i which exhibits the highest mutual information with the target. At every subsequent step the variable with the highest mutual information with the target, that is the highest relevance, and at the same time the lowest average mutual information with the already selected variables $\mathbf{X_S}$ is selected. Therefore, the variable maximizing

$$s_j = I(X_j; X_T) - \frac{1}{|\mathbf{X_S}|} \sum_{X_k \in \mathbf{X_S}} I(X_j; X_k) \tag{3.18}$$

is added to the set $\mathbf{X_S}$. In a final step, the score of a variable pair X_i, X_j is computed by taking the maximum between s_i and s_j. As with relevance networks, edges with a values lower than a given threshold are removed from the network. The complexity of the MRNet algorithm is between $O(n^2)$ and $O(n^3)$ depending on how many features are selected, [MKLB07]. The three algorithms ARACNE, CLR and MRNET have been implemented in the R/Bioconductor package *MINET* [MLB07a].

3.3.2.1.5 C3NET and BC3NET

A more prudent path is taken by the *Conservative Causal Core (C3NET)* algorithm. The main principle is to only include the most significant edges in the network [AES10]. In the first step the mutual information is tested for statistical significance using a hypothesis test. Then for each variable $X_i \in \mathbf{X}$, out of the edges $X_i - X_k$ with significant mutual information scores, the one with the highest value is added to the network.

$$X_j = \arg \max_{\{X_k : I(X_i, X_k) \text{ significant}\}} I(X_i, X_k) \tag{3.19}$$

This method has been implemented in the CRAN/R package *c3net* [AES11].

A bagging variant of the C3NET algorithm has been proposed in [dMSES12], namely *BC3NET*. This method proceeds in four steps, during the first the data set is resampled to obtain a number of independent data sets, say b. For each of these data sets, a network is obtained using C3NET. Thirdly, an aggregated network is constructed by using a combination of the b networks. Finally, the final network is obtained by cropping the aggregated network to only keep the significant edges. This algorithm has been implemented in the CRAN/R package *bc3net* [dMSES12].

3.3.2.2 Directed networks based on interaction information

As presented in Section 2.5.3, interaction information is an extension of mutual information taking into consideration three variables. Unlike mutual information this quantity can also be negative, more specifically negative interaction information can be used to detect triplets of variables linked to the *explaining away effect*. This states that a given common effect creates a dependency between two variables [Nea03].

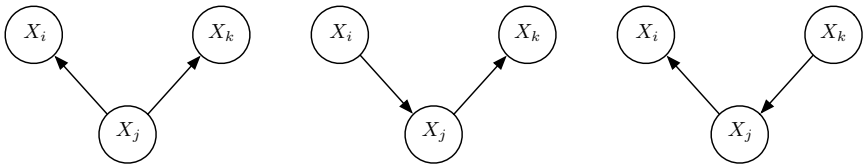
Property 3.9 (Explaining away effect)
Once the value of a common effect is known, it creates a dependency between its causes because knowing that one of the two causes occurred reduces the likelihood of the other one.

The following link between negative interaction information and v-structures has been presented in [CGK+01, Mey08].

Property 3.10
Given three variables X_i, X_j and X_k and a structure $X_i - X_j - X_k$. If $\mathcal{I}(X_i, X_j, X_k) < 0$, then

$$X_i \rightarrow X_j \leftarrow X_k. \tag{3.20}$$

The remaining three directed graphs that can be constructed based on the skeleton $X_i - X_j - X_k$ are depicted in Figure 3.8 and yield positive interaction information values. The connection between v-structures and negative interaction information is exploited



(a) fork (b) chain 1 (c) chain 2

Figure 3.8: Triplets of variables with positive interaction information $\mathcal{I}(X_i, X_j, X_k)$ corresponding to the conditional dependency relation $X_i \perp\!\!\!\perp X_k | X_j$.

by the two hereafter presented network inference methods and and causal filter criterion.

3.3.2.2.1 Mutual information 3

The **mutual information 3 (MI3)** algorithm computes the 'three-way mutual information' score for each triplet of variables X_i, X_j and X_k

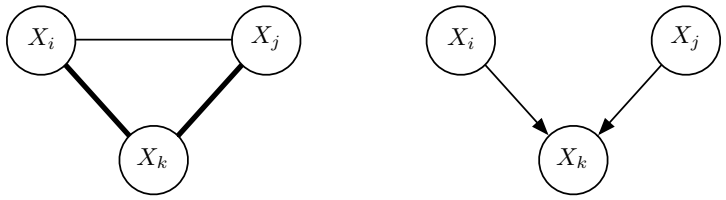
$$MI3(X_i, X_j, X_k) = I(X_i; X_j, X_k) - \mathcal{I}(X_i, X_j, X_k) \tag{3.21}$$

which is the difference between the mutual information $I(X_i; X_j, X_k)$ (equation (2.35)) and the interaction information $\mathcal{I}(X_i, X_j, X_k)$ (Definition 2.28) [LHW08]. The rationale

behind this combination of scores is that the former detects pairs of regulators X_j, X_k that describe well the target X_i. On the other hand, the latter will return a low negative score whenever the pair X_j, X_k is more informative with the target than the individual variables. Therefore, the score will yield high values for v-structures. Conflicting orientations are resolved by taking the arc's direction with the higher of the two MI3 values [LHW08].

3.3.2.2.2 Synergy augmented CLR

A second approach using interaction information based on maximizing its negative value is presented in [WLW+09], by using the so-called **synergistic regulation index (SRI)**. The main assumption behind the SRI is that whenever the interaction information of a triplet X_i, X_j, X_k is negative, the collider is that variable in a triplet exhibiting the highest pairwise mutual information with the pair of the other two variables, see Figure 3.9. This is equivalent the lowest pairwise mutual information occurring between



(a) The edge's thickness represent's the mutual information of the associated variables.

(b) The variable with the highest mutual information with the other two is the collider.

Figure 3.9: Underlying idea of the SRI algorithm.

the two parents X_i and X_j. Thus, the following score measures the confidence that X_i cooperatively regulates X_k.

$$SRI(X_i; X_k) = \max_{\substack{X_j \in \mathbf{X}. \\ X_j \neq X_i \neq X_k \\ I(X_i;X_j) < I(X_i;X_k) \\ I(X_i;X_j) < I(X_j;X_k)}} \{-\mathcal{I}(X_i, X_j, X_k)\}. \tag{3.22}$$

The interaction information for a triplet has to be 'negative' and 'significantly high'. The SRI score was used in combination with the CLR algorithm, see Section 3.3.2.1.3, to obtain a directed network, replacing the mutual information correction by the sum of the mutual information and the SRI

$$z_i = \max\left(0, \frac{I(X_i; X_k) + SRI(X_i; X_k) - \hat{\mu}_i}{\hat{\sigma}_i}\right), \tag{3.23}$$

where $\hat{\mu}_i$ and $\hat{\sigma}_i$ are the mean and the standard deviation of the sum's $I(X_i; X_k) + SRI(X_i; X_k)$, $k = 1, \ldots, n$, empirical distribution.

3.3.2.2.3 min-Interaction Max-Relevancy

[BM10] propose the use of interaction information in a causal filter criterion. Inspired by mRMR, the *min-Interaction Max-Relevancy* (mIMR) criterion selects at each step the variable that is at the same time the most informative with the target and also exhibits low average negative interaction information with the target and the already selected variables. Let X_T denote the target variable and $\mathbf{X_S}$ the already selected variables. Then, the algorithm selects at the next step the variable that maximizes

$$\arg \max_{X_k \in \mathbf{X}_+ \setminus \mathbf{X_S}} \left\{ I(X_k; X_T) - \frac{1}{|\mathbf{X_S}|} \sum_{X_i \in \mathbf{X_S}} \mathcal{I}(X_k, X_T, X_i) \right\}, \tag{3.24}$$

where $\mathbf{X}_+$ denotes the set of variables with positive mutual information.

3.4 Ensemble methods

3.4.1 GENIE3

GEne Network Inference with Ensemble of trees (GENIE3) [HTIWG10] is an inference method which ranks variables based on tree ensembles where each variable in the data set plays the role of target variable once. The Random Forests technique is used to build an ensemble of trees, where each tree is built on a bootstrap sample. The ensemble of trees for one target variable is then used to compute rankings for the remaining variables. This leads to n variable rankings which are subsequently used to compute an interaction ranking. The complexity is $\mathcal{O}(nTkm \log m)$, where n is the number of variables, T the number of trees, m the sample size and k the number of input variables. The final network is a directed network in the sense that the scores, that is the interaction rankings, for a pair of variables X_i, X_j do not have to be equal for both directions. In order to obtain the directed network, the direction with the higher value is kept. By default, the R implementation which is available from the author's webpage[1] builds a model with 1000 trees per target variable and $k = \sqrt{(n-1)}$. GENIE3 was winner of both, the DREAM5 *Network Inference* and the DREAM4 *In Silico Size 100 Multifactorial sub-challenge* [MPS+10a].

3.4.2 Combining methods

In [MCKf+12], a large set of techniques was applied to three different data sets based on the DREAM challenges [MPS+10a]. The authors compare the performance of each single method to the result the combination of them obtains. They showed that even though some single methods might beat the combination on one data set, only the combination of methods performs well on all data sets. Other conclusions drawn from this study by the authors:

i. The combination of many methods is robust against inclusion of poorly performing method if the number of these is not too high, the authors recommend $< 20\%$.

ii. The inclusion of well performing methods was always beneficial. Therefore it is still important to develop good inference methods.

[1] http://www.montefiore.ulg.ac.be/~huynh-thu/software.html

3.5 Kernel-based methods

The main concept behind kernel-based methods is to represent the data using kernel matrices. These matrices can be interpreted as adjacency matrices and thus as graphical representations of the relationships between the variables in the data. The two major reason for using kernel representations are: i) different types of data can be represented in the same mathematical framework and thus can be combined easily [YVK04, YG08] and ii) non-linear relationship between variables can be easier detected after the kernel transformation [STC04].

In this section we will discuss how to represent different data types with kernels, techniques to combine the kernel matrices and finally methods to derive networks from these kernel matrices. We present two main categories of methods: unsupervised (Section 3.5.2) and supervised (Section 3.5.3). For methods in the former category, the kernel transformation is applied to the data and a network is obtained by applying different techniques such as thresholding. For the latter category, prior knowlegde is used as a scaffold in order to determine a kernel matrix that matches the prior knowlegde as closely as possible.

3.5.1 Basic definitions and properties

"A function that returns the inner product between the images of two inputs in some feature space is known as kernel function" [STC04].

Definition 3.11 *[STC04]*
*A **kernel** is a function K that for all $\mathbf{x}, \mathbf{z} \in \mathbf{X}$ satisfies*

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle , \tag{3.25}$$

where ϕ is a mapping from $\mathbf{X}$ to an (inner product) feature space F

$$\phi : \mathbf{x} \to \phi(\mathbf{x}) \in F. \tag{3.26}$$

Property 3.12 *[STC04]*
Kernel matrices are positive semi-definite.

Theorem 3.13 *[STC04]*
A function $K : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ which is either continuous or has a countable domain, can be decomposed into a feature map ϕ into a Hilbert space F applied to both its arguments followed by the evaluation of the inner product in F (equation (3.25)), if and only if it finite and positive semi-definite (Appendix B.4).

3.5.1.1 Different types of input data in kernel representation

The idea of kernel representation is that patterns present in the data can be detected more easily in the feature space than from the actual data itself [STC04]. Two typical

data sets in bioinformatics are expression data and adjacency matrices of known connections obtained for example by experiments or literature mining. These different types of data sets require different kernel functions as the underlying structure is fundamentally different. We consider an $m \times n$ data matrix $\mathbf{D}$, where m is the number of samples and n is the number of variables. The prior matrix $\mathbf{P}$ is of dimension $n \times n$ and $p_{ij} = p_{ji} = 1$ if there is a known interaction $X_i - X_j$ and 0 otherwise.

3.5.1.1.1 Data matrices

A data matrix $\mathbf{D}$ of dimension $m \times n$ can be transformed into a symmetric positive definite kernel function K, such that for two columns $\mathbf{x}$ and $\mathbf{z}$ the Gaussian radial basis function kernel is defined as

$$K(\mathbf{x}, \mathbf{z}) = \exp\left\{-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right\}. \tag{3.27}$$

A second natural choice is the linear kernel

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{x} \cdot \mathbf{z} = \sum_{k=1}^{m} \mathbf{x}_k \mathbf{z}_k. \tag{3.28}$$

When considering the variables in the data set to be $X_1, \ldots, X_n$, the kernel matrix then is of dimension $n \times n$, with values $K(\mathbf{x}_i, \mathbf{x}_j)$, $\forall i, j \in 1, \ldots, n$.

3.5.1.1.2 Adjacency matrices

When an $n \times n$ adjacency matrix $\mathbf{P} = (p_{ij})$ is being transformed into a kernel, the following kernel is often used

$$K(P) = \exp\{\beta(\mathbf{P} - \mathbf{A})\}, \tag{3.29}$$

where $\beta > 0$ and $\mathbf{A}$ is the diagonal matrix of node connectivity where a_{ii} is the number of adjacencies of variable X_i [KL02]. This is known as **diffusion kernel**. However, the evaluation of this kernel requires the computation of the matrix exponential and thus the diagonalization of the Laplacian $\mathbf{P} - \mathbf{A}$ [STC04] which has a complexity of $O(n^3)$ and returns a dense matrix [TSS05].

3.5.1.2 Combination of different data source in the kernel space

The data integration takes as input the different kernel representations and combines them into a new kernel. The most straight-forward way to integrate l kernels $K_1, \ldots, K_l$ is to compute the sum over all kernels $K = \sum_{i=1}^{l} K_i$, see [PWCG01, YVNK03, YVK04]. Another possibility is to average the kernels [DBTvOM07] to obtain an integrated kernel K

$$K = \frac{1}{l} \sum_{i=1}^{l} \frac{K_i}{\beta_i}, \quad \beta_i > 0, \tag{3.30}$$

where β_i is used to correct the kernels' different scaling. This circumvents that some kernels might be dominating or underrepresented. The authors in [DBTvOM07] proposed to choose the β_i proportional to the trace (the sum of the diagonal) of the concerned kernel K_i.
When some data sets seem more relevant to the question at hand, it could be sensible to include a weighting parameter $\mu_i > 0$ accounting for this fact [DBTvOM07]

$$K = \frac{1}{l} \sum_{i=1}^{l} \mu_i \frac{K_i}{\beta_i}, \quad \beta_i > 0. \tag{3.31}$$

A second aim of this parameter is to automatically downweigh data sets that exhibit a lot of noise [DBTvOM07].

3.5.2 Determining an undirected network: unsupervised kernel methods

The simplest way to obtain a network from the kernel K, is to consider the value at the i, jth position of the kernel as a measure of similarity and take the highest values to derive the network [YVK04] (Algorithm 14).

Algorithm 14: Unsupervised kernel method

Input: data, vertex set $\mathbf{V}$, threshold, kernel K
Output: Undirected network: set of edges $\mathbf{E}$

```
14.1 Initialize E = ∅;
14.2 foreach pair of variables X and Y in V do
14.3 |   Compute K(x, y);
14.4 |   if K(x, y) > threshold then
14.5 |   |   Add the edge X − Y to E;
14.6 |   end
14.7 end
```

Based on spectral clustering [NJW01] and subsequently on kernel principal component analysis [SSM98], a network is built in the feature space by considering the problem of finding clusters of size two, that is pairs of variables. Given the kernel K, the network is obtained by first projecting all variables onto the subspace defined by the first principal component and then selecting those pairs which are similar [YVK04] (Algorithm 15).
An alternative to use kernels in unsupervised learning is to apply kernels in canonical correlation analysis (CCA) [Aka01, BJ03, Ver03, YVNK03]. The main idea of CCA is that whenever a multiple input, multiple output regression system needs to be solved, it may help to linearly transform the input features into a lower dimensional space and maximizing the correlation in that space. The problem is then defined as determining the transformation such that the correlation in the new space is maximized. Since the correlation will only detect linear dependencies, the main incentive to apply a kernel

Algorithm 15: Unsupervised kernel clustering

Input: data, vertex set $\mathbf{V}$, kernel $K : \mathbf{V}^2 \to \mathbb{R}$, set $\mathcal{H}$ of real-valued functions $\{f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}), (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n\}$ endowed with the norm $\|f\|_{\mathcal{H}} = \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$, number of principal components l

Output: Undirected network: set of edges $\mathbf{E}$

15.1 Initialize $\mathbf{E} = \emptyset$;
15.2 **foreach** *variable* $X_i \in \mathbf{V}$ **do**
15.3 Determine the projection $f^{(1)}(\mathbf{x}_i)$ onto the first principal direction by minimizing $\|f^{(1)}\|_{\mathcal{H}}$ under the constraint $\sum_{i=1}^{n} f^{(1)}(\mathbf{x}_i)^2 = 1$;
15.4 **foreach** $k \in 2, \ldots, l$ **do**
15.5 Determine the k-th projection $f^{(k)}(\mathbf{x}_i)$ the same way as the first with the additional orthogonality constraint $\sum_{i=1}^{n} f^{(k')}(\mathbf{x}_i) f^{(k)}(\mathbf{x}_i) = 0$ if $k' < k$;
15.6 **end**
15.7 **end**
15.8 Carry out clustering on the vectors $[f^{(1)}(\mathbf{x}_1), \ldots, f^{(l)}(\mathbf{x}_1)], \ldots, [f^{(1)}(\mathbf{x}_n), \ldots, f^{(l)}(\mathbf{x}_n)]$ to select similar variables X_i, X_j which are then added to $\mathbf{E}$.;

to the variables before maximizing the correlation is to make possible the detection of non-linear relations between the variables.

3.5.3 Including prior knowledge: supervised kernel methods

Unlike the unsupervised approach which only uses the information embedded in the kernel K, prior knowledge may be additionally used to infer the network. In practice, the feature mapping is constrained such that it matches as closely as possible the prior knowledge [VY04].

3.5.3.1 Using the prior to define a distance criterion in the feature space

[VY04] proposes a two step approach which the points are mapped into the feature space using a distance measure that takes the prior knowledge into account. This is done by assuring that known interactions are mapped to be close to each other in the feature space. In a second step the network is inferred by adding those edges to the prior network that are close to each other in the feature space as well.

3.5.3.2 Using the prior with kernel CCA

In [YVK04], each of the n variables $X_1, \ldots, X_n$ will be projected into $l < n$ principal components with the goal of defining a feature space in which interacting variables are close as described in Algorithm 15. This idea can be extended to include prior knowledge, therefore belonging to the class of supervised techniques. Assuming that prior

knowledge is perfect for the first $k < n$ variables, the goal is to identify the interactions with/between the remaining variables. This is done by extending Algorithm 15 to minimize two functions simultaneously on the known part of the network while exhibiting at the same time a high correlation between each other. The first function representing the prior knowledge and the second the data matrix.

The problem of this approach is that for the first k variables, it is assumed that all interactions in the network are correct and moreover that there are no interactions missing.

3.5.3.3 Searching for a completed prior matrix

Another approach to supervised kernel methods is presented in [TAA03]. Based on kernel representations for all data sets, the goal is to fill the unknown parts of the complete adjacency matrix based on the partial adjacency matrix representing the prior knowledge. The authors employ an EM algorithm to identify that matrix which is closest in terms of KL-divergence[1]. An extension to this idea has been proposed in [KTA05], that takes into account multiple data sets and not only fills the partial adjacency matrix but also identifies which data sets contain the most informational content.

[1] Kullback-Leibler divergence [CT90]

3.6 Conclusion

In this state-of-the-art section we outlined the most important methods to infer dependencies between variables by using different graphical models. We presented the important advances that have been made with regards to

- inference of undirected networks based on information theory,
- orientation of edge based on information theoretic measures,
- prior integration in Bayesian networks.

However, there are still open issued to be solved mainly with respect to the inference from data sets with high variable/sample ratio. The two most important open problems are:

i. How can we integrate prior knowledge with methods that have been developed to infer undirected or directed networks from such data?

ii. How can we then quantitatively validate these inferred networks?

The next three chapters are dedicated to our contributions. In Chapter 4 we tackle these two problems by developing a framework that allows prior integration with methods based on information theory and by using experimental knock-down data to validate the thus inferred networks. This framework is put into practice in Chapter 5 using experimental perturbation data obtained for colorectal colon cancer cell lines and publicly available patient tumor data. Finally, the contributions presented in Chapter 6 are threefold. We suggest extensions to existing methods in order to improve the robustness of mRMR feature selection (Section 6.1) and the orientation based on interaction information (Section 6.2). We conclude our contributions with an experimental study evaluating the influence of entropy estimation on the quality of networks inferred from genomic data (Section 6.3).

Chapter 4

An integrated methodology for causal network inference and validation

The advent of high throughput biomedical data and its widespread availability called for methods that could shed light on the question of how genes interact with each other. As presented in the state-of-the-art chapter of this thesis (Chapter 3), a manifold of methods was developed to tackle this problem mainly focusing on the inference of undirected gene-gene networks from genomic data. As presented, there exist only few techniques that can infer directed networks from these data sets with the previously discussed problems of high variable to sample ratio and large amounts of noise (Section 1.2). One possible alleviation to the difficulties posed by expression data is the integration of prior research into the inference procedure. In the best case this prior knowledge complements the information contained in genomic data to infer a resulting network that is closer to the true network. Even though Bayesian network inference algorithms (Section 3.2) infer directed networks by combining data and priors, current implementations have been designed for data sets with few variables and many samples.

As promising as the integration of prior knowledge is in order to facilitate the inference of directed networks from expression data, it comes with a price: any independent validation of an inferred network cannot rely on this prior knowledge. Currently it is common practice to validate inferred networks by checking whether high scoring edges in the inferred network correspond to known interactions. Whenever the overlap is deemed sufficiently high, other high scoring but unknown interactions are considered good candidates for subsequent experimental validation. This validation procedure is not only inapplicable to networks that were inferred from a combination of genomic data and prior knowledge due to the introduced bias towards known interactions but furthermore this procedure does not offer an easy assessment of networks inferred with different algorithms.

In this chapter we present a comprehensive framework that provides solutions to these two problems. The first part of our framework (Section 4.1.2) is devoted to the design of an inference method which infers directed networks combining high variable/low sample genomic data and prior knowledge. For the first part, we design a method that combines prior knowledge via a linear combination scheme with a feature selection procedure and an orientation scheme that extract information from the genomic data. In the second part of the framework (Section 4.1.3) we develop an independent, quantitative assessment for the inferred networks. We use experimental knock-down data in a cross-validation inspired setting, splitting the data into training data (the samples not related to the knock-down) and test data (the samples related to the knock-down). We use the training data for the inference of a directed network together with prior knowledge while we use the test data to independently quantify the quality of the inferred network.

Our framework does not only include the inference algorithm and the subsequent validation but furthermore the necessary tools to experimentally carry out the inference using our R/Bioconductor package *predictionet* (Section 4.2.1) and a web application named *Predictive Networks* (Section 4.1.1) to retrieve prior knowledge from biological databases and PubMed abstracts.

The different parts of this chapter are part of the following publications.

- [ODF+13] Catharina Olsen, Amira Djebbari, Kathleen Fleming, Niall Prendergast, Renee Rubio, Frank Emmert-Streib, Gianluca Bontempi, Benjamin Haibe-Kains & John Quackenbush. *Inference of predictive gene networks from biomedical literature and gene expression data.* Submitted to Genomics, 2013.
- [HKOD+12] Benjamin Haibe-Kains, Catharina Olsen, Amira Djebbari, Gianluca Bontempi, Mick Correll, Christopher Bouton & John Quackenbush. *Predictive Networks: A Flexible, Open Source, Web Application for Integration and Analysis of Human Gene Networks.* Nucleic Acids Research, 2011.
- [HKOD+12] Benjamin Haibe-Kains, Catharina Olsen, Gianluca Bontempi & John Quackenbush. *predictionet: Inference for predictive networks designed for (but not limited to) genomic data*, 2012. R package version 1.1.5.
- [OHKQB13] Catharina Olsen, Benjamin Haibe-Kains, John Quackenbush & Gianluca Bontempi. *On the Integration of Prior Knowledge in the Inference of Regulatory Networks.* Accepted for publication, 2013.

4.1 Methodology

Three methodological contributions are the heart of our framework outlined in Figure 4.1 and will be described in detail throughout this section: the retrieval of prior knowledge using the *Predictive Networks* web application (Section 4.1.1), the inference of directed networks from genomic data and said prior knowledge (Section 4.1.2) and finally the purely data-driven validation of the inferred networks (Section 4.1.3).

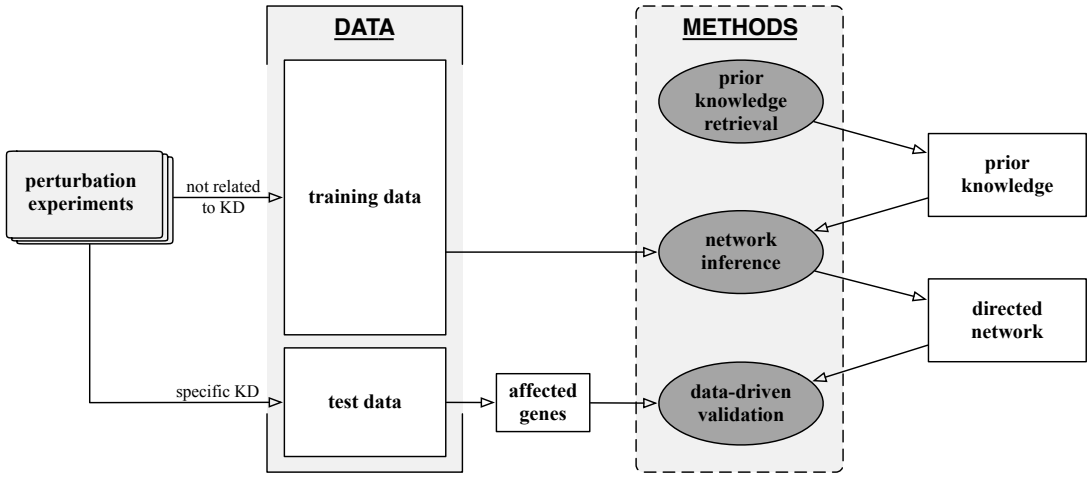


Figure 4.1: The comprehensive analysis framework, using *Predictive Networks* to retrieve prior knowledge, *predictionet* to infer a directed network from genomic data and prior knowledge and knock-down (KD) data in a cross-validation scheme for validation. From the test data we determine the set of affected genes which will then be used to build a confusion matrix and then to compute a performance score.

The key concept of our framework is the cross-validation inspired use of our experimental perturbation data. This data was generated by knocking down in turn a number of relevant genes and collecting multiple biological replicates for each of these knock-down experiments (Section 5.1).

We separated those samples that are related to one specific gene knock-down from the rest. The samples not related to the specific knock-down experiment constitute the *training data* and are used together with the prior knowledge to infer a directed network. The samples related to the specific knock-down are kept for the subsequent validation, this is the *test data*.

From the test data we can determine the set of genes that are actually affected by the perturbation experiment. This list contains the genes whose expression significantly changes with respect to the control samples in which no perturbation was inflicted upon any gene. This set of *affected genes* will then be used in our validation as these genes should be ef-

fects of the knocked down gene in the inferred network (Section 4.1.3.1). We extend this validation by furthermore proposing a set of additional quantitative evaluation strategies such as comparing the inferred network to random networks (Section 4.1.3.2) and evaluating the prediction ability using linear regression models (Section 4.1.3.3).

4.1.1 Prior knowledge retrieval

We developed the *Predictive Networks* web application [HKOD+12] with three main goals in mind. Primarily, it should allow the easy retrieval of prior knowledge for any list of genes. Secondly the web application should offer the possibility to infer a network making use of this prior knowledge together with uploaded genomic data. Lastly, it should offer the user different options to visualize the results. In this section, we will focus on the first part whereas the inference will be discussed in Section 4.1.2 and the visualization in Section 4.2.2.

4.1.1.1 Extracting prior knowledge

The first goal of *Predictive Networks* is to identify known gene-gene interactions from different repositories focusing on two principal resources: i) published biomedical literature including PubMed abstracts and PubMed Central's full-text articles and ii) biological databases including Pathway Commons [CGD+11]. As different as these two sources are structurally, as different are also the approaches to access the relevant information. While information extraction from biological databases requires little processing, the information contained in published literature can only be extracted using text-mining tools. The first step to identify gene-gene interactions by means of text mining is the definition of triples of the form

[subject; predicate; object].

The *subject* and the *object* correspond to gene names, synonyms and 'common names' as well as standard Gene Symbols. We defined a list of *predicates*, such as *'regulates', 'is inhibited by'* and furthermore negative interactions such as *'does not regulate'*. After having uploaded a list of gene, the application allows to download the list of identified gene-gene interactions together with their source. The implementation of *Predictive Networks* was carried out by Entagen[1].

4.1.1.2 Prior representation

As presented in the state-of-the-art chapter, representing prior knowledge about the existence of certain interactions between genes as priors over networks is considered to be one of the most difficult tasks in Bayesian learning. Our aim is to develop an inference technique which allows for easy integration of such prior knowledge without the need for more complicated approaches such as those explained in Section 3.2.2.2.

[1]Entagen, Newburyport, MA, 01950, USA

When prior information about specific interactions such as *'gene X_i regulates gene X_j'* is available, this can be graphically represented as $X_i \to X_j$. Whenever a set of prior information for the set of variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ is available, these are then represented using a weighted adjacency matrix $\mathbf{P} = (p_{ij})_{i,j=1,\ldots,n} \in [-1, 1]$. Values $p_{ij} = 0$ corresponds to no prior knowledge about an edge between the two variables X_i and X_j, a positive value p_{ij} corresponds to having prior knowledge about the existence of the edge $X_i \to X_j$. The higher this value, the higher is the confidence that this edge is present in the true network. When p_{ij} is negative, there is evidence for the absence of the edge from X_i to X_j, the smaller the value the stronger the evidence. We compute the strength of an interaction based on the number of times this interaction was cited in the literature and in biological databases. To reduce the influence of intensely studied interactions, we truncate the counts returned by *Predictive Networks* (Section 5.3).

4.1.2 Directed networks from genomic data and prior knowledge

The main advantage of network inference methods based on feature selection techniques employing pairwise information theoretic measures of association is that these methods are fast enough to be applied to data sets with high variable to sample ratios. We start this section by presenting a method integrating network inference based on feature selection with prior knowledge. Then we present the integration of prior knowledge into an orientation scheme. Figure 4.2 presents a flowchart of the different parts in our inference procedure which starts with expression data and prior knowledge and results in a directed network.

By adding prior knowledge to the inference process we aim at benefitting from the respective strengths of network inference based on feature selection using information theoretic measures of association and the integration of prior knowledge:

i. fast inference of networks from typical biological datasets with high variable to sample ratio,

ii. complement information available in the data with information from prior knowledge,

iii. obtain a more robust network.

The complete algorithm's pseudo-code is described in Algorithm 16.

4.1.2.1 Undirected networks with priors

One successful approach to infer undirected networks from biomedical data is the application of feature selection techniques using information theoretic measures of association (Section 3.3). Typically, each variable in the data set is considered to be the target variable once and the most relevant features with respect to this target variable are selected. In order to integrate prior knowledge into this procedure, we will modify the rankings using prior knowledge and a weighting scheme which allows to take into account the

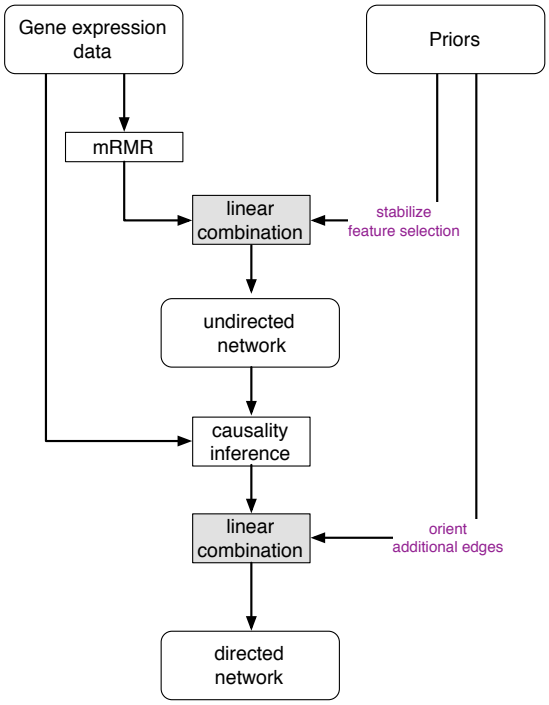


Figure 4.2: Our prior integration approach: One the one side we start using mRMR to obtain a feature ranking for each of the variables in the expression data. Then we combine this with prior knowledge in a linear combination scheme to obtain an undirected network. Subsequently, we use an orientation scheme based on interaction information and combine this in a linear combination with the prior as well to obtain the final network. The prior knowledge serves two different purposes: to stabilize the feature selection in the first step and to orient additional edges in the second step.

confidence in the prior. Given the ranking based on the chosen feature selection technique and the prior knowledge, we add a variable to the final set of selected variables in the following cases (ordered by descending final ranking).

- The feature is ranked high in the feature selection procedure and there is high confidence from prior knowledge in this interaction.
- The feature is strongly supported by either the data or the prior knowledge. Depending on the weight given to either source, one might be more influential than the other.
- Both sources only support the feature weakly but there are no strongly supported features.

Relevance of a feature with the target variable is often defined as mutual information between these two variables. Using this information theoretic measure allows fast estimation of the relevance score between two variables while at the same time offers the

possibility to detect both linear and non-linear relationships between variables. More recent methods add a layer of complexity trying to take the target variable's context into account (CLR), avoiding indirect relations (ARACNE) and avoiding to select redundant variables with respect to the previously selected variables (MRNET) (Section 3.3.2.1).

Similarly to MRNET we base our algorithm on the minimum redundancy maximum relevance (mRMR) criterion [DP05] which selects a variable $X^* \in \mathbf{X} \setminus \mathbf{X_S}$ by maximizing the relevance with the target variable X_T while at the same time minimizing the redundancy with the previously selected variables $\mathbf{X_S}$ using the following maximization

$$X^* = \arg \max_{X_i \in \mathbf{X} \setminus \mathbf{X_S}} \left\{ I(X_i; X_T) - \frac{1}{|X_\mathbf{S}|} \sum_{X_k \in X_\mathbf{S}} I(X_i; X_k) \right\}. \tag{4.1}$$

Therefore, a variable's X_i mRMR score s_i is computed by evaluating the difference between its relevance with the target $I(X_i; X_T)$ and its redundancy with the previously selected variables $\frac{1}{|X_\mathbf{S}|} \sum_{X_k \in X_\mathbf{S}} I(X_i; X_k)$

$$s_i = I(X_i; X_T) - \frac{1}{|X_\mathbf{S}|} \sum_{X_k \in X_\mathbf{S}} I(X_i; X_k). \tag{4.2}$$

The mRMR scores of a pair of variables X_i and X_j, s_i and s_j respectively, are not necessarily symmetric and thus the final mRMR score between a pair of variables is then determined by choosing the maximal score.

Computing the mRMR score for each variable results in an adjacency matrix $\mathbf{M} = (m_{ij})_{i,j=1,\dots,n}$. We scale this matrix to have values in the interval $[-1, 1]$ by dividing it by $\max |m_{ij}|$. We now use the prior knowledge to modify the obtained rankings in the rescaled mRMR matrix $\mathbf{M}$ according to the strength of the two sources' information. As seen before in Section 4.1.1.2, the prior knowledge can be represented by a weighted adjacency matrix $\mathbf{P} = (p_{ij})_{i,j=1,\dots,n}$ with $p_{ij} \in [-1, 1]$.

We then combine the two matrices $\mathbf{P}$ and $\mathbf{M}$ such that a weight $w \in [0, 1]$ determines the degree of confidence in the prior versus that in the data

$$w \cdot \mathbf{P} + (1 - w) \cdot \mathbf{M} \quad \in [-1, 1]. \tag{4.3}$$

This linear combination potentially changes the ranking of each variable balancing thus information from the data via the mRMR matrix $\mathbf{M}$ and the information contained in the prior knowledge matrix $\mathbf{P}$.

The network is pruned to keep only a predefined number of adjacencies for each variable, hereafter denoted by *maxparents*.

Our prior integration scheme is fulfilling the key principles of a good algorithm defined in [GHB13]:

- Only that part of the prior which is also supported by the data should be included.
- Using a structure prior must not limit the ability to learn the part of the network for which no prior information exists.
- The weight given to prior knowledge can be controlled.

The combination of the two matrices **M** and **P** can be interpreted as two estimators of the true network. The network **M** constructed from data exhibits high variance whereas the network **P** based on prior knowledge high bias. As described in Section 1.3.1, the combination of such estimators results in an improved estimation compared to either one by itself. Furthermore, the two matrices can be interpreted in the Bayesian framework: the matrix obtained from prior counts can be interpreted as prior probability (when scaling it appropriately) and the mRMR matrix as the evidence from data.

4.1.2.2 Directed networks with priors

After having identified an undirected network from genomic data and prior knowledge, we now use a second step to orient this network. As before, we will combine a criterion based on information theory with prior knowledge in a linear combination scheme controlling the weight given to the data versus that given to the prior knowledge via the parameter $w \in [0, 1]$.

Given an undirected network, triplets of variables can be oriented based on different conditional (in)dependence relations (Section 3.3.2.2). On the one hand, three connected variables $X_i - X_j - X_k$ with $X_i \neq X_k$ for which $X_i \not\perp\!\!\!\perp X_k | X_j$ can be oriented into a v-structure. On the other hand, this conditional independence relation can equivalently be expressed using interaction information

$$\mathcal{I}(X_i, X_j, X_k) < 0. \tag{4.4}$$

The three remaining possible structures for a triplet of variables, fork and chains in two directions correspond to the dependence relation $X_i \perp\!\!\!\perp X_k | X_j$ and equivalently to $\mathcal{I}(X_i, X_j, X_k) > 0$. Using negative interaction information to orient edges is the basis of both the MI3 algorithm (Section 3.3.2.2.1 [LHW08]) and SRI (Section 3.3.2.2.2 [WLW+09]). Sometimes an undirected edge is part of more than one triplet of variables. In this case its score is the minimum between the interaction information over all such triplets.

We start the second part of our algorithm by computing the interaction information $\mathcal{I}(X_i, X_T, X_j)$ for each triplet of variables $X_i - X_T - X_j$ with $X_i \neq X_j$. The score for

the edge $X_j \to X_T$ is given by

$$c_{jT} = \max_{X_i \in \mathbf{X_C}} \{-\mathcal{I}(X_i, X_T, X_j)\}, \tag{4.5}$$

where $\mathbf{X_C} = \{X_i : X_i - X_T - X_j,\ X_i \neq X_j\}$. Subsequently, the adjacency matrix $\mathbf{C} = (c_{ij})_{i,j=1,\ldots,n}$ can be filled for each target variable X_T with the corresponding scores c_{jT}. After rescaling the matrix $\mathbf{C}$ and thus computing $\mathbf{C} = \frac{\mathbf{C}}{\max_{i,j\in[1,\ldots,n]} |c_{ij}|}$, we can proceed to integrate the prior knowledge using the weighting factor $w \in [0, 1]$ which is the degree of confidence in the prior's quality

$$w \cdot \mathbf{P} + (1 - w) \cdot \mathbf{C} \quad \in [-1, 1]. \tag{4.6}$$

Pruning the number of edges to keep only those with positive causality score and to a maximal number of parents *maxparents* for each variable provides the final network.

Algorithm 16: Inferring a directed network from genomic data and prior knowledge

Input: data, prior knowledge $\mathbf{P}$, prior weight w, maximum number of parents for each variable *maxparents*

Output: Directed network

16.1

`/* Undirected network */`

16.2 Compute scaled mRMR matrix $\mathbf{M}$ from data;

16.3 Combine $\mathbf{M}$ and $\mathbf{P}$: $w \cdot \mathbf{P} + (1 - w) \cdot \mathbf{M}$;

16.4 Keep *maxparents* top scoring adjacencies for each variable in $\mathbf{V}$;

16.5

`/* Directed network */`

16.6 Compute scaled causality score matrix $\mathbf{C}$ from data given undirected network;

16.7 Combine $\mathbf{C}$ and $\mathbf{P}$: $w \cdot \mathbf{P} + (1 - w) \cdot \mathbf{C}$;

16.8 Keep *maxparents* top scoring parents with positive causality score for each variable in $\mathbf{V}$;

4.1.3 Validation

One of the big problems when inferring networks from biomedical data is that the majority of interactions is unknown. This implies that comparing the inferred network to the complete true underlying network is impossible. The two most common approaches to validate the inferred networks are i) the experimental validation of the inferred interactions and ii) the comparison to the known interactions. However, these two approaches both have drawbacks. The experimental validation is costly and time consuming whereas the comparison to known interactions cannot be carried out when using this knowledge as prior information to bias the inference process. Furthermore, inference methods that

do not rely on prior knowledge can be evaluated by comparing the resulting network to the known interactions. However, this procedure can be problematic due to the prior knowledge being far from complete in most cases and in the worst case not related to the studied problem.

Due to these problems, we present a novel, purely data-driven approach based on experimental knock-down data to quantitatively validate inferred networks. When knocking down one gene in an experiment, we can observe the effect this knock-down has on the remaining genes. This then allows us to identify a set of truly affected genes. We then check whether these genes are found downstream from the knocked down gene in the inferred network. Subsequently we can compute a performance score for each inferred network. Thus we can determine whether inferred interactions are indeed present in the medical problem we are studying.

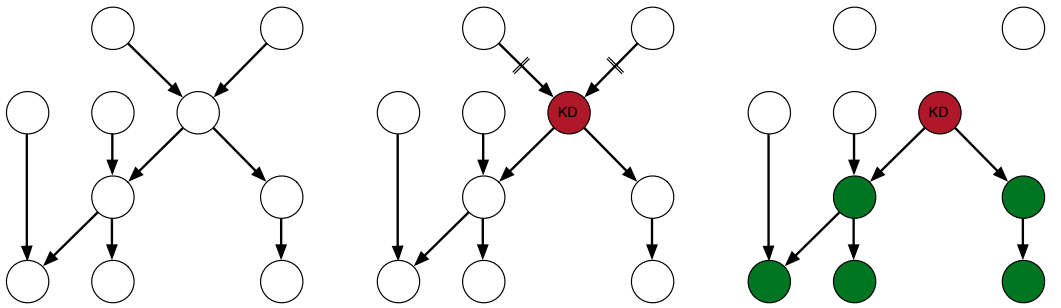
The second part of our data-driven validation is the comparison of the inferred network with randomly generated networks. It is difficult to interpret any obtained score for an inferred network. One way to assure that a result is meaningful is to compare the inferred network's score to the scores obtained using state-of-the-art methods. A second strategy is to compare the obtained result to results obtained using random networks. The generation of random networks allows us to establish a baseline result which should be attained using any inference technique.

The final validation we propose is going one step further by using the inferred network as basis for regression models. Each target variable is modeled using the set of inferred parents. These regression models are then used to predict the expression values of the target variable. This validation is carried out in cross-validation, using the training set to infer the network and to build the regression models and the test data for the subsequent predictions.

4.1.3.1 Using knock-down data

Whenever we discuss causal relationships between gene X_i and X_j such that $X_i \rightarrow X_j$ in a network, we mean implicitly that a manual change of X_i will affect the expression value of X_j and subsequently the expression values of all other children, grandchildren, etc. (Figure 4.3). In theory, this type of experimental data is very informative and thus would allow us to infer a better network compared to networks inferred from observational data only. However, a more realistic scenario due to the time and cost constraints is that only a few genes are perturbed. In this case, the inference might result in networks that do not represent reality. Therefore, we propose a different approach to make use of this type of data.

Knowing which gene was manually modified, we can identify the set of genes that are significantly affected by the manipulation, in Figure 4.3(c) these are the genes colored green.



(a) unperturbed network (b) knocking down one gene (c) network after perturbation

Figure 4.3: Three states of a network: the unperturbed network 4.3(a). Knocking down one gene 4.3(b) and thus removing the KD's ancestors from influencing the knocked down gene. The network after perturbation 4.3(c): those vertices that are still affected by the KD are highlighted in green.

More precisely, we compare the expression values of a gene in the unperturbed state with that after modification. If the difference is significant, then it belongs to the childhood (CH) of the modified gene. In the following we will denote this set of affected genes by $\mathbf{X_A}$. Unlike the childhood of the perturbed gene, its parents, grandparents, and other ancestors will not be affected by the perturbation and their expression should not be different compared to their expression in the unperturbed state.

After inference, we can classify the genes in the network using the confusion matrix in Table 4.1. All those genes that were inferred as part of the perturbed gene's childhood and at the same time were affected by the perturbation are classified as true positives (TP). Genes that were affected by the perturbation but are not inferred as part of perturbed gene's childhood but instead for example as a parent will be classified as false positive (FP). The genes that were not inferred as part of the childhood are classified as false negatives (FN) if they were affected by the perturbation and therefore should have been part of the childhood. Otherwise they are classified as true negatives (TN).

	$\in \mathbf{X}_A$	$\notin \mathbf{X}_A$
$\in$ childhood	TP	FP
$\notin$ childhood	FN	TN

Table 4.1: Confusion matrix used to classify genes in an inferred network, more precisely to compare the list of genes in the childhood of the perturbed gene with those known to be truly affected by this perturbation ($\mathbf{X}_A$).

In Figure 4.4, we graphically show the procedure using a list of affected genes X_1, X_4 and X_5 and an inferred network in which the perturbed gene's childhood consists of X_3, X_4, X_5 and X_6. In this example X_4 and X_5 are true positives because they are

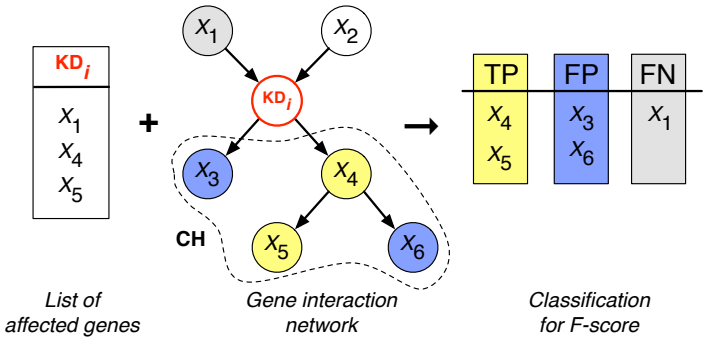


Figure 4.4: Given are a set of genes affected by a knocked down gene KD_i and a gene interaction network. Affected genes in the CH are classified as true positives (TP). All other genes in CH are classified as false positives (FP). Affected genes not knock-down's CH are classified as false negatives (FN).

in the list of truly affected genes and at the same time part of the perturbed gene's childhood. Two genes, X_3 and X_6, are inferred as part of the childhood but are not directly affected by the perturbation therefore they are classified as false positives. Finally X_1 is truly affected but inferred as parent of the perturbed gene and thus a false negative.

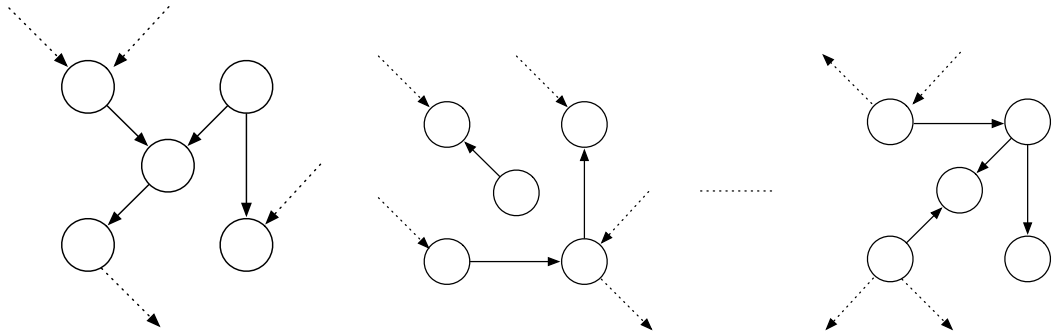
With the confusion matrix in hand we can compute different quality measures to quantitatively evaluate the corresponding network. Standard quality measures are presented in Section B.3.1.

4.1.3.2 Using random networks

An important step in a validation process is to verify that the obtained networks' quality could not be obtained by chance. The first possible strategy is to estimate the network's null distribution by randomizing the data and applying the same inference algorithm to each of these data sets. A second approach is to randomly swap connections in the network [MSOI$^+$02]. For example replacing the connections $X_1 \to X_2$ and $X_3 \to X_4$ by $X_1 \to X_4$ and $X_3 \to X_2$. A third approach was used in [MPS$^+$10b] in which edges were predicted randomly. The authors showed the difficulty to beat the performance of these randomly guessed interactions. As the first approach would be computationally heavy, we decided to generate random networks using a modified version of the second approach.

In order to verify whether the inferred network could have been obtained by chance, we decided to generate a large number of random networks which mimic the inferred network. In our inference algorithm we set the parameter *maxparents* which determines the maximum number of parents each variable in the network can maximally have. We keep this parameter also for the random networks. As we carry out a feature selection procedure which does not impose a predefined distribution on the edges, we designed the random network generation to select uniformly a number of parents for each variable

while leaving the total number of edges in the network equal to the total number of edges in the inferred network. Obtaining relevant results with this approach ensures that our two-step procedure makes sensible use of the information in the data and prior knowledge. Otherwise the inferred network will not be significantly better than the random addition of at most *maxparents* parents to each node.



(a) inferred network (b) k random networks

Figure 4.5: We infer one network using *predictionet* 4.5(a); we then generate k random networks 4.5(b) in which each node has the same maximum number of parents and the total number of edges is the same as the number of edges in the inferred network

After generating k random networks, we compute for each random topology the F-score (Appendix B.3.2) for the chosen childhood distance and compute the p-value via

$$p = \frac{\sum_{i=1}^{k} \mathbb{1}_{\left(F_{\text{random}_i} \geq F_{\text{inferred}}\right)}}{k} \tag{4.7}$$

testing whether the obtained F-score from the inferred topology is significantly better than what could be reached when choosing edges randomly. Apart from verifying that the F-score obtained by the inferred network is sufficiently high, this comparison to random networks serves another purpose, it restricts the number of edges to be included in the network: if the number of edges is too high, it will be easier to beat the inferred network.

4.1.3.3 Predictions

One important problem in network medicine is to use gene-gene networks to predict pathways related to a studied disease. Once we understand how genes interact, targeted treatment can be developed taking out key genes in these pathways. Our rationale in predicting gene expressions is that a gene's parents should be good predictors. Under this assumption we can also use the prediction framework to validate the inferred network. This offers another purely data-driven solution to the validation problem. The main idea is to use a gene's parents to predict the expression level of this gene. We assume linear dependencies between the genes and split the data into training and test

set. The training set is used to fit the parameters of the model and the test set is then used for the prediction and the subsequent validation using different quality measures such as R^2 and NRMSE (Section B.3.2).

Adopting linear dependencies between nodes, we fit the model for each gene $X_i, i = 1, \ldots, n$

$$X_i = \beta_0 + \sum_{X_j \in \mathbf{pa}(X_i)} \beta_j X_j, \tag{4.8}$$

where $\mathbf{pa}(X_i)$ denotes the set of parents of X_i returned by the structural step. The fitting of the model parameters β_j is obtained using the expression values of the corresponding variables $\mathbf{pa}(X_i)$.

4.1.3.4 Stability

When inferring networks from expression data, the algorithm has to cope with the large variable to sample ratio. This often leads to problems with the robustness of the result. Small changes in the data set might result in significant differences in the resulting network. We will measure the stability of a network as part of our validation strategy, assuming that an algorithm that infers edges more robustly is considered superior to a less robust algorithm. In a cross-validation scheme, we define the stability of an edge between variables X_i and X_j as the number of times it was inferred over the total number of repetitions

$$\text{stability}(X_i, X_j) = \frac{\#\text{ inferred}}{\#\text{ cross-validation repetitions}}. \tag{4.9}$$

This quantity assumes values in the interval $[0, 1]$, where 0 corresponds to an edge that has never been inferred and 1 to an edge that is inferred in all cross-validation repetitions.

Algorithm 17: Data-driven network validation

Input: data, specific KD, inferred network, set of affected genes for KD, childhood distance from the KD, number of random networks k

Output: Quality scores

17.1
`/* Scores based on confusion matrix */`

17.2 Determine the confusion matrix for KD based on the KD's childhood in the network and the set of affected genes;

17.3 Compute evaluation scores based on confusion matrix, for example F-score;

17.4
`/* p-value: inferred versus random networks */`

17.5 Generate k random networks;

17.6 **foreach** *random network* **do**

17.7 | Determine corresponding confusion matrix and F-score;

17.8 **end**

17.9 Compute p-value for inferred network via equation (4.7);

17.10
`/* Stability */`

17.11 Compute stability of the network by running a cross-validation scheme;

17.12
`/* Prediction scores */`

17.13 Build regression model for each gene using its parents in the network;

17.14 Make predictions using the obtained model;

17.15 Evaluate predictive ability using for example R^2 or NRMSE;

4.2 Implementation

4.2.1 R/bioconductor package *predictionet*

This section has been published in parts in [OHKQB13] and in the R/Bioconductor package *predictionet*'s vignette [HKOBQ12]. The package contains a set of scripts implementing the methods presented in this chapter. The available functionalities can be roughly grouped into three categories.

- *Network inference based on genomic data and prior knowledge:* The user can choose the weighting factor between information from data and from prior knowledge and the maximum number of parents for each gene.
- *Validation based on stability of the inferred network and on prediction scores based on a linear regression model using cross-validation:* The inference function has been extended to a cross-validation scheme which automatically computes the stability and the prediction scores.
- *Visualization by exporting to *.gml*[1] *file format:* Each network can be exported together with edge and node properties for subsequent visualization using tools such as Cytoscape [SOR+11] .

4.2.1.1 Inferring networks

In the following, we will describe how to infer a network using the prior matrix following the method described in Section 4.1.2. The function `netinf` is the heart of the package as it carries out the inference procedure. The necessary parameters of this function are presented in Table 4.2.

In what follows we present the package's functionalities using the data set and prior matrix provided in the package.

```
rm(list=ls())
library(predictionet)
goi=dimnames(annot.ras)[[1]][order(abs(log2(annot.ras[ ,"fold.change"])),
 decreasing=TRUE)][1:30]

mynet=netinf2(data=data.ras[,goi], priors=priors.ras[goi,goi],
 priors.count=TRUE, priors.weight=0.5, maxparents=4, causal=TRUE)
```

The returned object contains the following elements: `topology`, `edge.relevance` and `edge.relevance.global`, contain information concerning the inferred network: the adjacency matrix and the corresponding scores as defined in equation (4.6) after and before pruning, respectively.

[1] http://www.opengeospatial.org/standards/gml

parameter	definition
`data`	matrix of continuous or categorical values; observations in rows, features in columns
`priors`	matrix of prior information available for gene-gene interaction
`priors.count`	`TRUE` if priors specified by the user are number of citations (count) for each interaction, `FALSE` if probabilities or any other weight in $[0, 1]$ are reported instead
`priors.weight`	real value in $[0, 1]$ specifying the weight to put on the priors (0=only the data are used, 1=only the priors are used to infer the topology of the network)
`maxparents`	maximum number of parents allowed for each gene
`causal`	`TRUE` if a directed network should be inferred; `FALSE` to infer an undirected network

Table 4.2: Parameters of the `netinf` function.

4.2.1.2 Validation

An extension to the `netinf` function has been implemented in the `netinf.cv` function to automatically carry out

- a cross-validation procedure and
- a set of predictions using linear regression models.

This wrapper function makes use of three separate functions as described in Table 4.3.

function	definition
`predictionet.stability.cv`	performs a k-fold cross-validation procedure to compute the stability of each edge
`net2pred`	computes linear regression models on the basis of the inferred network
`netinf.predict`	uses the regression models to predict outcomes for each sample and each variable

Table 4.3: Functions needed for the cross-validation validation procedure.

4.2.1.2.1 Cross-validation

A k fold cross-validation scheme is implemented in the function `predictionet.stability.cv` such that for k times a network is inferred using $k-1$ folds of the training set and tested on the remaining fold. The first measure returned by the cross-validation is a matrix containing the stability (Section 4.1.3.4) of each edge in the inferred network `topology` using the entire dataset `edge.stability`.

```
mynet.stability.cv = predictionet.stability.cv(data=data.ras[,goi],
 priors=priors.ras[goi,goi],  priors.count=TRUE, priors.weight=0.5,
 maxparents=4, causal=TRUE, nfold=10,method="regrnet")
```

The object `topology.cv` contains a list of the topologies inferred during the k cross-validation runs. Furthermore the stability for edges inferred in any of the cross-validation repetitions is captured in `edge.stability.cv`.

Linear models
As described in Section 4.1.3.3, we use linear dependencies between the nodes in our prediction model. The fitting of the model's parameters β_j is obtained using the expression values of the corresponding variables $\mathbf{pa}(X_i)$. The linear models are can be computed using the function `net2pred`. In practice the linear models for an inferred network can be obtained using the following command.

```
mypred<-net2pred(net=mynet,data=data.ras[,goi])
```

In addition to the objects generated using the `netinf` function, the linear models were added in the object `lrm`. Three different quality measures for the predicted outcome are implemented: R^2, NRMSE and MCC.

The previously described functionalities are combined in the function `netinf.cv`. This function requires two additional parameters compared to those in Table 4.2, see Table 4.4, namely the number of cross-validation folds `nfold` and the number of categories `categories` needed for data discretization and the subsequent computation of the MCC.

parameter	definition
`nfold`	number of folds for the cross-validation
`categories`	number of categories used to discretize each variable

Table 4.4: Additional parameters of the `netinf.cv` function.

The wrapper function can be invoked using the following instruction.

```
mynet.cv = netinf.cv(data=data.ras[,goi], priors=priors.ras[goi,goi],
 priors.count=TRUE, priors.weight=0.5, maxparents=4, causal=TRUE,
 nfold=10, categories=3)
```

4.2.1.3 Visualization

The network object can be exported in `*.gml` format with the `netinf2gml` function as follows.

```
netinf2gml(mynet.cv)
```

Two files `predictionet.gml` and `predictionet.props` are generated as output. Importing these with Cytoscape [SOR+11] allows a visualization as presented in Figure 4.6 using the stability to color code the edges and the prediction score R^2 for the nodes.

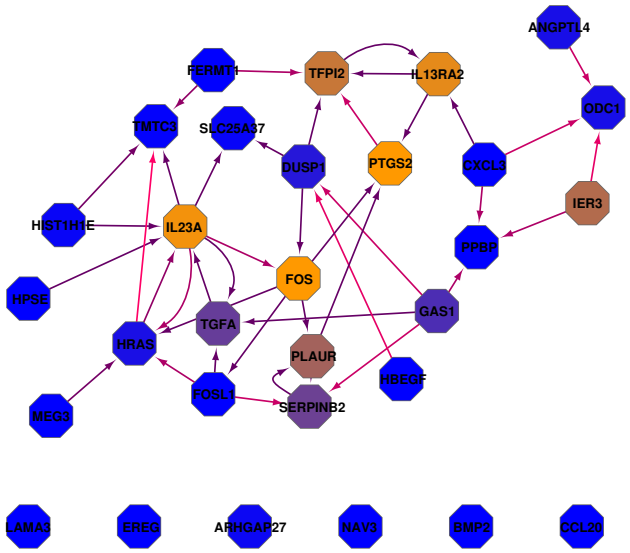


Figure 4.6: Visualization of the inferred network variables' prediction score using Cytoscape. Color coding of the nodes according to the R^2 prediction score, low values correspond to blue nodes, high prediction scores to yellow nodes. The edges are color coded according to their stability, low stability corresponds to red edges, high stability to purple edges.

4.2.2 Integration into web application *Predictive Networks*

The web application *Predictive Networks* [HKOD+12] was developed with two main goals in mind: i) An easy way to retrieve prior knowledge about gene interactions from biological databases and PubMed abstracts. ii) Conveniently infer networks using uploaded expression data and the available prior knowledge. In Figure 4.7, the welcome screen is presented providing the possibilities to upload a set of genes of interest and eventually the genomic data to carry out a full analysis.

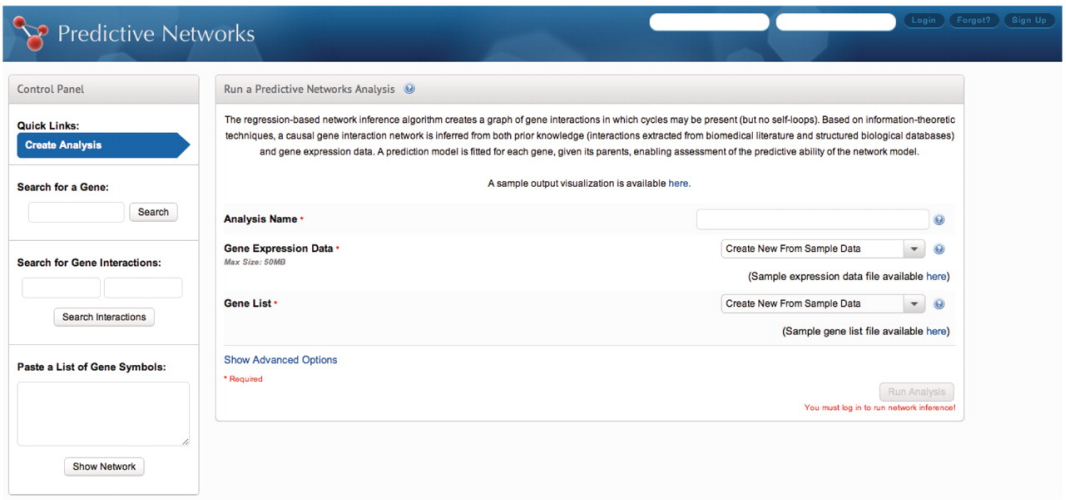


Figure 4.7: Front page of the *Predictive Networks* web application displaying the four entry points: the single gene, single gene-gene interaction and gene list searches, and the network inference analysis panel. The top left panel provides a series of quick links to ensure easy navigation between the different web pages which compose the *Predictive Networks* web application.

Predictive Networks contains a total of 81,022 interactions from PubMed documents and 1,323,776 interactions from biological databases, namely the Human Functional Interaction [WFS10] and the Pathways Common [CGD+10] databases (statistics retrieved on 2012-11-16).

The *predictionet* package was integrated into the web application allowing the user the same freedom to choose the maximum number of parents and the weight given to the prior. Once the inference is finished, the user can visualize the results by comparing the inferred edges to what has been provided as part of the prior knowledge and what parts of the networks are new (Figure 4.8). That is, when inferring a network with prior weight $w \in]0, 1[$, the edges which are part of the prior knowledge and part of this inferred network are colored yellow, edges which are in the inferred network are colored green and edges which have not been inferred but are part of the prior knowledge are colored red.

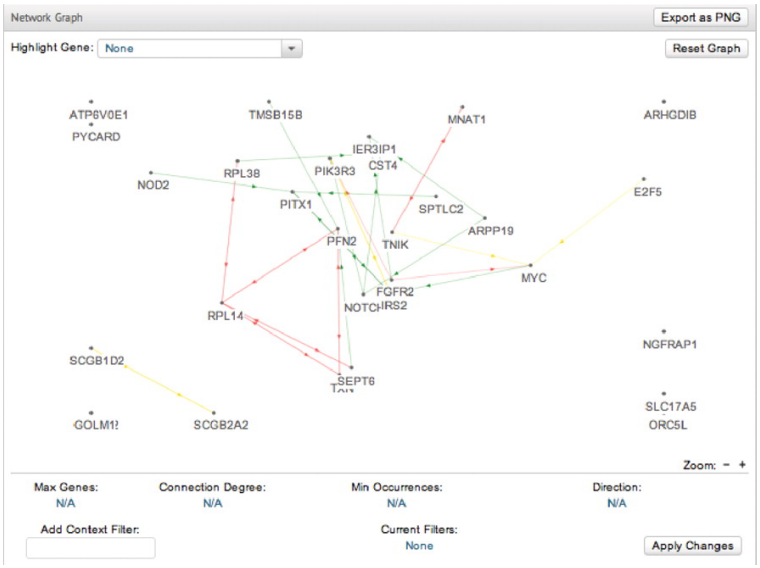


Figure 4.8: Analysis of the inferred edges: red edges are only present in the prior knowledge, green edges were not present in the prior knowledge but are inferred using the genomic data and yellow edges were known in the prior knowledge and inferred from the genomic data.

4.3 Conclusion

This chapter is the heart of the main contribution of this thesis. In this chapter we present the computational framework we developed to

- retrieve prior knowledge using *Predictive Networks*,
- infer directed networks using genomic data and prior knowledge for small sample/large variable data sets and
- validate these networks quantitatively following the proposed validation procedure using knock-down data.

Using *Predictive Networks* we can retrieve known gene-gene interactions from biological databases, PubMed abstracts and open access PubMed articles. Our inference algorithm extends mRMR feature selection to integrate prior knowledge and thus can handle data sets with large variable to sample ratio typical for genomic data. We use the obtained undirected network as basis for an orientation scheme that uses interaction information in combination with prior knowledge.

We implemented this procedure in the R/Bioconductor package *predictionet* and presented its usage in the final section of this chapter. While there exists a stand-alone version, it was also integrated with *Predictive Networks* to provide an easy way to upload data, retrieve prior knowledge and use the two sources directly to infer a directed network.

The final methodological contribution is a validation procedure based on the availability of experimental knock-down data. We will use these different components in the following chapter in a case study on colon cancer to infer directed networks from knock-down data and prior knowledge obtained via *Predictive Networks*.

Chapter 5

Causal discovery in colon cancer

We described in the introduction the impact that cancer has on public health. As one of the leading causes of death in the world, continuing research efforts are necessary to further understand the mechanisms that guide cancer growth and thus hopefully identify possible targets for medical treatment. In this chapter we address three important open problems in network inference using biological data related to the RAS pathway in colorectal cancer. We provide a detailed description of the experiments in Section 5.1.

Even though more and more network inference methods are being developed only little attention has been paid to the design of techniques that would allow a quantitative evaluation of the inferred networks' quality. In this chapter we show experimentally that with the proposed validation framework (Section 4.1.3), we can quantitatively assess the inferred networks.

The second problem is related to the use of prior research results via tools such as *Predictive Networks*. The retrieved knowledge is not problem-specific (this would be very limiting in the number of known interactions) and therefore we have to first ensure that the retrieved priors are relevant to colorectal cancer.

The final problem is to infer meaningful large scale directed networks from genomic data and prior knowledge. In this experimental section we show that *predictionet* can make use of these two sources to infer better networks compared to those from either source alone. We show that *predictionet* outperforms *GeneNet*[1] and furthermore that that the combination of the two sources as described in Section 4.1.2 is beneficial both in terms of F-score values and in terms of achieved significance with respect to random networks using our validation framework.

We tackle these three problems by applying our methodological contributions to two types of experimental colon cancer data: i) data obtained from perturbation experiments performed on two different cell-lines and ii) publicly available patient tumor data. Our

[1] We refer to the proposed method by the name of its implementation in R.

analysis pipeline is depicted in Figure 5.1.

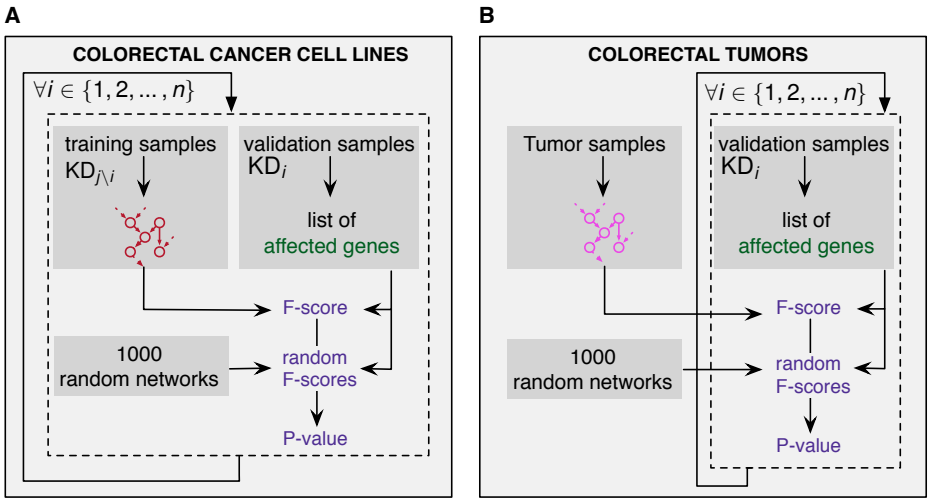


Figure 5.1: Analysis pipeline devised for the knock-down study. **A** using cell line data: for each of the knock-down i: the samples are split into training samples which are the samples not related to knock-down i and validation samples which are the samples related to knock-down i. For each of the knock-downs, one network is inferred and then validated. **B** using tumor samples: the validation samples are again the samples related to knock-down i. However, the training samples are the tumor samples from which one network is inferred. For both settings, the validation includes i) computing the F-score using the validation samples of knock-down i, ii) generating random networks and computing their F-scores, iii) computing a p-value to test whether the inferred network performs significantly better than the random networks.

In the first step (Figure 5.1**A**), we use experimental knock-down data in a cross-validation scheme by

i. setting samples related to one knock-down experiment aside (*validation samples*),

ii. using the validation samples to determine the set of affected genes by means of statistical tests,

iii. inferring one directed network from the remaining samples (*training samples*) and prior knowledge (Section 5.4),

iv. validating the inferred network using the set of affected genes as described in Section 4.1.3.

We repeat steps (i)-(iv) for each of the knock-downs and thus obtain one network and one set of validations (F-score and p-value) for each of the knocked down genes.

In the second step (Figure 5.1**B**), we proceed with the first two steps as before for each of the knocked down genes. This provides us again with a set of affected genes for each knocked down gene. However, instead of inferring one network for each of the knocked down genes using the corresponding training samples, we use patient tumor data to infer one directed network. We then validate this network by evaluating the quality of each knocked down gene's childhood as described in Section 4.1.3. In Section 5.5 we show that we obtain similar results for networks that were inferred from patient tumor data as for those inferred from cell line samples.

5.1 The knock-down data

We performed gene perturbation experiments (gene knock-downs) focusing on the RAS pathway, a well-studied and key pathway in colorectal cancer [BSBDN+07, SHS05]. We used two colorectal cancer cell lines, SW480 and SW620 [LSM+76], to measure the effects of targeted perturbations in the RAS pathway. These targeted perturbations are performed using RNAi knock-downs systematically silencing eight key genes in the RAS pathway as identified from PubMed and BioCarta in 2007: CDK5, HRAS, MAP2K1, MAP2K2, MAPK1, MAPK3, NGFR and RAF1, Figure 5.2[1]. In order to efficiently

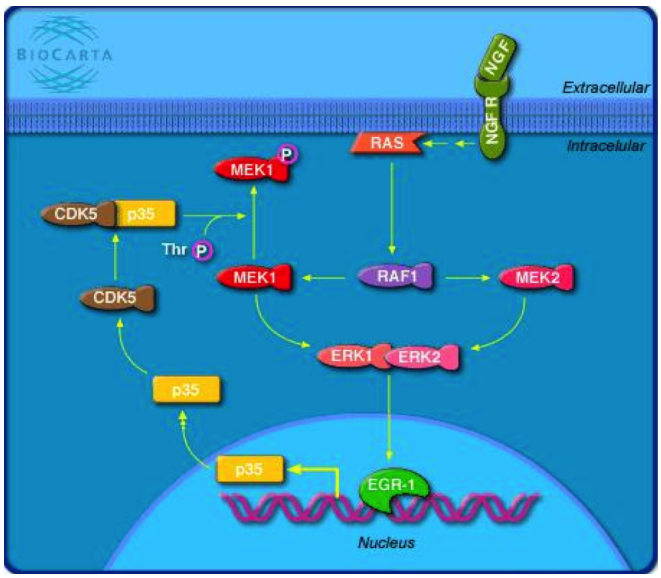


Figure 5.2: The RAS pathway as in BioCarta 2007.

silence each of the eight genes in turn we designed short hairpin RNA complexes with complementary sequence to the mRNA sequence of these genes of interest[2]. In each perturbation experiment the shRNAs are delivered to the cell via a lentivirus, which activates the enzyme dicer. The shRNAs assemble then with RISC (RNA Induced Silencing Complex), which unwinds and dissociates the sense strand; the resulting complex is an activated RISC complex bound to the anti-sense strand. This complex then binds to its target mRNA sequence and cleaves it, so the degraded mRNA cannot be translated and resulting gene expression is silenced. We then collected the media of the tissue culture cells after 72 hours post transfection. Three types of controls have been used:

- Empty Vector (EV): Vector containing no shRNA
- Non Target: Vector containing an shRNA that does not target any known human genes.

[1]Synonyms: MAPK1-ERK2, MAPK3-ERK1, MAP2K1-MEK1, MAP2K2-MEK2
[2]Supplementary File to [ODF+13]: rtpcr_primers.csv

- Non Transduced: these are SW480 and SW620 cells that did not get infected with any virus.

The experiments were done in six biological replicates where both the SW480 and SW620 cell lines were extracted for gene expression profiling[1]. We performed genome-wide gene expression profiling of control (no perturbation) and perturbed cells (gene knock-down) using the Affymetrix GeneChip HG-U133 PLUS2 platform. CEL files were normalized using *frma* [MBI10]. We used the *jetset* package to select a unique probeset for each gene symbol, which resulted in a set of 19,218 unique gene symbols; further annotations were obtained using *biomaRt* [DMK$^+$05]. The raw and normalized data will be deposited on the NCBI Gene Expression Omnibus repository [BST$^+$05] with the publication of [ODF$^+$13].

With the goal of inferring a larger network than that between the described eight genes, we needed to identify a set of genes linked to the RAS pathway in addition to the core list of eight key genes. We achieved this by comparing gene expression profiles of quiescent cell lines with cell lines over-expressing RAS [BYC$^+$05]. These gene expression data were generated using the Affymetrix GeneChip HG-U133 PLUS2 platform and normalized using MAS5 [Aff04]; the normalized data are available from GEO (GSE3151). Similarly to our perturbation dataset, we used the *jetset* and *biomaRt* packages to annotate each probeset. We used a Wilcoxon Rank Sum test to statistically compare the ten control cell lines with the ten cell lines over-expressing RAS in order to select the genes that are most differentially expressed (false discovery rate FDR (Appendix B.3.1) $< 10\%$ and fold change $>= 4$), which resulted in a list of 332 genes including HRAS. The RAS-associated genes and the eight knock-down genes (for a total number of 339 genes selected for further analysis) are listed together with their corresponding statistics in [ODF$^+$13][2].

[1] Supplementary File to [ODF$^+$13]: rubio2011_marray_demo.csv
[2] Supplementary File to [ODF$^+$13]: sig_ras.xls

5.2 Parameter selection

Besides the data set, the parameters required by *predictionet* are the maximum number of parents and the prior weight. We set the maximum number of parents to ten and infer one network for each knock-down using *predictionet* with different prior weights $w \in \{0, 0.25, 0.5, 0.75, 0.95, 1\}$. The weight $w = 0$ corresponds to inferring the network from data only. In order to verify *predictionet*'s performance, we compare it to directed networks inferred using *GeneNet* (Section 3.1.3).

We used the Wilcoxon Rank Sum test (knock-down vs. non target control, Wilcoxon Rank Sum test, $FDR < 10\%$) to identify the list of genes statistically significantly affected by the knock-down of each of the eight core genes of the RAS pathway. The number of affected genes for each of the eight knock-down genes are presented in Table 5.1.

KD	CDK5	HRAS	MAP2K1	MAP2K2
# affected genes	73	122	33	38
KD	MAPK1	MAPK3	NGFR	RAF1
# affected genes	117	59	99	61

Table 5.1: Number (#) of statistically significantly genes affected by KD (out of 339 genes) with FDR $< 10\%$

A difficulty to handle is to find the right size of the childhood, that is to answer the question which distance from the modified gene is still considered to be in the childhood and thus should be affected by the perturbation. It is usually a trade-off between having too few genes when only considering a very small distance such as direct children and a too large childhood when choosing a too large distance. In that case the number of genes might approach the number of genes in the network. In our experiments, we will consider distances one, two and three which correspond to only direct children (CH1), direct children and grandchildren (CH2) and children, grandchildren and great-grandchildren (CH3).

Concerning the comparison of networks inferred with *GeneNet*, we need to prune this network to allow a fair comparison. We chose to take the number of edges obtained with *predictionet* using prior weight of 0.5 for two reasons. Firstly, although GeneNet is not designed to natively integrate prior knowledge, we sought to show that combining data and prior using *predictionet* will yield better results than state-of-the-art methods that only use data. Secondly, F-score values tend to be higher for networks with more edges, therefore choosing a prior weight which results in an advantageous number of edges, such as 0.5, seems reasonable.

5.3 Extraction of prior knowledge from biomedical literature and databases with *Predictive Networks*

We used the *Predictive Networks* (PN) web-application to identify gene-gene interactions reported in the biomedical literature and in biological databases (Section 4.1.1); these known interactions are referred to as *priors*.

subject	predicate	object	direction	evidence	sentence	article	network
PTEN	inhibits	MDM2	right	positive	These results suggest that PTEN inhibits MDM2 and protects p53 through both p13k/Akt-dependent and -independent pathways.	PubMed:14559824	Medline Abstracts
RBBP8	interacts with	BRCA1	right	positive	Originally identified in a yeast two-hybrid screen for proteins that bind to the BRCA1 C-terminus (BRCT) domain of BRCA1 and confirmed throughbinding experiments, RBBP8 interacts with BRCA1 during Gof the cell cycle.	PubMed:19007437,PubMed Central:2582801,Other:1471-2091-9-S1-S7,DOI:10.1186/1471-2091-9-S1-S7	Pubmed Open Access Texts,Pathway Commons
LIF	enhances	POMC	right	positive	Another important endotoxin-inducible cytokine in FS cells is LIF., LIF enhances POMC expression and ACTH secretion in synergy with CRH.	DOI:10.1111/j.1365-2826.2007.01616.x,PubMed Central:2229370,PubMed:18081553	Pubmed Open Access Texts,Medline Abstracts
LIF	enhances	POMC	right	positive	Another important endotoxin-inducible cytokine in FS cells is LIF., LIF enhances POMC expression and ACTH secretion in synergy with CRH.	DOI:10.1111/j.1365-2826.2007.01616.x,PubMed Central:2229370,PubMed:18081553	Pubmed Open Access Texts,Medline Abstracts
CDK5	phosphorylates	DAB1	right	positive	CDK5 phosphorylates DAB1 on serine 491 independent of reelin signaling, suggesting an intersection of pathways.	DOI:10.1007/s10989-006-9034-3,PubMed:19617920,Other:9034,PubMed Central:2710985	Pubmed Open Access Texts,Medline Abstracts
CDK5	phosphorylates	DAB1	right	positive	CDK5 phosphorylates DAB1 on serine 491 independent of reelin signaling, suggesting an intersection of pathways.	DOI:10.1007/s10989-006-9034-3,PubMed:19617920,Other:9034,PubMed Central:2710985	Pubmed Open Access Texts,Medline Abstracts
UHRF1	interacts with	DNMT1	right	positive			Pathway Commons
LIF	activated	STAT3	right	positive	Genomic targets of LIF activated STAT3 were therefore identified by ChIP-chip analysis of AtT-20 cells treated with LIF for 20 minutes.	PubMed Central:2562516,DOI:10.1371/journal.pgen.1000224,Other:08-PLGE-RA-0414R2,PubMed:18927629	Pubmed Open Access Texts,Medline Abstracts
LIF	activated	STAT3	right	positive	Genomic targets of LIF activated STAT3 were therefore identified by ChIP-chip analysis of AtT-20 cells treated with LIF for 20 minutes.	PubMed Central:2562516,DOI:10.1371/journal.pgen.1000224,Other:08-PLGE-RA-0414R2,PubMed:18927629	Pubmed Open Access Texts,Medline Abstracts

Figure 5.3: Sample output for the 339 RAS associated genes using *Predictive Networks*. The output specifies the subject, predicate, object triplet, the direction of the interaction, the type of evidence and the sentence containing the triplet and the corresponding reference.

We looked for known gene interactions involving at least one gene contained in the RAS signature. Among the RAS-associated genes, 323 are present in the PN web-application for a total of 37,212 identified interactions, in particular, 602 interactions occurred between two RAS-associated genes. In Figure 5.3, a sample output of nine interactions is presented.

Each interaction is characterized by their citation count, that is number of positive evidence minus number of negative evidence, note that negative evidence is characterized by a predicate such as *'does not regulate'*. To reduce influence of very large citation counts, all counts are truncated by the 95^{th} percentile of their absolute values and subsequently divided by this quantile. These transformed counts are stored into the *prior matrix* $\mathbf{P}_{n\times n}$ where n is the number of RAS-associated genes and where each value of $\mathbf{P}$ lies in $[-1, 1]$.

5.4 Application to colorectal cancer cell line data

Following the pipeline illustrated in Figure 5.1A, we first inferred a network for each of the eight knock-downs using the training samples from perturbation data (Section 5.1). An example network is presented in Figure 5.4 for the knock-down of HRAS and prior weight equal to 0.5.

We observe in Figure 5.4 that truly affected genes are present within the set of children as well as with the grandchildren (yellow nodes). Due to the size of the network we

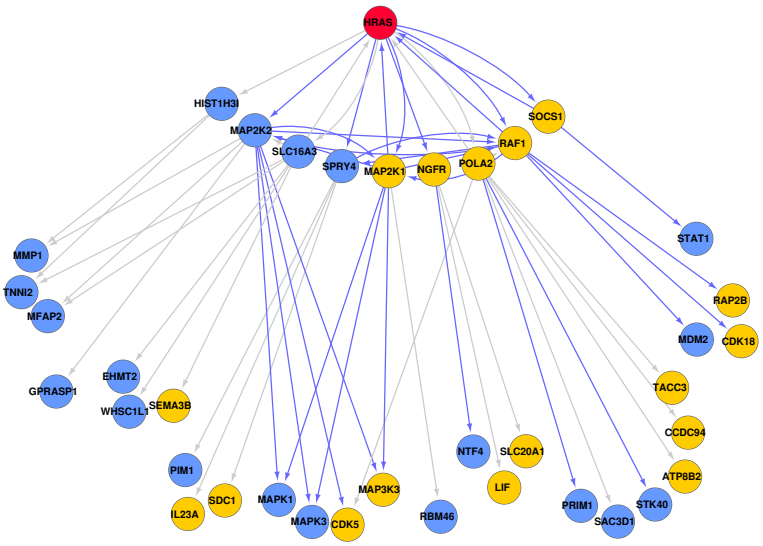


Figure 5.4: Children and grandchildren of HRAS (red node) inferred using *predictionet*, equal weight between training data and prior knowledge (prior weight $w = 0.5$). The yellow nodes (genes) are the ones identified as significantly affected by HRAS based on the validation samples. The remaining nodes, colored in blue, have been predicted as affected during network inference while they were not identified as significantly affected in the validation samples. Blue edges are known interactions (priors) while grey edges represent new interactions.

cannot visualize the affected genes that are not part of the children or grandchildren. However, our validation framework allows us to compute a performance score for each of these networks (one network per knock-down and per prior weight). The obtained characteristics of these networks are provided in Table 5.2: the number of edges, the number of genes in CH2, the F-score and the number of true positives.

A summary of the results consisting of the networks' F-scores inferred using GeneNet and *predictionet* with the selected prior weights are presented in Figure 5.5A. Each of the bars is color coded to represent the significance of the F-scores when compared with the

1000 randomly generated topologies. Figure 5.5B shows the percentage of true positives with respect to the total number of affected genes for each of the methods, color coded by their "source", which is either only genomic data, only prior knowledge, both, or new ones.

5.4.1 Priors are informative

To the best of our knowledge, the informational value of priors retrieved from biomedical literature and biological databased has not yet been quantitatively assessed in the context of gene network inference. Undisputedly, these known interactions are often the result of biological experiments that are valid in the context in which they have been performed. This does not necessarily mean that they carry information with respect to our specific biological data set.

We observe that networks inferred from priors only (prior weight equal to one) are informative as they yielded significant F-scores for all the knock-downs except NGFR (Figure 5.5A and Table 5.2). For NGFR there is only one direct child which is not sufficient to compute a meaningful F-score.

5.4.2 Networks from genomic data only: *predictionet* vs *GeneNet*

We started inferring networks for each of the knock-down experiments using only the genomic data and no prior knowledge. We can observe in Figure 5.5 and Table 5.2 that *predictionet* with prior weight equal to zero obtains significantly better F-scores than random networks for CDK5, HRAS and NGRF and with a p-value < 0.1 for MAPK1. *GeneNet* does not perform significantly better than random networks for any of the eight KDs. This could be explained by the low number of samples in the data sets.

	CDK5				HRAS			
	#edges	CH2	TP	Fscore	#edges	CH2	TP	Fscore
GeneNet	1006	2	1	0.026667	1058	0	0	0
0	586	16	6	0.13483	661	16	7	0.10145
0.25	802	43	12	0.2069	861	28	12	0.16
0.5	1006	62	18	0.26667	1058	37	17	0.21384
0.75	1024	78	24	0.31788	1077	37	17	0.21384
0.95	1029	78	24	0.31788	1080	37	17	0.21384
1	313	49	13	0.21311	313	29	16	0.21192
	MAP2K1				MAP2K2			
	#edges	CH2	TP	Fscore	#edges	CH2	TP	Fscore
GeneNet	1044	0	0	0	1025	3	1	0.04878
0	637	0	0	0	610	21	3	0.10169
0.25	831	34	5	0.14925	811	49	9	0.2069
0.5	1044	46	5	0.12658	1025	61	11	0.22222
0.75	1062	46	5	0.12658	1043	61	11	0.22222
0.95	1066	46	5	0.12658	1048	61	11	0.22222
1	313	36	5	0.14493	313	33	5	0.14085
	MAPK1				MAPK3			
	#edges	CH2	TP	Fscore	#edges	CH2	TP	Fscore
GeneNet	1017	0	0	0	1016	0	0	0
0	607	6	4	0.065041	593	13	0	0
0.25	818	66	27	0.29508	783	53	5	0.089286
0.5	1017	100	36	0.3318	1016	87	22	0.30137
0.75	1033	100	36	0.3318	1031	94	22	0.28758
0.95	1035	100	36	0.3318	1033	94	22	0.28758
1	313	58	22	0.25143	313	48	11	0.20561
	NGFR				RAF1			
	#edges	CH2	TP	Fscore	#edges	CH2	TP	Fscore
GeneNet	1014	0	0	0	1025	0	0	0
0	578	7	5	0.09434	603	11	1	0.027778
0.25	787	18	9	0.15385	792	26	5	0.11494
0.5	1014	10	6	0.11009	1025	58	13	0.21849
0.75	1027	10	6	0.11009	1044	59	13	0.21667
0.95	1031	10	6	0.11009	1049	59	13	0.21667
1	313	1	0	0	313	29	9	0.2

Table 5.2: Inference using knock-down data in cross-validation: # denotes the number of edges in the inferred network; CH2 denotes the number of genes in the KD's childhood consisting of children and grandchildren; TP and F-score denote the number of true positives and F-score for the childhood consisting of children and grandchildren.

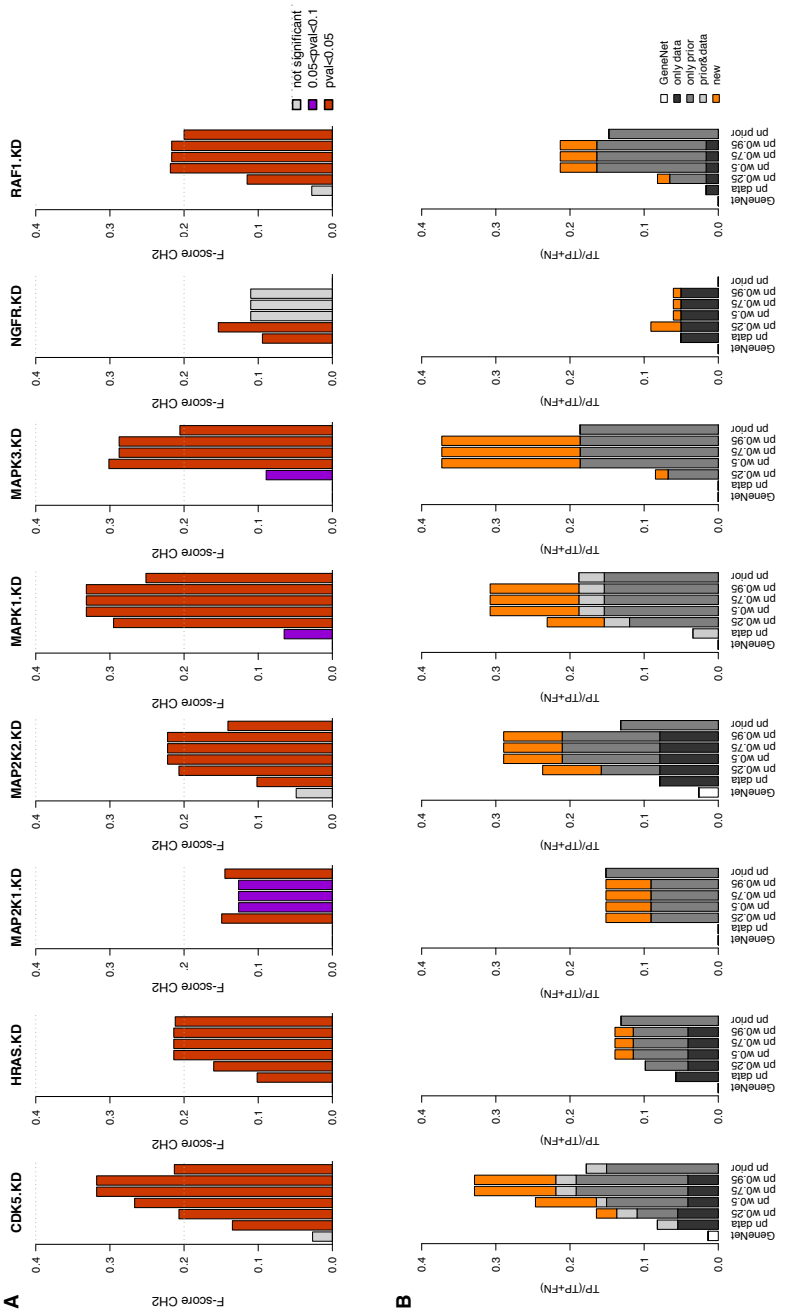


Figure 5.5: In cancer cell lines, performance of gene interaction networks inferred from genomic data only (*GeneNet* and *predictionet* (pn) with priors weight $w = 0$), *predictionet* with priors only (prior weight $w = 1$) and *predictionet* using combinations of both data sources (prior weight $w = \{0.25, 0.5, 0.75, 0.95\}$). Each column reports the the performance of the network validated for each KD. (**A**) Bars represent the F-scores of each network in each validation experiment; they are colored with respect to their significance, that is in red and purple when network's F-score is higher than 5% and 10% of random networks, respectively. (**B**) Bars' heights represent the percentage of true positives with respect to the total number of affected genes for each KD's network; they are colored by their origin: black for true positives identified in the network inferred from genomic data only, dark grey from priors only, light grey in both, and orange for true positives that are uniquely found in networks inferred by combining genomic data and priors.

5.4.3 Improved networks when combining data and priors

To test whether combining prior knowledge with genomic data improves the quality of the inferred gene interaction networks, we statistically compare F-scores obtained for networks inferred from

- data only (*GeneNet* and *predictionet* with priors weight $w = 0$),
- priors only (priors weight $w = 1$)
- a combination of data and priors (priors weight $w \in]0, 1[$).

We observe in Figure 5.5A that networks inferred from a combination of priors and genomic data yield consistently higher F-scores than networks inferred from data only (Wilcoxon signed rank test comparing the eight values obtained for data only with the eight values obtained for prior weight $w = 0.5$ yielded $p = 0.004$). When we compare the networks inferred combining genomic data and prior knowledge to networks inferred from priors only, we observe improvement for five out of eight KDs (CDK5, MAP2K2, MAPK1, MAPK3 and NGFR), which yield significance (Wilcoxon signed rank test $p = 0.01$ for priors weight $w = 0.5$). Moreover the networks inferred from combined data sources are significantly better than random networks in most cases, except for NGFR for which the prior knowledge is limited (Figure 5.5A).

We further assess the benefit of combining data sources by counting how many true positives can only be found by combining priors and genomic data, that is they are not present in the data only and/or priors only networks (Figure 4.4). In other words it does not suffice to combine the data/priors only networks to get these true positives. Figure 5.5B represents the portion of true positives that can be found in the networks inferred from genomic data only, priors only or the combination of both. We observe in Figure 5.5B that there is little overlap between true positives identified in networks inferred from genomic data only or priors only, suggesting that priors and genomic data provide very different information regarding gene interactions.

5.5 Application to colorectal tumor data

Having shown that the knock-down experiments provide a way to independently quantify the quality of an inferred network, we now seek to apply our validation framework and network inference approach in a large dataset of 292 colorectal human tumors[1]. Following our analysis pipeline in Figure 5.1, we infer a gene interaction network using the entire dataset as training set and use the KD experiments to assess networks' quality. This is a challenging task as colorectal cell lines are imperfect models for their patient's tumors counterpart [GCV$^+$11, MVT$^+$06].

The networks inferred from colorectal tumor data are denser than those inferred from cell lines (Table D.3); this is expected due to the larger sample size of the tumor dataset ($\sim$300 vs $\sim$100 for the colorectal tumor and cell lines, respectively). Despite the difference in network density, the F-scores are not statistically significantly different to those observed in the cell lines experiments (Wilcoxon signed rank test $p \geq 0.10$, Figure 5.6). Interestingly, *GeneNet* performs better on the tumor data than on the KD data, probably due to sample size. However, only for MAPK1 it yields significant results compared to random networks (Figure 5.6). On the contrary, networks inferred using combination of genomic data and priors yield significant F-scores in most cases, except for NGFR which is consistent with the results from the cell line experiments. Combining data with the prior knowledge improved F-scores for CDK5, MAP2K1, MAP2K2, MAPK1, MAPK3 and RAF1 which also corresponds to the results obtained from cell lines data.

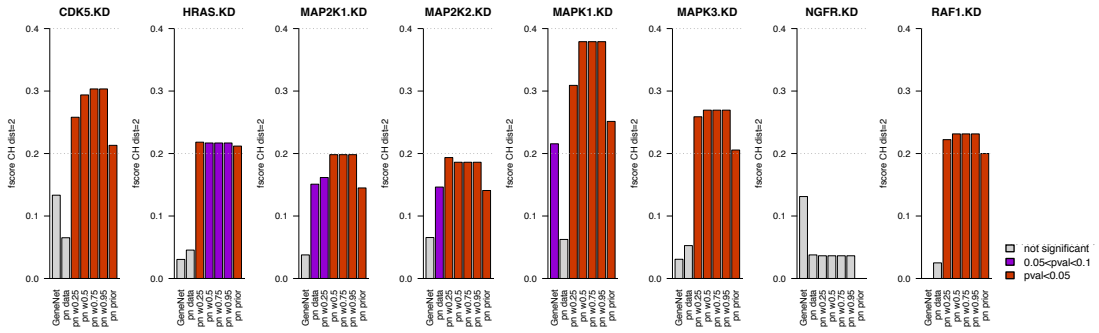


Figure 5.6: In tumor data, performance of gene interaction networks inferred from genomic data only (*GeneNet* and *predictionet* (pn) with priors weight $w = 0$), *predictionet* using priors only (priors weight $w = 1$) and *predictionet* using a combination of both data sources (priors weight $w = \{0.25, 0.5, 0.75, 0.95\}$). Each column reports the the performance of the network validated in each KD. Bars represent the F-scores of each network in each validation experiment; they are colored with respect to their significance, that is in red and purple when network's F-score is higher than 5% and 10% of random networks, respectively

[1] `https://expo.intgen.org/geo/`

5.6 Comparison of gene interaction networks inferred from colorectal cell lines and tumors

Given that the networks inferred from colorectal cancer cell lines and tumor data yielded similar F-scores, we compared their topologies to identify the edges inferred in both datasets and those specific to either cell lines or tumors. For this section we inferred one network using the complete KD data set. This cell line network and the tumor network shared on average 22% of edges depending on the methods (4%, 5%, 20%, 31%, 33%, 33% for *GeneNet*, *predictionet* with priors weight $w = 0, 0.25, 0.5, 0.75, 0.95$, respectively; Table D.1). As expected, the proportion of common edges increases with priors weight; however the networks shared less than one third of their edges, suggesting that either the gene interactions present in cell lines and tumors significantly differ from each other and/or that the sample size is not sufficient to infer networks that are generalizable to multiple datasets.

Moreover, we observed that most of the common interactions involve one of the eight KD genes (41%, $p < 0.001$) for *predictionet* with priors weight $w = 0.5$), suggesting using data from targeted experiments can indeed help understand diseases such as cancer. The network we obtained from samples related to cell line experiments is similar to that inferred from real tumor samples. Therefore, in the future we can experimentally generate more data in order to understand diseases for which data collection is difficult for example due to the number of patients affected by that disease. This was also supported by recent studies which suggested different ways to overcome the large variable to sample ratio in genomic data [BBAIdB07, TB07]. We illustrated this result in Figure 5.7 which represents the gene interaction network around HRAS, indeed showing that most common interactions involve at least one of the KD gene.

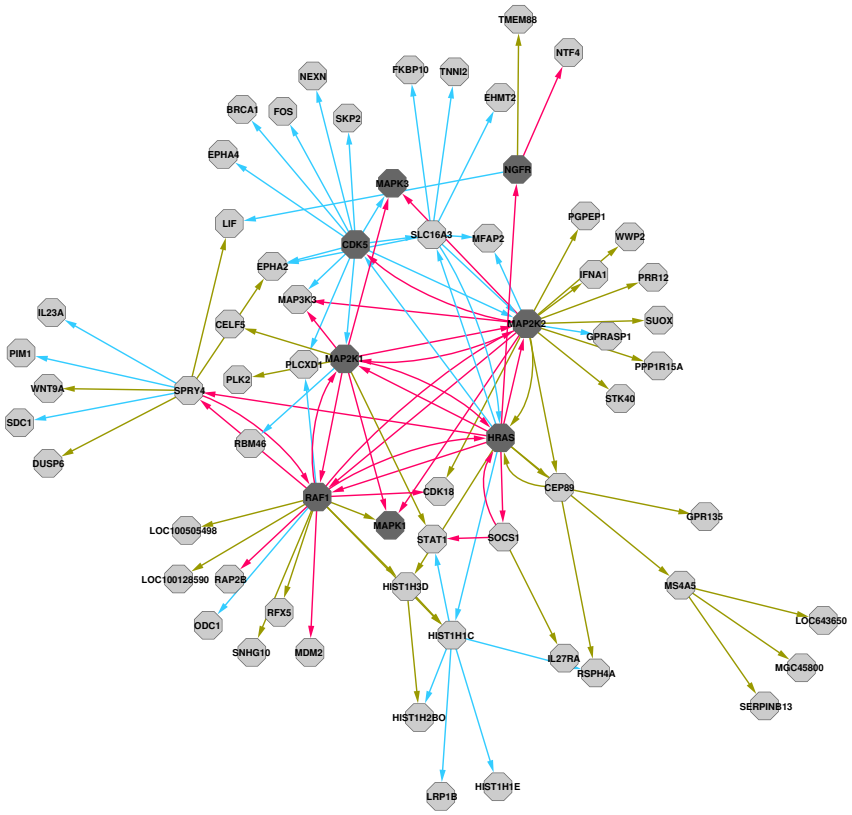


Figure 5.7: Children and grandchildren of HRAS, pink edges appear in both networks: KD using all data and expO both with prior weight equal to 0.5, blue edges only in KD and yellow only in expO; only those variables are included which are members HRAS's childhood in either the network built on the complete KD data or on the expO data set. This adjacency matrix is obtained by first adding arcs from HRAS to its children and then for each of HRAS's children adding arcs from the child to all of its children. Comparing the two networks, one inferred from all KD samples and one inferred from all expO samples, the overlap is significant $p \ll 0.05$ using Fisher's exact test [KD05, AWP+09].

5.7 Conclusion

In the first contribution chapter of this thesis, we presented a new framework that allows to validate inferred networks by using i) experimental knock-down data to compute the inferred network's performance (F-score) and ii) to assess network's performance in terms of p-values using random networks as null hypothesis to ensure statistical significance of the results. These two parts are complementary as, on one hand, only a relatively sparse networks are likely to be significantly better than random networks while, on the other hand, networks with more interactions are more likely to yield higher F-score.

In this chapter we used the proposed framework to show how difficult it is to infer networks solely based on genomic data using both *GeneNet* and *predictionet*. Furthermore, we provide evidence for the quality of prior knowledge retrieved with the *Predictive Networks* web-application. Finally, we are able to show that combining genomic data and prior networks let us achieve higher F-scores than one of the sources achieves by itself. Furthermore, these improved networks are also significantly better than random networks. When using real tumor data, we get comparable results which suggests that generalization is possible in the sense that cell-line experiments can be used for validation of patient data and furthermore that cell line experiments could help to increase the sample size of expression data sets.

In the remainder of the thesis, we work on extending certain parts of our inference procedure in order to counter some of the method's weaknesses such as the variability of the feature selection procedure. In the third contribution chapter, we propose improvements

- in the feature selection step using an ensemble approach (Section 6.1). Using an ensemble approach we will try to select features more robustly than standard mRMR feature selection does.
- in the use of interaction information, (Section 6.2). By exploiting also triplets with positive interaction information together with computing the score on a genes neighborhood allows us to possibly orient more edges than using only the maximal negative interaction information.

One bottleneck for methods based on the computation of mutual information is its estimation from data. Methods range from measures based on pairwise correlation to more complicated approaches based on plug-in estimators (Section 2.6). They are mostly selected in network inference procedures by their ability to handle the high number of variables typical in genomic data but their influence on the quality of inferred network is mostly unknown. In Section 6.3, we investigate this influence in an experimental study using both generated and publicly available biological data sets.

Chapter 6

Contributions – Extensions

For our main contribution *predictionet* , we mainly extended state-of-the-art techniques that can handle genomic data to integrate prior knowledge: mRMR feature selection to identify the network's skeleton and minimum interaction information to orient the skeleton's edges. We developed the methods presented in this chapter to overcome some of the state-of-the-art techniques' shortcomings.

i. *Extending the feature selection step:* When applied to genomic data, standard feature selection strategies' performances often suffer when small changes occur in the data set. In the worst case, this problem could compromise the inferred model's generalizability a typical strength of feature selection strategies. To overcome this problem, we developed an ensemble mRMR approach which selects at each step a set of features with high mRMR scores instead of only the highest scoring variable. We show in this thesis that this allows to select features more robustly: small changes in the data set do not lead to big changes in the selected features using the parallel implementation of the R/Bioconductor package *mRMRe*.

   - [DJPCO$^+$13] Nicolas De Jay*, Simon Papillon-Cavanagh*, Catharina Olsen, Gianluca Bontempi & Benjamin Haibe-Kains. *mRMRe: an R package for parallelized mRMR ensemble feature selection.* Bioinformatics, 2013.

ii. *Extending the orientation step:* The state-of-the-art orientation algorithms based on interaction information only exploit triplets with negative interaction values. This strategy however leaves certain parts of the network undirected. This is especially important for those parts in the network which consists of chains of variables. In the second part of this chapter, we devise an algorithm based on interaction information that takes into account triplets with positive interaction information. This allows us to orient higher number of variables and thus infers better networks than current state-of-the-art orientation algorithms.

   - [OMB13] Catharina Olsen, Patrick E. Meyer & Gianluca Bontempi. *A scalable heuristic to orient arcs in undirected networks.* Submitted to BMC Bioinformatics, 2013.

- [OMB09b] Catharina Olsen, Patrick E. Meyer & Gianluca Bontempi. *Poster: Inferring causal relationships using information theoretic measures.* In Proceedings of the 5th Benelux Bioinformatics Conference (BBC09), 2009.

iii. *Studying the impact of entropy estimation on the quality of inferred networks:* So far we have assured the improvement of the feature selection and orientation steps of our algorithm. However, these algorithms rely on the estimation of information theoretic quantities such as entropy, mutual information and interaction information. These estimations are typically the bottleneck of the network inference algorithms when it comes to the computational cost. Therefore, too complex estimation techniques cannot be applied and we have to rely on those estimators presented in Section 2.6. In this experimental study, we show that the choice of entropy/mutual information estimator greatly influences the performance of the inference algorithms. Some guidelines on how to choose an estimator are derived.

- [OMB09a] Catharina Olsen, Patrick E. Meyer & Gianluca Bontempi. *On the Impact of Entropy Estimation on Transcriptional Regulatory Network Inference Based on Mutual Information.* EURASIP Journal on Bioinformatics and Systems Biology, 2009.
- [OMB08] Catharina Olsen, Patrick E. Meyer & Gianluca Bontempi. *On the impact of entropy estimator in transcriptional regulatory network inference.* Proceedings of WCSB08, 2008.

6.1 Ensemble mRMR

It has been shown in the literature that adding a bagging step to a network inference method can improve its performance by reducing the variability of the solution (Sections 3.4.1, 3.3.2.1.2 and 3.3.2.1.5). The other path taken by researchers is to combine different inference techniques in order to obtain a better overall network (Section 3.4.2). In this section we present a third alternative to generate multiple solution which can then be combined into one final solution.

6.1.1 Methodology

The rationale of mRMR feature selection is that at each step the best scoring feature is added to the set of selected features based on maximizing the relevance with the target variable and at the same time minimizing the the redundancy with the previously selected variables (Section 3.3.2.1.4). Often however, the mRMR score of the best and therefore selected feature is only marginally better than the the second best feature's score, etc. To build the ensemble mRMR models, all solutions will be considered at each step to build a tree of solutions. That is: in the first step, the variables will be added to the target variable X_T in the order of their relevance with X_T. The ensemble mRMR feature selection strategy continues by selecting the set of features having the highest mRMR scores for each branch of the tree and adding the features ordered by their mRMR score with the target and the previously selected features in the current branch. At each step, a verification step is included, such that the added feature does not generate a model equivalent to an earlier generated branch.

Algorithm 18: ensemble mRMR

Input: data, vertex set $\mathbf{V}$, target variable X_T, number of levels l, levels $\{k_1, \ldots, k_l\}$

Output: list $\prod_{i=1}^{l} k_i$ of models

18.1 Select the k_1 variables having the highest relevance with the target X_T;
18.2 **foreach** $i \in [2, l]$ **do**
18.3 **foreach** *of the* $\prod_{j=1}^{i-1} k_j$ *branches* **do**
18.4 Select the k_i variables with the highest mRMR values with the target given the previously selected variables in the corresponding branch;
18.5 **end**
18.6 **end**

Ideally, the best feature and all statistically not worse features should be selected. One option to obtain this set of best features is to employ a bootstrap approach at each step which would render the method computationally demanding. Therefore, we start by generating the tree with a user-defined maximum number of levels l and a user-defined number of features to be added at each level $k_i, i \in [1, l]$, Algorithm 18. The structure

of an ensemble mRMR tree is depicted in Figure 6.1 using parameters $l = 2$ and $k_1 = 3$ and $k_2 = 2$.

Each branch of an ensemble mRMR tree corresponds to one mRMR model. Let the root of the tree be X_T and each node be labeled as in Figure 6.1. The mRMR model corresponding to the first branch includes the nodes X_T, X_1 and X_4 and to the second branch X_T, X_1 and X_5, etc. Therefore, ensemble mRMR infers a set of models for each variable in the data set.

The standard mRMR feature strategy only returns the first branch of the tree by always selecting the variable with the highest mRMR score (the grey branch in Figure 6.1).

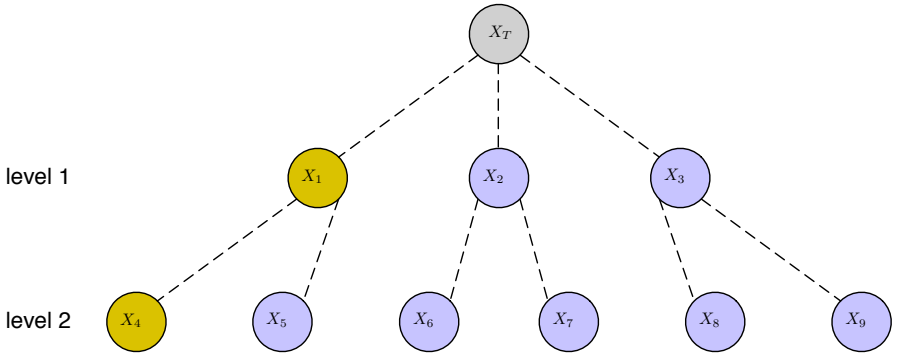


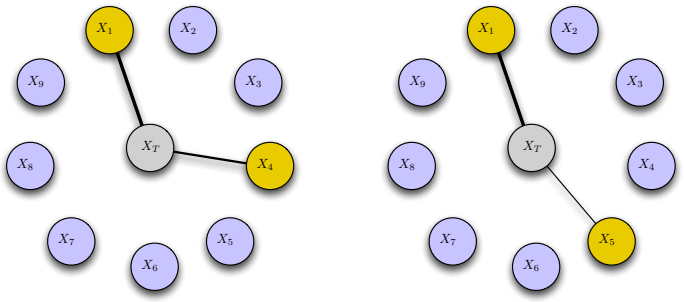
Figure 6.1: Example mRMR ensemble tree with $l = 2$, $k_1 = 3$ and $k_2 = 2$, the nodes that the standard mRMR procedure selects are colored yellow.

Different strategies can be employed to combine these models into a network. The most basic option is to integrate an edge into the network whenever it appears in a least one model. In this section we use the maximum obtained mRMR score as a convenient way of combining these into a final adjacency matrix.

The output of the mRMR algorithm is the set of adjacencies for each target variable, the corresponding mRMR values can be easily computed applying equation (3.18). A target variable's adjacency is added to the final adjacency matrix using the maximum mRMR score for this adjacency over all models (Figure 6.2).
We combine the different mRMR models by adding an inferred edge using the maximum score over all ensemble mRMR models.

6.1.2 Experimental study

In this section, we will experimentally highlight ensemble mRMR's advantages compared to the classic mRMR feature selection used in the MRNET network inference algorithm. We implemented mRMR in the R/Bioconductor package *mRMRe* [DJPCO$^+$13] and will



(a) first branch (b) second branch (c) remaining branches

Figure 6.2: The different branches in the ensemble mRMR tree correspond to different mRMR models. In each of these models grey is the target variable X_T, the selected variables are yellow and the magnitude of the mRMR score between two variables is represented via the corresponding edge's thickness.

use this package to infer the ensemble networks in this section. We will experimentally show that

- the inferred networks are more robust against changes in the data set, that is adding samples or removing them has a smaller influence on the final network's topology.
- the network's skeleton can be inferred without having to prune it before orientation. This pruning is an open problem in network inference which implies finding a good threshold to balance the number of true and false positives.

In order to show the latter, we will test whether the networks inferred with the ensemble network perform similarly to those inferred using MRNET in terms of maximal F-score.

6.1.2.1 Robustness feature selection

To experimentally show that indeed ensemble networks are more robust against small changes in the data set, we will carry out a set of experiments in which we infer the networks using only a certain percentage ($\{50, 75, 90, 95, 99\}\%$) of the available samples. We repeat this procedure 50 times. We generated two data sets with 100 variables and 100 samples each using two different data generators: Syntren (Appendix C.1) and GNW (Appendix C.4).
From each of the inferred networks, we only keep the top scoring edges (we show the results for (a) the top 20 scoring edges in the network and (b) the top 50 scoring edges in the network) and compare how many different edges were inferred over the 50 repetitions.

We compare the number of different edges inferred with MRNET to those inferred using ensemble mRMR with different number of levels and different number of adjacencies per

# levels	adjacencies per level
2	2,2 3,3 2,1
3	2,2,2 3,3,3 3,2,1

Table 6.1: Levels used in ensemble mRMR.

level, Table 6.1.

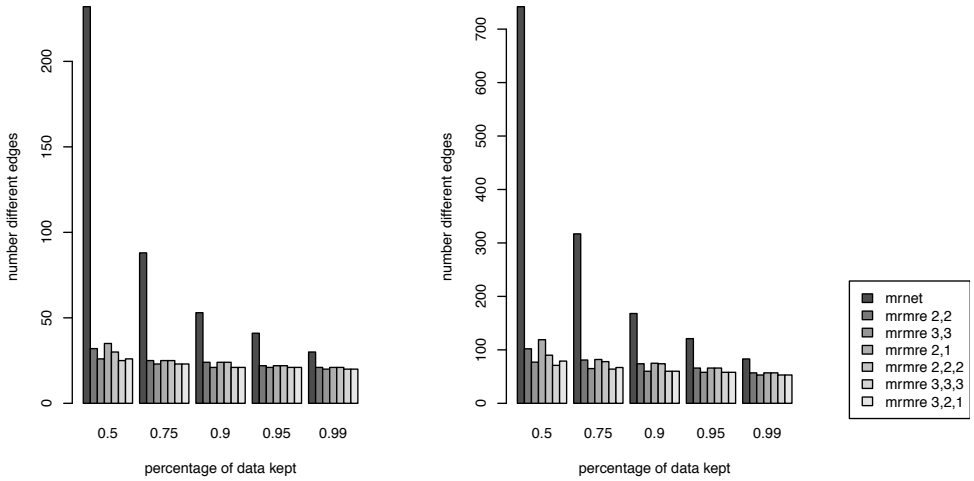
In Figure 6.3(a) we present the results of these computations using the GNW generated data. We are mostly interested in the number of different edges each method inferred when resampling a certain percentage (without replacement) of the data set. Most importantly, we can observe that classic mRMR selects a higher number of different edges for each of the percentages and for 20 and 50 top edges. Furthermore, this number increases with decreasing percentages. On the contrary, all ensemble mRMR setups result in lower number of different edges. The number increases as well with decreasing sample numbers, however this increase is much lower.

A similar picture is drawn when looking at the results using now Syntren generated data, Figure 6.3(b). However when comparing the results for the two generators, a more difficult question to answer is which ensemble mRMR setup performs best. The optimal choice of ensemble parameters seems to depend on the data and further studies will have to be carried out.

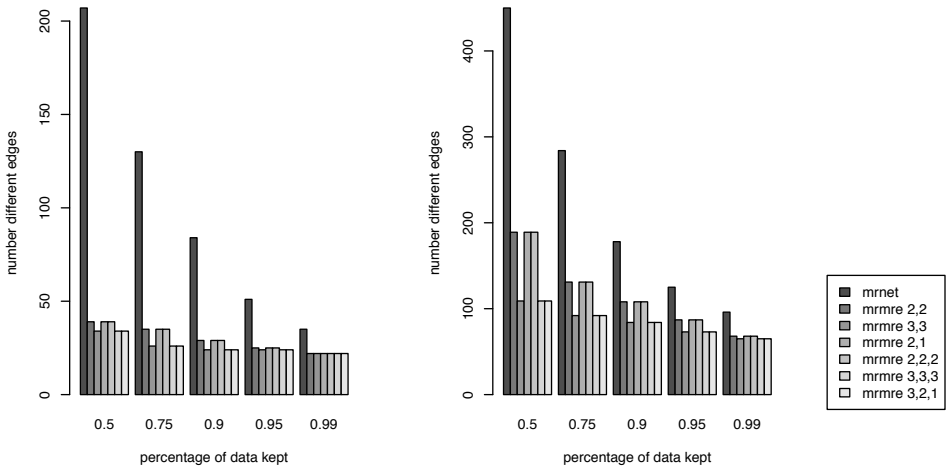
6.1.2.2 Quality of the inferred networks

Having inferred a more robust network with the ensemble mRMR approach does not imply that the inferred networks are also better in terms of F-scores. Therefore, for each of the 50 inferred networks in each setup we computed the maximum F-score and plotted the mean values in Figure 6.4.

The two main observations are: i) the ensemble mRMR networks perform better than the networks using standard MRNET and ii) the F-score increases with increasing percentage for all methods. If this preliminary result can be confirmed in a larger study, using ensemble mRMR not only performs well but also provides a convenient way to infer only the best edges (a much sparser network than the standard MRNET).



(a) Data generated using GNW



(b) Data generated using Syntren.

Figure 6.3: Robustness to changes in the data set using MRNET and *ensemble mRMR* with six different setups as specified in the legend and Table 6.1. Five different data modifications were carried out: randomly selecting 50% of the samples, 75%, 90%, 95% and 99%.

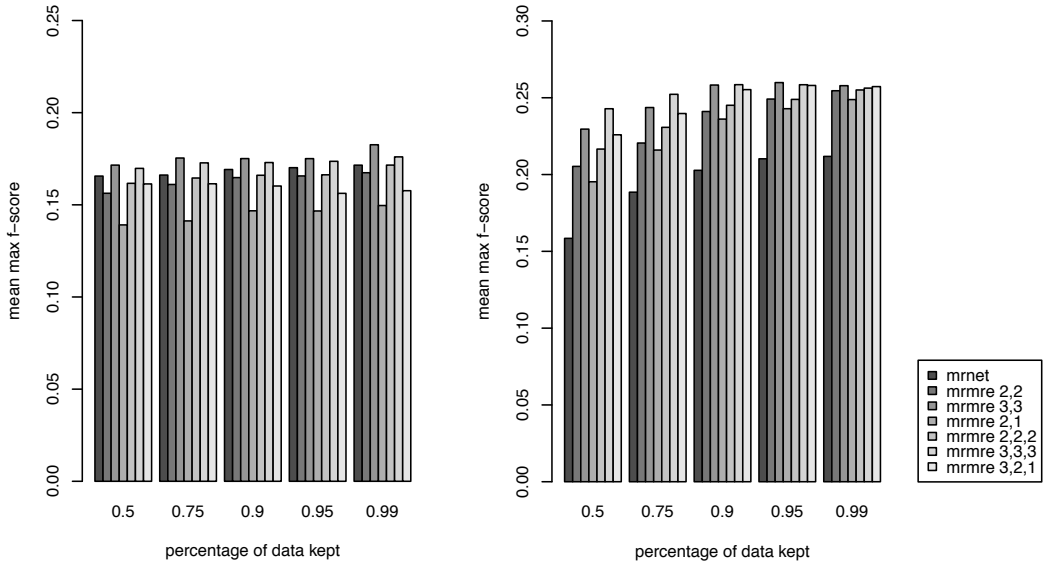


Figure 6.4: F-scores for data set using MRNET and *ensemble mRMR* with six different setups as specified in the legend and Table 6.1. Five different data modifications were carried out: randomly selecting 50% of the samples, 75%, 90%, 95% and 99%. Left: data generated using Syntren, right: data generated using GNW.

6.1.3 Conclusion

In a preliminary study we have shown that ensemble mRMR is a powerful alternative to standard mRMR feature selection. It is more robust to missing data, performs similarly or better in terms of F-scores and restricts the number of inferred interactions. The last point is important because state-of-the-art network inference techniques require a threshold to prune the obtained network.

In addition to this preliminary study on generated data, we used the mRMRe package in [DJPCO$^+$13] on real biological data. In this study, we confirm that ensemble mRMR selects a more stable gene list compared to classic mRMR. Furthermore, we show that i) the ensemble approach outperforms the classic mRMR in terms of prediction accuracy and ii) more biological relevant genes are selected by the ensemble method.

6.2 An orientation method based on interaction information

One way to orient edges of a skeleton is to compute interaction information for each triplet $X_i - X_j - X_k$ with $X_i \neq X_k$. Whenever the interaction information is negative, the triplet is then oriented into a v-structure $X_i \rightarrow X_j \leftarrow X_k$. Whenever, an edge belongs to more than one triplet, the current state-of-the-art algorithms orient the edge according to the triplet with minimal interaction information (Section 3.3.2.2). However, these algorithms ignore all triplets for which the interaction information is positive because this score can correspond to any of the three remaining configurations

$$X_i \leftarrow X_j \rightarrow X_k, \qquad X_i \rightarrow X_j \rightarrow X_k, \qquad X_i \leftarrow X_j \leftarrow X_k. \tag{6.1}$$

Therefore, there is not one unambiguous orientation to be derived from this information alone. However, whenever one of the edges is already oriented such that

$$X_i \rightarrow X_j - X_k \qquad \text{or} \qquad X_i - X_j \leftarrow X_k \tag{6.2}$$

the only possible orientation is a chain. Because otherwise it would be a v-structure and this would have implied a negative interaction information value.

The main problem we have to overcome with our algorithm is to orient edges for which there is ambiguous information, that is triplets supporting one direction but also triplets supporting the other direction. Because we use negative and positive interaction information it does not suffice anymore to consider that direction with the minimal score. In this section we present an algorithm named OMbIT (Orientation Method based on Information Theory) which exploits triplets with both negative and positive interaction information able to orient additional triplets compared to currents state-of-the-art strategies.

6.2.1 Methodology

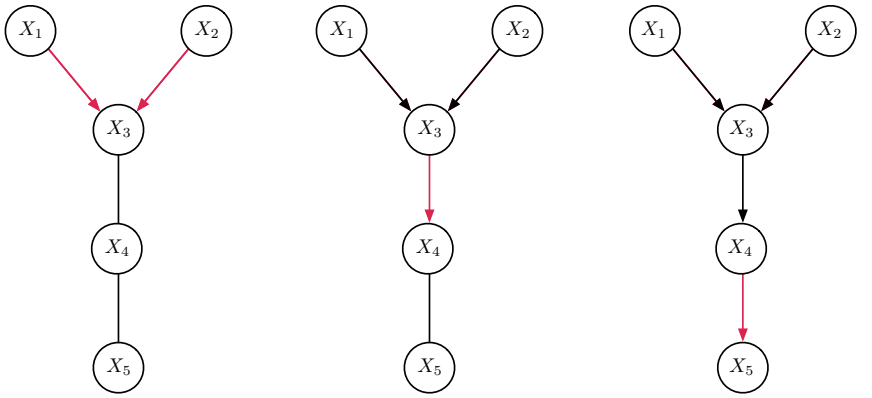
The rationale of our algorithm is based on the following two concepts

i. Instead of focusing on the highest scoring triplet, all adjacent variables should be taken into account.

ii. Making use of triplets with positive interaction information.

6.2.1.1 V-structures

In practice, we compute the average interaction information over all eligible triplets. In the first step these are the candidate v-structures. In Figure 6.5(a), there is one such



(a) step 1: v-structures (negative II)

(b) step 2: chains (positive II)

(c) step 3: continued chains (positive II)

Figure 6.5: OMbIT orients edges in three steps.

triplet $X_1 - X_3 - X_2$. To compute the score between X_i and X_j we take all triplets (X_i, X_j, X_k) such that

$$X_i - X_j - X_k \quad \text{and} \quad X_i \not\sim X_k \quad \text{and} \quad \mathcal{I}(X_i, X_j, X_k) < 0. \tag{6.3}$$

The set of variables X_k fulfilling equation (6.3) will be denoted by $\mathbf{X}_{\mathbf{K}^-}$.

$$s^{ij}_{avg} = \frac{1}{|\mathbf{X}_{\mathbf{K}^-}|} \sum_{X_k \in \mathbf{X}_{\mathbf{K}^-}} |\mathcal{I}(X_i, X_j, X_k)|. \tag{6.4}$$

We will use this equation as a one of the methods for our experimental section, the *average* method and as first step of the OMbIT algorithm.

6.2.1.2 Chains

The reason for considering the absolute value is that not only negative but also positive interaction information can be used within the same framework. Indeed, the higher the absolute value the stronger the evidence. Positive interaction information for a triplet (X_k, X_i, X_j) can be taken into account whenever X_i is a collider in a v-structure

$$\exists X_{k'} : X_k \to X_i \leftarrow X_{k'},\ k' \neq j. \tag{6.5}$$

The set of eligible triplets are in step 2 potential chains. In Figure 6.5(b), these are $X_1 \to X_3 - X_4$ and $X_2 \to X_3 - X_4$.

$$X_k \to X_i - X_j \quad \text{and} \quad X_k \not\sim X_j \quad \text{and} \quad \mathcal{I}(X_i, X_j, X_k) > 0 \tag{6.6}$$

The set of variables X_k fulfilling both equation (6.5) and equation (6.6), will be denoted by $\mathbf{X}_{\mathbf{K}^+}$, see Figure 6.5(b). Subsequently, the scoring function can be written as:

$$s^{ij}_{OMbIT} = \frac{1}{|\mathbf{X}_{\mathbf{K}^- \cup \mathbf{K}^+}|} \sum_{X_k \in \mathbf{X}_{\mathbf{K}^- \cup \mathbf{K}^+}} |\mathcal{I}(X_i, X_j, X_k)|. \tag{6.7}$$

6.2.1.3 Continued chains

During the final step of our algorithm, we try to counter the problem that variables with less than two parents will not be identified in the first two steps, $X_3 \rightarrow X_4 - X_5$ in Figure 6.5(c). Whenever the current target variable has only one parent oriented as part of the previous steps in the algorithm but also some children, these children cannot have been oriented. We can exploit this fact by orienting all edges for which there is evidence for a chain (positive interaction information) and one previously found parent. This step will be repeated until we do not identify new children.

Taking Figure 6.5 as an example: assume that $X_1 \rightarrow X_3 \leftarrow X_2$ and $X_3 - X_4 - X_5$ with $\mathcal{I}(X_1, X_3, X_4) > 0$, $\mathcal{I}(X_2, X_3, X_4) > 0$ and $\mathcal{I}(X_3, X_4, X_5) > 0$. Then, the algorithm (Algorithm 19) can sequentially orient $X_3 \rightarrow X_4$ and $X_4 \rightarrow X_5$ because these connections exhibit no negative interaction information. The last two steps are neglected in [LHW08, WLW$^+$09, BM10].

The algorithmic complexity of OMbIT is given by $\mathcal{O}(t^2 + n^2)$ with t being the number of triplets in the undirected network and n representing the number of variables. In other words: the algorithm is adapted to sparse networks, typically observed in biology [TB07].

The estimation of independence measures such as mutual information and interaction information is in general the bottleneck of inference algorithms. Therefore, by limiting this part of the computation to bivariate terms using equation (2.40) to estimate the interaction information between triplets of variables, we improve the overall speed of our orientation algorithm.

This interaction information estimator together with the three different heuristics were implemented as extension to the R/Bioconductor package *MINET* .

6.2.2 Experiments

In this section, we experimentally compare OMbIT to two different basic heuristics based on interaction information, the first one uses the minimum interaction information as a score (*min*) and the second the average absolute value of the interaction information over all eligible triplets *avg*, equation (6.4). Furthermore, a score-based algorithm Hill-Climbing (*hc*, Section 3.2.2) using the Bayesian information criterion (Section 3.2.2.1) and lastly a constraint-based algorithm (*iamb*, Section 3.3.1).

Algorithm 19: OMbIT

Input: data, vertex set $\mathbf{V}$, undirected network

Output: Oriented network

Initialize $n \times n$ dimensional matrices $\mathbf{S}$ and **counts** with zeros;
Compute interaction information for each triplet of variables with $X_i - X_j - X_k$ and $X_i \not- X_k$;
for all triplets X_i, X_j, X_k with $\mathcal{I}(X_i, X_j, X_k) < 0$ **do**
    $\mathbf{S}[i, j] = \mathbf{S}[i, j] + |\mathcal{I}(X_i, X_j, X_k)|$;
    $\mathbf{S}[k, j] = \mathbf{S}[k, j] + |\mathcal{I}(X_i, X_j, X_k)|$;
    counts$[i, j]$ = **counts**$[i, j] + 1$;
    counts$[k, j]$ = **counts**$[k, j] + 1$;
end for
c =number of edges with no evidence
while entering the loop OR the value of c changed in last iteration **do**
    for all triplets (X_i, X_j, X_k) with previous evidence for $X_i \to X_j$ and $X_j - X_k$ and $X_i \not- X_k$ and $\mathcal{I}(X_i, X_j, X_k) > 0$ **do**
        $\mathbf{S}[j, k] = \mathbf{S}[j, k] + \mathcal{I}(X_i, X_j, X_k)$;
        counts$[j, k]$ = **counts**$[j, k] + 1$;
    end for
    c =number of edges with no evidence
end while
compute average $\mathbf{S}_{avg}$ using $\mathbf{S}$ and **counts**
orient all edges $X_i \to X_j$ for which there is evidence and $\mathbf{S}_{avg}[i, j] > \mathbf{S}_{avg}[j, i]$, if both values are equal, infer a bidirectional edge;

In the first part, we will use synthetic data sets obtained using two different data generators: Syntren and GNW (Appendix C.1 and C.4) . The experiments' workflow is presented in Figure 6.6: starting with the removal of the orientation from the true network, thus obtaining what we will call hereafter **true skeleton**, we proceed by orienting the edges in this undirected network using different algorithms.

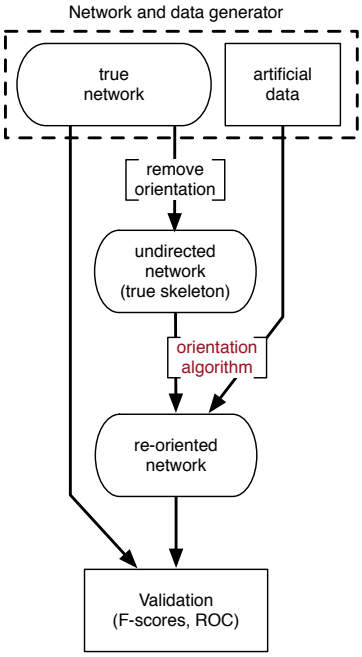


Figure 6.6: Setup experiments: focusing on orientation by taking true skeleton as input for all orientation algorithms.

The second part of this section will be devoted to four real biological data sets which were published as part of [mCRE$^+$10]. We will use the known interactions as defined in [mCRE$^+$10] as true skeleton and infer from this which are the transcription factors. Due to the sparsity of prior knowledge which is due to knowing only a subset of all true interaction in any real organism, the performance of orientation algorithms will be lower than what is obtained in the first part. In both parts, we will show that OMbIT performs similarly well or better than its competitors on data sets with a small number of variables and better on data set with a high number of variables.

6.2.2.1 Validation

For our validation of the oriented networks, the true positives (TP) comprise the number of edges oriented in the correct direction. The false positives (FP) comprise the number of edges oriented in the wrong direction. Edges present in the skeleton but not oriented are false negatives (FN). It is possible that the true networks generated by Syntren or

GNW contain bidirectional edges. In this case we count a directed edge (either of the two directions or bidirectional) as one true positive and a missing edge as false negative. In the case of a bidirectional inferred edge, if the true edge is directed, it will count as one true positive and one false positive, see Table 6.2 for the corresponding confusion matrix and use the F-score as scoring metric (Appendix B.3.1).

inferred direction\true direction	$\longrightarrow$	$\longleftarrow$	$\longleftrightarrow$	missing
$\longrightarrow$	TP	FP	TP	FP
$\longleftarrow$	FP	TP	TP	FP
$\longleftrightarrow$	TP,FP	TP,FP	TP	FP
missing	FN	FN	FN	TN

Table 6.2: Confusion matrix used for F-score computation

6.2.2.2 Arc orientation on generated data sets

Data sets
We generated several data sets containing up to 1000 variables with Syntren (Section C.1) and GNW (Section C.4). Syntren generates artificial microarray data sets from known biological networks such as E.coli and S.cerevisiae. We chose E.coli because the maximum number of possible variables in a network is higher than with S.cerevisiae: 1330 compared to 690. We simulated a range of networks and data sets using the E.coli source network provided together with the generator and number of variables/genes from 100 up to 1000 (details in Table 6.3) while keeping the number of samples constant to 100. For each specified variable/sample combination, we generated 10 different networks.

GNW extracts subnetworks using as basis transcriptional regulatory networks of E.coli or S.cerevisiae avoiding auto-regulatory interactions. We also used as source network E.coli and generated sets of ten data sets/networks containing the same number of variables as for Syntren, here E.coli offers a maximum of 1565 variables. With this generator, the number of samples is fixed to the number of variables. We chose to work with multifactorial perturbation data which are static steady-state measurements obtained by slightly perturbing all genes simultaneously, as used in the analysis in [HTIWG10].

Results
In this section we present the results obtained by orienting the true skeleton with different orientation methods: OMbIT, minimum interaction information (*min*), average interaction information (*avg*), a score-based (*hc*) and constraint-based (*iamb*) algorithm. The computations were carried out using the R-packages *ombit* for the methods based on interaction information and *bnlearn* [Scu10] for both, the constraint-based and the score-based methods. In Figure 6.7, the results for the two generators, the three different number of variables and the five different orientation methods are presented. Each

generator	# variables	# samples	# networks
Syntren	100 500 1000	100	10
GNW	100 500 1000	100 500 1000	10

Table 6.3: Characteristics of generated data sets

box represents the F-scores over the ten generated networks/data sets. The results for

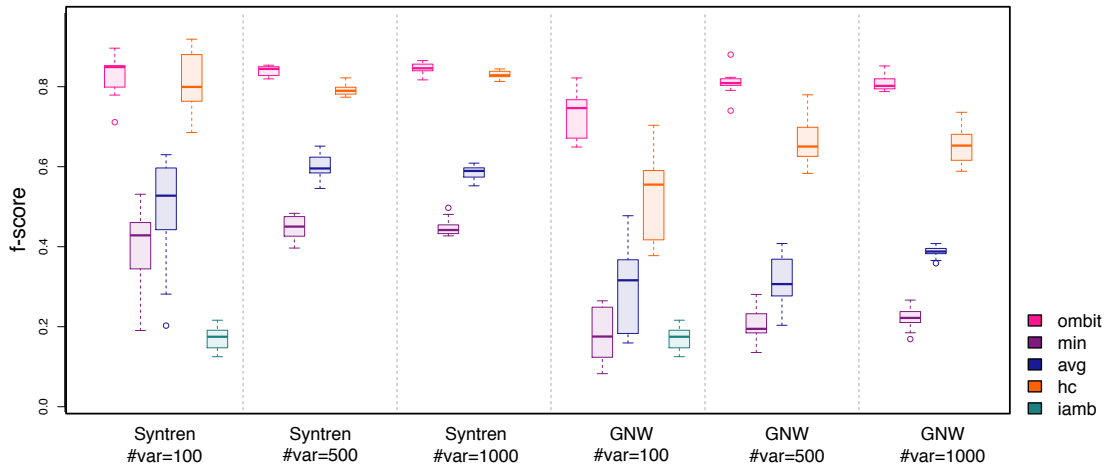


Figure 6.7: Each box represents the F-scores over ten repetitions of generated data sets orienting the true skeleton, generators and number of variables are specified on x-axis, orientation method can be determined by the box's color.

the different orientation methods are similar for both data generators. For both generators, OMbIT performs best whenever the number of variables is higher than 100. Hill-Climbing performs well on Syntren-generated data sets. However, it seems to be more challenging to infer orientations for GNW-generated data sets. While OMbIT still obtains F-score with median 0.8, at least for the data sets with 500 and 1000 variables, the remaining methods perform worse than on Syntren-generated data sets. With growing number of variables, the F-scores are increasing for all methods and both generators. However, only OMbIT performs well for both data generators.

6.2.2.3 Arc orientation on biological data sets

The main difference when inferring networks, undirected or directed, from biological data is that the true network is unknown or in the best case partially known. This makes it naturally much more difficult to validate any obtained result. In our analysis, we orient the true skeleton applying the same methods as before: *ombit, min, avg, hc* and *iamb*. In a second step we replace the true skeleton with an undirected network obtained using the MRNET inference algorithm to show that also in the more realistic setup that the undirected is unknown, arc orientations can be inferred using OMbIT.

Data sets
The biological data sets used in this experimental study have been published in [mCRE$^+$10]. The redfly network of known interactions consists of 138 variables and 233 directed edges. For the inference, we use four different data sets whose characteristics are presented in Table 6.4. The number of variables correspond to the number of variables the specified data set has in common with the redfly network.

name data set	# variables	# samples	# known interactions
MA	128	28	188
flyatlas	91	26	55
rnaseq	132	11	201
rnaseq2	135	30	209

Table 6.4: Names and characteristics of data sets related to the fly network. The number of variables is restricted by the number of variables present in the redfly set of known orientations.

Results – using the true skeleton
Real-world data - in this case the true skeleton - does not provide the complete image of the network but rather a subset of true connections. The F-scores for the different orientation methods are presented in Table 6.5. We can observe that OMbIT performs

	ombit	min	avg	hc	iamb
MA	**0.6904**	0.6217	0.6165	0.4918	0.1321
flyatlas	**0.5**	0.4789	0.4789	0.3385	0.25
rnaseq	**0.6414**	0.5673	0.562	0.4669	NA
rnaseq2	**0.6731**	0.6333	0.6288	0.5512	0.0889

Table 6.5: F-scores for different orientation methods for the four datasets orienting true skeleton's edges. In bold font the highest scoring orientation method.

better than the remaining methods for all data sets The lowest performance is reached

when orienting edges for the *flyatlas* data set. The percentage of undirected edges is 38 percent (31 not oriented out of 55 known interactions). This could be explained by the fact that this network contains a lot of connected pairs of variables that are not connected with any other variables, therefore there are not enough edges in the true skeleton such that an orientation could be successful. For the other three data sets, the percentages of undirected edges is lower but also here, there is room for improvement with approximately eight to ten percent of undirected edges.

	ombit	min	avg	hc	iamb
MA	16	30	30	88	170
flyatlas	21	22	22	42	45
rnaseq	16	26	26	85	NA
rnaseq2	17	32	32	80	195

Table 6.6: number of non-oriented edges

Results – inferring the undirected network from data
The next logical step is to infer the skeleton from data instead of using the known interactions to build the skeleton. The evaluation of the orientation procedure faces two difficulties:

- The inference will not infer all known interactions, therefore generating false negatives not due to the orientation itself. Thus, we evaluate only the orientation of edges known to be in the true skeleton.
- The inference method always required a threshold to select only the most relevant interactions. The trade-off for the subsequent orientation is to keep enough of the true interactions while at the same time being sparse enough for the orientation to find possible v-structures. In what follows, we chose to keep 50 % of the edges inferred using MRNET.

As the main interest of this study's part is to assess the quality of the orientation, we present in Table 6.7 the classification for the oriented edges that are present in the set of known interactions. The true positives are those edges which are correctly oriented, the false positives those that are oriented in the wrong directions and the false negatives those edges which are present in the true interactions and also in the undirected network but for which there no orientation was obtained.

From the results in Table 6.7 it can be observed that the methods based on interaction information perform better than the score- and constraint-based methods. However, the difference between the three methods using interaction information is small on these four data sets. This can be explained by the low number of chains in these networks. This means that there are mainly v-structures to be oriented. In this case, there is no significant difference between the three methods based on interaction information.

	ombit			min			avg		
	#tp	#fp	#fn	#tp	#fp	#fn	#tp	#fp	#fn
MA	28	19	0	26	19	2	26	19	2
flyatlas	6	7	0	5	8	0	5	8	0
rnaseq	37	25	0	37	24	1	36	25	1
rnaseq2	38	25	0	37	25	1	37	25	1
	iamb			hc			# not in		
	#tp	#fp	#fn	#tp	#fp	#fn	undirected		
MA	7	6	34	5	5	42	137		
flyatlas	4	2	7	3	3	10	41		
rnaseq	9	16	36	NA	NA	NA	135		
rnaseq2	12	7	44	1	1	62	142		

Table 6.7: keeping 50% of the edges inferred using MRNET

6.2.3 Conclusion

In this section, we present a novel arc orientation algorithm which efficiently orients edges in an undirected network with possibly thousands of variables. In comparison with other heuristics based only on negative interaction information, we considerably improve the quality of the oriented networks both on synthetic data and on biological data. The reasons for the improvement being two-fold: including the entire neighborhood of the target variable is better than only taking that triplet with the highest negative interaction information, that is *avg* being always at least as good as *min*. Secondly, taking advantage of additional information from triplets with positive interaction allows to orient chains of variables which is not possible with the state-of-the-art usage of negative interaction information alone. The benefit of this has been shown for both synthetic data and biological data as long as there are enough interactions in the true skeleton.

6.3 Experimental study of estimators for information theory

The feature selection techniques we presented to infer networks from genomic networks rely on the computation of pairwise mutual information values. As the distribution of the variables is unknown these mutual information values need to be estimated from data (Section 2.6). There exist different strategies such as assuming that the variables follow a multivariate normal distribution or using plug-in estimators in which the probability distribution is estimated via frequencies using different binning strategies. More complicated estimators are usually not employed in network inference algorithms as this step is the bottleneck for the algorithms speed.

We start our experimental study by evaluating the estimators' influence on the inferred networks' quality using artificial data sets generated with Syntren (Section C.1). We then infer networks for publicly available biological data sets and validate the results by using firstly experimental ChIP-chip experiments and secondly known interactions.

6.3.1 Synthetic datasets

6.3.1.1 Data and experimental setup

We start our experimental section by evaluating networks inferred from generated data sets. We used Syntren (Section C.1) to generated 12 data sets with varying numbers of variables and samples, Table 6.8 for a systematic description. We varied the number of genes between 50 and 300 and the number of samples from 100 to 300.

In order to test the influence of additive Gaussian noise (having 50% variance of the observed values) and missing values, we first introduced them separately and subsequently together to the data sets.

In order to study the impact of missing values, we removed expression values from the generated data sets. The number of missing values is distributed according to the $\beta(a, b)$ distribution with parameters $a = 2$ and $b = 5$. The maximal allowed number of missing values is a third of the entire data set. We chose this distribution instead of the uniform distribution, because the latter could have favored the empirical estimator.

We used five different estimators: the empirical, the Miller-Madow, the shrink, Pearson and Spearman correlation (Section 2.6). We applied two different discretization strategies, equal frequency and equal width (Section 2.6.3.1), to the data sets for the first three estimators which require discrete data.

We compared the performance of three different network inference algorithms: CLR, ARACNE and MRNET (Section 3.3.2.1).

dataset	1	2	3	4	5	6	7	8	9	10	11	12
# variables	300				200				100			
# samples	300	200	100	50	300	200	100	50	300	200	100	50

Table 6.8: Generated datasets. Number of genes n, number of samples m, datasets ecoli_n_m

We repeated each experiments ten times and computed the mean F-scores and evaluated the significance of the results using a paired t-test.

All computations were carried out using the R-package MINET[1] [MLB07b].

6.3.1.2 Results

The results of the benchmark using synthetic data are collected in Table 6.9. We present the F-scores (Section B.3.1) for each combination of inference method, mutual information estimator and nature of the data set (noisy versus not noisy, complete versus missing data). Note that the maximal F-score is highlighted, together with the F-scores which are not significantly different from the best.

We analyze the results according to four different aspects: the impact of the estimator, the impact of the discretization, the impact of the inference algorithm and the influence of sample and network size.

The section concludes with the identification of the best combination of inference algorithm and estimator.

Impact of the estimator:
In case of complete data sets with no noise, the empirical and the Miller-Madow estimator with equal frequency binning lead to the highest F-scores for the MRNET and the ARACNE inference methods. The Spearman correlation is not significantly different from the best, in case of ARACNE, and close to the best in case of MRNET. The CLR method is less sensitive to the estimator and the best result is obtained with the Pearson correlation.

In case of noisy data or missing value (NA) configurations, the Pearson correlation and the Spearman correlation lead to the highest F-score for all inference methods. A slightly better accuracy of the Pearson correlation can be observed in presence of missing values. The Spearman correlation outperforms the other estimators in MRNET and ARACNE when complete yet noisy data sets are considered. In CLR, Pearson and Spearman lead

[1] http://cran.r-project.org/web/packages/minet

Method		MRnet			
Estimator		no noise no NA	noise no NA	no noise NA	noise NA
Pearson		0.2006	0.1691	**0.1790**	**0.1611**
Spearman		0.3230	**0.1771**	0.1464	0.1333
Emp	eqf	**0.3420**	0.1551	0.1136	0.0868
Emp	eqw	0.2028	0.1650	0.1036	0.0822
MM	eqf	**0.3396**	0.1524	0.1140	0.0924
MM	eqw	0.1909	0.1592	0.1068	0.0883
Shr	eqf	0.3306	0.1506	0.1150	0.0788
Shr	eqw	0.1935	0.1574	0.1090	0.0839
		Aracne			
Pearson		0.1117	0.1082	0.1054	**0.1069**
Spearman		**0.1767**	**0.1156**	**0.1167**	**0.1074**
Emp	eqf	**0.1781**	0.1042	0.0993	0.0765
Emp	eqw	0.1287	0.1082	0.0892	0.0727
MM	eqf	**0.1786**	0.1032	0.0985	0.0783
MM	eqw	0.1217	0.1049	0.0931	0.0767
Shr	eqf	0.1736	0.1000	0.1009	0.0697
Shr	eqw	0.1152	0.1045	0.0898	0.0717
		CLR			
Pearson		**0.2242**	**0.1941**	**0.2231**	**0.1911**
Spearman		**0.2197**	**0.1915**	0.1806	0.1582
Emp	eqf	0.2123	0.1729	0.1847	0.1397
Emp	eqw	0.2098	0.1724	0.1799	0.1327
MM	eqf	0.2128	0.1729	0.1860	0.1427
MM	eqw	0.2083	0.1723	0.1845	0.1384
Shr	eqf	0.2096	0.1670	0.1864	0.1311
Shr	eqw	0.2030	0.1659	0.1822	0.1333

Table 6.9: MINET results: *noise* stands for Gaussian additive noise, *NA* for missing values, *eqf* for equalfrequency and *eqw* for equalwidth. In bold: maximum F-scores and significantly not different values.

Estimator		Method		
		MRnet	Aracne	CLR
Pearson		0.1775	0.1081	**0.2081**
Spearman		**0.1950**	**0.1285**	0.1863
Emp	eqf	0.1744	0.1145	0.1774
Emp	eqw	0.1384	0.0997	0.1737
MM	eqf	0.1746	0.1147	0.1786
MM	eqw	0.1363	0.0881	0.1759
Shr	eqf	0.1688	0.1111	0.1735
Shr	eqw	0.1360	0.0953	0.1711

Table 6.10: For each method and estimator the mean over the four different setups: no NA, no noise; no NA, noise; NA, no noise; NA noise. In bold face the best mean F-score.

the ranking without being significantly different.

Impact of the discretization:
In case of complete data sets with no noise, the equal frequency binning approach outperforms the equal width binning approach for all discrete estimators. The gap between the two discretization methods is clearly evident in MRNET and less striking in ARACNE and CLR. In case of noisy or missing data configurations, differences are attenuated.

Impact of the inference algorithm:
In case of complete data sets with no noise, the MRNET inference technique outperforms the other algorithms. The situation changes in presence of noisy or missing values. Here CLR appears to be the most robust by returning the highest F-scores for all combinations of noise and missing values.

Conclusion:
A concise summary of the previously discussed results is displayed in Table 6.10 which averages the accuracy over the different data configurations. It emerges that the most promising combinations are represented by the MRNET algorithm with the Spearman estimator and the CLR algorithm with the Pearson correlation. The former seems to be less biased because of its good performance in front of non-noisy data sets, while the latter seems to be more robust since less variant in front of additive noise.

6.3.2 Biological data sets

The second part of the experimental session aims to assess the performance of the two selected techniques applied to a real biological task.

We proceeded by i) setting up a data set which combines several public domain microarray data sets about the yeast transcriptome activity, ii) carrying out the inference with

Dataset	Number of samples	Origin
1	7	[DIB97]
2	7	[CDE+98]
3	77	[SSZ+98]
4	4	[FBBR99]
5	173	[GSK+00]
6	52	[GHM+01]
7	63	[HMJ+00]
8	300	[HMJ+00]
9	8	[ODB00]
10	20	[GUV+07]

Table 6.11: Number of samples and bibliographic references of the yeast microarray data used for network inference

the two selected techniques, and iii) assessing the quality of the inferred network with respect to two independent sources of information: the list of interactions measured by means of an alternative genomic technology and a list of biologically known gene interactions.

The data set was built by first normalizing and then joining ten public domain yeast microarray data sets, whose number of samples and origin is detailed in Table 6.11. The resulting data set contains the expression of 6352 yeast genes in 711 experimental conditions.

6.3.2.1 Assessment by ChIP-chip technology

The first validation of the network inference outcome is obtained by comparing the inferred interactions with the outcome of a set of ChIP-chip experiments. The ChIP-chip technology, detailed in [BL04], measures the interactions between proteins and DNA by identifying the binding sites of DNA-binding proteins. The procedure can be summarized as follows. First, the protein of interest is cross-linked with the DNA site it binds to, then double-stranded parts of DNA fragments are extracted. The ones which were cross-linked to the protein of interest are filtered out from this set, reverse cross-linked and their DNA are purified. In the final step, the fragments are analyzed using a DNA microarray in order to identify gene-gene connections. For our purposes it is interesting to remark that the ChIp-chip technology returns for each pairs of genes a probability of interaction. In particular we use, for the validation of our inference procedures, the ChIp-chip measures of the yeast transcriptome provided in [HGL+04].

6.3.2.2 Assessment by biological knowledge

The second validation of the network inference outcome relies on existing biological knowledge and in particular on the list of putative interactions in Saccaromyces Cere-

	AUC
Harbison	0.6632
CLR Pearson	0.5534
MRNET Spearman	0.5433
MRNET Miller-Madow	0.5254
CLR Miller-Madow	0.5207

Table 6.12: AUC: Harbison, CLR with Gaussian, MRNET with Spearman, CLR with Miller-Madow, MRNET with Miller-Madow

visiae published in [SWCvH04]. This list contains 1222 interactions involving 725 genes and in the following we will refer to this as the Simonis list.

6.3.2.3 Results

In order to being able to compare inferred interactions to those in the Simonis list of known interactions, we limited our inference procedure to the 725 genes contained in the list. The quantitative assessment of the final results is displayed by means of a receiver operating characteristics (ROC) and the associated area under the curve (AUC) (Section B.3.1).

Note that this assessment considers as true only the interactions contained in the Simonis list.

Figure 6.8 displays the ROC curves and Table 6.12 reports the associated AUC for the following techniques: the ChIP-chip technique (Harbison), the MRNET-Spearman correlation combination, the CLR-Pearson combination, the CLR-Miller-Madow combination, the MRNET-Miller-Madow combination and the random guess.

A first consideration to be made about these results is that network inference methods are able to infer better interactions than a random guess in real biological settings. Also the two combinations which appeared to be the best in synthetic data sets confirmed their supremacy over the Miller-Madow based techniques also in real data.

However, the weak performance of the networks inferred from microarray data requires some specific considerations.

i. With respect to the ChIP-chip technology it is worth mentioning that the information coming from microarray data sets is known to be less informative than the one coming from the ChIP-chip technology. Microarray data sets remain nowadays however more easily accessible to the experimental community and techniques able to extract complex information from them are still essential for system biology purposes.

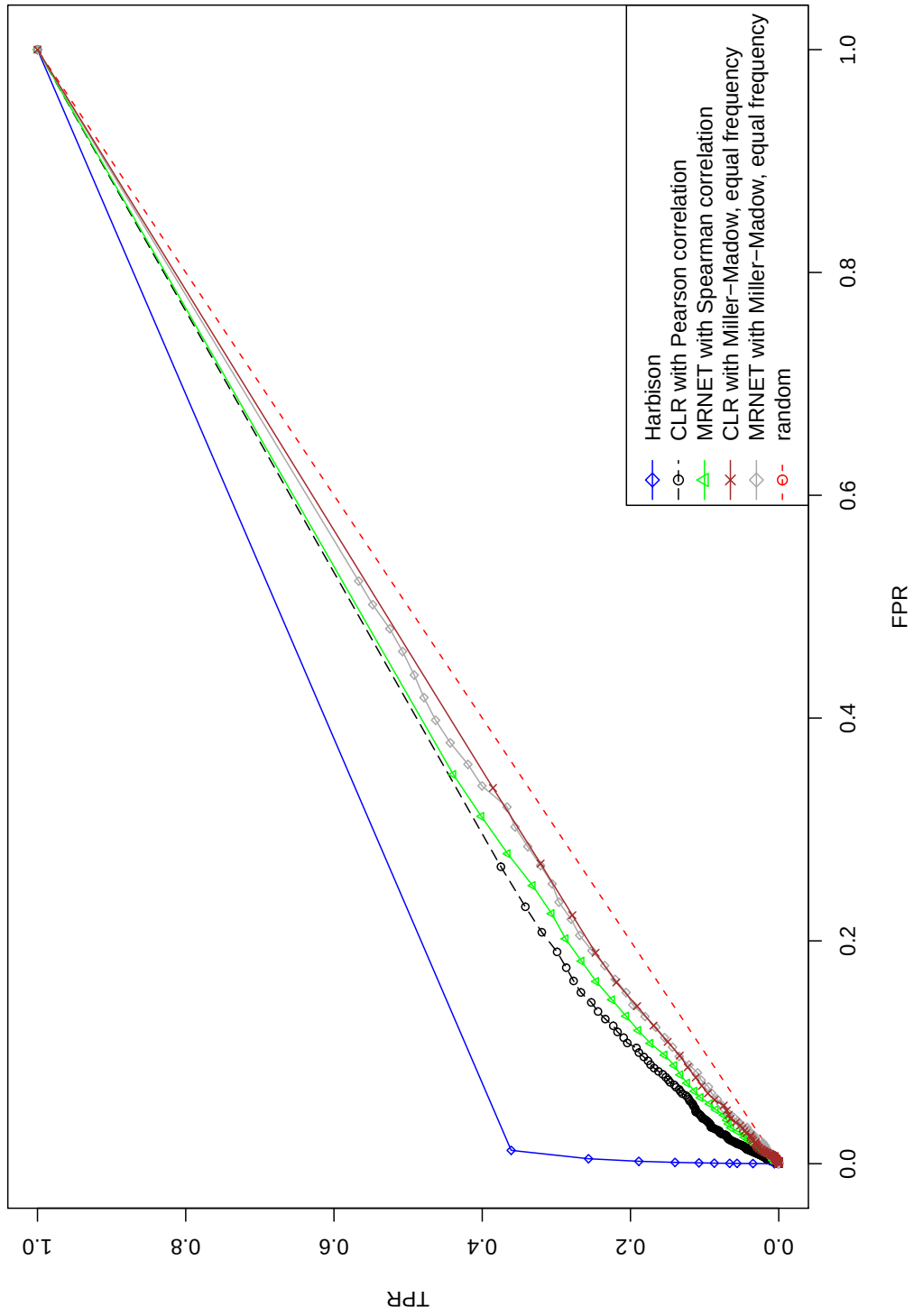


Figure 6.8: ROC curves: Harbison network, CLR combined with Pearson correlation, MRNET with Spearman correlation, CLR combined with the Miller-Madow estimator using the equal frequency discretization method, MRNET with Miller-Madow using equal frequency discretization and random decision

ii. Both the microarray data set we set up for our comparison and the list of known interactions we used for assessment are strongly heterogeneous and concern different functionalities in yeast. We are confident that more specific analysis on specific functionalities could increase the final accuracy.

iii. Like in any biological validation of bioinformatics methods, the final assessment is done with respect to a list of putative interactions. It is likely that some of our false positives could be potentially true interactions or at least deserve additional investigation.

6.3.3 Conclusion

We investigated in this experimental study the influence of entropy estimation of the quality of the inferred networks. The main conclusion from our study is that the estimators based on correlation are most robust against noise and missing values. Furthermore, the performance of each network inference algorithm is dependent on the used estimators,thus requiring to carefully select the best estimator for each employed network inference algorithm. This conclusion was seconded in [dMSES11] in which the authors presented a novel inference algorithm, C3NET (Section 3.3.2.1.5), and investigated the estimators' influence on its performance using generated data sets.

Chapter 7

Conclusion

7.1 Summary of main results

We started this thesis by presenting a state-of-the-art chapter on network inference and prior integration. This helped us to identify strengths and weaknesses of different methods and led to the development of *predictionet*, a method that combines genomic data and prior knowledge to infer directed networks. This method was implemented in the R/Bioconductor package of the same name. Furthermore, we developed a purely data-driven strategy to validate directed networks based on experimental knock-down data.

Based on the ideas used in *predictionet*, we developed a number of extensions. Our inference is based on mRMR feature selection and on arc orientation using interaction information. A weakness of the former is its sensitivity to changes in the data sets thus making these models less generalizable. We tackle this problem by developing an ensemble mRMR feature selection strategy. The second extension we propose is to fully exploit the informational content of the interaction information in order to orient more edges than the state-of-the-art technique. A bottleneck for inference methods based on the computation of information theoretic quantities such as entropy, mutual information and interaction information is the estimation of these quantities. Therefore, the employed estimators are not the most complex but rely instead on assumptions such as Gaussianity of the data. In an experimental study we investigate the influence of these estimators on the performance of inference algorithms and determine which combinations of estimators and inference algorithms perform best facing high amounts of noise and missing values.

7.1.1 Prior integration and network validation

Prior information about which genes interact with each other are stored in biological databases and research articles. In this thesis, we develop a comprehensive framework with the appropriate tools to i) infer directed networks from a combination of genomic data and prior knowledge and ii) validate these networks using experimental knock-down

data. The inference part of framework consists of:

- *Predictive Networks*: a web application to retrieve prior knowledge from biological databases and PubMed abstract and articles.
- *predictionet*: an inference technique that combines genomic data and prior knowledge. In the feature ranking step, the ranking based on mRMR is modified using the prior knowledge. The arc orientation then integrates prior knowledge to influence the orientation score computed based on interaction information. We implement this method in the R/Bioconductor package *predictionet*.

We tackle the problem of network validation using our experimental knock-down data. This data set consists of eight different single gene knock-downs in multiple biological replicates plus control samples. We use the samples related to the knock-down of a specific gene to determine the affected genes. The network is inferred using the remaining samples. We then compute a performance score which determines a ratio of affected genes that are present in the knocked down gene's childhood within the inferred network. This evaluation then allows us to compare different methods quantitatively and purely data driven.

In our experimental study based on perturbation experiments on two colon cancer cell lines, we were able to show

- that the retrieved priors are relevant and
- that prior knowledge is beneficial to the inferred network's quality.

7.1.2 Ensemble mRMR

Based on the principle that ensemble methods reduce the variability of the result and thus perform better than the methods separately, we develop an ensemble mRMR feature selection strategy. Contrary to the usual approach of either bootstrapping or using different inference techniques, we use the same feature selection strategy on the entire data set but select more than one feature in each iteration. The rationale is that given the few samples, the best scoring feature might actually not be better but just as good as the next best feature(s). We use this idea to build trees of models that can then be combined in different ways to construct the final model. In a preliminary experimental study we show the usefulness of ensemble mRMR feature selection with respect to the robustness of the models with respect to changes in the data set.

7.1.3 Causal inference

We show that arc orientation based on minimum interaction information can be improved by

- taking the node's entire neighborhood into account instead of the highest scoring triplet and

- making use of triplets with positive interactions to orient chains in additions to v-structures.

In the experimental study we propose to use a fast estimator based on the assumption that the data follows a multivariate Gaussian distribution. We show that our novel arc orientation method, named OMbIT, performs better than state-of-the-art methods on both synthetic and biological data sets.

7.1.4 Influence of entropy estimation on inferred networks

In this experimental study we show the influence of entropy estimation on network inference. We show that depending on the entropy estimator results vary considerably and furthermore that the consistently best results are obtained using estimators based on correlation.

7.2 Future work

In this section we will first outlay a number of short term goals mainly based on the work presented in this thesis and then mention several more general areas in which we our techniques could contribute.

7.2.1 Methodology

Our network inference method *predictionet* integrates prior knowledge in both steps, the structure identification and the orientation phase using a weighting parameter $w \in [0, 1]$. We have observed that the best result is not always obtained using the same values for w but instead the strength of prior integration depends on the data as well as on the quality of prior knowledge in the studied problem. In the future, we aim at determining this w in a i) purely data driven framework and ii) investigate whether there is an optimal value that can be derived theoretically. This could be inspired by the optimal λ^* determined for the shrinkage entropy estimator (Section 3.1.2.1). Furthermore, we aim at extending the prior integration framework using more than two sources similarly to the approach kernel methods are taking. With the multitude of available data sets we need to take of the possibly complementary information contained in these data sets.

Another possibility to integrate prior knowledge in network inference based on the estimation of information theoretic quantities directly with the mutual information estimation. We aim to investigate whether this integration provides a i) theoretically sound estimator and ii) improves the final network compared to our original work.

Another important direction to investigate is whether different representations of prior knowledge lead to significantly different results or whether these differences can be controlled solely by the prior weight parameter.

Throughout this thesis we have often assumed linear dependencies. This assumption usually does not hold up in practice and we need to extend for example the prediction part to non-linear models.

7.2.2 Experiments

With the growing availability of next generation sequencing data, our validation approach should be validated using RNA-sequencing data. It provides a convenient strategy to compare networks obtained from microarray data and those obtained from RNA-sequencing data [GDFL13].

We presented an experimental study on colon cancer using only eight knock-down genes in two cell lines. However, we assessed their effect on other genes on a genome-wide scale. Very recently, a new database, named COLT[1], of perturbation experiments is making available perturbation data sets from more than 70 Pancreatic, Ovarian and Breast human cancer cell lines covering ∼16000 human genes. This is the ideal data set to test our validation framework more extensively and possibly identify new candidates for targeted treatment.

Not only can we use this publicly available data to test our framework but use our parallel implementation of *ensemble mRMR* in the feature selection step of *predictionet*. This will allow us to not only show its usefulness when inferring networks from generated data but on a large scale, real data case study. Furthermore, this study will allow us to determine in a data-driven fashion the optimal setting for the number of levels in a tree and the number of variables to be selected in each of the branches and lastly the optimal combination scheme for the different branches.

Unfortunately such vast data is not available for most diseases. In those cases, inferred networks can help the design of the perturbation study, that is identify good targets and moreover select different targets for multivariate experiments.

7.2.3 Perspectives

The future of cancer research lies in *Network Medicine* as it is not single genes but gene networks that guide pathologies [BGL11]. This will eventually entail that treatments are administered based on a patient's *interactome*, in other words according to a tumor's subtype classified based on the inferred network.

However, until now the main focus of research in our field were the identification of important gene-gene interactions and their subsequent verification mainly in laboratory experiments. Due to the inherent difficulty of genomic data, researchers in our domain first needed to understand which strategies are key to model gene-gene interactions from genomic data. These are:

[1] http://colt.ccbr.utoronto.ca/cancer/

(i) the integration of prior knowledge to improve the inferred networks' quality,

(ii) the integration of ensemble/bagging strategies to reduce the variability of the resulting networks and

(iii) the combination of different good methods into consensus models which showed on average to be outperforming the single methods [MCKf$^+$12].

In this thesis we have designed methods that address the first two requirements. Furthermore, the methods we designed are modular and can thus be combined as needed. For example, ensemble mRMR can be combined with OMbIT or/and with prior prior knowledge. With respect to the validation of the inferred gene-gene interactions we proposed a novel data-diven validation framework which allows to assess any inferred network and thus reducing the need for subsequent experimental validation.

The next logical step will be to make use of the networks beyond the understanding of which gene-gene interactions can be modeled from the given data. This will require a paradigm shift in the research field from the problem of 'inferring the correct network' to the problem of 'inferring a useful network'. In order to reach this point, I see three important steps ahead of us.

Before being able to classify a cancer subtype based on a network, we first need to understand the differences in the networks inferred from healthy samples and those inferred from tumor samples. With our inference algorithm, we will be able to infer these two networks and identify the most significant differences. Once these differences are understood, we can start building different networks for different tumor subtypes.

The second step is related to drug design. With the network in hand we can identify which pathways are compensating for each other in case one is blocked (for example after administering a single drug). Ideally, we will use our networks as predictive models to understand when this effect can occur. However, until now this task has been too challenging. A possible solution could be to integrate different types of data, for example about protein activity, which would mean to overcome our simplifying assumptions that cell activity can be captured by modeling gene-gene interactions from expression data.

The last step concerns the identification of possible drug combinations from networks that are able to make use of these predictive models. In order to identify candidated drug combinations, we will need to extend our data from single knock-down to multiple overexpression (a common phenomenon in cancer related genes) and knock-down experiments. In order to decide which genes to perturb in which fashion (overexpression versus knock-down), we will be able to use the predictive models to identify the best candidate combinations.

Filling these steps with life will be the next great challenge for our field.

List of Figures

List of Tables

Bibliography

[ADH09] S. Anjum, A. Doucet, and C.C. Holmes. A boosting approach to structure learning of graphs with and without prior knowledge. *Bioinformatics*, 25:2929–2936, November 2009.

[AES10] G. Altay and F. Emmert-Streib. Inferring the conservative causal core of gene regulatory networks. *BMC Systems Biology*, 4(1):132, 2010.

[AES11] G. Altay and F. Emmert-Streib. Structural influence of gene networks on their inference: Analysis of c3net. *Submitted*, 2011.

[Aff04] Affymetrix. GeneChip Expression Analysis: Data Analysis Fundamentals. Technical report, 2004.

[Aka01] S. Akaho. A kernel method for canonical correlation analysis. In *In Proceedings of the International Meeting of the Psychometric Society (IMPS2001*. Springer-Verlag, 2001.

[ALCD08] E. Almasri, P. Larsen, G. Chen, and Y. Dai. Incorporating literature knowledge in bayesian network for inferring gene networks with gene expression data. In *Proceedings of the 4th international conference on Bioinformatics research and applications*, ISBRA'08, pages 184–195, Berlin, Heidelberg, 2008. Springer-Verlag.

[Ana07] D. Anastassiou. Computational analysis of the synergy among multiple interacting genes. *Molecular systems biology*, 3, February 2007.

[ATS+03] C. Aliferis, I. Tsamardinos, A. Statnikov, C.F. Aliferis M. D, Ph. D, and I. Tsamardinos Ph. D. Hiton, a novel markov blanket algorithm for optimal variable selection, 2003.

[AWP+09] S. Ahn, R.T. Wang, C.C. Park, A. Lin, R.M. Leahy, K. Lange, and D.J. Smith. Directed mammalian gene regulatory networks using expression and comparative genomic hybridization microarray data from radiation hybrids. *PLoS Comput Biol*, 5(6):e1000407, 06 2009.

[BB03] P. Bühlmann and Yu B. Boosting with the l2 loss: Regression and classification. *Journal of the American Statistical Association*, 98:324–339, 2003.

[BBAIdB07] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo. How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3(1):78, 2007.

[BdlFM02] P. Brazhnik, A. de la Fuente, and P. Mendes. Gene networks: how to put the function in genomics. *Trends in biotechnology*, 20(11):467–472, November 2002.

[Bel03] A.J. Bell. The co-information lattice. In *Proc. Fourth Int'l Symp. Independent Component Analysis and Blind Signal Separation (ICA '03)*, pages 921–926, 2003.

[BGL11] A.-L. Barabasi, N. Gulbahce, and J. Loscalzo. Network medicine: a network-based approach to human disease. *Nat Rev Genet*, 12(1):56–68, January 2011.

[BH95] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[BJ03] F.R. Bach and M.I. Jordan. Kernel independent component analysis. *J. Mach. Learn. Res.*, 3:1–48, 2003.

[BK00] A.J. Butte and I.S. Kohane. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput*, 5:418–429, 2000.

[BL04] M.J. Buck and J.D. Lieb. Chip-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, 83, 2004.

[BM10] G. Bontempi and P.E. Meyer. Causal filter selection in microarray data. In *ICML*, pages 95–102, 2010.

[Bos89] J.L. Bos. ras oncogenes in human cancer: a review. *Cancer research*, 49(17):4682–4689, September 1989.

[Bou93] R.R. Bouckaert. Probalistic network construction using the minimum description length principle. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, ECSQARU '93, pages 41–48, London, UK, UK, 1993. Springer-Verlag.

[BSBDN+07] S. Benvenuti, A. Sartore-Bianchi, F. Di Nicolantonio, C. Zanon, M. Moroni, S. Veronese, S. Siena, and A. Bardelli. Oncogenic activation of the RAS/RAF signaling pathway impairs the response of metastatic colorectal cancers to anti-epidermal growth factor receptor antibody therapies. *Cancer Res*, 67(6):2643–2648, March 2007.

[BST+05] T. Barrett, T.O. Suzek, D.B. Troup, S.E. Wilhite, W.-C. Ngau, P. Ledoux, D. Rudnev, A.E. Lash, W. Fujibuchi, and R. Edgar. NCBI GEO: mining millions of expression profiles–database and tools. *Nucleic acids research*, 33(Database issue):D562–6, January 2005.

[BT12] G. Borboudakis and I. Tsamardinos. Incorporating causal prior knowledge as path-constraints in bayesian networks and maximal ancestral graphs. In *ICML*. icml.cc / Omnipress, 2012.

[BTA04] L.E. Brown, I. Tsamardinos, and C.F. Aliferis. A novel algorithm for scalable and accurate bayesian network learning. *Medinfo*, 11(Pt 1):711–715, 2004.

BIBLIOGRAPHY

[BTS+00] A.J. Butte, P. Tamayo, D. Slonim, T.R. Golub, and I.S. Kohane. Discovering functional relationships between rna expression and chemotherapeutic susceptibility using relevance networks. *Proc Natl Acad Sci U S A*, 97(22):12182–12186, October 2000.

[BYC+05] A.H. Bild, G. Yao, J.T. Chang, Q. Wang, A. Potti, D. Chasse, M.-B. Joshi, D. Harpole, J.M. Lancaster, A. Berchuck, J.A. Olson, J.R. Marks, H.K. Dressman, M. West, and J.R. Nevins. Oncogenic pathway signatures in human cancers as a guide to targeted therapies. *Nature*, 439(7074):353–357, November 2005.

[CDE+98] S. Chu, J.L. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P.O. Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282, 1998.

[CGD+10] E.G. Cerami, B.E. Gross, E. Demir, I. Rodchenkov, O. Babur, N. Anwar, N. Schultz, G.D. Bader, and C. Sander. Pathway Commons, a web resource for biological pathway data. *Nucleic acids research*, 39(Database):D685–D690, December 2010.

[CGD+11] E.G. Cerami, B.E. Gross, E. Demir, I. Rodchenkov, Ö. Babur, N. Anwar, N. Schultz, G.D. Bader, and C. Sander. Pathway commons, a web resource for biological pathway data. *Nucleic Acids Research*, 39(suppl 1):D685–D690, January 2011.

[CGH94] D. Chickering, D. Geiger, and D. Heckerman. Learning bayesian networks is NP-hard. Technical report, Microsoft Research, 1994.

[CGK+01] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, .., 2001.

[CH92] G.F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 09(4):309–347, October 1992.

[CH12] T. Claassen and T. Heskes. A bayesian approach to constraint based causal inference. In N. de Freitas and K.P. Murphy, editors, *UAI*, pages 207–216. AUAI Press, 2012.

[Chi95] D.M. Chickering. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence, Montreal, QU*, pages 87–98. Morgan Kaufmann, August 1995.

[CR06] R. Castelo and A. Roverato. A robust procedure for gaussian graphical model search from microarray data with p larger than n. *J. Mach. Learn. Res.*, 7:2621–2650, 2006.

[CS00] R. Castelo and A. Siebes. Priors on network structures. Biasing the search for Bayesian networks. *Int. J. Approx. Reasoning*, 24(1):39–57, 2000.

[CT90] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1990.

[CW93] D.R. Cox and N. Wermuth. Linear dependencies represented by chain graphs. *Statistical Science*, 8(3):204–218, 1993. With comments and a rejoinder by the authors.

[CWK+02] E. Chudin, R. Walker, A. Kosaka, S.X. Wu, D. Rabert, T.K. Chang, and D.E. Kreder. Assessment of the relationship between signal intensities and transcript concentration for affymetrix genechip arrays. *Genome Biol*, 3(1):RESEARCH0005, 2002.

[DBTvOM07] T. De Bie, L.-C. Tranchevent, L.M.M. van Oeffelen, and Y. Moreau. Kernel-based data fusion for gene prioritization. *Bioinformatics*, 23(13):i125–i132, July 2007.

[DCTC09] B. Di Camillo, G. Toffolo, and C. Cobelli. A gene network simulator to assess reverse engineering algorithms. *Annals of the New York Academy of Sciences*, 1158:125–142(18), March 2009.

[DIB97] J.L. DeRisi, V.R. Iyer, and P.O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338), 1997.

[DJPCO+13] N. De Jay*, S. Papillon-Cavanagh*, C. Olsen, N. El-Hachem, G. Bontempi, and B. Haibe-Kains. mrmre: an r package for parallelized mrmr ensemble feature selection. *Bioinformatics*, 2013.

[DKS95] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *International Conference on Machine Learning*, pages 194–202, 1995.

[DMK+05] S. Durinck, Y. Moreau, A. Kasprzyk, S. Davis, B. De Moor, A. Brazma, and W. Huber. BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21(16):3439–3440, August 2005.

[dMSES11] R. de Matos Simoes and F. Emmert-Streib. Influence of statistical estimators of mutual information and data heterogeneity on the inference of gene regulatory networks. *PLoS ONE*, 6(12):e29279+, December 2011.

[dMSES12] R. de Matos Simoes and F. Emmert-Streib. Bagging statistical network inference from large-scale gene expression data. *PLoS ONE*, 7(3):e33624, 03 2012.

[DP05] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3:185–205, 2005.

[DQ08] A. Djebbari and J. Quackenbush. Seeded bayesian networks: Constructing genetic networks from microarray data. *BMC Systems Biology*, 2(1):57, 2008.

[Dra03] S. Draghici. *Data Analysis Tools for DNA Microarrays*. Chapman & Hall/CRC, June 2003.

[DSSK04] C.O. Daub, R. Steuer, J. Selbig, and S. Kloska. Estimating mutual information using b-spline functions - an improved similarity measure for analysing gene expression data. *BMC Bioinformatics*, 5:118, 2004.

[Dyk70] R. Dykstra. Establishing the positive definiteness of the sample covariance matrix. *The Annals of Mathematical Statistics*, 1970.

[Edw00] D. Edwards. *Introduction to Graphical Modelling*. Springer, 2000.

[EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[FBBR99] T.L. Ferea, D. Botstein, P.O. Brown, and R.F. Rosenzweig. Systematic changes in gene expression pattern following adaptive evolution in yeast. *Proc.Natl.Acad.Sci.*, 96, 1999.

[FBHM04] A. Fuente, N. Bing, I. Hoeschele, and P. Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18):3565–3574, December 2004.

[Fel68] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, January 1968.

[FHea07] J.J. Faith, B. Hayete, and J.T. Thaden et al. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biology*, 5, 2007.

[FLNP00] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.

[FNP99] N. Friedman, I. Nachman, and D. Pe'er. Learning bayesian network structure from massive datasets: The "sparse candidate" algorithm. In *UAI*, pages 206–215, 1999.

[FSGM+07] A. Fujita, J.R. Sato, H.M. Garay-Malpartida, R. Yamaguchi, S. Miyano, M.C. Sogayar, and C.E. Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC systems biology*, 1:39, xx 2007.

[GAC+08] I. Guyon, C.F. Aliferis, G.F. Cooper, A. Elisseeff, J.-P. Pellet, P. Spirtes, and A.R. Statnikov. Design and analysis of the causation and prediction challenge. *Journal of Machine Learning Research - Proceedings Track*, 3:1–33, 2008.

[GAE07] I. Guyon, C. Aliferis, and A. Elisseeff. Causal feature selection. *Computational Methods of Feature Selection*, 2007.

[GC99] C. Glymour and G.F. Cooper, editors. *Computation, Causation, and Discovery*. AAAI Press, 1999.

[GCV+11] J.-P. Gillet, A.M. Calcagno, S. Varma, M. Marino, L.J. Green, M.I. Vora, C. Patel, J.N. Orina, T.A. Eliseeva, V. Singal, R. Padmanabhan, B. Davidson, R. Ganapathi, A.K. Sood, B.R. Rueda, S.V. Ambudkar, and M.M. Gottesman. Redefining the relevance of established cancer cell lines to the study of mechanisms of clinical anti-cancer drug resistance. *Proceedings of the National Academy of Sciences*, 108(46):18708–18713, November 2011.

[GDFL13] F.M. Giorgi, C. Del Fabbro, and F. Licausi. Comparative study of RNA-seq- and microarray-derived coexpression networks in arabidopsis thaliana. *Bioinformatics*, 29(6):717–724, March 2013.

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[Geu10] P. Geurts. Bias vs variance decomposition for regression and classification. In *Data Mining and Knowledge Discovery Handbook*, pages 733–746. 2010.

[GHB13] A. Greenfield, C. Hafemeister, and R. Bonneau. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics*, 29(8):1060–1067, April 2013.

[GHM+01] A.P. Gasch, M. Huang, S. Metzner, D.Botstein, S.J. Elledge, and P.O. Brown. Genomic expression responses to dna-damaging agents and the regulatory role of the yeast atr homolog mec1p. *Mol.Biol.Cell*, 12, 2001.

[Gra69] C.W.J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, August 1969.

[GSK+00] A.P. Gasch, P.T. Spellman, C.M. Kao, O. Carmel-Harel, M-B-Eisen, G. Storz, D. Botstein, and P.O. Brown. Genomic expression programs in the respomse of yeast cells to environmental changes. *Mol.Biol.Cell*, 11, 2000.

[GUV+07] P. Godard, A. Urrestarazu, S. Vissers, K. Kontos, G. Bontempi, J. van Helden, and B. André. Effect of 21 different nitrogen sources on global gene expression in the yeast saccharomyces cerevisiae. *Molecular and Cellular Biology*, 27:3065–3086, 2007.

[GVVDM07] O. Gevaert, S. Van Vooren, and B. De Moor. A framework for elucidating regulatory networks based on prior information and expression data. *Annals of the New York Academy of Sciences*, 1115(1):240–248, 2007.

[Hau06] J. Hausser. Improving entropy estimation and the inference of genetic regulatory networks. Master's thesis, Dépt Biosciences and Bâtiment Louis Pasteur and Avenue Jean Capelle and F- Villeurbanne Cedex, August 2006.

[Hay94] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, 1994.

[Hec95] D. Heckerman. A tutorial on learning with bayesian networks. Technical report, Microsoft Research, 1995.

[HGC95] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[HGJY02] A.J. Hartemink, D.K. Gifford, T.S. Jaakkola, and R.A. Young. Combining location and expression data for principled discovery of genetic regulatory network models. In *Proc. of the Pacific Symp. on Biocomputing*, volume 7, pages 437–449, 2002.

[HGL+04] C.T. Harbison, D.B. Gordon, T.I. Lee, N.J. Rinaldi, K.D. Macisaac, T.W. Danford, N.M. Hannett, J.-B. Tagne, D.B. Reynlds, J. Yoo, E.G. Jennings, J. Zeitlinger, D.K. Pokholok, M. Kellis, P.A. Rolfe, K.T. Takusagawa, E.S. Lander, D.K. Gifford, E. Fraenkel, and R.A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 2004.

[HK70] A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[HKOBQ12] B. Haibe-Kains, C. Olsen, G. Bontempi, and J. Quackenbush. *predictionet: Inference for predictive networks designed for (but not limited to) genomic data*, 2012. R package version 1.1.5.

BIBLIOGRAPHY

[HKOD+12] B. Haibe-Kains, C. Olsen, A. Djebbari, G. Bontempi, M. Correll, C. Bouton, and J. Quackenbush. Predictive networks: a flexible, open source, web application for integration and analysis of human gene networks. *Nucleic acids research*, 40(D1):D866–D875, January 2012.

[HMJ+00] T.R. Hughes, M.J. Marton, A.R. Jones, C.J. Roberts, R. Stoughton, C.D. Armour, H.A. Bennett, E. Coffey, H. Dai, Y.D. He, M.J. Kidd, A.M. King, M.R. Meaer, D. Slade, P.Y. Lum, S.B. Stepaniants, D.D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S.H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102, 2000.

[HR13] M.A. Hernan and J.M. Robins. *Causal Inference*. Monographs on Statistics and Applied Probability. Taylor & Francis Group, 2013.

[HTF03] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, corrected edition, July 2003.

[HTIWG10] V.A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts. Inferring regulatory networks from expression data using tree-based methods. *Plos ONE*, 5(9):e12776, sept 2010.

[HV05] R. Hoffmann and A. Valencia. Implementing the iHOP concept for navigation of biomedical literature. *Bioinformatics (Oxford, England)*, 21 Suppl 2, September 2005.

[HW07] D. Husmeier and A.V Werhli. Bayesian integration of biological prior knowledge into the reconstruction of gene regulatory networks with bayesian networks. *Comput Syst Bioinformatics Conf*, 6:85–95, 2007.

[HY10] Z. He and W. Yu. Stable feature selection for biomarker discovery. *Computational Biology and Chemistry*, 34(4):215 – 225, 2010.

[IHG+03] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. In *In Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB 03*, pages 104–113. IEEE, 2003.

[IRMM10] A. Ikin, C. Riveros, P. Moscato, and A. Mendes. The gene interaction miner: a new tool for data mining contextual information for protein-protein interaction analysis. *Bioinformatics (Oxford, England)*, 26(2):283–284, January 2010.

[ISG+02] S. Imoto, K. Sunyong, T. Goto, S. Aburatani, K. Tashiro, S. Kuhara, and S. Miyano. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *Proc. 1st IEEE Computer Society Bioinformatics Conference*, 1:219–227, 2002.

[JN07] F.V. Jensen and T.D. Nielsen. *Bayesian Networks and Decision Graphs (2nd ed.)*. Springer, 2007.

[KB08] K. Kontos and G. Bontempi. Nested q-partial graphs for genetic network inference from small n, large p microarray data. In *Bioinformatics Research and Development, Second International Conference, BIRD 2008, Vienna, Austria, July 7-9, 2008, Proceedings*, volume 13 of *Communications in Computer and Information Science*, pages 273–287. Springer, 2008.

[KB09] K. Kontos and G. Bontempi. An improved shrinkage estimator to infer regulatory networks with gaussian graphical models. In *Proceedings of the 2009 ACM symposium on Applied Computing*, SAC '09, pages 793–798, New York, NY, USA, 2009. ACM.

[KD05] P. Khatri and S. Draghici. Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, 21(18):3587–3595, 2005.

[KEBC05] M. Kaern, T. C. Elston, W. J. Blake, and J. J. Collins. Stochasticity in gene expression: from theories to phenotypes. *Nat Rev Genet*, 6(6):451–64, 2005.

[KFGT07] D. Koller, N. Friedman, L. Getoor, and B. Taskar. Graphical models in a nutshell. In L. Getoor and B. Taskar, editors, *An Introduction to Statistical Relational Learning*. MIT Press, 2007.

[KJ97] R. Kohavi and G.H. John. Wrappers for feature subset selection, 1997.

[KK51] J. F. Kenney and E. S. Keeping. *Mathematics of Statistics Part Two*. D. Van Nostrand Company Inc, 2nd edition, 1951.

[KL02] R.I. Kondor and J.D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 315–322, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[Kon09] K. Kontos. *Gaussian Graphical Model Selection for Gene Regulatory Network Reverse Engineering and Function Prediction*. PhD thesis, Université Libre de Bruxelles, Belgium, 2009.

[KSOA99] M. Kendall, A. Stuart, K.J. Ord, and S. Arnold. *Kendall's Advanced Theory of Statistics:Volume 2A -Classical Inference and and the Linear Model (Kendall's Library of Statistics)*. A Hodder Arnold Publication, 6 edition, April 1999.

[KT81] R. Krichevsky and V. Trofimov. The performance of universal coding. *IEEE Trans. Inform. Theory*, 27:199–207, 1981.

[KTA05] T. Kato, K. Tsuda, and K. Asai. Selective integration of multiple biological data for supervised network inference. *Bioinformatics*, 21:2488–2495, May 2005.

[Lau96] S.L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

[LHW08] W. Luo, K.D. Hankenson, and P.J. Woolf. Learning transcriptional regulatory networks from high throughput gene expression data using continuous three-way mutual information. *BMC bioinformatics*, 9(1):467+, November 2008.

[LSM+76] A. Leibovitz, J.C. Stinson, W.B. McCombs, C.E. McCoy, K.C. Mazur, and N.D. Mabry. Classification of human colorectal adenocarcinoma cell lines. *Cancer Res*, 36(12):4562–4569, December 1976.

[LW03] O. Ledoit and M. Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621, December 2003.

[LZB+09] Y. Li, Y. Zhu, X. Bai, H. Cai, W.i Ji, and D. Guo. Retrn: a retriever of real transcriptional regulatory network and expression data for evaluating structure learning algorithm. *Genomics*, August 2009.

[Mar03] D. Margaritis. *Learning Bayesian Network Model Structure from Data*. PhD thesis, Carnegie Mellon University, 2003.

[MBI10] M.N. McCall, B.M. Bolstad, and R.A. Irizarry. Frozen robust multiarray analysis (fRMA). *Biostatistics (Oxford, England)*, 11(2):242–253, April 2010.

[MC07] N.D. Mukhopadhyay and S. Chatterjee. Causality and pathway search in microarray time series experiment. *Bioinformatics*, 23(4):442–449, 2007.

[McD09] J.H. McDonald. *Handbook of Biological Statistics*. Sparky House Publishing, Baltimore, Maryland, USA, second edition, September 2009.

[McG54] W.J. McGill. Multivariate information transmission. *Psychometrika*, 1954.

[MCKf+12] D. Marbach, J.C. Costello, R. Kü ffner, N.M. Vega, R.J. Prill, D.M. Camacho, K.R. Allison, M. Kellis, J.J. Collins, G. Stolovitzky, the DREAM5 Consortium, and Y. Saeys. Wisdom of crowds for robust gene network inference. *NATURE METHODS*, 9(8):796–804, 2012.

[mCRE+10] The modENCODE Consortium, S. Roy, J. Ernst, P.V. Kharchenko, P. Kheradpour, N. Negre, M.L. Eaton, J.M. Landolin, C.A. Bristow, L. Ma, M.F. Lin, S. Washietl, B.I. Arshinoff, F. Ay, P.E. Meyer, N. Robine, N.L. Washington, L. Di Stefano, E. Berezikov, C.D. Brown, R. Candeias, J.W. Carlson, A. Carr, I. Jungreis, D. Marbach, R. Sealfon, M.Y. Tolstorukov, S. Will, A.A. Alekseyenko, C. Artieri, B.W. Booth, A.N. Brooks, Q. Dai, C.A. Davis, M.O. Duff, X. Feng, A.A. Gorchakov, T. Gu, J.G. Henikoff, P. Kapranov, R. Li, H.K. MacAlpine, J. Malone, A. Minoda, J. Nordman, K. Okamura, M. Perry, S.K. Powell, N.C. Riddle, A. Sakai, A. Samsonova, J.E. Sandler, Y.B. Schwartz, N. Sher, R. Spokony, D. Sturgill, M. van Baren, K.H. Wan, L. Yang, C. Yu, E. Feingold, P. Good, M. Guyer, R. Lowdon, K. Ahmad, J. Andrews, B. Berger, S.E. Brenner, M.R. Brent, L. Cherbas, S.C.R. Elgin, T.R. Gingeras, R. Grossman, R.A. Hoskins, T.C. Kaufman, W. Kent, M.I. Kuroda, T. Orr-Weaver, N. Perrimon, V. Pirrotta, J.W. Posakony, B. Ren, S. Russell, P. Cherbas, B.R. Graveley, S. Lewis, G. Micklem, B. Oliver, P.J. Park, S.E. Celniker, S. Henikoff, G.H. Karpen, E.C. Lai, D.M. MacAlpine, L.D. Stein, K.P. White, and M. Kellis. Identification of functional elements and regulatory circuits by drosophila modencode. *Science*, 330(6012):1787–1797, 2010.

[Mee95] C. Meek. Causal inference and causal explanation with background knowledge. *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pages 403–441, August 1995.

[Mey08] P.E. Meyer. *Information-Theoretic Variable Selection and Network Inference from Microarray Data*. PhD thesis, Université Libre de Bruxelles, Belgium, 2008.

[Mil55] G.A. Miller. Information theory in psychology: Problems and methods. *Free Press*, 1955.

[MKLB07] P.E. Meyer, K. Kontos, F. Lafitte, and G. Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP Journal on Bioinformatics and Systems Biology*, Special Issue on Information-Theoretic Methods for Bioinformatics, 2007.

[MLB07a] P.E. Meyer, F. Lafitte, and G. Bontempi. *minet: Mutual Information Network Inference*, 2007. R package version 1.8.0.

[MLB07b] P.E. Meyer, F. Lafitte, and G. Bontempi. *minet: Mutual Information Network Inference*, 2007. R package version 1.1.3.

[MLB08] P.E. Meyer, F. Lafitte, and G. Bontempi. MINET: An open source R/Bioconductor package for mutual information based network inference. *BMC Bioinformatics*, 2008.

[MM05] J.A. Birchler M.A. Matzke. Rnai-mediated pathways in the nucleus, 2005.

[MMF09] D. Marbach, C. Mattiussi, and D. Floreano. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology. To appear.*, 2009.

[MNea06] A.A. Margolin, I. Nemenman, and K. Basso et al. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7, 2006.

[MOB11a] P.E. Meyer, C. Olsen, and G. Bontempi. Transcriptional network inference based on information theory. In *Applied Statistics for Network Biology: Methods in Systems Biology*. Wiley, 2011.

[MOB11b] A.A. Miranda, C. Olsen, and G. Bontempi. Fourier spectral factor model for prediction of multidimensional signals. *Signal Processing*, 91(9):2172 – 2177, 2011.

[MPS+10a] D. Marbach, R.J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, 107(14):6286–6291, 2010. WingX.

[MPS+10b] Daniel Marbach, Robert J. Prill, Thomas Schaffter, Claudio Mattiussi, Dario Floreano, and Gustavo Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, 107(14):6286–6291, 2010.

[MR05] O. Maimon and L. Rokach, editors. *The Data Mining and Knowledge Discovery Handbook*. Springer, 2005.

[MRF+08] S. Mostafavi, D. Ray, D.W. Farley, C. Grouios, and Q. Morris. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, 9(Suppl 1):S4+, 2008.

[MRL95] Y.-I. Moon, B. Rajagopalan, and U. Lall. Estimation of mutual information using kernel density estimators. *Phys. Rev. E*, 52:2318–2321, September 1995.

[MSOI+02] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

BIBLIOGRAPHY

[MT99] D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems 12*, pages 505–511. MIT Press, 1999.

[Mur01] K. Murphy. Learning bayes net structure from sparse data sets. Technical report, Comp. Sci. Div., UC Berkeley, 2001.

[MVT+06] L. Masramon, E. Vendrell, G. Tarafa, G. Capellà, R. Miró, M. Ribas, and M.A. Peinado. Genetic instability and divergence of clonal populations in colon cancer cells in vitro. *Journal of cell science*, 119(Pt 8):1477–1482, April 2006.

[MWL+06] A.A. Margolin, K. Wang, W.K. Lim, M. Kustagi, I. Nemenman, and A. Califano. Reverse engineering cellular networks. *Nat Protoc*, 1(2):662–671, 2006.

[Nea03] R.E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, April 2003.

[Net12] The Cancer Genome Atlas Network. Comprehensive molecular characterization of human colon and rectal cancer. *Nature*, 487(7407):330337, 2012.

[NJW01] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[NSB02] I. Nemenman, F. Shafee, and W. Bialek. Entropy and inference, revisited. *arXiv*, Jan 2002.

[ODB00] N. Ogawa, J. DeRisi, and P.O. Brown. New componenets of a system for phosphate accumulation and polyphosphate metabolism in saccaromyces cerevisiae revealed by genomic expression analysis. *Mol.Biol.Cell*, 11, 2000.

[ODF+13] C. Olsen, A. Djebbari, K. Fleming, N. Prendergast, R. Rubio, F. Emmert-Streib, G. Bontempi, B. Haibe-Kains, and J. Quackenbush. Inference of predictive gene networks from biomedical literature and gene expression data. *Submitted to Genomics*, 2013.

[OGP+10] D. Onichtchouk, F. Geier, B. Polok, D.M. Messerschmidt, R. Mossner, B. Wendik, S. Song, V. Taylor, J. Timmer, and W. Driever. Zebrafish pou5f1-dependent transcriptional networks in temporal control of early development. *Molecular Systems Biology*, 6, March 2010.

[OHKQB13] C. Olsen, B. Haibe-Kains, J. Quackenbush, and G. Bontempi. On the integration of prior knowledge in the inference of regulatory networks. Accepted for publication., 2013.

[OMB08] C. Olsen, P.E. Meyer, and G. Bontempi. On the impact of entropy estimator in transcriptional regulatory network inference. *Proceedings of WCSB08*, 2008.

[OMB09a] C. Olsen, P.E. Meyer, and G. Bontempi. On the impact of entropy estimation on transcriptional regulatory network inference based on mutual information. *EURASIP Journal on Bioinformatics and Systems Biology*, 2009.

[OMB09b] C. Olsen, P.E. Meyer, and G. Bontempi. Poster: Inferring causal relationships using information-theoretic measures. In *Proceedings of the 5th Benelux Bioinformatics Conference (BBC09)*, 2009.

[OMB13] C. Olsen, P.E. Meyer, and G. Bontempi. A scalable heuristic to orient arcs in undirected networks. *preprint*, 2013.

[ORS07] R. Opgen-Rhein and K. Strimmer. From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC Systems Biology*, 1(1), 2007.

[Pan03] L. Paninski. Estimation of entropy and mutual information. *Neural Computation*, 2003.

[PCJH+13] S. Papillon-Cavanagh, N. De Jay, N. Hachem, C. Olsen, G. Bontempi, H. Aerts, J. Quackenbush, and B. Haibe-Kains. Comparative study and validation of genomic predictors for anticancer drug sensitivity. *Journal of the American Medical Informatics Association*, 2013.

[PE08] J.-P. Pellet and A. Elisseeff. Using markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9:1295–1342, 2008.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, September 1988.

[Pea00] J. Pearl. *Causality : Models, Reasoning, and Inference*. Cambridge University Press, March 2000.

[PRM+03] B.E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and D'alche F. Buc. Gene network inference using dynamic bayesian-networks. *Bioinformatics*, 19(Suppl. 2), 2003.

[PV91] J. Pearl and T. Verma. A theory of inferred causation. In *KR*, pages 441–452, 1991.

[PWCG01] P. Pavlidis, J. Weston, J. Cai, and W.N. Grundy. Gene functional classification from heterogeneous data. In *Proceedings of the fifth annual international conference on Computational biology*, RECOMB '01, pages 249–255, New York, NY, USA, 2001. ACM.

[Ris86] J. Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, 14(3):1080–1100, 1986.

[Scu10] M. Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 7 2010.

[SG02] T. Schürmann and P. Grassberger. Entropy estimation of symbol sequences, 2002.

[SGS01] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, Cambridge, MA, USA, second edition, January 2001.

[Shi02] B. Shipley. *Cause and Correlation in Biology : A User's Guide to Path Analysis, Structural Equations and Causal Inference*. Cambridge University Press, August 2002.

[SHS05] S.S. Sridhar, D. Hedley, and L.L. Siu. Raf kinase as a target for anticancer therapeutics. *Molecular cancer therapeutics*, 4(4):677–685, April 2005.

[Sil86] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, April 1986.

[SJS06] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Proceedings of the AAAI'06 workshop on Evaluation Methods for Machine Learning*, 2006.

[SKCR12] A. Sîrbu, G. Kerr, M. Crane, and H.J. Ruskin. Rna-seq vs dual- and single-channel microarray data: Sensitivity analysis for differential expression and clustering. *PLoS ONE*, 7(12):e50986, 12 2012.

[SKD+02] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 18 Suppl 2, 2002.

[SO09] A. Shermin and M.A. Orgun. Using dynamic bayesian networks to infer gene regulatory networks from expression profiles. In *Proceedings of the 2009 ACM symposium on Applied Computing*, SAC '09, pages 799–803, New York, NY, USA, 2009. ACM.

[SOR+11] M.E. Smoot, K. Ono, J. Ruscheinski, P.-L.L. Wang, and T. Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics (Oxford, England)*, 27(3):431–432, February 2011.

[SORS09] J. Schäfer, R. Opgen-Rhein, and K. Strimmer. *GeneNet: Modeling and Inferring Gene Networks*, 2009. R package version 1.2.4.

[SPP+05] 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(5721):523–529, 2005.

[SS05a] J. Schäfer and K. Strimmer. An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, March 2005.

[SS05b] J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4, 2005.

[SS11] M. Scutari and K. Strimmer. Introduction to graphical modelling, June 2011.

[SSM98] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, July 1998.

[SSZ+98] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol.Biol.Cell*, 9, 1998.

[STC04] J. Shawe-Taylor and Nello Christianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[SWCvH04] N. Simonis, S.J. Wodak, G.N. Cohen, and J. van Helden. Combining pattern discovery and discriminant analysis to predict gene co-regulation. *Bioinformatics*, 20, 2004.

[TA03] I. Tsamardinos and C. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*. Morgan Kaufmann Publishers, 2003.

[TAA03] K. Tsuda, S. Akaho, and K. Asai. The em algorithm for kernel matrix completion with auxiliary data. *J. Mach. Learn. Res.*, 4:67–81, December 2003.

[TAS03a] I. Tsamardinos, C. Aliferis, and A. Statnikov. Time and sample efficient discovery of markov blankets and direct causal relations. In *Proceedings of the 9th CAN SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 673–678, 2003.

[TAS03b] I. Tsamardinos, C. Aliferis, and A.R. Statnikov. Algorithms for large scale markov blanket discovery. In *The 16th International FLAIRS Conference, St*, pages 376–380. AAAI Press, 2003.

[TB07] J. Tegnér and J. Björkegren. Perturbations to uncover gene networks. *Trends in genetics : TIG*, 23(1):34–41, January 2007.

[TBA06] I. Tsamardinos, L.E. Brown, and C. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, October 2006.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[TSS05] K. Tsuda, H. Shin, and B. Schölkopf. Fast protein classification with multiple networks. *Bioinformatics*, 21(2):59–65, January 2005.

[VdBVLN+06] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7, 2006.

[Ver03] J.-P. Vert. Graph-driven features extraction from microarray data using diffusion kernels and kernel cca. *Advances in Neural Information Processing Systems*, 2003.

[Vic02] J.D. Victor. Binless strategies for estimation of information from neural data. *Physical Review E*, 66(5), November 2002.

[VP91] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *UAI '90: Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 255–270, New York, NY, USA, 1991. Elsevier Science Inc.

[VtVDVdV+02] L.J. Van t Veer, H. Dai, M.J. Van de Vijver, Y.D. He, A.A.M. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, and S.H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, January 2002.

[VY04] J.-P. Vert and Y. Yamanishi. Supervised graph inference. In *NIPS*, 2004.

[WB06] A. Wille and P. Bühlmann. Low-order conditional independence graphs for inferring genetic networks. *Statistical Applications in Genetics and Molecular Biology*, 5(1):1, 2006.

BIBLIOGRAPHY

[WFS10] G. Wu, X. Feng, and L. Stein. A human functional protein interaction network and its application to cancer data analysis. *Genome biology*, 11(5):R53, 2010.

[Whi90] J. Whittaker. *Graphical Models in Applied Multivariate Statistics (Wiley Series in Probability & Statistics)*. John Wiley & Sons, March 1990.

[WL90] N. Wermuth and S.L. Lauritzen. On substantive research hypotheses, conditional independence graphs and graphical chain models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52(1):pp. 21–50, 1990.

[WLB10] L. Wang, P. Li, and T.P. Brutnell. Exploring plant transcriptomes using ultra high-throughput sequencing. *Briefings in Functional Genomics*, 9(2):118–128, 2010.

[WLW+09] J. Watkinson, K. Liang, X. Wang, T. Zheng, and D. Anastassiou. Inference of regulatory gene interactions from expression data using Three-Way mutual information. *Annals of the New York Academy of Sciences*, 1158(1):302–313, March 2009.

[WNR+07] L. Wu, P. Neskovic, E. Reyes, E. Festa, and W. Heindel. Classifying n-back eeg data using entropy and mutual information features. In *ESANN*, 2007.

[WZV+04] A. Wille, P. Zimmermann, E. Vranova, A. Furholz, O. Laule, S. Bleuler, L. Hennig, A. Prelic, P. von Rohr, L. Thiele, E. Zitzler, W. Gruissem, and P. Bühlmann. Sparse graphical gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. *Genome Biology*, 5(11):R92, 2004.

[YG08] K.Y. Yip and M. Gerstein. Training set expansion: an approach to improving the reconstruction of biological networks from limited and uneven reliable interactions. *Bioinformatics (Oxford, England)*, 25(2):243–250, January 2008.

[YVK04] Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20(1):363–370, January 2004.

[YVNK03] Y. Yamanishi, J.-P. Vert, A. Nakaya, and M. Kanehisa. Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. *Bioinformatics*, 19(suppl 1):i323–i330, 2003.

[YW03] Y. Yang and G.I. Webb. On why discretization works for naive-bayes classifiers. In *Proceedings of the 16th Australian Joint Conference on Artificial Intelligence (AI*, pages 440–452, 2003.

[ZH03] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2003.

[ZPZ+13] Y. Zhang, Y. Pu, H. Zhang, Y. Su, L. Zhang, and J. Zhou. Using gene expression programming to infer gene regulatory networks from time-series data. *Computational Biology and Chemistry*, (0):–, 2013.

Appendices

A Notations

In this thesis, bold face notation is used for sets of variables. The transpose of a matrix/vector $\mathbf{X}$ will be denoted by $\mathbf{X}^T$. Estimators of a quantity such as p will be denoted using the hat symbol $\hat{p}$.

A.1 Probabilities

X, Y, Z	upper case letters denote random variables
x, y, z	lower case letters denote their realizations
$\mathcal{X}, \mathcal{Y}, \mathcal{Z}$	corresponding probability spaces
p	probability distribution
$f_{X,Y}$	joint probability density function of X and Y
f_X	marginal density of X
$f_{Y\|X}$	conditional probability density of Y given X
$\mathbb{E}(X)$	expectation value of X
μ	mean of a distribution
a	parameter of Dirichlet distribution
Φ	cumulative distribution function of the standard normal distribution
$\psi(z)$	Digamma function of z
ρ_{XY}	Pearson correlation between X and Y
σ_X	standard deviation of X
$var(X)$	variance of X
$cov(X, Y)$	covariance of X and Y
Σ	covariance matrix
Ω	concentration matrix
ω_{ij}	entries of concentration matrix Ω
$\rho_{XY\|\mathbf{Z}}$	partial correlation between X and Y conditional on $\mathbf{Z}$
$\rho_{ij\|\mathbf{K}}$	partial correlation between X_i and X_j conditional on $X_\mathbf{K}$
$X \perp\!\!\!\perp Y$	X is independent of Y
$X \perp\!\!\!\perp Y \| \mathbf{Z}$	X is conditional independent of Y given $\mathbf{Z}$
$\mathbf{S}_{XY}$	separating set: $X \perp\!\!\!\perp Y \| \mathbf{S}_{XY}$

A.2 Information theory

$H(X), H(p)$	equivalent notation of entropy of X following a distribution p
$H(X, Y)$	joint entropy of X and Y
$H(X\|Y)$	conditional entropy of X given Y
$I(X; Y)$	mutual information of X and Y
$I(X; Y_1, \ldots, Y_n)$	mutual information of X and $Y_1, \ldots, Y_n$
$I(X; Y\|Z)$	mutual information of X and Y given Z
$\mathcal{I}(X, Y, Z)$	interaction information of X, Y and Z

A.3 Graphs

$A, B, \ldots, V_i, \ldots$,X,Y,Z	nodes in the graph
$\mathbf{V}$	set of nodes/vertices
$\mathbf{E}$	set of edges
$\mathcal{G} = (\mathbf{V}, \mathbf{E})$	graph
$\mathcal{G}^{(q)}$	q-partial graph
$E(\mathcal{G})$	energy of graph $\mathcal{G}$
$\mathbf{adj}(X)$	set of adjacent nodes of X
$\mathbf{pa}(X)$	set of parent nodes of X
$an(X), \mathbf{an}(X)$	ancestor, set of ancestors of X
$de(X), \mathbf{de}(X)$	descendant, set of descendants of X
$\mathbf{MB}(X)$	Markov blanket of X

A.4 Network inference

$\mathbf{X} = \{X_1, \ldots, X_n\}$	set of all features
$\mathbf{X}^{\setminus i}$	set of all features $\mathbf{X}$ without X_i
X_T	target variable
$\mathbf{D}$	data set
m	total number of experiments/number of samples in the data set
n	number of variables/genes in the data set
β_i	regression coefficient
$\beta_j^{(i)}$	regression coefficient regressing all X_j, $j \neq i$ onto X_i
$\beta_j^{(i),t}$	t-th iteration of regression coefficient regressing all X_j, $j \neq i$ onto X_i
$\|\mathbf{X}\|_p$	L_p norm of $\mathbf{X}$
$BN = (\mathcal{G}, \mathbf{p})$	Bayesian network consisting of $\mathcal{G}$ and $\mathbf{p}$
$L(\mathcal{G}, \mathbf{p}, \mathbf{D})$	likelihood function
$s(\mathcal{G}, \mathbf{D})$	scoring metric
$p(\mathcal{G})$	prior probability of graph $\mathcal{G}$
$\#(x_i)$	number of occurrences of x_i
λ, $\hat{\lambda}^*$	shrinkage parameter and its optimal estimator
$\mathbf{T}, \mathbf{U}$	estimators: unrestricted estimate, shrinkage target

A.5 Validation

TP, TN	number of true positives, true negatives
FP, FN	number of false positives and false negatives
TPR, FPR	true positive rate, false positive rate
F	F-score

B Preliminaries

B.1 Probabilities

Let X be a continuous random variable. Its **probability density function** (pdf), denoted by $f_X(x)$, satisfies the following two properties

R1 $f_X(x) \geq 0$, $\forall x \in \mathbb{R}$ (non-negativity),

R2 $\int_{-\infty}^{\infty} f_X(x)\,\mathrm{d}x = 1$.

Theorem .1 *[Dyk70] Suppose $\mathbf{X}_1, \ldots, \mathbf{X}_m$ is a random sample for an n-variate normal distribution whose covariance matrix is of full rank. Then the sample covariance matrix $\sum_{i=1}^{m}(\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})^T$ is positive definite with probability one if and only if $m > n$.*

B.2 Information theory

The definitions presented in Section 3.3.2 can be extended to continuous random variables X and Y. Let f be the probability density functions of X [CT90]. The entropy of X is defined as

$$H(X) = -\int_{\mathbb{R}} f(x) \log f(x) dx.$$

Similar extensions exist for the conditional entropy and the mutual information for pairs of continuous random variables X and Y

$$H(X|Y) = -\int_{\mathbb{R}}\int_{\mathbb{R}} f(x,y) \log f(x|y) dxdy$$

and

$$I(X,Y) = \int_{\mathbb{R}}\int_{\mathbb{R}} f(x,y) \log \frac{f(x,y)}{f(x)f(y)} dxdy.$$

B.3 Machine Learning

B.3.1 Quality measures: networks

A quantitative assessment of an inferred network's quality starts with the classification of the network's edges as true positives (TP), true negatives (TN), false positives (FP) or false negatives (FN) and the subsequent recording of the corresponding counts in a confusion matrix.

With such a confusion matrix in hand we can compute different quality measures for the network: precision, recall, specificity and F-score. The *precision* is defined as the

fraction of inferred true edges among all inferred edges

$$precision = \frac{TP}{TP + FP}. \tag{1}$$

On the other hand, *recall* is the fraction of inferred true edges among all true edges

$$recall = \frac{TP}{TP + FN}. \tag{2}$$

Recall is also known as *sensitivity* or *true positive rate* (TPR).

The *F-score* is the weighted average of precision and recall [SJS06]

$$F_\beta = \frac{(1 + \beta^2) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall} = \frac{(1 + \beta^2) \cdot TP}{(1 + \beta^2) \cdot TP + \beta^2 \cdot FN + FP}. \tag{3}$$

Choosing $\beta = 1$ equally balances precision and recall. Other typical values are $\beta = 0.5$ and $\beta = 2$.

The fraction of truly missing edges out of all nonexistent edges is known as *specificity* or *true negative rate* (TNR)

$$specificity = \frac{TN}{TN + FP}. \tag{4}$$

The fraction of falsely identified edges out of all nonexistent edges is known as *false positive rate* (FPR)

$$FPR = \frac{FP}{TN + FP}. \tag{5}$$

The *accuracy* is defined by the fraction of all correctly identified genes, true positives and true negatives, among all genes

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{6}$$

In network inference, the algorithms typically return a score for each of the edges. By varying the threshold on these scores, the networks sparsity can be controlled. Whereas the F-score is a quality measure for a specific threshold, the *receiver operating characteristic* (ROC) curve plots the TPR versus the FPR for varying thresholds. The *area under the curve* (AUC) computes the are under the ROC curve and thus summarizes the information contained in the ROC.

In hypothesis testing, a *false positive* occurs when a true null hypothesis is rejected.

The *false discovery rate* (FDR) controls this type of error in multiple testing procedures [BH95]. It is defined as expected proportion of the erroneously rejected hypotheses amongst all rejected hypotheses [BH95].

B.3.2 Quality measures: predictions

The R^2 measure of quality is defined as

$$R^2 = 1 - \frac{\sum_i (X_i - \hat{X}_i)^2}{\sum_i (X_i - \bar{X})^2}. \tag{7}$$

The root-mean-square error (RMSE) is defined as

$$RMSE = \sqrt{\mathbb{E}((X_i - \hat{X}_i)^2)}. \tag{8}$$

The normalized root-mean-square error (NRMSE) is then

$$NRMSE = \frac{RMSE}{\max_i\{\hat{X}_i\} - \min_i\{\hat{X}_i\}}. \tag{9}$$

The original Matthew's correlation coefficient (MCC) was used to evaluate binary classifications

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{10}$$

B.4 Kernels

A symmetric matrix is **positive semi-definite** if and only if

$$\mathbf{v}'\mathbf{A}\mathbf{v} \geq 0 \tag{11}$$

for all non-zero vectors $\mathbf{v}$ [STC04]. It is **positive definite** if and only if

$$\mathbf{v}'\mathbf{A}\mathbf{v} > 0 \tag{12}$$

for all non-zero vectors $\mathbf{v}$.

B.5 Time-series data

When using time-series data to model causal relationships, it is clear that a cause must preceed its effects in time. However, the interval of time between the occurence of the cause and the manifestation of the effect may vary.

A schematic setup a time-series experiment is depicted in Figure B.1.

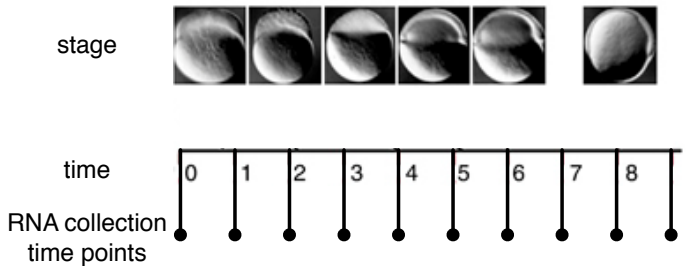


Figure B.1: Measurements are taken at different times in the cell cycle, adaption of figure in [OGP+10].

The data is a vector of observations each made during different points in time. In the literature, methods to model this temporal aspects include auto-regressive models [MC07, FSGM+07] based on Granger causality [Gra69] and dynamic Bayesian networks [PRM+03, SO09].

Time-series data can be used for validation by train the model on a number of time points and using the remaining samples to test the quality of the network as done for a very small network of twelve genes in [ZPZ+13].

C Data generators in bioinformatics: an overview

C.1 Syntren [VdBVLN+06]

Goal	Implementation	Method	Restrictions
• create synthetic transcriptional regulatory networks • simulate gene expression data that approximate experimental data	JAVA application[1]; batch mode via config file **Options** • Subnetwork selection method: neighbor addition, cluster addition • amount of biological and experimental noise • probability for complex 2-regulator interactions • number of external nodes and of correlated external nodes	• Network topologies are generated by selecting subnetworks from previously described regulatory networks • Interaction kinetics are modeled by equations based on Michaelis-Menten and Hill kinetics	• number of nodes restricted by input *sif file • maximal number of nodes in the included network files: max = 1330 (E.Coli), max = 690 S.cerevisiae • bug in network generation from own file (not under active development anymore)

C.2 retrn [LZB+09]

Goal	Implementation	Method	Restrictions
• generate network which possesses scale free property • generate corresponding gene expression data	MATLAB (also runs with Octave) **Options** • cluster or parent addition • possible to specify the number of parent nodes for each node **Usage** • [expreg expname expdata] = retrn(num, varargin); varargin could be either 'ParentAdd' (default) or 'ClusterAdd' • expreg returns the generated network, expname returns the gene names contained in this network, expdata return gene expression data of these genes	• extract subnetworks from known transcription network • data reflects the temporal relationship in gene expression • in each iteration a node is selected randomly and all of its parents are added to the graph and only the nodes with at least one link to the current graph are retained	• max number samples =25 • max number of variables =1163 • creates self-loops

C.3 netsim [DCTC09]

Goal	Implementation	Method	Restrictions
• generate network topology, regulatory rules and expression profiles • reproduce characteristic features of regulation in real gene networks (scale-free, high clustering coefficient, low characteristic path length) • produce time series data	- R package[1] **Usage** • net<- simulatenet(N=50, Xmax=X, times=seq(0,10,10)), net$expr.data returns random initial conditions, second column the final stationary state	• using a new hierarchical modular topology model which mimics the three topological features (scale-free degree distribution, high clustering coefficient, low characteristic path length) • integrates fuzzy logic with differential equations • two models: one describing the network topology, the second generating the epression profiles	

C.4 GeneNetWeaver[MMF09]

Goal	Implementation	Method	Restrictions
time series data or stationary data	Java Web Start[1] **Options** • Networks: Example (scale free network 64 nodes, 207 edges), Ecoli (1502 nodes, 3587 edges), Yeast (4441 nodes, 12873 edges) • Subnetwork extraction: Random vertex (for each subnetwork, the method start from a different random seed node of the source network), selection from list (manual selection from the list of all nodes) • neighbor selection: greedy (choose the neighboring node which maximizes the modularity), random among top (one of the nodes with top p% (specified by the user) modularity is randomly selected) • autoregulatory interactions can be removed • noise: Gaussian noise with chosen standard deviation is added after simulation • data generation: – steady-state – trajectories (from wild-type to the new steady-state in the perturbation experiments; duration and number of time points) – time series (shows how the networks recover from external perturbations; duration and number of time points of each time series)	• extract subnetwork from either Yeast (max 4441 nodes) or Ecoli (max 1502 nodes) • samples are generating by knocking out one gene at the time (*null-mutants.tsv: set to zero, *heterozygous.tsv: set to half of the original value) • network: *gold-standard.tsv or *gold-standard-signed (including in a third column information whether down- or up-regulated)	no batch mode

D Supplementary Information to [ODF+13]

D.1 Figures

D.1.1 F-scores KD data

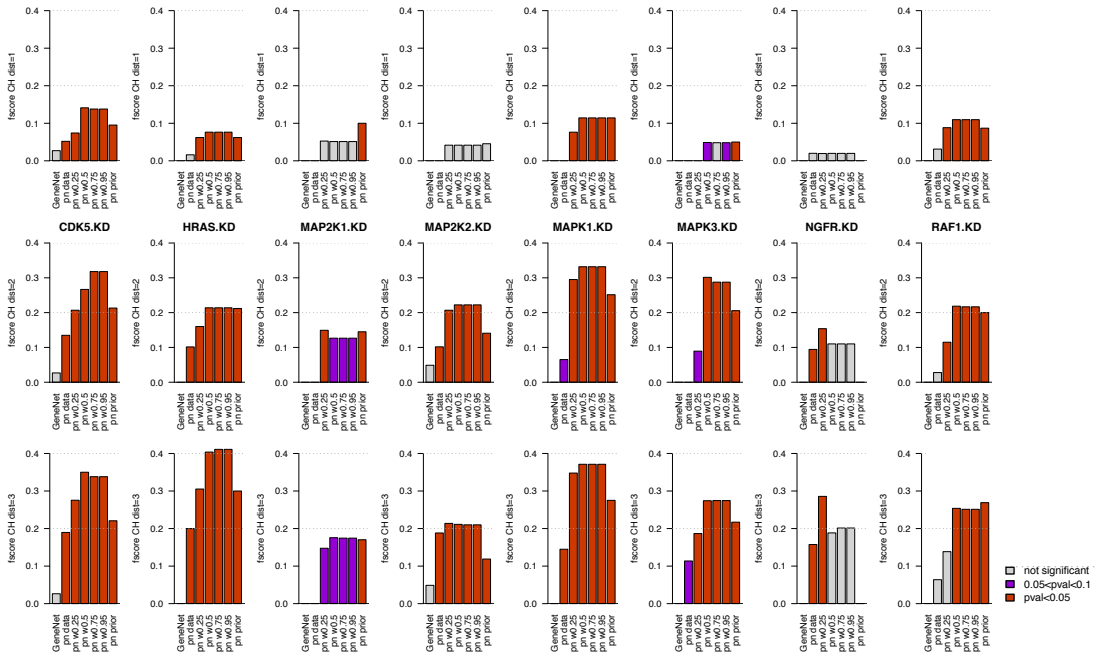


Figure D.1: In cancer cell lines, performance of gene interaction networks inferred from genomic data only (*GeneNet* and *predictionet* with priors weight $w = 0$), priors only (priors weight $w = 1$) and combination of both data sources (priors weight $w = \{0.25, 0.5, 0.75, 0.95\}$). Each column reports the the performance of the network validated in each KD. Bars represent the F-scores of each network in each validation experiment; they are coloured with respect to their significance, that is in red and purple when network's F score is higher than 5% and 10% of random networks, respectively. Rows correspond to childhood sizes one, two and three from top to bottom.

D.1.2 Percentages KD data

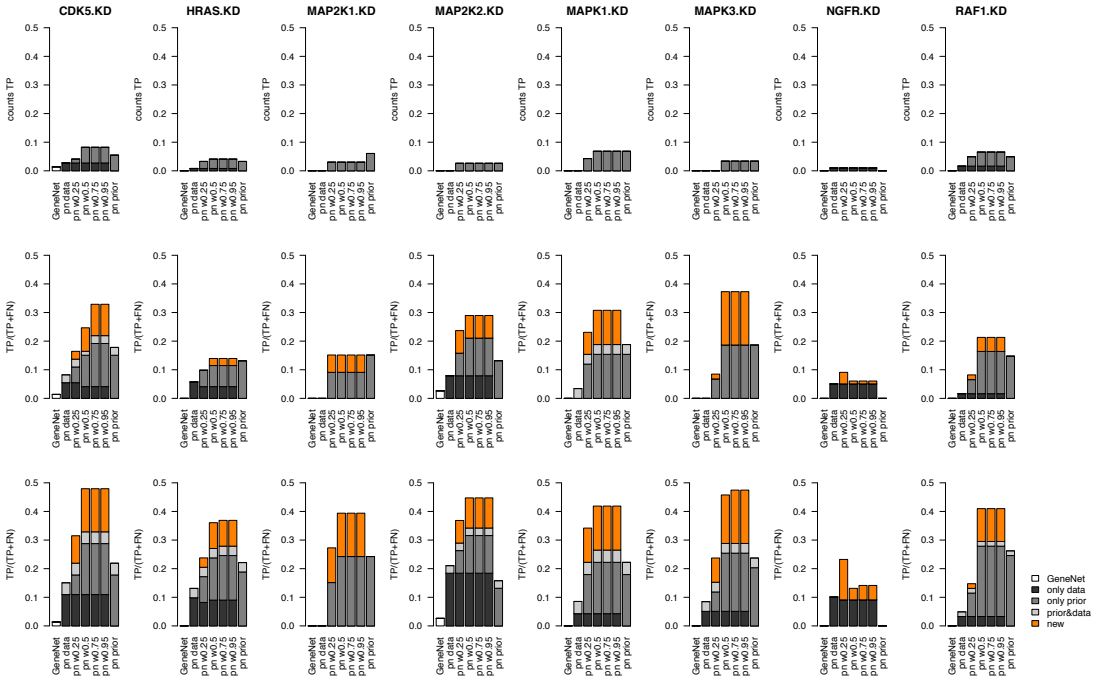


Figure D.2: Bars' heights represent the percentage of true positives with respect to the total number of affected genes for each KD's network; they are coloured by their origin: black for true positives identified in the network inferred from genomic data only, dark grey from priors only, light grey in both, and orange for true positives that are uniquely found in networks inferred by combining genomic data and priors. Rows correspond to childhood sizes one, two and three from top to bottom.

D.1.3 F-scores tumor data

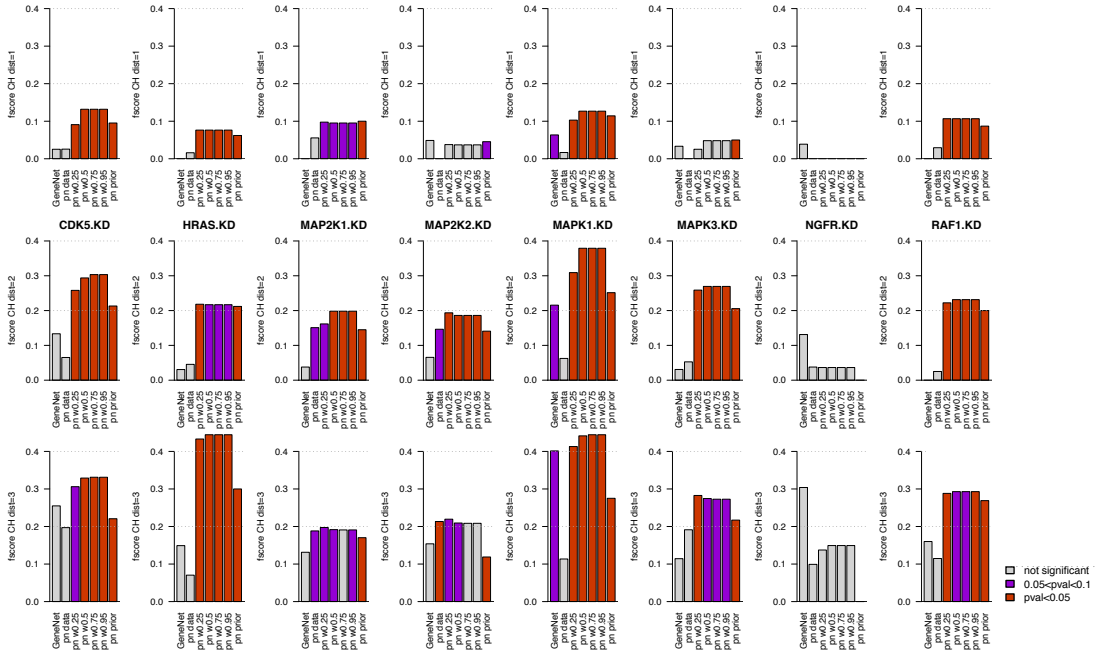


Figure D.3: In tumor data, performance of gene interaction networks inferred from genomic data only (*GeneNet* and *predictionet* with priors weight $w = 0$), priors only (priors weight $w = 1$) and combination of both data sources (priors weight $w = \{0.25, 0.5, 0.75, 0.95\}$). Each column reports the the performance of the network validated in each KD. Bars represent the F scores of each network in each validation experiment; they are colored with respect to their significance, that is in red and purple when network's F score is higher than 5% and 10% of random networks, respectively. Rows correspond to childhood sizes one, two and three from top to bottom.

D.2 Tables

	# common edges	# edges KD	# edges expO
GeneNet	40	1030	1616
0	31	612	1292
0.25	167	821	1523
0.5	322	1029	1615
0.75	344	1046	1620
0.95	348	1050	1620
1	313	313	313

Table D.1: Number of common edges in networks inferred from i) complete KD data (no CV) and ii) expO data; total number of edges in both networks separately.

	CDK5	HRAS	MAP2K1	MAP2K2	MAPK1	MAPK3	NGFR	RAF1
0	1	3	0	2	0	0	0	0
0.25	16	16	22	27	42	28	1	15
0.5	47	21	34	38	72	57	1	19
0.75	52	21	34	38	72	59	1	31
0.95	52	21	34	38	72	59	1	31
1	49	29	36	33	58	48	1	29

Table D.2: Number of common genes in the KD's childhood (distance=2) in networks inferred from i) complete KD data (no CV) and ii) expO data.

		CDK5			HRAS			MAP2K1			MAP2K2		
	# edges	CH2	TP	Fscore	CH2	TP	Fscore	CH2	TP	Fscore	CH2	TP	Fscore
GeneNet	1616	32	7	0.13333	9	2	0.030534	20	1	0.037736	23	2	0.065574
0	1292	19	3	0.065217	10	3	0.045455	20	4	0.15094	44	6	0.14634
0.25	1523	82	20	0.25806	43	18	0.21818	66	8	0.16162	86	12	0.19355
0.5	1615	104	26	0.29379	44	18	0.21687	78	11	0.1982	91	12	0.18605
0.75	1620	105	27	0.30337	44	18	0.21687	78	11	0.1982	91	12	0.18605
0.95	1620	105	27	0.30337	44	18	0.21687	78	11	0.1982	91	12	0.18605
1	313	49	13	0.21311	29	16	0.21192	36	5	0.14493	33	5	0.14085

	MAPK1			MAPK3			NGFR			RAF1		
	CH2	TP	Fscore	CH2	TP	Fscore	CH2	TP	Fscore	CH2	TP	Fscore
GeneNet	50	18	0.21557	6	1	0.030769	23	8	0.13115	10	0	0
0	11	4	0.0625	17	2	0.052632	7	2	0.037736	19	1	0.025
0.25	103	34	0.30909	111	22	0.25882	11	2	0.036364	56	13	0.22222
0.5	131	47	0.37903	134	26	0.26943	11	2	0.036364	60	14	0.2314
0.75	131	47	0.37903	134	26	0.26943	11	2	0.036364	60	14	0.2314
0.95	131	47	0.37903	134	26	0.26943	11	2	0.036364	60	14	0.2314
1	58	22	0.25143	48	11	0.20561	1	0	0	29	9	0.2

Table D.3: Inference using colon cancer data expO data, number of true positives and F-score computed for childhood containing children and grandchildren (CH2).