UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Dipartimento di Scienze Statistiche
SCUOLA DI DOTTORATO DI RICERCA IN SCIENZE STATISTICHE
CICLO XXVII

Graphical modelling of biological pathways

Direttore della Scuola: Prof.ssa Monica Chiogna

Supervisore: Prof.ssa Monica Chiogna

Co-supervisore: Prof.ssa Chiara Romualdi

Dottoranda: Vera Djordjilović

31/01/2015

Acknowledgements

I would like to thank my supervisor, Professor Monica Chiogna, for her full support and skilful guidance throughout this PhD project. Her patience, encouragement and advice followed me in the past three years. I cannot imagine anyone doing a better job of enabling independence and recognizing when more guidance is needed. I thank my co-supervisor, Professor Chiara Romualdi, for providing valuable insights from the biological perspective, making this interdisciplinary work possible. I thank Sofia Massa, who not only introduced me to the beautiful world of the Oxford University, but also provided valuable suggestions that changed the course of my thesis. I thank Milan Studený and Jirka Vomlel, who kindly welcomed me at their research Institute in Prague and showed interest in my research problem. The substantial part of the thesis, dealing with structure learning of graphical models, stemmed from that collaboration.

I thank my colleagues from the 27th PhD cycle, Mareg, Marco, Valentina, Gianluca, Leonardo and Paola. I feel incredibly lucky to have shared these three years of ups and downs with such an amazing group of people. I thank Ronaldo, Lorenzo, Giulio, Giovanni, my awesome and inspiring colleagues, for their support. I thank Jelena, Agi, Elvin, Mirko, Umberto, Leonardo and Claudio for their help and encouragement at various times during this project. I thank Ivan for helping me take the leap three years ago.

Finally, I thank my family for being my pillars of support.

Abstract

Biological pathways underlie the basic functions of a living cell. They are complex diagrams featuring genes, proteins and other small molecules, showing how they work together to achieve a particular biological effect. From a technical point of view, they are networks represented through a graph where genes and their connections are, respectively, nodes and edges of a graph.

The main research objective of this thesis is to develop a framework for simulating effects of gene silencing. To this end, we propose a three step approach. First, we refine the structure of a pathway via our CK2 algorithm. Next, we assess the uncertainty in the refined structure. Finally, we simulate gene silencing through intervention analysis in causal graphical models. The proposed approach showed promising results when applied to the problem of predicting the effect of the knockdown of the *nkd* gene in *Drosophila Melanogaster*.

Abstract

I pathway biologici sono alla base del funzionamento delle cellule viventi. Tali pathway sono diagrammi complessi che coinvolgono geni, proteine e altre piccole molecole, mostrando come essi svolgano un ruolo congiunto nel raggiungimento di uno specifico effetto biologico. Da un punto di vista tecnico, questi network sono rappresentati mediante diagrammi dove i geni e le loro connessioni sono, rispettivamente, nodi e archi.

Il principale obiettivo di questa ricerca è sviluppare una tecnica per simulare gli effetti del silenziamento genico. A tal fine, proponiamo un approccio basato su tre passi. Nel primo passo, raffiniamo la struttura di un pathway attraverso il nostro algoritmo CK2. In seguito, nel secondo passo, valutiamo l'incertezza nella struttura raffinata. Infine, nel terzo passo, simuliamo il silenziamento genico tramite intervention analysis nei modelli grafici causali. L'approccio proposto mostra risultati promettenti se applicato al problema della previsione dell'effetto del silenziamento del gene *nkd* della *Drosophila Melanogaster*.

Table of contents

Table of contents ix

List of figures xi

List of tables xiii

1 Introduction 1

1.1 Overview 1

1.2 Main contributions of the thesis 3

2 The motivating problem 5

2.1 Gene silencing 5

2.2 Biological pathways 6

2.3 From pathways to graphical models 8

2.4 Uncertainty in the graphical structure 9

3 The data 11

3.1 Protocol 12

3.2 Exploratory analysis 13

3.3 A note about technical variability 17

4 The statistical background 21

4.1 Conditional independence and graphs 21

4.1.1 Markov properties on undirected graphs 22

4.1.2 Markov properties on directed acyclic graphs 24

4.2 Appendix: Lexicon and notation 26

5 Guided learning in graphical models 29

5.1 Guided structural learning 30

5.1.1 Background on the K2 algorithm 30

5.1.2 The CK2 algorithm 32
5.1.3 Notes and observations 34
5.2 Empirical comparison with alternative learning strategies 36
5.2.1 Categorization of expression measurements 36
5.2.2 Evaluation of predictive accuracy 38
5.2.3 Learning algorithms 39
5.2.4 Results 42
5.2.5 Conclusions 43
5.3 Model uncertainty 48
5.4 Shrinkage 51
5.4.1 Background 52
5.4.2 The proposal 53
5.4.3 Simulation studies 56
5.4.4 Discussion 61

6 Intervention analysis 63
6.1 Intervention calculus 64
6.2 Application to real data 67
6.3 Notes and observations 72

7 Conclusions 75

References 77

List of figures

2.1 Prostate cancer: pathway taken from the Kyoto Encyclopedia of Genes and Genomes (KEGG). 7

3.1 WNT pathway with measured genes indicated in red. 14
3.2 *Drosophila Melanogaster* experiment: Mean expression values of the twelve genes in the knockdown and control group. 15
3.3 *Drosophila Melanogaster* experiment: Estimated densities for the twelve genes in the knockdown (red) and the control group (blue). 16
3.4 *Drosophila Melanogaster* experiment: Coefficients of biological (left) and technical variation (right) of the twelve genes in the knockdown (red) and the control group (blue). 18

4.1 An example of DAG 25
4.2 Non uniqueness of conditional independence properties entailed by a DAG 25
4.3 Paths and cycles. A sequence (X_1, X_2, X_3, X_4) is a partially directed path. A sequence (X_3, X_4, X_1, X_2) is a directed path. (X_1, X_2, X_3, X_1) is a 3-cycle. An example of a chain is sequence (X_1, X_2, X_4). 27
4.4 V-structure. 28

5.1 B score as a function of the number of edges in the DAG. 45
5.2 DAGs used in the first simulation study (ordered clockwise from the upper left corner): the true structure used to generate data, the first misspecified structure with the added edges shown in blue, the second misspecified structure with the deleted edges shown in red, the third misspecified structure corresponding to a randomly generated DAG. . . 57
5.3 DAG representing the B cell pathway. 62

6.1 The pathway DAG (left) and the refined DAG obtained by CK2 (right). 68

6.2 The consensus DAG. 69
6.3 The predictions based on the consensus DAG. We mark as successes the genes in which the hypothesis of equality of the predicted mean after intervention and the mean in the knockdown condition is not rejected. 70
6.4 The predictions of the mean expression values after the silencing of the *nkd* gene based on the pathway DAG (left) and the refined DAG obtained by CK2 (right). We mark as successes the genes in which the hypothesis of equality of the predicted mean after intervention and the mean in the knockout condition is not rejected. 73
6.5 The bootstrap distribution of silencing effects on *daam*. 73

List of tables

3.1 Sequence primers. 12

3.2 Selected genes involved in WNT pathway. 13

3.3 *Drosophila Melanogaster* experiment: Estimated mean expression values ($\hat{\mu}$) of the twelve genes in the knockdown and control group and respective biological ($\hat{\sigma}_\mu$) and technical ($\hat{\sigma}$) standard deviations. 19

3.4 *Drosophila Melanogaster* experiment: p-values of tests of equality of means in the knockdown and control group. 20

5.1 Evaluation of the prediction accuracy: the B Score. 46

5.2 The (B) score for ten randomly generated orderings passed to K2. . . . 47

5.3 Average size of the Markov Blanket for different algorithms. 47

5.4 Root mean square error (and standard deviation) of different covariance estimators; correctly specified structure 58

5.5 Root mean square error (and standard deviation) of different covariance estimators: the first misspecified structure (additional edges). 59

5.6 Root mean square error (and standard deviation) of different covariance estimators: the second misspecified structure (missing edges). 59

5.7 Root mean square error (and standard deviation) of different covariance estimators: the third misspecified structure. 60

5.8 The B cell pathway model: root mean square error (and standard deviation) of different covariance estimators, multiplied by 10^2. 60

6.1 *Drosophila melanogaster* experiment: p-values of tests of the equality of predicted means and means in the knockdown condition. 71

6.2 Summary of the bootstrap distribution of silencing effects: the number of bootstrap samples in which the estimated model implied no effect of *nkd* silencing, the confidence interval for the mean after silencing, and for reference, estimated means of the genes in the knockdown group $\bar{Y}_k$. 72

Chapter 1

Introduction

1.1 Overview

Molecular pathways underlie the basic functions of a living cell. They are elaborate diagrams featuring genes, proteins and other small molecules, showing how they work together to achieve a particular biological effect. From a technical point of view, they are networks with explicit biological interactions and can be represented through a graph where genes and their connections are, respectively, nodes and edges. Pathways are often identified piecemeal over extended periods of time, by a variety of researchers, and stored in public databases such as KEGG (Ogata et al., 1999) or Biocarta (Nishimura, 2001).

One of the key questions pertaining to pathways is the importance of individual participating genes. What happens if one gene is switched off? To answer this question, scientists perform experiments called gene silencing, in which an expression of a particular gene is forced to a minimal non-lethal level. Although a gold standard in functional genomics, this technique is still not in widespread use due to the high costs both in terms of financial and time resources. The need to find a theoretical surrogate to gene silencing is the motivation of this work.

In this thesis, we argue that a theoretical surrogate might be found by relying on proper statistical modelling. The statistical components of such an approach should comprise a statistical model for the biological pathway (including the gene to be silenced) and data on expression levels of genes appearing in the pathway. The issue of quality of such data is, obviously, crucial, as reliability of results highly depends on good estimation of the model. Having these ingredients, our solution boils down to modelling biological pathways by means of graphical models (Lauritzen, 1996), and simulating gene silencing as external interventions in directed graphical models.

In developing such an approach, we faced numerous problems, of different nature and relevance with respect to the main aim of the work. One of the first problems that we experienced is that graphical models derived from pathways are often poorly supported by experimental data. This finding was confirmed by an extensive exploratory analysis of real data that led us to propose a new shrinkage estimator of a covariance matrix for a given graphical model. To improve the graphical model representation, we followed the route of refining the the pathway information in the light of data. The prediction of the effects of silencing through intervention analysis needed therefore to take into account the uncertainty related to the refinement, which we did by resorting to resampling strategies.

The resulting work appears to be a collection of seemingly unrelated proposals, but they all bloomed with the purpose of finding a suitable statistical framework to perform gene silencing. The reader might, at times, feel confused or even lost as to how different pieces fit together and how they contribute to the big picture. It might be reassuring for the reader to know that the same feelings were author's loyal companions throughout the development of this project. The overwhelming complexity of the underlying biological problem calls for careful and time requiring interdisciplinary work, and this thesis, hopefully, makes the first steps of that journey. A discussion of some of the issues to be tackled in the future is given in Chapter 7.

The outline of the thesis is as follows. Chapter 2 covers the biological background of our motivating problem: gene silencing. In Chapter 3, we give a description of the key experiment used throughout this thesis: the *nkd* gene silencing in fruit flies. A brief introduction to graphical models, for readers not familiar with the topic, is given in Chapter 4. Chapter 5 covers the proposed solutions to various problems that we faced. They can all be framed within the big topic of learning in graphical models. First, in Section 5.1, we introduce a new algorithm, that we call CK2, for refining the graphical structure of a pathway. In Section 5.2, we empirically compare CK2 to a number of different structure learning algorithms, and offer guidelines as to when a particular approach might be preferred over its alternatives. Next, we propose a method for evaluating uncertainty in the refined structure in Section 5.3. Finally, we cover the guided penalized estimation of the covariance matrix in a Gaussian graphical model by proposing a novel shrinkage estimator in Section 5.4. In Chapter 6, we turn our attention to simulating gene silencing through intervention analysis. We first recall general notions about causal models, and then focus on the estimation of the effects of silencing. In Section 6.2, we apply our approach to the data from the fruit fly experiment. We demonstrate how different parts described in the thesis come together in the task of simulating effects of *nkd* gene silencing. Chapter 7 contains main conclusions drawn from this project up to date and possible directions for future research.

1.2 Main contributions of the thesis

Main contributions of the thesis can be summarized as follows.

1. Definition of a partially supervised learning algorithm of directed acyclic graphs, CK2, applicable in situations when some prior information pertaining to the topology of the graph is available. An application to the refinement of the existing graphical structures obtained from molecular pathways is provided.

2. Comparative study of predictive accuracy of different structure learning algorithms applied to gene expression data. Proposal of a data driven categorization of the expression measurements.

3. Definition of a consensus DAG.

4. Introduction of a new penalized approach for the estimation of the covariance matrix in the Gaussian graphical models in "$p > n$" setting. Numerical evaluation of the proposed estimator.

5. Definition of a three step procedure for the estimation of effects of an intervention when prior information on the ordering of variables is available. Application to the gene silencing experiments.

6. Application and validation of the novel approach on real data from *Drosophila Melanogaster* silencing of gene *nkd*. To this aim, a tailored experiment was performed and a new dataset created.

Chapter 2

The motivating problem

2.1 Gene silencing

Nearly a decade ago, Craig Mello and Andrew Fire were honored the Nobel prize for their discoveries related to gene silencing, a process that allows cells to selectively turn off specific genes. Research in this area jumpstarted a new biological field, termed RNA interference, by opening up previously inaccessible areas of research. Today, scientists routinely use this powerful method to study the functions of specific genes and gene silencing is being successfully used as a tool for functional genomics. One of the most exciting applications of such methodology is in biomedical research. Scientists are using manipulation of genes to study the progress of thousands of genetically based diseases at the molecular level. The hope is that by better understanding how a certain gene contributes to a particular disease, researchers can then take the knowledge a step further and look for drugs that act on that gene. Another essential application is in drug development. The silencing technology may lead to the discovery of the next generation of blockbuster therapies for curing numerous diseases based on novel targets from the human genome.

Although gene silencing is highly advantageous for both biomedical research and drug development, it also contains a number of limitations, some of which related to technical aspects and some to the costs of the experiments. Recent years have witnessed constantly growing efforts for producing technologies that break down the costs and provide high quality results. Clearly, in the everyday lab practice, if potential effects of silencing could be investigated before physically performing the experiment, this could enable a more efficient design and organization of the experiment leading to considerable savings in terms of time and money.

To be able to simulate or predict effects of gene silencing, a model describing well the relationships between genes is essential. Technological advances seen in the ultimate two decades, related to high throughput analysis, resulted in a vast amount of data that are used in an attempt to elucidate the mechanisms underlying the complex interplay of different genes. Some of that information is stored in the form of diagrams of biological pathways. In other words, pathway diagrams capture (a part of) our knowledge about the interactions between genes (and proteins and other metabolites) and for our purposes, they provide valuable information that is used in addition to the gene expression data to build a model for the system of considered genes.

2.2 Biological pathways

Biological pathways can be described as sets of linked biological components interacting with each other over time to generate a single biological effect, such as a change in enzyme activity, a change in gene expression or a change in ion channel activity. A number of diseases are associated with defects in these pathways, motivating a growing body of research that aims to deepen our understanding. In fact, pathways are often identified piecemeal over extended periods of time and by a variety of researchers. Figure 2.1 represents an example of such pathways, the Prostate cancer pathway taken from the Kyoto Encyclopedia of Genes and Genomes (KEGG). It is composed by edges and nodes, which have the following meanings. Rectangles represent gene products, mostly proteins, but also RNA and complexes. The edges between rectangles represent functional interactions. They can be both directed and undirected. Circles are other types of molecules, mostly chemical compounds, while the large white rectangles are the links to other pathways. We note that in the thesis we will consider only signalling pathways; the interpretation of the diagram might be different for metabolic pathways, and thus the methods that we consider would have to be adapted accordingly.

A vast variety of databases containing information such as that shown in Figure 2.1 exist. They present biological interactions in a graphical format comparable to the representation present in text books, as well as in standard formats allowing the exchange between different software platforms and further processing by network analysis, visualization and modeling tools. The Pathguide resource serves as a good overview of current pathway databases (Bader et al., 2006). It lists more than 200 pathway repositories; over 60 of those are specialized on reactions of the human species. However, only half of them provide pathways and reactions in computer-readable formats needed for automatic retrieval and processing. Pathway annotations comprise

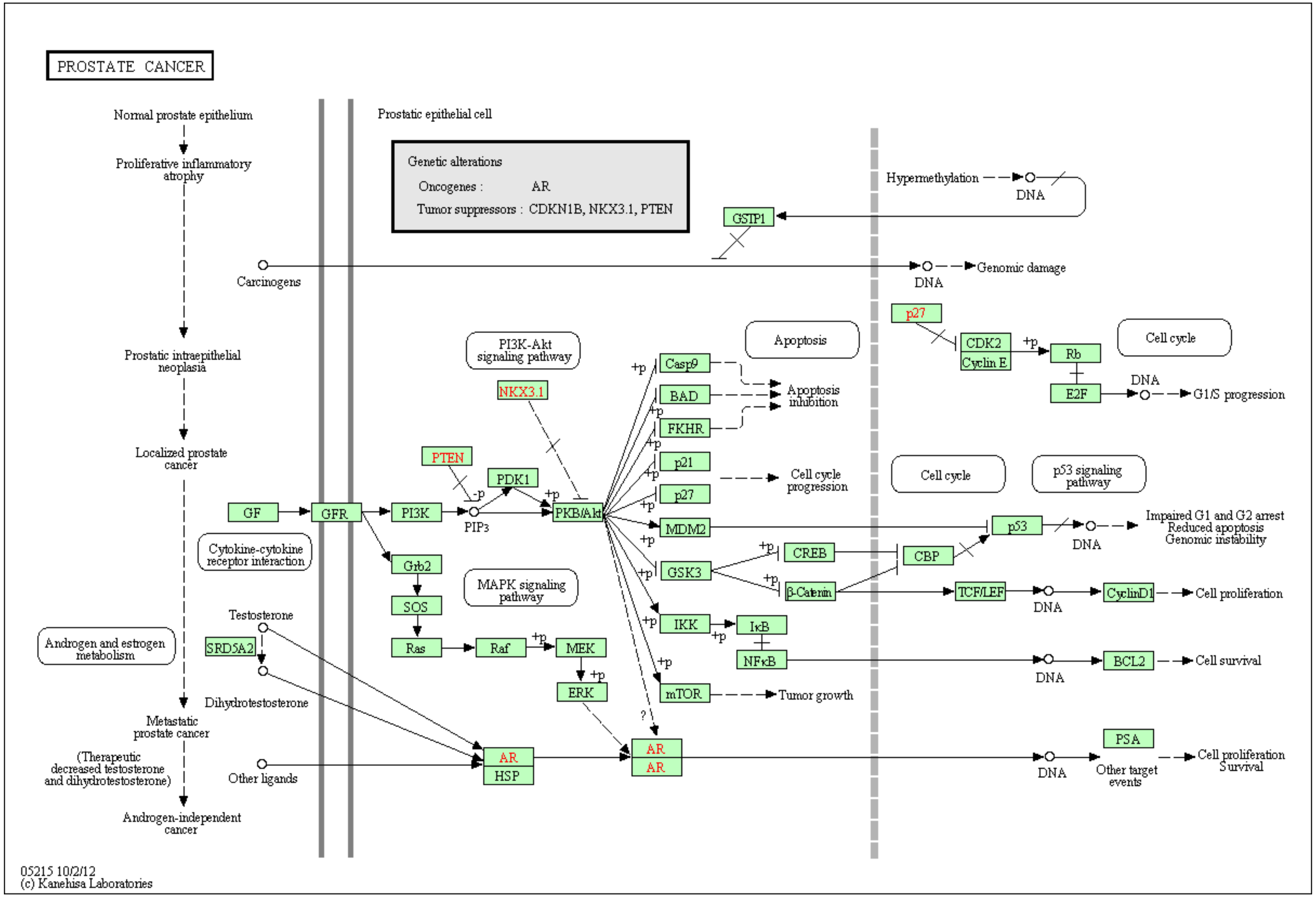

Fig. 2.1 Prostate cancer: pathway taken from the Kyoto Encyclopedia of Genes and Genomes (KEGG).

a myriad of interactions, reactions, and regulations which is often too rich for the conversion to a graph. In particular, challenges are posed by the presence of chemical compounds mediating interactions and by different types of gene groups (e.g. protein complexes or gene families) that are usually represented as single nodes. Moreover, different databases are characterized by different annotations and only a part of the whole set of reactions are confirmed by all the repositories.

Among the widely used databases, we find Biocarta (Nishimura, 2001), KEGG (Kanehisa and Goto, 2000), NCI/Nature Pathway Interaction Database (Schaefer et al., 2009) and Reactome, (Joshi-Tope et al., 2005; Vastrik et al., 2007). Reactome (Vastrik et al., 2007), backed by the European Bioinformatics Institute (EBI), is one of the most complete repositories; it is frequently updated and provides a semantically rich description of each pathway. KEGG Pathways (Kanehisa and Goto, 2000) provides maps for both signaling and metabolic pathways, supplemented by 19 highly interconnected databases with genomic, chemical and phenotypic information. BioCarta (www.biocarta.com) and NCI (Schaefer et al., 2009) are available from the NCI/Nature Pathway Interaction Database database web page.

2.3 From pathways to graphical models

Initially, models translating gene regulatory networks into mathematical structures were deterministic, most prominent examples include systems of differential equations (Glass and Kauffman, 1973) and Boolean models (Thomas, 1991). Deterministic models, built on detailed biological knowledge, have proven useful for testing existing theories about biological systems. The last two decades saw a significant change of course: the rapid development of novel technologies triggered interest in approaches that would enable one to go beyond testing existing theories. One would like to make the full use of the newly available experimental data, and to formulate new testable theories. This notion shifted attention towards probabilistic models.

Graphical models seem especially suitable for the task and Friedman et al. (2000) first introduced the idea of using directed acyclic graphs (DAGs) for the purpose of modelling gene networks. Embracing a graphical models paradigm, in addition to capturing inherent variability present in biological systems, enables one to learn models from experimental data and to formulate new hypotheses about relations between genes.

In order to incorporate pathways into graphical models, a descriptive diagram needs to be translated into a mathematical graph, either directed or undirected. The choice of the type of graph depends on the goal of the statistical study. Due to the descriptive nature of pathways and their inherent complexity, there is no simple recipe for conversion that can be applied in every situation. For this reason, close collaboration with biologists is preferred at this step (Djordjilović et al., 2013). Here, we address some most common issues and for possible solutions we refer the reader to Chapter 7.

First of all, gene expression experiments rarely measure expression values for all genes in a given pathway. Therefore, a preliminary action is to construct a subpathway obtained by the intersection of the nodes in the pathway and the genes in the dataset. This suggests that the topological properties of the obtained graph can be considered dataset-specific, and that links in the subpathway might lofse the functional meaning characterizing the links in the pathway. Secondly, biological pathways may contain cycles. This presents a problem if a pathway is to be converted to a directed graph, since almost all approaches based on directed graphs do not allow loops. Finally, an additional difficulty is represented by compounds and complexes. Compounds are not measured in the microarray experiment, and should be removed. As for complexes, some are protein complexes (proteins linked by protein-protein interactions) and some contain alternative members (gene families, genes that share similar biological functions). They need to solved on a case to case basis.

2.4 Uncertainty in the graphical structure

Although pathways represent our up-to-date knowledge of the cellular processes, we can not always assume that the obtained mathematical graph will be the optimal structure of a graphical model. There are a number of reasons to consider this graph a tentative model and we describe some of them here.

First of all, most of the interactions featured in pathways are interactions between proteins rather than genes. It has been a common practice so far to assume that mRNA levels determine with high accuracy the cellular protein abundance. Researchers would use gene expression level as a proxy for the protein level, tacitly assuming the linear relationship between the two. However, recent technological advances have allowed for measuring protein abundance directly. Experiments that investigated both the mRNA and protein levels showed that the relationship between the two is less strong than anticipated. In other words, the role of post-transcriptional mechanisms involved in turning a mRNA into a protein is substantial (Vogel and Marcotte, 2012).

Secondly, pathway representation, although detailed, is by no means exhaustive; in part because there are still many undiscovered features and in part because the most realistic model is usually too cumbersome to be useful. Every graphical representation of a signaling pathway can be seen as a compromise between accuracy and complexity. Furthermore, as already stated, pathways represent joint work of the global scientific community, and most of the information is obtained experimentally, which implies that the possibility of errors cannot be excluded.

Lastly, it should be kept in mind that signalling pathways represent the transduction of a signal; signals that coordinate cell actions. They are, thus, inherently dynamic systems and by measuring gene expression levels at a single time point, we obtain a snapshot of a cell profile, neglecting the time aspect.

Chapter 3

The data

Within the scientific method, the experiments are the gold standard for studying causal relationships. Not surprisingly, experiments are also the method of choice when testing approaches related to causal inference. In particular, in order to evaluate an approach that aims to predict effects of gene silencing, one would compare the model based predictions against the "truth", represented by the silencing experiment. The problem is that gene silencing or knockdown experiments are costly and involve a lot of time and effort. For that reason, this type of data is not easily available. In order to allow us to test our approach to predicting the effects of gene silencing, the Department of Biology of the University of Padova performed one such experiment. They silenced the naked cuticle gene (*nkd*) in a fruit fly (*Drosophila Melanogaster*).

The data consist of expression levels of 15 genes measured in the treatment (knock-down of the *nkd*) and the control (sometimes also referred to as *wild type*) group. The number of observations in both groups is 15. The experiment has been performed so that both a measure of technical and biological variability in gene expression are obtained. To measure technical variability, in both groups there were 5 different cell lines and 3 measurements per each cell line. Given that within a cell line there are no biological differences, differences in gene expression can be attributed to technical artifacts.

The common concern regarding expression data is that they are inherently noisy. We note that compared to the microarray sequencing, the technique used in this experiment, the real time PCR, is more specific and precise, and is often considered a gold standard for detection and quantification of the gene expression.

The technical details concerning the experiment are given in the experimental protocol below, while in Section 3.2, we take a look at some initial exploratory analysis.

3.1 Protocol

Cell cultures The *Drosophila* S2R+ cell line were derived from a primary culture of late stage (20-24 h old) *D. melanogaster* embryos (Schneider, 1972), obtained from *Drosophila* Genomics Resource Center (DGRC) (http://dgrc.cgb.indiana.edu/). S2R+ cells grow at 25°C without CO_2 in Schneider's medium (Life Technologies) with 10% heat-inactivated fetal bovine serum (FBS) (Sigma-Aldrich) as a loose, semi-adherent monolayer, showing a doubling time of about 48 h.

dsRNA production and RNAi procedures

dsRNAi synthesis was performed employing the T7 Megascript kit (Life Technologies) (Flockhart et al., 2012; Ni et al., 2009). The oligonucleotides primers used to synthesize dsRNA starting from cDNA were nkd_T7 forward (F) and reverse (R) (primer sequences are reported in Table 3.1). These primers give two complementary 650 bp RNA products that anneal as temperature decreases, forming a final 650 bp dsRNA. About 2×10^6 cells suspended in 1 ml of serum-free medium were mixed with 2 µg/ml dsRNA, plated in a 24 wells plate and incubated at room temperature (RT) for 1 h. Subsequently, one volume of complete medium 2X was added and cells were grown in the presence of dsRNA for 4 days at 25 °C.

Table 3.1 Sequence primers.

Gene	Sequence
nkd_T7-F	5'-TTAATACGACTCACTATAGGGAGATGT ACAAGCACGGCAAATACTCAA-3'
nkd_T7-R	5'-TTAATACGACTCACTATAGGGAGATGT ATTTTCGCTGTTGCTGTCATC-3'
nkd-F	5'-ACCCGAACCATCAAATGC-3'
nkd-R	5'-GTTTCGAGGCAGTGGTCCT-3'
Rp49-F	5'-TCGGTTACGGATCGAACAA-3'
Rp49-R	5'-GACAATCTCCTTGCGCTTCT-3'

RNA isolation and qRT-PCR experiments to determine Nkd silencing

Total RNA was extracted from approximately 2×10^6 cells using Trizol (Life Technologies). RNA samples were checked for integrity by capillary electrophoresis (RNA 6000 Nano LabChip, Agilent Technologies). For each sample, 1 µg of RNA was used for first-strand cDNA synthesis, employing 10 mM deoxynucleotides, 10 µM oligo-dT and SuperScript II (Life Technologies). qRT-PCRs were performed in triplicate in a 7500 Real-Time PCR System (Life Technologies) using SYBER Green chemistry (Promega). The $2^{-\Delta Ct}$ (RQ, relative quantification) method implemented in the 7500

Real Time PCR System software was used to calculate the relative expression ratio (Livak and Schmittgen, 2001). The nkd oligonucleotides primer used were nkd F and R. Rp49 was used as endogenous control and the oligonucleotides employed were Rp49 F and R.

RNA isolation and qRT-PCR experiments to determine Nkd silencing
Experiments were performed using the TaqMan probes for selected genes (Table 2 and Figure 1) in the 7500 Real-Time PCR System (Applied Biosystems). PCR were performed in triplicate for each sample (five samples) for a total of 15 data point for each gene. Ribosomal protein L32 (Rpl32) and RNA polymerase II 140kD subunit (RpII 140) were used as reference genes. After the RNA retrotrascription using a mix of oligod(T) and random primers 10 ng of cDNA were used in each PCR amplification. The original expression level for each gene in each well was expressed as $2^{-\Delta Ct}$ where ΔCt indicates $Ct($ gene of interest$) - Ct($average of reference gene$)$.

Table 3.2 Selected genes involved in WNT pathway.

Gene	Description
arr or CG5912	low density lipoprotein receptor-related protein 5/6
pont or CG4003	RuvB-like protein 1 (pontin 52)
nkd or CG11614	naked cuticle
daam or CG14622	Dishevelled Associated Activator of Morphogenesis
dco or CG2048	discs overgrown or casein kinase 1, epsilon
fz3 or CG16785	frizzled 3
dally or CG4974	division abnormally delayed
sgg or CG2621	shaggy or glycogen synthase kinase 3 beta
arm or CG11579	armadillo or catenin beta 1
psn or CG18803	presenilin 1
rho1 or CG8416	Ras homolog gene family, member A
rok or CG9774	rho-associated kinase
por or CG6205	porcupine
dsh or CG18361	segment polarity protein dishevelled
wg or CG4889	wingless-type MMTV integration site family, member 1

3.2 Exploratory analysis

After preliminary examination of the data, the expressions of the three genes, *arr*, *dsh* and *wg*, were deemed too low by biologists (signaling that the corresponding genes were not expressed in neither condition), and thus they are excluded from further

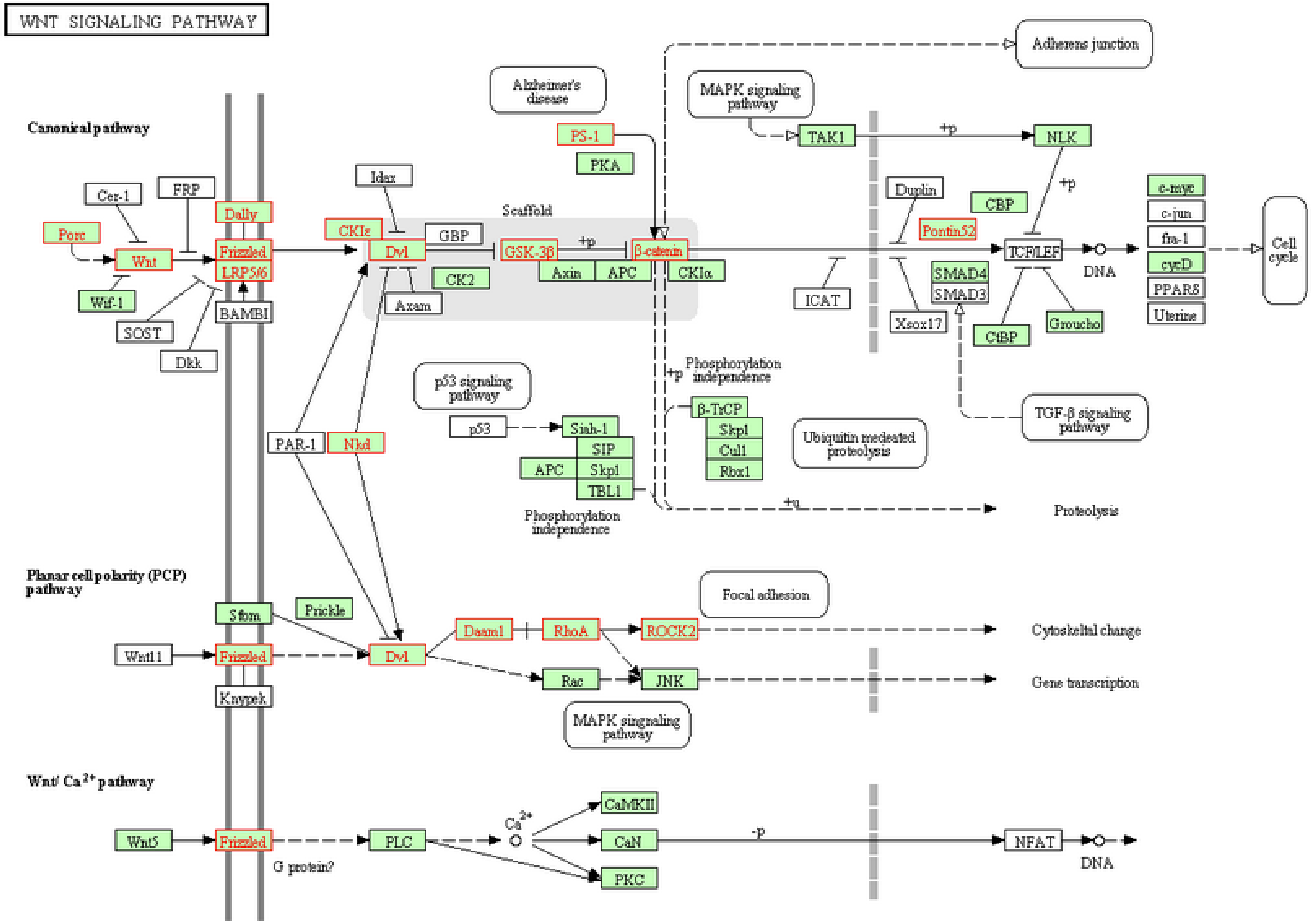

Fig. 3.1 WNT pathway with measured genes indicated in red.

analysis.

There were two missing values in our dataset, one in each group. Both missing values belonged to the silenced gene, *nkd*. Since missing values greatly complicate further analysis, and in addition, the *nkd* gene has a prominent role in the experiment, we excluded the incomplete observation from each group. That left us with $n = 14$ observations of $p = 12$ genes in two experimental conditions, and this is the dataset that will be considered in the remainder of the thesis.

Figure 3.2 shows if and how the mean expression values of considered genes change in response to the silencing of the *nkd*. We see that the change in mean is most visible in the *dally* gene, whose expression is higher in the knockdown group. We discuss this observation in greater detail in Chapter 6. The remaining genes either change slightly or barely so (*arm*, *por*, *dco*), at least in terms of the mean value. To

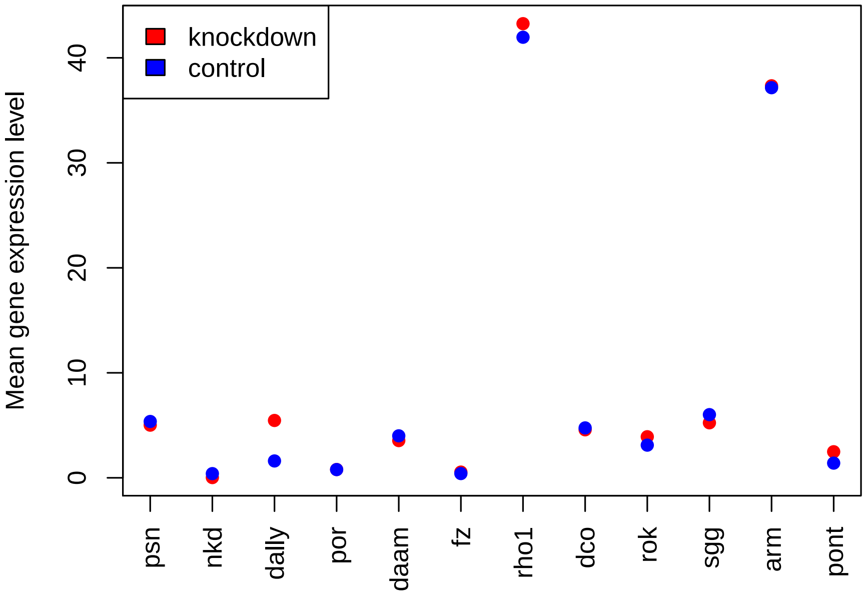

Fig. 3.2 *Drosophila Melanogaster* experiment: Mean expression values of the twelve genes in the knockdown and control group.

see the effect of silencing on the whole distribution, we can have a look at Figure 3.3. It shows estimated densities for the knockdown and the control group for each gene. The plot corresponding to the silenced gene, *nkd*, provides a clear visual idea of how the intervention shifted its mean towards zero and drastically reduced its

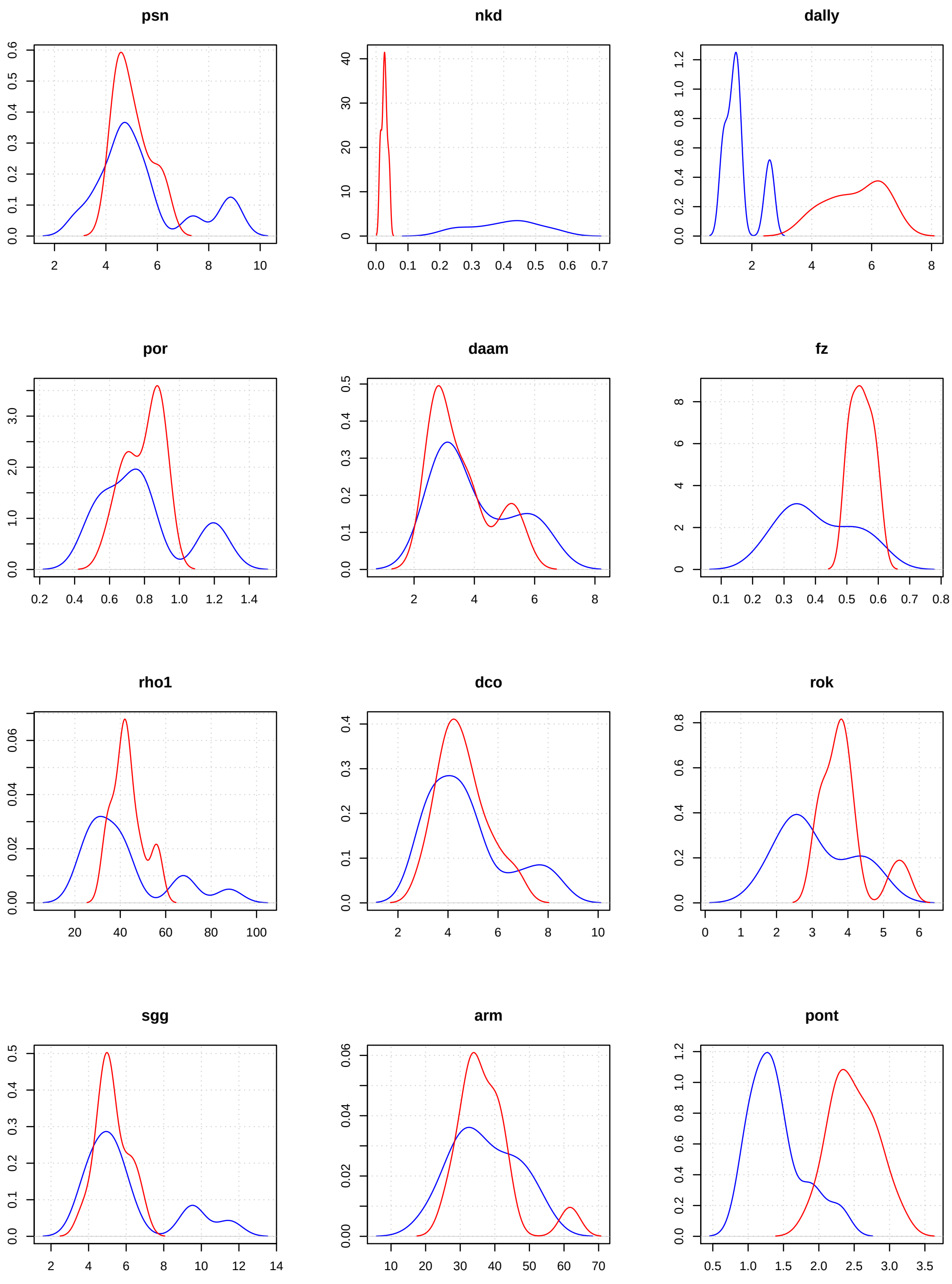

Fig. 3.3 *Drosophila Melanogaster* experiment: Estimated densities for the twelve genes in the knockdown (red) and the control group (blue).

variance. Another interesting observation is that genes *dally*, *fz* and *pont* have very well separated distributions in the knockdown and control group, suggesting they are closely related to, or perhaps regulated by *nkd*.

3.3 A note about technical variability

As mentioned in the introductory section, the design of the experiment was such to allow for estimation of the technical variability. The observations in both the control and the knockdown group consist of five groups of two or three measurements on one sample. To model gene expression measurements obtained in this way, we can employ a random effects model that we specify so as to be as close as possible to the technical realization of the experiment. To illustrate this notion, we restrict our attention to one generic gene in an arbitrary, but fixed, treatment group (control or knockdown). With the treatment group being fixed, *groups* in the following refer to the groups of replicate measurements of the same biological sample. We model the underlying data generating process as a two step procedure: in the first step, a mean value of each of the five samples is drawn from a normal distribution whose variance corresponds to the biological variability; in the second step, the additional, independent noise is added to these group means. Let X_{ij} denote the jth replicate in the ith group of that particular gene in our dataset, where $i = 1, \ldots, I$ and $j = 1, \ldots, n_i$, where n_i is the number of replicate measurements of the ith sample (2 or 3 according to i), and $I = 5$ is the number of samples. If we adopt notation $\sim (\mu, \sigma^2)$ to denote a random variable with mean μ and variance σ^2, we set

$$\begin{array}{rcl} \mu_i & \sim & (\mu, \sigma_\mu^2) \\ X_{ij} & = & \mu_i + \epsilon_{ij}, \quad \epsilon_{ij} \sim (0, \sigma^2); \quad \mu_i \perp\!\!\!\perp \epsilon_{ij}, \quad i = 1, \ldots, I; j = 1, \ldots, n_i, \end{array}$$

where μ_i is the mean of the ith group, while μ, the mean expression, σ_μ^2, the biological variability, and σ^2, the technical variability, are model parameters.

An unbiased estimate of μ is the sample mean $\bar{X} = \sum_{i=1}^{I} \sum_{j=1}^{n_i} X_{ij}/n$. The estimates of the two variance parameters are usually found by the moment method, by equating the mean squares in the analysis of variance to the means of their sampling distributions. The well-known decomposition of the total residual variance is

$$\sum_{i=1}^{I} \sum_{j=1}^{n_i} \left(X_{ij} - \bar{X} \right)^2 = \sum_{i=1}^{I} \sum_{j=1}^{n_i} \left(X_{ij} - \bar{X}_i \right)^2 + \sum_{i=1}^{I} n_i \left(\bar{X}_i - \bar{X} \right)^2,$$

where $\bar{X}_i$ is the mean of the ith group. The terms on the right hand side are usually referred to as *within group variance* (SSW) and *between groups variance* (SSB). The mean square of the within group variance is exactly the estimate of the technical variability σ^2

$$\hat{\sigma}^2 = \frac{SSW}{n-I},$$

where $n = \sum_{i=1}^{I} n_i$ is the total number of observations. As for the between group variance, its expected value is

$$\mathsf{E}\left[\sum_{i=1}^{I} n_i \left(\bar{X}_i - \bar{X}\right)^2\right] = \frac{n^2 - \sum_{i=i}^{I} n_i^2}{n}\sigma_\mu^2 + (I-1)\sigma^2.$$

An unbiased estimate of σ_μ^2 is, therefore

$$\hat{\sigma}_\mu^2 = \frac{n(I-1)}{n^2 - \sum_{i=1}^{I} n_i^2}\left(\frac{SSB}{I-1} - \frac{SSW}{n-I}\right).$$

The estimates of the mean and variances in the dataset are provided in Table 3.3. It shows estimated model parameters for each gene. Figure 3.4 shows estimated coefficients of biological and technical variation for each gene in the two treatment groups.

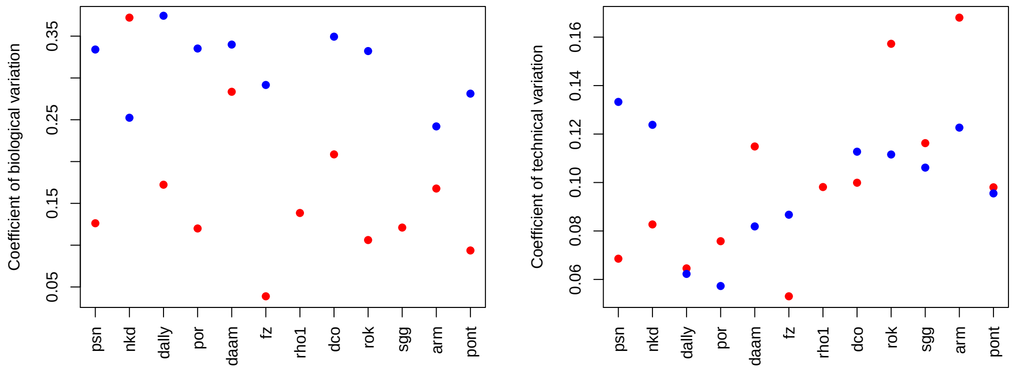

Fig. 3.4 *Drosophila Melanogaster* experiment: Coefficients of biological (left) and technical variation (right) of the twelve genes in the knockdown (red) and the control group (blue).

We note that the silencing of the *nkd* lowered the sample variance for all genes (except already mentioned *dally*). This might seem counter-intuitive, since the knockdown group was affected by an external intervention, and we might expect the intervention to introduce an additional source of variability. However, the cells responded by lowering the variability of (almost all) the remaining genes. We can also note that the mean level of expression is very different across genes: this is to be expected, since the expressions vary according to the varied biological functions of the respective genes.

Table 3.3 *Drosophila Melanogaster* experiment: Estimated mean expression values ($\hat{\mu}$) of the twelve genes in the knockdown and control group and respective biological ($\hat{\sigma}_\mu$) and technical ($\hat{\sigma}$) standard deviations.

	Knockdown			Control		
	$\hat{\mu}$	$\hat{\sigma}_\mu$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}_\mu$	$\hat{\sigma}$
psn	5.02	0.63	0.34	5.37	1.79	0.72
nkd	0.03	0.01	0.00	0.40	0.10	0.05
dally	5.46	0.94	0.35	1.61	0.60	0.10
por	0.79	0.10	0.06	0.79	0.26	0.05
daam	3.55	1.01	0.41	4.00	1.36	0.33
fz	0.55	0.02	0.03	0.41	0.12	0.04
rho1	43.26	5.99	4.24	41.96	19.13	7.51
dco	4.58	0.95	0.46	4.76	1.66	0.54
rok	3.91	0.42	0.62	3.11	1.03	0.35
sgg	5.24	0.63	0.61	6.02	2.50	0.64
arm	37.34	6.26	6.27	37.16	8.99	4.56
pont	2.49	0.23	0.24	1.40	0.39	0.13

We can use these estimates to test the hypothesis of equality of mean expression values in the knockdown and control group on the univariate level. Under the hypothesis of equality of the means and normality of the random quantities involved in the model, the test statistic will be asymptotically distributed as a standard normal variate. The p-values associated with these test statistics for each gene are given in Table 3.4. We can see that the silenced *nkd* gene, as expected, has the most significant p-value. The other genes that demonstrate significant difference in the mean between the two groups are *dally*, *fz* and *pont*.

Table 3.4 *Drosophila Melanogaster* experiment: p-values of tests of equality of means in the knockdown and control group.

	p-value
psn	0.35
nkd	2.4×10^{-15}
dally	3.4×10^{-14}
por	0.47
daam	0.28
fz	6.4×10^{-3}
rho1	0.45
dco	0.42
rok	0.07
sgg	0.26
arm	0.49
pont	4.2×10^{-07}

Chapter 4

The statistical background

This section is a review of the key concepts in graphical modelling. The main terminology and the notation are collected in the appendix.

4.1 Conditional independence and graphs

The link between graphs and statistical models is the concept of conditional independence whose properties were first studied in a formal fashion by Dawid (1979). The author showed that many concepts central to statistical inference, such as sufficiency or ancillarity, can be defined in terms of generalized conditional independence. Conditional independence is defined for random events, for σ-algebras of events and for random variables. Here, we limit our attention to random variables.

Definition 4.1.1 *We say that random variables X and Y are conditionally independent given Z and write $X \perp\!\!\!\perp Y \mid Z$ if and only if*

$$\mathsf{P}(X \in A, Y \in B \mid Z) = \mathsf{P}(X \in A \mid Z)\mathsf{P}(Y \in B \mid Z), \tag{4.1}$$

for any A and B measurable in the sample space of X and Y, respectively.

Equivalently, we can say that X and Y are conditionally independent given Z if and only if $\mathsf{P}(X \in A \mid Y, Z) = \mathsf{P}(X \in A \mid Z)$. This alternative definition has an intuitive interpretation: once we know the value of Z, the distribution of X does not further depend on the value of Y. Unconditional independence can be seen as a special case of the above definition for Z trivial.

When X, Y and Z are all discrete random variables the condition 4.1 simplifies to

$$\mathsf{P}(X = x, Y = y \mid Z = z) = \mathsf{P}(X = x \mid Z = z)\mathsf{P}(Y = y \mid Z = z),$$

where the equality holds for all z such that $\mathsf{P}(Z = z) > 0$. When all three variables are continuous we have

$$f(x, y \mid z) = f(x \mid z)f(y \mid z),$$

where the equality holds almost surely.

Pearl and Paz (1985) discovered the connection between the relation of conditional independence of random variables and a ternary relation defined on the sets of vertices of an undirected graph induced by a certain separation criterion. This led them to study conditional independence of variables with the help of graphs, where each variable is associated to a node, and edges of a graph describe the possibility of conditional dependence. Their work was largely motivated by the idea of probabilistic reasoning in expert systems (Pearl, 1988) and in fact, it is this application of graphical methods that has rendered graphical models popular in the artificial intelligence community from the late 1980s to this day.

In what follows, we will consider a collection of random variables $\{X_v, v \in V\}$, so that each random variable corresponds to a node of the graph $G = (V, E)$. We will assume that the cardinality of V is p, and when no ambiguity may arise, the same notation X_i will be used for both random variables and nodes of a graph.

4.1.1 Markov properties on undirected graphs

Here, we report briefly three Markov properties associated with undirected graphs; for a detailed treatment we refer the interested reader to Lauritzen (1996). A distribution of $\{X_v, v \in V\}$ is said to obey the

1. *pairwise Markov property* relative to G if for any par $X, Y \in V$ non-adjacent

$$X \perp\!\!\!\perp Y \mid V \setminus \{X, Y\};$$

2. *local Markov property* relative to G if for any vertex $X \in V$

$$X \perp\!\!\!\perp V \setminus \mathrm{cl}(X) \mid \mathrm{bd}(X);$$

3. *global Markov property* relative to G if for any triple (A, B, C) of disjoint subsets of V, such that C separates A from B

$$A \perp\!\!\!\perp B \mid C.$$

It is not difficult to show that the global Markov property implies a local Markov property, which in turn implies the pairwise Markov property. Markov properties are closely related to the factorization of the joint density. A density $f(x)$ is said to *factorize* according to G if for all complete subsets $a \subset V$ there exists non-negative functions ϕ_a that depend on x only through x_a, and there exists a product measure $\mu = \otimes_{\alpha \in V} \mu_\alpha$ on the sample space of X, such that X has a density f with respect to μ, where f is of the form

$$f(x) = \prod_{a \text{ complete}} \phi_a(x).$$

Since this factorization is not unique, one can without loss of generality assume that only cliques appear among sets a

$$f(x) = \prod_{c \in \mathcal{C}} \phi_c(x),$$

where $\mathcal{C}$ represents the set of cliques of G.

It can be shown that if f factorizes then it satisfies the global Markov property (and thus all other weaker Markov properties). In the case when $\{X_v, v \in V\}$ has a strictly positive and continuous density, it can be shown that all of these properties are equivalent.

The undirected graphical models for multivariate normal distribution are called *Gaussian graphical models.* Given an undirected graph G, the Gaussian graphical model for $\mathbf{X}$ assumes that $\mathbf{X}$ follows a multivariate normal distribution and further obeys conditional independence properties implied by the graph. Since in this case the density is continuous and strictly positive, the global, local and pairwise Markov property, as well as the factorization property are equivalent.

Conditional independence relations implied by the graph G are easily represented by parameters of the normal distribution, more precisely by the structure of the inverse of the variance matrix. To see this, consider the density of the normal distribution with mean vector $\boldsymbol{\mu}$ and concentration matrix $\mathbf{K}$

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\mathbf{K}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} k_{ij} (x_i - \mu_i)(x_j - \mu_j) \right\}, \tag{4.2}$$

where k_{ij} are elements of $\mathbf{K}$. From here, we can see that whenever k_{ij} is zero, the joint density factorizes into two components of which one contains x_i and the other x_j. In this case, according to the factorization criterion, the variables X_i and X_j are conditionally independent given the remaining ones. Therefore, a normally distributed random vector $\mathbf{X}$ obeys the pairwise Markov property with respect to G if and only if $k_{ij} = 0$ for all pairs (X_i, X_j) non-adjacent in G.

If we denote by $S^+(G)$ the set of all $p \times p$ symmetric positive definite matrices with null elements corresponding to the missing edges of G, we can define the family of Gaussian graphical models as

$$\mathcal{M}(G) = \left\{ \mathbf{X} \sim \mathrm{N}_p(\boldsymbol{\mu}, \mathbf{K}^{-1}) : \boldsymbol{\mu} \in \mathbb{R}^p, \mathbf{K} \in S^+(G) \right\}.$$

4.1.2 Markov properties on directed acyclic graphs

Directed acyclic graphs have a long history in statistics dating back to the work of Wright (1934) and his method of path coefficients. Statistical models based on DAGs (sometimes referred to as Bayesian networks) have proved useful in a number of interesting applications, including probabilistic expert systems, genetics, forensics, causal inference, and machine learning.

Although all three Markov properties defined for undirected graphs have their counterparts in directed acyclic graphs, the local directed Markov property has proved to be most useful. This is not limiting in practical applications, since the condition under which the three properties are equivalent (strictly positive joint density) is usually satisfied. The local Markov property on the directed acyclic graph states that X_v is conditionally independent of its non descendants given its parents

$$X_v \perp\!\!\!\perp \{\mathrm{nd}(X_v) \setminus \mathrm{pa}(X_v)\} \mid \mathrm{pa}(X_v), \quad v \in V.$$

This property is equivalent to the factorization of the joint density. We say that the joint density f factorizes with respect to the graph G if it can be written as a product of $|V|$ univariate conditional densities

$$f(x) = \prod_{v \in V} f\left[x_v \mid \mathrm{pa}(x_v)\right].$$

An instance of a DAG on five nodes is shown in Figure 4.1. If the joint density $f(x_1, \dots, x_5)$ factorizes with respect to this graph, then it can be written as

$$f(x_1, \dots, x_5) = f(x_1) f(x_2 \mid x_1) f(x_3 \mid x_1) f(x_4 \mid x_3) f(x_5 \mid x_3, x_4),$$

where $f(\cdot)$ denotes a generic probability function. Local Markov property applied to this graph gives the following conditional independence relations: $X_2 \perp\!\!\!\perp \{X_3, X_4, X_5\} \mid X_1$; $X_3 \perp\!\!\!\perp X_2 \mid X_1$; $X_4 \perp\!\!\!\perp \{X_1, X_2\} \mid X_3$ and $X_5 \perp\!\!\!\perp \{X_1, X_2\} \mid \{X_3, X_4\}$. Clearly,

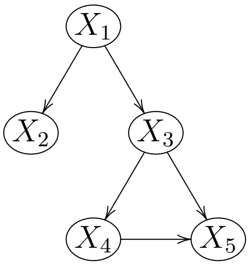

Fig. 4.1 An example of DAG

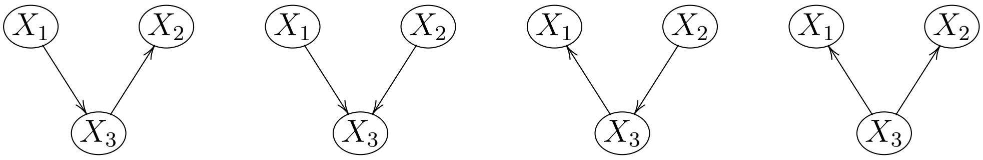

Fig. 4.2 Non uniqueness of conditional independence properties entailed by a DAG

these relations entail other conditional independence properties, such as $X_1 \perp\!\!\!\perp X_5 \mid X_3$ and $X_1 \perp\!\!\!\perp X_4 \mid X_3$. In order to obtain an exhaustive list of relations, one can apply the d-separation criterion, see Pearl (1988) or an alternative moralization criterion, see Lauritzen et al. (1990).

If density $f(x)$ of $\{X_v, v \in V\}$ is normal and factorizes according to a DAG G, we say that the $\{X_v, v \in V\}$ is a Gaussian Bayesian network. Since all conditional densities are themselves normal we have that, for every variable X_v, the conditional distribution given its parents, $\mathrm{pa}(x_v)$, is normal

$$X_v \mid \mathrm{pa}(X_v) \sim N(\alpha_v + \boldsymbol{\beta_v}^T \mathrm{pa}(X_v), \sigma_v^2), \qquad v \in V,$$

where σ_v^2 is the residual variance, independent of $\mathrm{pa}(X_v)$, and $\boldsymbol{\beta}_v$ is the vector of regression coefficients.

Every DAG determines a set of conditional independence relations among variables. It turns out that different graphs can lead to the same set of independencies. As a simple illustration consider a DAG on 3 nodes with two arrows. By applying the local Markov property on the first, second and fourth graph, we obtain $X_1 \perp\!\!\!\perp X_2 \mid X_3$. We say that these three graphs are equivalent from the probabilistic point of view. The term is adequate since it can be shown that DAGs entailing the same set of conditional independence properties form an equivalence class. The non uniqueness of the graphical representation has important practical consequences in statistical inference: in the problem of inferring the graphical structure from data, the underlying DAG is not identifiable; we cannot distinguish between DAGs in the same equivalence class.

Characterization of the graphs that determine the same set of conditional independencies has been proposed in Bonissone et al. (1991) and is reported in Theorem 4.1.1.

Theorem 4.1.1 **(Verma and Pearl, 1991.)** *Two DAGs defined over the same set of vertices induce the same set of conditional independence relations if and only if they have the same skeleton and the same set of V-structures.*

4.2 Appendix: Lexicon and notation

Graph. A *graph* G is a pair (V, E), where V is a finite set of nodes (or, equivalently, vertices) and $E \in V \times V$ is the set of edges. Edges can be undirected (both (X, Y) and (Y, X) are in E) and undirected (exactly one of the edges $\{(X, Y), (Y, X)\}$ is in E). If a graph has only undirected edges it is called an *undirected* graph, whereas if all the edges are directed it is called *directed*. A subset of graphs containing both directed and undirected edges, the so-called *chain* graphs, will be defined in the following.

Subgraph. If $A \subset V$ is a subset of a vertex set it induces a *subgraph* $G_A = (A, E_A)$, where $E_A \subset E$ is obtained from E so that only edges with both endpoints in A are kept. A graph is *complete* if all its edges are joined by an edge. A subset is complete if it induces a complete subgraph. A complete set that is maximal (with respect to $\subset$) is called a *clique*.

Neighbours of a node. If there is an edge between X and Y, they are said to be *neighbors* or *adjacent*. The set of neighbors of a vertex X is denoted $\mathrm{ne}(X)$. If there is neither edge nor arrow between X and Y they are said to be *non-adjacent*. If there is an arrow pointing from X to Y, then X is said to be a *parent* of Y and Y is said to be a *child* of X. The set of parents of Y is denoted $\mathrm{pa}(Y)$ and the set of children of X is denoted $\mathrm{ch}(Y)$.

The expression $\mathrm{pa}(A), \mathrm{ch}(A), \mathrm{ne}(A)$ denote the parents, children and neighbors of veritces in A that are not themselves elements of A. The *boundary* $\mathrm{bd}(A)$ of a subset

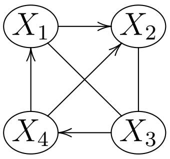

Fig. 4.3 Paths and cycles. A sequence (X_1, X_2, X_3, X_4) is a partially directed path. A sequence (X_3, X_4, X_1, X_2) is a directed path. (X_1, X_2, X_3, X_1) is a 3-cycle. An example of a chain is sequence (X_1, X_2, X_4).

A is a subset of $V \setminus A$ of parents and neighbors of vertices in A

$$\mathrm{bd}(A) = \mathrm{ne}(A) \cup \mathrm{pa}(A).$$

The *closure* of A is defined as $\mathrm{cl}(A) = A \cup \mathrm{bd}(A)$. We say that a set of nodes C *separates* sets A and B in an undirected graph G if every path from a node in A to a node in B contains at least one node from C.

Paths and cycles. A *path* of length n from X to Y is a sequence of distinct vertices $X_0 = X, X_1, \ldots, X_n = Y$, such that (X_i, X_{i+1}) belongs to E, for $i = 0, 1, \ldots, n-1$. Thus, a path can never cross itself and it can never go against the direction of the arrows. In general, we can distinguish undirected, partially directed and directed paths. If all edges are directed, we call it a *directed path*. A *partially directed path* is such that it can contain directed and undirected edges (see Figure 4.3). Clearly, a directed path is a special case of a partially directed path. An undirected path has all edges undirected. A *chain* of length n from X to Y is a sequence $X_0 = X, X_1, \ldots, X_n = Y$ of distinct vertices, such that $X_i \rightarrow X_{i+1}$ or $X_i \leftarrow X_{i+1}$ for all $i = 0, \ldots, n-1$. A *n-cycle* is a path of length n, such that $X = Y$. The cycle is said to be *directed* if it contains an arrow.

Directed acyclic graph. Directed graphs without cycles (called directed acyclic graphs or DAGs) play an important role in statistics and will be used throughout this work. In addition to graph theoretic objects already defined, a few additional terms will prove useful when dealing with DAGs. For a node X, we define the set of its descendants, $\mathrm{de}(X)$, as a set of all nodes Y, such that there is a directed path between X and Y. We further define the set of non-descendants of X as $\mathrm{nd}(X) = V \setminus \{\mathrm{de}(X) \cup \{X\}\}$. The *skeleton* of a DAG G is an undirected graph obtained from G by replacing all the arrows with undirected edges. A *V-structure* (or an unshielded collider) is a three nodes structure consisting of a child node and two unmarried

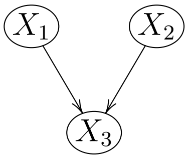

Fig. 4.4 V-structure.

parents, see Figure 4.4.

Topological ordering of nodes. Topological ordering of vertices of a directed acyclic graph is such that if a variable X is an ancestor of a variable Y in a graph G, then X precedes Y in that ordering. Obviously, such an ordering is generally non unique but always exists, see for example Bondy and Murty (2010).

Chain graph. *Chain* graphs contain both directed and undirected edges and can be seen as a generalization of both directed and undirected graphs. The vertex set V of the chain graph is partitioned into numbered subsets, the so-called chain components $V = V_1 \cup \ldots \cup V_T$ such that all edges between nodes belonging to the same subset are undirected while all edges between different subsets are directed, pointing from the set with a lower number toward the set with a higher number. Such graphs are characterised by having no partially directed cycles. An undirected graph is a special case of a chain graph when there is a single chain component, while a directed acyclic graph is a special case of a chain graph, when all chain components consist of a single vertex.

Moral graph. For a chain graph G, we define its *moral graph* G^M as the undirected graph with the same vertex set but with X and Y adjacent in G^M if and only if either $X \to Y$ or $Y \to X$ or if there are Z, W in the same chain component such that $X \to Z$ and $Y \to W$. In the special case of a DAG, moralization consists of first marrying the unmarried parents (see Figure 4.4) and then replacing arrows with undirected edges.

Chapter 5

Guided learning in graphical models

As stated in the previous chapters, we are motivated by the need of defining a statistical framework for gene silencing. We intend to develop such framework within graphical modelling. The first step is therefore eliciting a graphical model for a set of interacting genes. Information about interactions among genes is present in the pathway diagram, so one could simply translate this diagram into a graph (directed or undirected). It is often the case, however, that this structure can be significantly improved upon in the light of experimental data. In this Chapter, we discuss this issue, with the aim of combining available biological knowledge with the data on gene expression measurements. This aim falls within the scope of learning in graphical models.

When talking about learning in graphical models, we distinguish two broad classes of problems: estimating the parameters of a model when a graphical structure is given, and learning the structure of the model from data. While the former is usually considered a traditional problem of statistical inference, the latter is usually covered in the machine learning literature. In Section 5.1, we propose a method for guided structure learning in situations when some prior information is available, but somehow vague to be fully trusted. In Section 5.2, we look at various structure learning algorithms and empirically compare their performance to that proposed in 5.1. In Section 5.3, we propose to assess the uncertainty in the learned structure via resampling. Finally, in Section 5.4, we tackle a guided penalized estimation of large dimensional matrices in Gaussian graphical models.

5.1 Guided structural learning

The problem of inferring the network of genes from gene expression data (commonly referred to as "reverse engineering") has received much attention in the computational biology literature in the last two decades (see Bansal et al., 2007, for a comparative review). Here, we do not aim at inferring a network from data only; instead, we want to inform the learning strategy about the relations among genes that are described in a pathway.

Our solution is based on a modification of a very popular structure learning algorithm, the so-called K2 algorithm (Cooper and Herskovits, 1992). K2 algorithm is one of the first solutions to the problem of learning DAGs from data. Before, DAGs were usually constructed by hand, in close collaboration with domain experts. Obviously, that approach had limitations associated with the size of the considered network and the availability of the domain knowledge. This motivated the search for a procedure that would construct DAGs automatically. In their highly original work, Cooper and Herskovits (1992) proposed a new Bayesian score function, the K2 score, that scores individual DAGs reflecting how well they fit the observed data. The task is then reduced to finding the structure that maximizes the considered score. Unfortunately, the search is not trivial, since the number of possible structures grows exponentially with the number of nodes. To reduce the search space, the K2 algorithm takes as an input in addition to the data, the ordering of variables.

Although specification of the topological ordering of variables might seem limiting in some cases, in our context it proves to be a desirable property as it provides an opportunity to include prior knowledge about the graphical structure. We therefore transform the pathway into a DAG and then pass its ordering to the algorithm. The main strength of this proposal is its simplicity and low computational cost: biological knowledge is incorporated without specifying a prior distribution on the space of all possible graphs, which proves to be very difficult in practice. Our proposal uses prior information without relying on a fully Bayesian approach.

5.1.1 Background on the K2 algorithm

The K2 algorithm belongs to the score based approaches to structure learning. It uses a Bayesian scoring function to score individual graphs and indicate how well they fit the data. One of the main results presented in the paper Cooper and Herskovits (1992) is the analytical expression giving, under very mild conditions regarding the uniform prior distribution, the posterior probability of a DAG structure. The K2 score is based on that result.

Theorem 5.1.1 *Let there be n i.i.d observations of p discrete variables $X_1, \ldots, X_p$, where X_i has r_i possible values $(v_{i_1}, \ldots, v_{ir_i})$. Let G be a DAG containing these variables. Let pa_i denote the set of parents of X_i in G. Let w_{ij} denote the jth unique realization of pa_i. Suppose there are q_i such unique realizations. Define N_{ijk} to be the number of observations in which X_i assumes the value v_{ik} and pa_i has the value w_{ij}. Let*

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}.$$

If the prior distribution is uniform over a set of all possible DAGs on p nodes, then K2 score of the structure G corresponds up to a constant to its posterior probability and is given by

$$g(G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!. \tag{5.1}$$

For the proof, we refer the interested reader to the original article.

The K2 algorithm searches for the graph that maximizes the K2 score among all DAGs. To explore the space of possible structures, it makes use of the one step greedy search strategy (the justification for this strategy is given in Section 5.1.2). It also requires a topological ordering of the variables. We recall, a topological ordering of a directed graph is an ordering of its nodes, such that for every directed edge $X_i \to X_j$, X_i precedes X_j in the ordering.

The algorithm looks for an optimal parent set for every variable starting from the first one in the topological ordering. For a fixed node, the algorithm starts from the empty parent set, and considers the set of candidate parents consisting of all the variables that precede the fixed node in the ordering. The parent whose addition most increases the score of the resulting structure is added to the parent set. When no addition of a single parent can increase the score, it stops adding parents to the variable and moves to the next variable in the ordering. Since an ordering of the nodes is known beforehand, the search space under this constraint is much smaller than the entire space. The following pseudocode expresses this heuristic search approach.

K2 algorithm is designed for discrete data, while our gene expression measurements are continuous. To bypass this issue, we can make our observations categorical (we pursue this in the next Section) or modify the algorithm to allow for continuous data. Here, we focus on the latter.

Algorithm 1 K2 algorithm

```
{Input: $n$ observations of $p$ variables $X_1, \dots, X_p$, a topological ordering, an upper bound for the number of parents $u$}
for $i = 0$ to $p$ do
    $\mathrm{pa}_i \leftarrow \varnothing$
    $P_{old} \leftarrow g(i, \mathrm{pa}_i)$
    $OKToProceed \leftarrow$ true
    while $OKToProceed$ and $|\mathrm{pa}_i| < u$ do
        let $z$ be the node in $Pred(x_i) \setminus \mathrm{pa}_i$ that maximizes $g(i, \mathrm{pa}_i \cup \{z\})$
        $P_{new} \leftarrow g(i, \mathrm{pa}_i \cup \{z\})$
        if $P_{new} > P_{old}$ then
            $P_{new} \leftarrow P_{old}$
            $\pi_i \leftarrow \mathrm{pa}_i \cup \{z\}$
        else $OKToProceed \leftarrow$ false
        end if
    end while
    print "parents of the node: ", $x_i$, "are", $\mathrm{pa}_i$
end for
```

5.1.2 The CK2 algorithm

Our strategy consists of specifying the ordering of nodes according to the considered biological pathway (more precisely, according to a DAG derived from the said pathway) and using the K2 algorithm. However, one limitation of the K2 score is that it is defined only for discrete, or more precisely, categorical variables. Gene expression measurements are usually continuous. One possibility is categorization, which is the direction that we take in the next Section. However, in some instances this is not the best course of action (see 5.1.3) and so, in what follows, we extend the algorithm allowing for continuous data. We call our proposal CK2.

To extend the method to continuous data one would need to modify the scoring function. We thus considered alternative scoring criteria applicable to continuous data. When considering alternatives, we restricted our attention to criteria that balance the goodness of fit and model parsimony. The first one is the Bayesian information criterion, which is, in its most general form, given by

$$\log(\text{maximized likelihood}) - \frac{\log(n)}{2} \times (\text{n. of estimated parameters}).$$

In the case of a Gaussian bayesian network, this becomes

$$\mathrm{BIC} = n\left[\log|\hat{\mathbf{K}}| - \mathrm{tr}(\mathbf{S}\hat{\mathbf{K}})\right] - \log(n)\sum_{i=1}^{p}\mathrm{pa}_i, \tag{5.2}$$

where $\mathbf{S}$ is the sample covariance matrix and $\hat{\mathbf{K}}$ is the maximum likelihood estimate of the concentration matrix. The BIC criterion belongs to the Bayesian scoring metrics family and can be seen as an asymptotic approximation of the posterior probability of the structure, i.e., the approximation of the full posterior probability integrated over all possible parametrizations of the conditional densities for the given structure.

The second scoring criterion that we consider is the Akaike information criterion differing in the multiplicative factor of the penalization term

$$\mathrm{AIC} = n\left[\log|\hat{\mathbf{K}}| - \mathrm{tr}(\mathbf{S}\hat{\mathbf{K}})\right] - 2\sum_{i=1}^{p}\mathrm{pa}_i.$$

In Section 5.2, we will consider further possibilities based on the quantity

$$\log(\text{maximized likelihood}) - \text{multiplier} \times \sum_{i=1}^{p}\mathrm{pa}_i.$$

The asymptotic properties of the K2 algorithm, which depend on the consistency of the scoring criterion and on the search strategy, are not affected in CK2. To see this, we first recall the *consistency* of a scoring criterion.

Definition 5.1.1 (Consistency of a scoring criterion.) *Assume that data are generated by some distribution P^* whose underlying DAG is G^* (in other words, the set of conditional independence relations that hold in P^* coincides with the set of conditional independence relations implied by G^*). We say that scoring function is consistent if the following properties hold as the number of observations goes to infinity, with probability that approaches 1:*

- *the structure G^* will maximize the score;*
- *all structures that are not equivalent to G^* will have strictly lower score.*

Haughton et al. (1988) showed that both BIC and AIC are consistent scoring criteria. Chickering and Meek (2002) and Chickering (2003) derived optimality results stating that the greedy search used in conjunction with any consistent scoring criterion will, as the number of observations goes to infinity, identify the true structure (up to an equivalence class, see 5.1.3). Therefore, asymptotic properties are preserved.

Two aspects are also worth mentioning when choosing a search strategy: the computational cost and the ease of implementation. The greedy search, that we use in CK2 algorithm, allows to comfortably deal with the number of DAGs that grows exponentially with the number of nodes, and is also easy to implement.

5.1.3 Notes and observations

Equivalence of DAGs. We have seen earlier in Section 4.1.2 that different DAGs can encode the same set of conditional independence relations and cannot thus be distinguished on the basis of observations alone. In statistical terms, this means that the true DAG is identifiable up to an equivalence class. As a consequence, structure learning algorithms usually output an object representative of the whole equivalence class. A CPDAG (complete partially directed acyclic graph) contains both directed and undirected edges. An edge between nodes i and j, present in the skeleton, is directed in a CPDAG if and only if the orientation of that edge is the same across all DAGs in the equivalence class. Otherwise, it remains undirected.

The approaches that work with a given ordering of variables (such as CK2), avoid identifiability issue: the information provided by the topological ordering orients all the edges, so that output is a unique DAG.

The topological ordering assumption. In some cases, the ordering is clearly determined by the temporal aspect; in the majority of others, the need to specify the ordering of variables beforehand is limiting. In our setting, it represents an opportunity to include the biological knowledge about the system of studied genes. Neverthelss, all results and inferences drawn on the basis of the refined pathway are then, as a consequence, conditional on that topological ordering. Several issues might arise. First of all, the topological ordering may be misspecified due to a number of reasons, such as inaccuracy of pathway representation or the choices made in a translation from a pathway diagram to a fully directed graph. Secondly, a DAG induces only a partial order on the set of nodes, that is, not every pair of nodes is necessarily ordered with respect to the relation of precedence (the most evident example is the set of nodes of having no ancestors in a given DAG). In order to apply the algorithm we need to extend it to a total order. The number of topological orderings of a given DAG is a # P complete problem. In other words, while finding a single topological ordering of a DAG is a simple task, finding all of them is anything but simple; in fact, the # P complete class is sometimes referred to as the class of easy problems with hard counting solutions. This greatly complicates the sensitivity analysis designed to address the issue of the uncertainty pertaining to a single chosen topological ordering. We return to this issue in Section 5.2.4 and Chapter 7.

Discrete vs. continuous observations. Gene expression measurements are usually continuous. Nevertheless, it is still a debated issue whether one should analyze these measurements directly or only after a categorization procedure. This question boils down to: are there only a limited number of states a gene can assume (for instance, “not expressed”, “under-expressed”, “normally expressed”, “over-expressed”) that are then affected by a number of noise sources; or the gene expression values can be considered inherently continuous? In any case, we believe that both approaches are worth pursuing. In some cases, it might be more reasonable to categorize the measurements. One such situation arises when the spectrum of possible states of genes is wide enough, usually because gene expression is measured across different experimental conditions. We can then safely assume, that at least some of the genes will be affected by the change and will vary considerably (so that we observe different underlying states). On the other hand, when expression is measured only in wild type samples (steady state without any external intervention) it might be possible that the scale of variation is too limited to assume that different underlying states are present in the data. In that case, we can probably learn more by analyzing directly the continuous measurements, assuming that even moderate variations in expression are informative, and can help us gain some insight about relationships between studied genes.

Software availability. We note that, to the best of our knowledge, K2 algorithm has not been implemented in R to date. For that reason, we implemented the original version, along with the proposed CK2 in the form of R functions.

5.2 Empirical comparison with alternative learning strategies

Here, we compare our CK2 approach to a number of popular structure learning algorithms by applying them to the experimental data from the *Drosophila Melanogaster* experiment (see Chapter 3). Taking into account our primary aim, i.e., predicting the effects of gene silencing, we evaluate different algorithms on the basis of the predictive accuracy of the DAGs that they produce. Such DAGs need not necessarily provide a good description of the underlying biological mechanism, but this is not an issue of concern, since our goal is finding a good basis for making predictions in the presence of possibly incomplete or inaccurate biological information.

Structure learning algorithms can be roughly divided into two major approaches: search and score methods and constraints based methods. The first approach consists of a score function that evaluates each structure with respect to the data, and a search strategy employed to find the optimal structure according to this score. The CK2 algorithm, that we proposed, belongs to this class. The second approach to structure learning uses statistical tests such as chi-square or mutual information to find conditional independence relations among the variables; these are then used in conjunction with causality-driven orientation rules to construct DAGs (Pearl et al., 1991). The PC algorithm (Spirtes et al., 2000) is the most popular representative of this class of methods.

Most methods work with categorical variables, so in Section 5.2.1, we describe how the gene expression measurements could be categorized prior to the application of the structure learning algorithms. In Section 5.2.2, we explain how we assess the predictive accuracy of considered algorithms. A brief description of the algorithms compared in this study is given in Section 5.2.3. Finally, results are presented in Section 5.2.4, and some final remarks are given in Section 5.2.5.

5.2.1 Categorization of expression measurements

As most structure learning algorithms make use of categorical variables, an extensive empirical comparison requires to categorize our measurements. In the work that first introduced the idea of using DAGs for representing gene regulatory networks, Friedman et al. (2000) considered both discrete and continuous models. In the first case, they categorized the gene expression values to three categories (“under-expressed”, “normally expressed”, and “over-expressed”) prior to the analysis; in the second case they assumed a multivariate normal distribution for the gene expression measurements. They demonstrated strengths and weaknesses of both choices. It is clear that the first strategy attenuates the effect of the technical variability. On the other hand, it might lead to information loss, and is also sensitive to the choice of the categorisation procedure. The second strategy incurs no information loss, but is incapable of capturing non-linear relationships between genes. In particular, combinatorial relationships (one gene is over-expressed only if a subset of its parents is over-expressed, but not if at least one of them is under-expressed) can be modeled only with a discrete Bayesian network. The two approaches thus seem complementary and we believe that both can help researchers obtain the biologically relevant results, at least as a means of postulating testable scientific hypothesis.

When the goal of categorization is to obtain categories which are meaningful from the biological perspective, one would ideally have the control group (a previous experiment) which would serve as a reference for comparison; if the measured expression is significantly higher with respect to the control, it is labeled over-expressed. The threshold point for significance has to be set in advance, based on subject matter considerations and previous experiments. In Friedman et al. (2000) the ratio of the two measurements was considered significant if greater than $2^{0{,}5}$. Similar considerations apply for “under-expressed” genes, the threshold being $2^{-0{,}5}$.

When control data are not available, we propose to perform categorization based solely on data at hand. It is assumed that genes can assume only a few functional states, for example “under-expressed”, “normal”, and “over-expressed”. The actual measurements depend on these functional states and the amount of biological variability and technical noise. A plausible model for such data is a mixture of K normal distributions, each centered at one of the K functional states

$$X_i \sim \sum_{k=1}^{K} \tau_{ik} \mathsf{N}(\mu_{ik}, \sigma_{ik}^2), \quad i = 1, \dots, p,$$

where X_i is an expression of the considered gene, μ_{ik} and σ_{ik}^2 are parameters corresponding to the k-th functional state, τ_{ik} the probability that an observation belongs to the k-th component ($\tau_{ik} \geq 0, \sum_{k=1}^{K} \tau_{ik} = 1$) and p is the number of considered genes. In the example above shown $K = 3$. However, it is not always plausible to assume that all K states are present in a single experiment, for example, certain genes remain normally expressed in a wide range of conditions, others can only be downregulated, etc. This led us to propose a data driven approach to categorization: a number of components, that can vary from one (corresponding to a gene with only one observed state) to K (all functional states are present in the data) is estimated from the data for each gene independently. The assumed model for the i-th gene is thus

$$X_i \sim \sum_{k=1}^{\hat{K}_i} \tau_{ik} \mathsf{N}(\mu_{ik}, \sigma_{ik}^2), \quad i = 1, 2, \ldots, p,$$

where $\hat{K}_i$ is the estimated number of components for the i-th gene, τ_{ik} are, as before, the weights of individual components, μ_{ik}, σ_{ik} are component specific parameters. The approach that simultaneously estimates the number of components in the mixture and parameters pertaining to different components and then classifies each observation according to the estimated model is called Model Based Clustering and was introduced by Fraley and Raftery (2002). We used its implementation in the R package `mclust` (Fraley et al., 2012). In what follows, we will denote $Y_i = (Y_{i1}, \ldots, Y_{i\hat{K}_i})$ the variable obtained from X_i through the proposed categorization, where $Y_{ij} = 1$, if X_i falls to category j, and zero otherwise.

5.2.2 Evaluation of predictive accuracy

As anticipated, we compare different algorithms in terms of their predictive accuracy. Given the small sample size, to assess the predictive accuracy we adopt a "leave-one-out" approach, where in each step the chosen learning algorithm is applied to the data from which the single observation j has been removed. In the second step, the removed observation is used to evaluate the predictive accuracy: prediction of the value of every variable is computed given the values of all other variables.

It is worth noticing that when considering conditional distributions of individual variables given the remaining ones in the context of DAGs, it is sufficient to restrict attention to the Markov blanket. The Markov blanket of a given node in a DAG, denoted mb(·), consists of the nodes' parents, children and other parents of its children. It is the set of variables which shields the given variable from the rest of the network. More formally, considering without loss of generality the variable Y_1, we have

$$\mathsf{P}(Y_1 \mid Y_2, \ldots, Y_p) = \mathsf{P}[Y_1 \mid \mathrm{mb}(Y_1)]$$

Given that the networks of genes are typically sparse, this offers a considerable computational relief, especially when the number of genes is large.

To measure the distance between the observed value and the predicted value for variable Y_i fixing all remaining variables to the values observed on the removed observation j, we use the Brier score, introduced in Brier (1950). Originally employed to assess the quality of rain forecasts, due to its simplicity and interpretability, this criterion found its application in various other fields. In general, if we denote ${}_jy_i = ({}_jy_{i1}, \ldots, {}_jy_{i\hat{K}_i})$ the observed value of variable Y_i in the jth observation, the Brier score is defined as

$$ {}_jb_i = \frac{1}{2}\sum_{k=1}^{\hat{K}_i}({}_j\hat{\pi}_{ik} - {}_jy_{ik})^2, \tag{5.3}$$

where ${}_j\hat{\pi}_{ik}$ is the predicted probability that Y_i falls into the category k. The Brier score measures the squared distance between the forecast probability distribution and the observed value. It can assume values between 0 (the perfect forecast) and 1 (the worst possible forecast).

In this empirical comparison, we will consider data from the fruit fly silencing experiment. To make the comparison less dependent on the small sample size, we use jointly the control and the knockdown data. We thus have $n = 28$ observations of $p = 12$ genes denoted $Y_1, \ldots, Y_p$. Adopting a “leave-one-out” approach, for every algorithm we have n predictions, one for each observation that is being left out. We measure the predictive accuracy of the algorithm with a scalar measure B

$$B = \sum_{j=1}^{n}\sum_{i=1}^{p} {}_jb_i. \tag{5.4}$$

Obviously, algorithms having lower score are preferred.

We compare algorithms designed for categorical and continuous data. The learning algorithms that work with continuous data produce predictions on the continuous scale. In order to make them comparable with categorical predictions, we combine discriminant analysis with the proposed categorization procedure. We classify continuous predictions into one of the gene specific components estimated in the initial categorization. More precisely, we apply the discriminant analysis to the prediction ${}_j\hat{X}_i$; the output is the estimated vector of probabilities $({}_j\hat{\pi}_{i1}, \ldots, {}_j\hat{\pi}_{i\hat{K}_i})$ that ${}_j\hat{X}_i$ falls into associated categories. We can then plug this vector in the expression for the Brier score (5.3).

5.2.3 Learning algorithms

Here, we introduce the algorithms that we consider in this empirical comparison.

PC algorithm

The PC algorithm (named after its inventors Peter Spirtes and Clark Glymour), is a popular constraint based algorithm, introduced in Spirtes et al. (2000). It starts from the full undirected graph on p variables, and then moves sequentially to remove edges not supported by the data.

- For each pair of variables Y_i and Y_j test whether they are independent. If so, remove the edge between them.
- For each pair of variables that are still connected, test $Y_i \perp\!\!\!\perp Y_j \mid Y_k$, for each $k \in \{1, 2, \ldots, p\} \setminus \{i, j\}$. If for some k the hypothesis is not rejected, remove the edge between Y_i and Y_j.
- For each pair of variables that are still connected, test the conditional independence given all possible sets of two variables. If for some conditioning set the test does not reject the independence hypothesis, remove the edge between them.
- For each pair of variables that are still connected, check whether they are conditionally independent given all possible sets of three variables. If so, remove the edge between them.

  ⋮
- Finally, for any pair of variables that are still connected check whether Y_i is conditionally independent of Y_j given all $p - 2$ other variables. If so, remove the edge between them.

The PC algorithm, as every other constraint based learning algorithm, relies on the use of conditional independence tests. The choice of which conditional independence test to use depends on the nature of considered variables. In the case of discrete variables, the most usual choice is the χ^2 test of independence in contingency tables, and this is our choice in this study. We note, however, that PC algorithm can be combined with any consistent statistical test of independence. The sparsity of the obtained structure depends on the chosen significance level of the tests: the higher the significance level the lower is the number of edges. Here, we used two levels, i.e., 5% and 20%.

According to what has been stated so far, the output of the PC algorithm seems to be an undirected graph that is a skeleton of the underlying DAG. However, the final step of the algorithm uses simple probabilistic considerations, as well as the property of acyclicity, to orient some of the edges. The output is thus a partially directed acyclic graph, representative of a certain equivalence class. The PC algorithm is freely available in the `pcalg` package (Kalisch et al., 2012), but is also implemented in the commercial software Hugin. In this work, we rely on the latter implementation.

Gobnilp

Recently a number of approaches to structure learning that do not rely on a search strategy, but explore the entire space of DAGs, have been proposed. Two major directions are dynamic programming and integer linear programming (ILP). For the former, see for instance Koivisto and Sood (2004). We consider a representative of the latter: the approach of Cussens (2011). Here, the problem of structure learning is translated into an optimization problem with a linear objective function and a set of linear constraints (including integrality constraints on the variables).

The approach is implemented in the freely available C program Gobnilp (Globally optimal Bayesian Network learning using Integer Linear Programming). Gobnilp works fairly fast with moderately sized problems, and is allowing the user to specify a large number of options and restrictions, such as the upper limit for the parent set size, prior knowledge about probabilities of different parent sets, etc. Currently, the default score is the BDeu: a particular case of the likelihood equivalent Bayesian Dirichlet score. We opted for the BIC, for reasons of comparability with competing approaches. We also employed two modifications of the BIC criterion, acting on the penalty term. In particular, we considered two alternative multipliers of the penalty, i.e., 10^{-3} and 10^{-9}. In the former, the penalty is considerably smaller than in the BIC, while in the latter the penalization is negligible so that this scoring criterion behaves as the likelihood function. Given the small sample size, many structures approximately maximize the likelihood, so a small but a non-zero penalty favours sparser structures. The output of the GOBNILP algorithm is the globally optimal structure; when more than one structure maximizies the score, the sparser one is preferred.

Detailed descriptions of K2 and CK2 are given in Section 5.1. Both of these algorithms, along with gene expression measurements and a topological ordering, have an additional input, i.e., the upper limit for the set of parents, which in this study was set to two.

To summarize, in this empirical study, we consider the following options.

PC The PC algorithm using χ^2 test of independence at the 5% significance level.

PC20 The PC algorithm using χ^2 test of independence at the 20% significance level.

K2 The original K2 algorithm.

K2-BIC A modified K2 algorithm, where the criterion used to score competing DAGs is BIC, while the search strategy remains the one step greedy search.

G-BIC The Gobnilp algorithm with the BIC scoring criterion.

G-BICm The Gobnilp algorithm with the modified BIC criterion (the penalty term is multiplied by a factor of 10^{-3}).

G-BICl The Gobnilp algorithm where the modified BIC criterion (the penalty term is multiplied by 10^{-9}).

CK2 The CK2 algorithm proposed in 5.1.2. The only algorithm in this study that is applied to the continuous measurements.

Full graph Corresponds to the complete directed acyclic graph, which is a directed acyclic graph whose skeleton is a complete graph. In other words, the set of conditional independence relations entailed by such a DAG is empty.

Empty graph Corresponds to the DAG containing no arrows. In other words, the variables of such a graph form a system of independent random variables. This is a very naive prediction method, but it may serve as a reference for comparison with more advanced methods.

5.2.4 Results

In Table 5.1, we report the B score for each of the considered methods. Variables *arm* and *rok* were excluded from the analysis, since in the categorized dataset they assumed only one value. In our study K2 reaches the minimal B score, followed by the Gobnilp's likelihood method G-BICl. The K2 algorithm with the BIC score, K2-BIC, together with the remaining Gobnilp methods, G-BICm and G-BIC, also perform reasonably well with a slightly inferior score with respect to the leading twosome. On the other hand, the PC algorithm in this study gives significantly less accurate predictions. One possible explanation is that on the level of 5% optimal structures have too few edges (see Table 5.3), which led us to consider the version with a higher nominal significance level of 20% (PC20). The B score improves, but is still unable to approach the best performing algorithms. The CK2 algorithm, seems to fail in this case. Its B score is almost comparable to the one of the full graph (Full). A closer look suggests that the gene *dally*, i.e., the gene that changes most dramatically between the conditions of wildtype and knockdown, is responsible for a large part of the cumulative B score.

It is interesting to note that of the two methods on categorized variables using the BIC score, K2-BIC and G-BIC, it is the former that minimizes the B score. This is a little surprising, since Gobnilp finds globally optimal structures, while K2-BIC uses the ordering of variables, and thus might suffer from misspecification. In addition to that, K2-BIC relies on the greedy search, possibly restricting the search space enough to miss the global optima. In fact, structures found by Gobnilp have a lower BIC criterion (and thus a better fit to the data), but are inferior when it comes to prediction. This observation, together with a success of the K2, suggests that possibly the subject matter knowledge employed to specify the ordering of variables is the reason behind their good performance. To test this hypothesis, we generated 20 random orderings and passed them to the K2 algorithm. Computed B scores for the first 10 of them are presented in the Table 5.2. We see that for the ordering number 8, the B score equals the one corresponding to the pathway ordering. Nevertheless, none of the ten reported (and twenty computed) scores manages to outperform it, providing support for the practice of using the prior information in the form of a topological ordering.

Figure 5.1 shows how the B score deteriorates with the addition of arrows to the optimal structure found by K2. Here, the B score is a function of the number of arrows present in the graph. It starts from the K2 structure, containing 15 arrows, and ends with the full graph, containing 66 arrows. Structures in between are obtained sequentially, by randomly adding a single arrow to the current structure. Obviously, the order of addition of arrows plays a role, and thus this is only one possible way in which the score might evolve between the two extreme points. Nevertheless, the increasing trend of the dependence is informative and independent of the order of arrow inclusion.

One of the reasons behind the success of the K2 algorithm might also be that it identifies DAGs with a relatively high number of edges. To examine this possibility, we computed the average size of the Markov blanket for all considered methods. The results are reported in Table 5.3. We see that K2 indeed has a comparatively large average Markov blanket size, but it is second to the Gobnilp's likelihood method. The ranking of methods with respect to their prediction accuracy suggests therefore that the density of the graphs inferred by K2 is not the only reason for its good performance.

5.2.5 Conclusions

We compared a number of different approaches of inferring a network of genes on the basis of gene expression measurements. In terms of prediction accuracy the most promising one seems to be the K2 algorithm that, in addition to the experimental data in the form of categorized measurements, requires information regarding the topological ordering of genes. K2 is followed by the Gobnilp's exact method with a likelihood scoring criterion. The possible reasons for the success of K2 are twofold:

its inferred graphs are more dense with respect to graphs inferred by other methods (the property related to the K2 scoring criterion), and the use of prior information that seems to point the search towards “better” models, at least when it comes to prediction considerations. On the other hand, we attribute the somewhat surprisingly low performance of CK2, in large part, to the use of the continuous measurements that makes predictions much more sensitive and less robust. In fact, as already noted, the gene *dally* is mostly responsible for its poor performance. The change of *dally* between controls and knockdowns is mitigated in the categorization process, resulting in a less dramatic impact on prediction. This suggests that the best course of action might be a data driven categorization proposed in 5.2.1.

The results are obviously sensitive to the categorization procedure. When the range of expression states is wide enough (which usually translates into observing samples in different experimental conditions, such as wild type vs. perturbations), we prefer the model based clustering approach to to the simpler quantile based approach, the former being more justified from the biological perspective.

An important aspect of this study is the small sample size ($n = 28$), atypical in machine learning applications, but very common in genomics setting. To overcome this issue, most authors propose approaches Bayesian in nature. For instance, Friedman et al. (2000) focus on local features of the network regarding two or three nodes and then use model averaging to find posterior probabilities of the features of interest. Imoto et al. (2004) use prior biological knowledge in addition to data on gene expression levels to obtain more reliable estimates of the gene regulatory networks. Even though the adequacy of this sample size for the goal of elucidating biological mechanisms at play is questionable, from the prediction perspective the results reported here are encouraging: learned graphs manage to bring considerable improvement over the procedure that does not assume or look for any conditional independence relations between genes (represented by the full graph). This is an important empirical conclusion that we draw from this study.

All considered approaches assume there are no missing values in the data, which is true of our experiment. Unfortunately, in many real life datasets this is not the case. In gene expression experiments the percentage of missing values is usually small, so that incomplete observations can be ignored without introducing a serious bias. When the percentage of incomplete observations is not negligible, the course of action should be based on a careful investigation of the nature of the missing values.

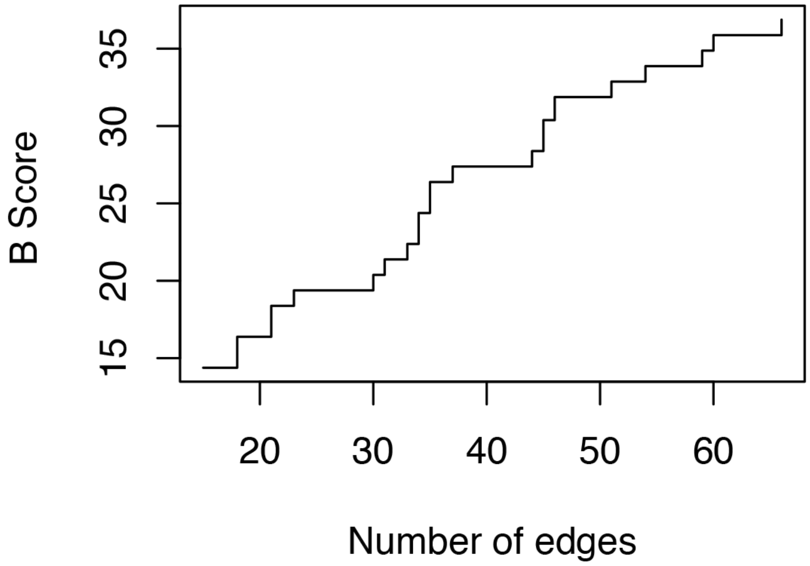

Fig. 5.1 *B* score as a function of the number of edges in the DAG.

Table 5.1 Evaluation of the prediction accuracy: the B Score.

	PC	PC20	K2-BIC	K2	G-BIC	G-BICm	G-BICl	CK2	Full	Empty
psn	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.01	3.00	2.88
nkd	4.65	2.61	0.00	0.00	1.00	0.00	0.00	8.50	3.00	7.53
dally	5.36	6.97	5.34	5.30	6.29	5.30	5.30	13.56	6.30	9.72
por	1.00	0.00	0.00	0.00	1.00	1.00	0.00	1.98	3.00	2.88
daam	4.81	3.81	4.87	3.95	5.13	3.95	3.44	2.99	5.44	6.15
fz	4.07	1.19	2.41	1.12	1.12	1.12	1.12	0.01	3.12	6.15
rho1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.41	3.00	2.88
dco	3.29	3.05	1.25	1.50	1.25	2.50	2.50	1.29	3.50	3.69
sgg	0.18	0.00	0.00	0.00	1.00	0.00	1.00	0.99	3.00	2.88
pont	1.10	3.16	1.50	1.50	1.50	1.50	1.50	1.98	3.50	7.37
	24.46	21.79	16.37	14.37	18.29	15.37	14.86	32.72	36.86	52.13

Table 5.2 The (B) score for ten randomly generated orderings passed to K2.

Ordering of variables												B
6,	5,	2,	8,	4,	1,	12,	7,	3,	9,	10,	11	22.87
6,	11,	9,	5,	2,	7,	1,	4,	8,	10,	12,	3	17.38
4,	7,	11,	3,	8,	2,	1,	10,	9,	6,	12,	5	15.38
9,	3,	12,	8,	11,	6,	1,	5,	4,	10,	7,	2	19.87
9,	10,	7,	8,	6,	4,	11,	3,	2,	5,	12,	1	15.38
6,	2,	11,	5,	8,	3,	12,	9,	4,	10,	1,	7	19.87
8,	3,	4,	11,	1,	12,	9,	7,	6,	5,	10,	2	19.87
7,	2,	3,	5,	11,	8,	6,	10,	4,	12,	1,	9	14.38
10,	12,	1,	2,	6,	5,	9,	7,	4,	8,	3,	11	20.38
9,	7,	8,	11,	1,	6,	10,	5,	2,	3,	12,	4	16.10

Table 5.3 Average size of the Markov Blanket for different algorithms.

	Average size
PC	0.98
PC20	1.32
K2-BIC	2.30
K2	2.64
G-BIC	2.15
G-BICm	2.49
G-BICl	2.71
CK2	4.10

5.3 Model uncertainty

An important question that arises when learning graphical structures from data is the degree of confidence we have in the learned graph. Asymptotic results guarantee that, in the limit, the "true" structure will be identified. In practice, we deal with finite samples, and so usually several networks have similar scores. This is particularly true when the sample size is limited. Unfortunately, this is the typical situation when learning biological networks. In a standard gene expression experiment, the number of genes is of the order of hundreds, while the number of biological samples is of the order of tens. For purposes of comparison, we note that in other applied settings, such as engineering or social sciences, the situation is quite different: either thousands of observations of a particular system are available (see, for example, the ALARM network featured in Cooper and Herskovits, 1992), or the number of considered variables is much lower (see, for example, The causes of publishing productivity study in Spirtes et al., 2000, chap. 5).

Especially when the sample size is small, whatever approach to structure learning is chosen, it is the case that many different structures will fit the data almost equally well. In case of a score based algorithm, one structure will usually maximize the scoring function, but many others will have only a slightly inferior score. Is it then justifiable to base our inference on a single "best" model? Many would respond negatively. Here, we propose a simple method to evaluate the level of uncertainty of the model learned by the chosen structure learning algorithm.

In what follows, we will consider the *network features*, which are simply aspects of interest of the underlying network. We will denote them by $f(G)$, to highlight that they are functions of the considered network. For instance, we might be particularly interested in a presence of a specific edge $X \to Y$, so that $f(G) = 1$, when such edge is present in G, and $f(G) = 0$ otherwise. Another interesting example is the relation of ancestry between two nodes, so that $f(G) = 1$, whenever there is a directed path between X and Y. Network features are not limited to pairwise relations, but for now, for reasons of simplicity, we consider binary features.

If we adopt a Bayesian approach to the problem, the quantity of interest is the posterior probability of the chosen feature

$$\mathsf{P}(f \mid \mathbf{X}) = \sum_{G} f(G)\mathsf{P}(G \mid \mathbf{X}),$$

where the sum is taken over the set of all possible structures G. Obviously, as we saw earlier, given the number of possible structures, summing over all of them is not feasible, save the special case of networks of up to five nodes. One possibility is to approximate this probability by considering a sample of size M from the posterior distribution on the space of structures

$$\mathsf{P}(f \mid \mathbf{X}) = \frac{\sum_{m=1}^{M} f(G_m)\mathsf{P}(G_m \mid \mathbf{X})}{\sum_{m=1}^{M} \mathsf{P}(G_m \mid \mathbf{X})}. \tag{5.5}$$

The question is how to obtain the sample $G_1, \ldots, G_M$. One possible answer is to use the general tool of Markov chain Monte Carlo (MCMC) simulation. In that case, we need to define a Markov chain over the space of possible structures whose stationary distribution is the posterior distribution $\mathsf{P}(G \mid \mathbf{X})$. A simpler and computationally inexpensive solution is offered by bootstrap. To apply the bootstrap approach, we resample observations with replacement from the original dataset many times, and we learn the graphical structure for each generated dataset. In this way, we obtain a number of DAGs which are all reasonable models for the data at hand. We apply the bootstrap approach as follows:

- Sample data with replacement from the original dataset M times, to obtain M samples $\mathbf{X}_m$, $m = 1, \ldots, M$.
- Apply the structure learning algorithm to each $\mathbf{X}_m$, to obtain $\hat{G}_m$, $m = 1, \ldots, M$.
- Approximate the probability of the feature by

$$\hat{\mathsf{P}}(f) = \frac{1}{M} \sum_{m=1}^{M} f(\hat{G}_m).$$

Simulation studies by Friedman et al. (1999) show that bootstrap based estimates are very reliable, so that features with high estimated probabilities are rarely false positives. Moreover, they correlate very well with the Bayesian posterior probabilities, shown in (5.5), even though structures are not weighted in proportion to their posterior probability.

We now turn our attention to our specific problem: the assessment of uncertainty in structure learning. We consider a set of variables $X_1, \ldots, X_p$. We are motivated by the pathway refining algorithm CK2. In particular, we use aforementioned strategy to address the issue of confidence in the learned structure. In order to do that, we define as the feature of interest the adjacency matrix

$$A = \{a_{ij}\}_{i=1,j=1}^{p} = \begin{cases} 1 & \text{if } X_i \in \mathrm{pa}_j; \\ 0 & \text{otherwise.} \end{cases}$$

There is a one to one correspondence between DAGs and adjacency matrices. In fact, the 1s in the columns of the matrix A correspond to parent sets; the column j gives parents of variable X_j. So, we set $f(G) = A$, and apply the bootstrap approach. We obtain a matrix $\hat{A} = \{p_{ij}\}$, where p_{ij} gives the proportion of the learned graphs in which the edge $X_i \to X_j$ is present. We can use this result to form the so-called *consensus* (or, average) DAG $\hat{G}$. We choose a threshold value c, and then include in $\hat{G}$ edges for which $p_{ij} > c$. The question is how to choose the threshold level in an optimal way. The high levels of c ensure the high reliability of the included edges, but might lead to missing some important links. On the other hand, by lowering c, we permit more spurious edges. One possible solution is to take into consideration the expected number of edges in the underlying biological network. It is common to assume that biological networks are sparse, and we are usually able to quantify this assumption rather accurately. This is the approach that we take in Section 6.2.

We interpret the consensus DAG as a graph that shows relationships that are robust with respect to small perturbations in the input data. It should be kept in mind that, as with a refined graph, the arrows lose a part of their physical interpretation. However, this is not worrisome, given that the structural discovery of the underlying mechanism is not our primary goal. Instead, we use this DAG as basis for the intervention analysis for simulating effects of gene silencing. We note that the similar idea of using a consensus network is also present in Shojaie et al. (2014). The main difference is that in their approach the resulting average graph is not necessarily a DAG, given that they consider multiple topological orderings when learning the structure from data.

5.4 Shrinkage

The starting point of many statistical procedures in graphical modelling is the estimation of the structured covariance matrix of a normally distributed random vector. In this Section, we tackle the estimation of the covariance matrix when the number of variables under consideration is of the same order of magnitude as the number of available statistical units, a situation encountered often in genomics data.

It is a well known fact that in high dimensional settings the usual estimate of the covariance matrix, the maximum likelihood estimate, or closely related sample covariance, may not be invertible. Even when it is invertible, it might be ill conditioned (the estimation error is amplified after inversion), a feature especially worrisome in graphical models where the concentration matrix plays a crucial role. To obtain an estimator that will be both invertible and well conditioned, Ledoit and Wolf (2004) proposed a shrinkage approach. In this case, the shrinkage estimator is a weighted average of the sample covariance matrix and the identity matrix (sometimes referred to as *the target*). By choosing the optimal shrinkage intensity (the weight given to the identity matrix), undesired properties of the two individual estimators, a high variance of the sample covariance and a bias of the identity matrix are balanced out. In addition to that, the estimate is always invertible, even when the sample covariance is a singular matrix. The strength of their approach lies in a theoretical result concerning the optimality of their estimator in the general asymptotics framework. While in a standard asymptotics framework, the number of variables p is fixed and the number of observations n goes to infinity, in general asymptotics both n and p go to infinity. The only condition is that the ratio p/n remains bounded. The authors note that this framework is more relevant to most real world applications where p and n are of comparable size, and report simulation studies showing that even for 20 observations and 20 variables asymptotic results apply.

We adapt the approach of Ledoit and Wolf to the graphical models setting. We do that by replacing the identity matrix with a different target, a target that encodes the presumed graphical structure. In other words, we propose to estimate the covariance matrix as an optimally weighted average of the the sample covariance matrix (the unconstrained estimate) and a target reflecting the graphical structure (constrained estimate). There are a number of ways to choose a target matrix encoding a graphical structure. We will propose three choices, each characterized by a different number of parameters.

Clearly, choosing the right weight (shrinkage intensity) to give to the target is essential. We follow the asymptotic approach of Ledoit and Wolf. Intuitively, when the sample covariance and the target estimate differ slightly the weight of the latter should be higher, while the large difference indicates that the lower dimensional target is misspecified.

The outline is the following. In Section 5.4.1, we review the standard results concerning estimation of a covariance matrix in Gaussian graphical models. In Section 5.4.2, we propose a new penalized estimator and discuss the estimation of the shrinkage parameter, while in Section 5.4.3, we study its properties via simulations.

5.4.1 Background

Let us assume that we have n observations, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, of a p-variate normal random vector $\mathbf{X}$, where possibly $p > n$, and, in addition, assume that there is an undirected graph $G = (V, E)$, such that the distribution of $\mathbf{X}$ is Markov with respect to G. Then the model assumed for the data is

$$(X_1, \ldots, X_p)' \sim \mathrm{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \boldsymbol{\mu} \in \mathbb{R}^p, \boldsymbol{\Sigma}^{-1} \in S^+(G),$$

where $S^+(G)$ is the set of all $p \times p$ symmetric positive definite matrices with null elements corresponding to the missing edges of G (see Lauritzen, 1996). In that case, the smaller the number of edges of G is, the greater is the number of zeros in the concentration matrix and thus the smaller is the number of free parameters to estimate.

Our goal is to estimate the covariance matrix $\boldsymbol{\Sigma}$. We report the result concerning the maximum likelihood estimate.

Theorem 5.4.1 *The maximum likelihood estimate of $\boldsymbol{\Sigma}$ exists if*

$$\mathbf{S} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})'$$

is positive definite, where $\bar{\mathbf{x}}$ is the sample mean vector. This happens with probability one when $n > p$. In the case it exists, the estimate is determined as the unique solution of the system of equations

$$\hat{\sigma}_{ii} = s_{ii}, \quad \hat{\sigma}_{ij} = s_{ij}, \quad i \in V, \{i, j\} \in E,$$

which also satisfies model constraints $\boldsymbol{\Sigma}^{-1} \in S^+(G)$.

Note that the condition $n > p$ is sufficient, but not necessary for the existence of the maximum likelihood estimate of $\mathbf{\Sigma}$. On the other hand, necessary, but not sufficient condition is $n > \max\{|c|, c \in \mathcal{C}\}$, where $\mathcal{C}$ is the set of cliques of G.

The above theorem specifies which equations to solve to find $\hat{\mathbf{\Sigma}}$ but does not give any advice on how to do so. In general, the system of equations needs to be solved by iterative methods. The algorithm usually employed is the Iterative Proportional Scaling (IPS), introduced by Dempster (1972), which iteratively adjusts the concentration matrix until the right zero pattern of the concentration matrix is obtained, all the while maintaining the equalities pertaining to the elements of the covariance matrix.

5.4.2 The proposal

In the situation when the sample size is small, so the maximum likelihood estimate of the covariance matrix either does not exist, or is expected to exhibit very high variance, we propose a shrinkage approach that incorporates explicitly information on the graphical structure. It amounts to shrinking the sample covariance matrix towards a constrained estimate embedding the conditional independence properties stored in the graphical structure. The proposed estimator is of the following form

$$\mathbf{U} = \lambda\mathbf{T} + (1-\lambda)\mathbf{S}, \tag{5.6}$$

where $\mathbf{S}$ is the sample covariance matrix, $\mathbf{T}$ is a target matrix and λ is the shrinkage parameter. In Ledoit and Wolf (2004) $\mathbf{T}$ is taken to be an identity matrix $\mathbf{I}$.

Our choice of matrix $\mathbf{T}$ is guided by two criteria: it should be positive definite, so that $\mathbf{U}$, being a convex combination of a positive semidefinite and a positive definite matrix, is also positive definite; and it should reflect the graphical structure of the considered distribution.

To define the first target, we consider first a matrix $\mathbf{T}_1^0$ parametrized by two parameters, one for the common variance and one for the common covariance

$$\mathbf{T}_1^0 = \begin{pmatrix} v & c & \cdots & c \\ c & v & \cdots & c \\ \vdots & \vdots & \ddots & \vdots \\ c & c & \cdots & v \end{pmatrix}.$$

Then, we impose constraints on its inverse such that $(\mathbf{T}_1^0)^{-1} \in S^+(G)$. As a result, we obtain a matrix $\mathbf{T}_1$, which satisfies both of our conditions.

A second target $\mathbf{T}_2$ could be obtained starting from the following initial matrix

$$\mathbf{T}_2^0 = \begin{pmatrix} v_1 & r\sqrt{v_1 v_2} & \cdots & r\sqrt{v_1 v_p} \\ r\sqrt{v_2 v_1} & v_2 & \cdots & r\sqrt{v_2 v_p} \\ \vdots & \vdots & \ddots & \vdots \\ r\sqrt{v_p v_1} & r\sqrt{v_p v_2} & \cdots & v_p \end{pmatrix},$$

which assumes constant correlation between variables but permits different variances. It requires estimation of $p+1$ parameters.

To obtain targets in practice, we estimate initial matrices from data and then pass them to the IPS algorithm to ensure that their inverses have the right zero structure.

The matrix $\mathbf{U}$ resulting from (5.6) will be invertible, but its inverse, in general, will not have the desired zero structure corresponding to the missing edges of G. Therefore in the last step, we apply the IPS algorithm to $\mathbf{U}$ to ensure that the model constraints are satisfied.

A key question in this procedure is the choice of the shrinkage parameter λ. If no prior knowledge is available to motivate a specific choice, it is not obvious what strategy should be adopted in eliciting the optimal value. We define the optimal value the one that leads to the estimator minimizing the expected distance from the true covariance matrix. As a measure of distance between two matrices we might consider, for example, the Frobenius distance, $\|\mathbf{A}-\mathbf{B}\|_F/p$, where $\|\mathbf{A}-\mathbf{B}\|_F = \sqrt{\mathrm{tr}[(\mathbf{A}-\mathbf{B})'(\mathbf{A}-\mathbf{B})]}$ Consider the squared distance between the shrinkage estimate and the true covariance matrix:

$$\begin{aligned} \mathrm{tr}[(\mathbf{U}-\Sigma)'(\mathbf{U}-\Sigma)] &= \mathrm{tr}[(\lambda\mathbf{T}+(1-\lambda)\mathbf{S}-\Sigma)'(\lambda\mathbf{T}+(1-\lambda)\mathbf{S}-\Sigma)] \\ &= \sum_{i=1}^{p}\sum_{i=1}^{p}[\lambda t_{ij}+(1-\lambda)s_{ij}-\sigma_{ij}]^2 \\ &= \sum_{i=1}^{p}\sum_{i=1}^{p}[\lambda(t_{ij}-\sigma_{ij})+(1-\lambda)(s_{ij}-\sigma_{ij})]^2 \end{aligned}$$

where we used the proposition on the trace of a product $\mathrm{tr}(\mathbf{AB}) = \sum\sum a_{ij}b_{ji}$, and in particular $\mathrm{tr}(\mathbf{A}'\mathbf{A}) = \sum\sum a_{ij}^2$. Let $R(\lambda)$ denote the expected value of the squared Frobenius distance from the true covariance matrix. We have:

$$\begin{aligned} R(\lambda) &= \mathsf{E}\{\mathrm{tr}[(\mathbf{U}-\boldsymbol{\Sigma})'(\mathbf{U}-\boldsymbol{\Sigma})]\} \\ &= \sum_{i=1}^{p}\sum_{i=1}^{p}\Big\{\lambda^2\Big[\mathsf{var}(t_{ij})+\mathsf{bias}^2(t_{ij})\Big]+(1-\lambda)^2\mathsf{var}(s_{ij})+ \\ &\quad 2\lambda(1-\lambda)\mathsf{cov}(t_{ij},s_{ij})\Big\}. \end{aligned}$$

The value of λ that minimizes the above function can be found by setting the first derivative of $R(\lambda)$ to zero. We obtain:

$$\begin{aligned}\lambda^* &= \frac{\sum_{i=1}^p \sum_{j=1}^p \left[\mathsf{var}(s_{ij}) - \mathsf{cov}(t_{ij}, s_{ij})\right]}{\sum_{i=1}^p \sum_{i=j}^p \left[\mathsf{var}(t_{ij}) + \mathsf{bias}^2(t_{ij}) + \mathsf{var}(s_{ij}) - 2\mathsf{cov}(t_{ij}, s_{ij})\right]} \\ &= \frac{\sum_{i=1}^p \sum_{j=1}^p \left[\mathsf{var}(s_{ij}) - \mathsf{cov}(t_{ij}, s_{ij})\right]}{\sum_{i=1}^p \sum_{j=1}^p \mathsf{E}\left(t_{ij} - s_{ij}\right)^2}. \qquad (5.7)\end{aligned}$$

We make a few observations about how the optimal shrinkage intensity, shown in (5.7), behaves.

- The smaller the variance of the elements of the sample covariance matrix, the smaller is the shrinkage intensity λ^*, implying that with an increasing sample size the effect of the target matrix diminishes and the shrinkage estimator converges to the maximum likelihood estimator.

- The smaller is the difference featured in the denominator, between the elements of the sample covariance and the target matrix, the more weight is given to the target matrix. Conversely, when the difference between the two is high, the shrinkage intensity decreases. This implicitly protects against a misspecified target.

- Finally, we note that λ^* is translation invariant but not scale invariant. The dependence on the scale, which is a general property of many regularization procedures (such as lasso or ridge regression), should be taken into account. When variances of variables are of different orders, the effect of shrinkage might vary considerably depending whether it is applied to a covariance or a correlation matrix.

The expression for the optimal shrinkage intensity λ^* contains quantities related to the distribution of the elements of the sample covariance matrix ($\mathsf{var}(s_{ij})$), as well as the joint distribution of the sample covariance and the constrained estimate ($\mathsf{cov}(t_{ij}, s_{ij})$ and $\mathsf{E}\left(t_{ij} - s_{ij}\right)^2$). To estimate these quantities we can adopt a bootstrap approach. We draw 200 samples with replacement from the original data and for each sample we compute both the sample covariance and the constrained estimate. On the basis of the bootstrap sample, we compute the variance of the elements of the sample covariance, as well as the covariance of the corresponding elements of the two estimators. By plugging in these estimates in (5.7), we can obtain the estimate of the optimal shrinkage intensity.

In some high dimensional datasets the cost of the bootstrap might be prohibitive. In those instances, we can consider a different target matrix containing no parameters to be estimated. This approach is closer to the standard shrinkage approach, where the sample covariance is shrank toward the identity matrix. Consider a simple initial matrix that reflects the presumed graphical structure. One can start from the matrix

$$\mathbf{T}_3^0 = \begin{pmatrix} 1 & 0.1 & \cdots & 0.1 \\ 0.1 & 1 & \cdots & 0.1 \\ \vdots & \vdots & \ddots & \vdots \\ 0.1 & 0.1 & \cdots & 1 \end{pmatrix}.$$

As before, one then passes $\mathbf{T}_3^0$ to the IPS algorithm to obtain $\mathbf{T}_3$. Since the target matrix is now deterministic, the expression (5.7) simplifies to

$$\lambda^* = \frac{\sum_{i=1}^p \sum_{j=1}^p [\mathsf{var}(s_{ij})]}{\sum_{i=1}^p \sum_{j=1}^p \mathsf{E}\left(t_{ij} - s_{ij}\right)^2}.$$

The expression in the denominator, featuring the expected value of the squared distance can be replaced with $\sum_{i=1}^p \sum_{j=1}^p \left(t_{ij} - s_{ij}\right)^2$ without affecting significantly the value of λ. As for the estimation of the variances featuring in the numerator, we proceed in the following way. Without loss of generality, we assume that considered variables have zero means. We define product random variables $W_{ij} = X_i X_j$, for $i, j = 1, \ldots, p$. We then have $\mathsf{E}(W_{ij}) = \mathsf{cov}(X_i, X_j)$. Let w_{kij} be the kth observation of the variable W_{ij}, $k = 1, \ldots, n$, and $\overline{W}_{ij}$ the corresponding sample mean. The sample covariance s_{ij} equals $n/(n-1)\overline{W}_{ij}$. We thus have

$$\begin{aligned} \mathsf{var}(s_{ij}) &= \frac{n^2}{(n-1)^2}\mathsf{var}(\overline{W}_{ij}) \\ &= \frac{n}{(n-1)^2}\mathsf{var}(W_{ij}). \end{aligned}$$

Substituting $\mathsf{var}(W_{ij})$ with its unbiased estimate we obtain

$$\widehat{\mathsf{var}}(s_{ij}) = \frac{n}{(n-1)^3} \sum_{k=1}^n \left(W_{kij} - \overline{W}_{ij}\right)^2.$$

5.4.3 Simulation studies

Performances of the proposed approach are studied via simulation in a number of different scenarios. For each scenario, we will compare three novel shrinkage estimators based on targets $\mathbf{T}_1, \mathbf{T}_2$ and $\mathbf{T}_3$ with the standard shrinkage estimator (where the target is the identity matrix $\mathbf{I}$). For purpose of comparison, we also include the sample covariance matrix $\mathbf{S}$. In fact, in a standard asymptotics framework, with increasing sample size the weight given to the shrinkage target should vanish, and so all considered shrinkage estimators should converge to the sample covariance. The sample covariance thus serves an additional purpose of a quick check of the convergence.

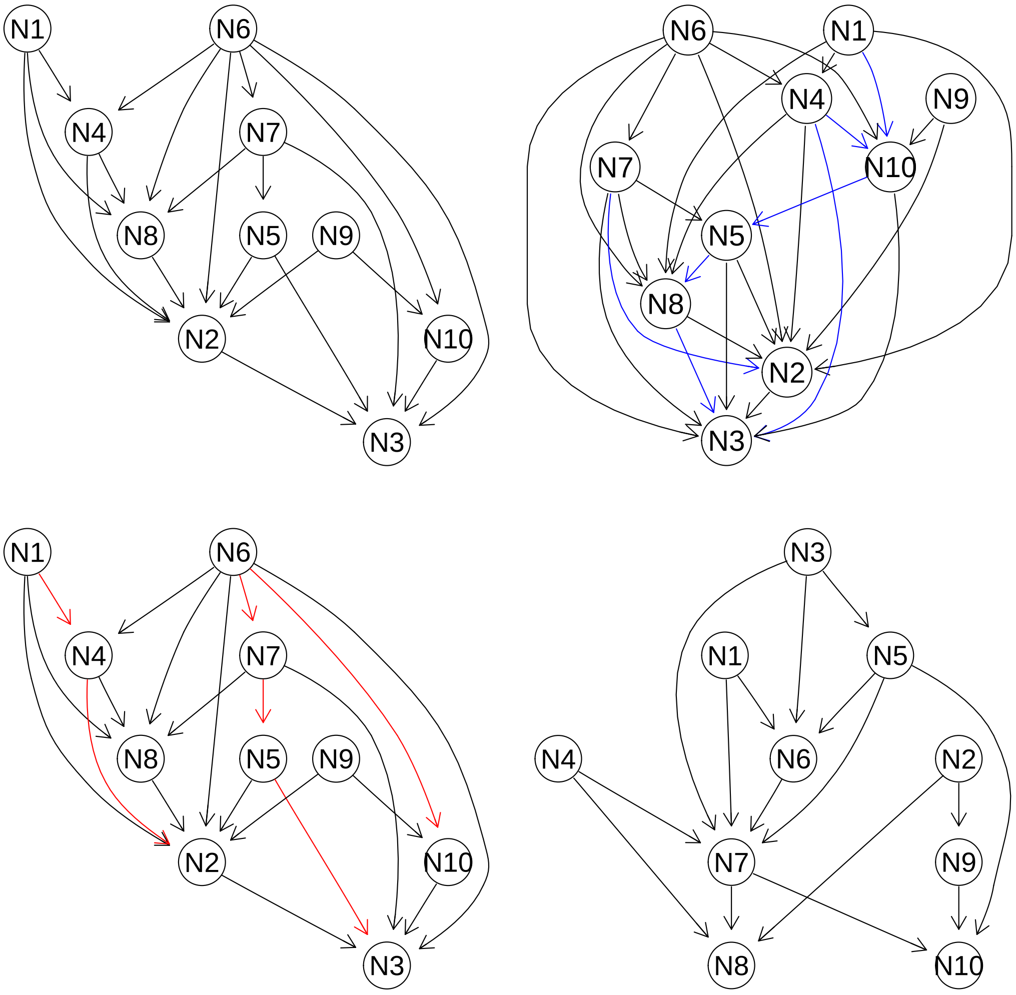

Fig. 5.2 DAGs used in the first simulation study (ordered clockwise from the upper left corner): the true structure used to generate data, the first misspecified structure with the added edges shown in blue, the second misspecified structure with the deleted edges shown in red, the third misspecified structure corresponding to a randomly generated DAG.

In the first simulation study, we consider a DAG whose structure is shown in Figure 5.2 in the upper left corner. This DAG, of 10 nodes and 21 edges, was taken from

Rau et al. (2013). Data were simulated according to equation (6.2). Parameters of the distribution are chosen as follows. Without loss of generality, variables have zero means. Non zero regression coefficients of the matrix **B** are drawn uniformly from $(-1, -0.25) \cup (0.25, 1)$. Residual variances are set to $\sigma_j^2 = j$, $j = 1, \ldots, 10$. The variance matrix is then computed according to (6.2). Once the parameters are chosen, 1000 datasets are simulated for four different sample sizes $n = 20, 30, 50, 100$. We consider the covariance matrix corresponding to the moralized graph and compute the root mean square error based on the Frobenius distance for each estimator. In this case target matrices $\mathbf{T}_1, \mathbf{T}_2$ and $\mathbf{T}_3$ are reflecting the true underlying structure. The shrinkage estimator based on the identity matrix **I** is computed via `cov.shrink` function of the `corpcor` R package (Schäfer et al., 2013). The results are shown in Table 5.4.

Table 5.4 Root mean square error (and standard deviation) of different covariance estimators; correctly specified structure

n	$\mathbf{T}_1$		$\mathbf{T}_2$		$\mathbf{T}_3$		**I**		**S**	
20	2.64	(0.82)	2.75	(0.80)	2.79	(0.82)	2.84	(0.75)	2.83	(1.02)
30	2.18	(0.68)	2.27	(0.67)	2.28	(0.68)	2.36	(0.67)	2.27	(0.75)
50	1.73	(0.55)	1.79	(0.55)	1.78	(0.55)	1.83	(0.54)	1.78	(0.60)
100	1.20	(0.36)	1.23	(0.36)	1.22	(0.35)	1.26	(0.36)	1.22	(0.36)

The best performing estimator in this scenario is the shrinkage estimator based on $\mathbf{T}_1$, characterized by two parameters. Interestingly enough, the second and the third target give comparable results, even though the latter involves considerably less computation. The third target, as well as the standard shrinkage target **I** perform only slightly better than the sample covariance **S** in terms of the expected loss, but their standard deviation is significantly lower. With 50 and 100 observations all estimators give similar results, implying that shrinkage approaches place almost no weight on the shrinkage target.

Next, we look at how the proposed shrinkage estimators perform when the target is misspecified. We consider the same datasets as above, but the three estimators shrink the sample covariance towards the “wrong” graphical structure. In the first instance, that structure is obtained by randomly adding seven edges to the true DAG (see the upper right corner of the Figure 5.2). The expected loss in this case is shown in Table 5.5. The results are almost identical as before demonstrating that adding edges does not burden the shrinkage estimators. This is not too surprising, since it is missing edges that carry information in graphical models; they characterize the conditional independence relations. By adding edges, we allow for dependence between

Table 5.5 Root mean square error (and standard deviation) of different covariance estimators: the first misspecified structure (additional edges).

n	$\mathbf{T}_1$		$\mathbf{T}_2$		$\mathbf{T}_3$		**I**		**S**	
20	2.65	(0.82)	2.72	(0.82)	2.76	(0.83)	2.84	(0.75)	2.83	(1.02)
30	2.20	(0.68)	2.24	(0.67)	2.25	(0.68)	2.36	(0.67)	2.27	(0.75)
50	1.73	(0.55)	1.77	(0.55)	1.77	(0.55)	1.83	(0.54)	1.78	(0.60)
100	1.21	(0.35)	1.23	(0.36)	1.22	(0.35)	1.26	(0.36)	1.22	(0.36)

conditionally independent variables, which amounts to passing to a broader family of models.

The exact opposite happens when we delete edges from a graph: we are forcing conditional independence between dependent variables (assuming that the distribution is faithful to the graph). In this case, we are dealing with a truly misspecified model. One such structure (see the lower left corner of the Figure 5.2) is obtained by randomly deleting six edges from the true DAG. Results are shown in Table 5.6. As expected, in this case the effect of misspecification on the three shrinkage estimators is visible, especially for larger sample size. Since the standard deviation remains the same (or slightly decreases), all the additional error comes from the bias caused by the misspecified model.

Table 5.6 Root mean square error (and standard deviation) of different covariance estimators: the second misspecified structure (missing edges).

n	$\mathbf{T}_1$		$\mathbf{T}_2$		$\mathbf{T}_3$		**I**		**S**	
20	2.74	(0.79)	2.85	(0.77)	2.88	(0.78)	2.84	(0.75)	2.83	(1.02)
30	2.32	(0.64)	2.41	(0.63)	2.41	(0.64)	2.36	(0.67)	2.27	(0.75)
50	1.92	(0.51)	1.97	(0.50)	1.96	(0.51)	1.83	(0.54)	1.78	(0.60)
100	1.48	(0.30)	1.50	(0.31)	1.49	(0.31)	1.26	(0.36)	1.22	(0.36)

The last scenario investigating the effects of a misspecified target considers a randomly generated graph on 10 nodes shown in the lower right corner of the Figure 5.2. Given that the target in this scenario is completely misspecified, we expect the worst behaviour of the novel estimators. The results in Table 5.7 confirm our intuition, but nevertheless, the mean square error is still reasonably low, implying that the choice of the shrinkage parameter λ is guarding us from giving too much weight to the target when data do not support its structure.

In the second simulation study, we consider a DAG describing a particular biological pathway, the B cell pathway. The DAG derived from this pathway contains 35 nodes and is shown in Figure 5.3. This graph, alongside measurements of the ex-

Table 5.7 Root mean square error (and standard deviation) of different covariance estimators: the third misspecified structure.

n	$\mathbf{T}_1$		$\mathbf{T}_2$		$\mathbf{T}_3$		$\mathbf{I}$		$\mathbf{S}$	
20	2.73	(0.73)	2.77	(0.76)	2.87	(0.72)	2.84	(0.75)	2.83	(1.02)
30	2.38	(0.61)	2.39	(0.61)	2.47	(0.61)	2.36	(0.67)	2.27	(0.75)
50	2.01	(0.47)	2.01	(0.47)	2.04	(0.47)	1.83	(0.54)	1.78	(0.60)
100	1.62	(0.28)	1.63	(0.28)	1.63	(0.28)	1.26	(0.36)	1.22	(0.36)

pression levels of the participating genes, is an example featured in the R package `topologyGSA` (Massa and Sales, 2013). We use these expression measurements to estimate the parameters of the DAG, and then use the estimated model to simulate 100 datasets for each of the considered sample sizes. As before, we compare three novel shrinkage estimators to the standard shrinkage estimator and the sample covariance matrix. The results are shown in the Table 5.8.

Table 5.8 The B cell pathway model: root mean square error (and standard deviation) of different covariance estimators, multiplied by 10^2.

n	$\mathbf{T}_1$	$\mathbf{T}_2$	$\mathbf{T}_3$	$\mathbf{I}$	$\mathbf{S}$
10	4.33 (0.51)	4.69 (0.74)	5.29 (0.68)	4.88 (0.45)	12.53 (1.59)
20	3.38 (0.41)	3.38 (0.40)	4.25 (0.38)	4.03 (0.25)	8.67 (0.77)
30	3.10 (0.40)	3.04 (0.43)	3.86 (0.41)	3.77 (0.30)	7.11 (0.72)
50	2.55 (0.35)	2.47 (0.32)	3.16 (0.28)	3.29 (0.22)	5.38 (0.37)
100	2.02 (0.27)	2.05 (0.23)	2.39 (0.20)	2.82 (0.16)	3.84 (0.28)
500	0.86 (0.12)	0.97 (0.14)	0.92 (0.12)	1.56 (0.09)	1.70 (0.10)
1000	0.57 (0.08)	0.64 (0.09)	0.60 (0.08)	1.14 (0.06)	1.20 (0.06)
2000	0.40 (0.05)	0.44 (0.06)	0.42 (0.06)	0.84 (0.04)	0.85 (0.04)

The first shrinkage estimator seems to give the best results in this simulation study as well. The estimator based on the second shrinkage target performs comparably well, sometimes slightly better than the first one. An interesting observation is that for the small sample sizes the standard shrinkage estimator outperforms the competing shrinkage estimator (the third target). This might be due to the very low sample size compared to the number of free parameters. Nevertheless, it would be of interest to study the conditions under which it is recommendable to apply the standard shrinking towards the identity matrix, even when considering structured matrices.

5.4.4 Discussion

In this Section, we studied the problem of estimation of structured covariance matrices associated with Gaussian graphical models. We proposed three novel shrinkage estimators, applicable in $p > n$ setting. We performed simulation studies that showed that they perform very well when the underlying structure is correctly specified, as well as when the true structure is a subgraph of an assumed graph. Their performance is affected when an underlying structure is misspecified. Nevertheless, the data driven procedure for eliciting the tuning parameter guards against placing a lot of weight on the misspecified target. Moreover, one of the conclusions drawn from the reported simulation studies is that using shrinkage estimators is recommended even when the sample covariance matrix is regular. Shrinkage estimators introduce bias, but when the sample size is small, the error is usually smaller than the one caused by the high variance of an unbiased estimator.

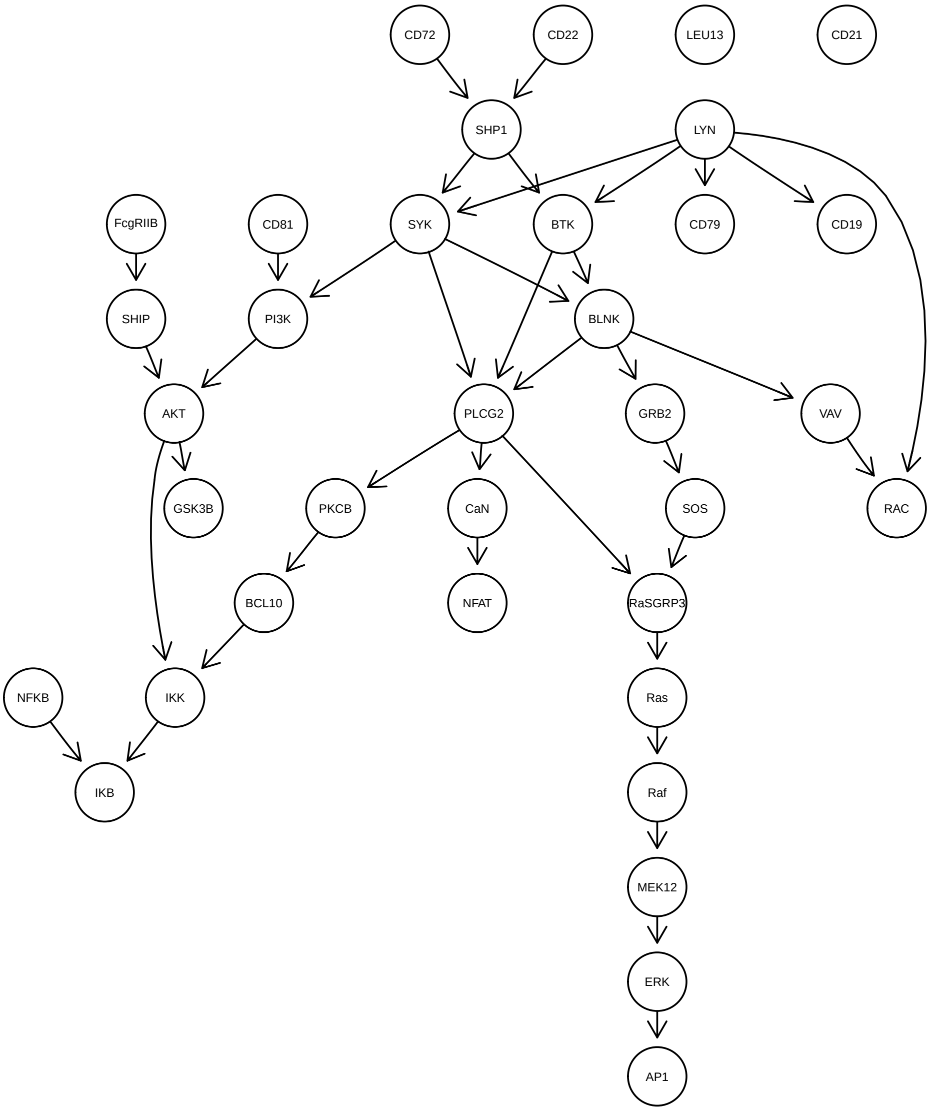

Fig. 5.3 DAG representing the B cell pathway.

Chapter 6

Intervention analysis

In the previous chapter, we studied the problem of finding a graphical structure, coherent with biological knowledge and reasonably supported by the data, able to provide a framework for gene silencing. Here, we deal with performing gene silencing and measuring its effects. To this aim, we will consider gene silencing as an intervention in a directed acyclic graph.

An intervention is an external manipulation of a subset of variables. Generally speaking, the interest usually lies in predicting how the system will change in response to the manipulation. In our case, the system is described through a suitable DAG. It is tempting to interpret directed arrows as implications of cause and effect relations, but this interpretation is not warranted in the standard graphical models framework. To illustrate this point, consider two dependent variables X and Y and assume that X precedes Y in time. The natural representation of their joint distribution would be $X \rightarrow Y$. However, there is nothing in the definition of the graphical model that disqualifies the model $X \leftarrow Y$. Despite it going against our intuition, the two models are on equal standing when it comes to representing their joint distribution. If the graphical representation is in line with a potential causal explanation, intervention becomes a key ingredient of causal inference. In fact, it is the natural causal interpretation of edges that made DAGs central to much of the work in causal inference. According to Lauritzen (2001) the connection between graphical models and causal concepts brought up a renewed interest in the study of causality. But, it should be always clear that whether such an interpretation is justifiable depends on subject matter considerations and should be addressed on a case to case basis.

For a detailed treatment of causal inference based on graphical models we refer the reader to Pearl (2000), Spirtes et al. (2000), Shafer (1996). We conclude with a quote from the highly interesting book by Shipley (2002):

> "In fact, with few exceptions, correlation does imply causation. If we observe a systematic relationship between two variables, and we have ruled out the likelihood that this is simply due to a random coincidence, then something must be causing this relationship... A more accurate sound bite for introductory statistics would be that a simple correlation implies an unresolved causal structure, since we cannot know which is the cause, which is the effect, or even if both are common effects of some third, unmeasured variable."

In Section 6.1, we give a brief overview of intervention calculus and in Section 6.2, we apply the proposed approach of simulating gene silencing to the data from *Drosophila Melanogaster* experiment.

6.1 Intervention calculus

We define a DAG G to be *causal* for a set of random variables $\{X_v, v \in V\}$, if the joint density factorizes with respect to G

$$f(x) = \prod_{v \in V} f[x_v \mid \mathrm{pa}(x_v)]$$

and it further holds for any $A \subset V$ that

$$\begin{aligned} f(x \mid\mid x_A^*) &= \left. \prod_{v \in V \setminus A} f[x_v \mid \mathrm{pa}(x_v)] \right|_{x_A = x_A^*} \\ &= \left. \frac{f(x)}{\prod_{a \in A} f\left[x_a^* \mid \mathrm{pa}(x_a)\right]} \right|_{x_A = x_A^*}, \end{aligned}$$

where the notation $f(x \mid\mid x_A^*)$ denotes the post-interventional distribution, i.e., the density of the distribution of X after manipulating X_A, forcing it to assume the value x_A^*.

The above formula is usually referred to as the *intervention formula* and can be found in various forms in Pearl (2000) and Spirtes et al. (2000). Note that this is only one possible definition of a causal DAG, see Spirtes et al. (2000) for an alternative definition in terms of direct causes.

The so-called intervention formula provides a recipe for determining the effects of interventions. Under the assumption that a DAG underlying the joint distribution is causal, the post-intervention distribution of the system can be found in terms of the pre-intervention distribution. As was noted before, the assumption that a DAG is causal is strong and not verifiable mathematically. It has to be justified in every particular instance using subject matter knowledge. In fact, much of the controversy about causal inference is not associated to the technical aspect of mathematical calculus, but, rather, to the underlying assumptions crucial for validity and interpretation of the obtained results.

The intervention that is most commonly considered is the one that manipulates a certain variable and sets its value to a constant. In general, we could be interested in more general types of interventions, be that a simultaneous acting on more than one variable or changing the distribution of the target variable, not necessarily setting it to a specific value. Here, we will focus on the latter.

Consider random variables $\{X_i, i = 1, \ldots, p\}$, so that $V = \{1, 2, \ldots, p\}$ and that their joint normal density factorizes with respect to a DAG G,

$$f(x_1, \ldots, x_p) = \prod_{i=1}^{p} f\left[x_i \mid \mathrm{pa}(X_i)\right],$$

Further, assume that G is causal for this system of variables. Let there be an intervention targeting variable X_k, $k \in \{1, 2, \ldots, p\}$, changing its marginal distribution to $f^*(x_k)$. According to the intervention formula, we can obtain the post-intervention distribution in terms of the pre-intervention conditional distributions and the changed marginal distribution of the targeted variable:

$$f^*(x_1, \ldots, x_p) = f^*(x_k) \prod_{i \neq k} f[x_i \mid \mathrm{pa}(X_i)].$$

Sometimes it might be useful to express post-intervention joint distribution $f^*(\cdot)$ in terms of the pre-intervention distributions $f(\cdot)$ and the post-intervention marginal distribution of the variable affected by the intervention, i.e., $f^*(x_k)$

$$f^*(x_1, \ldots, x_p) = \frac{f^*(x_k)}{f[x_k \mid \mathrm{pa}(X_k)]} f(x_1, \ldots, x_p).$$

We now look at estimation of effects of interventions. DAGs can be equivalently defined in terms of structural equations

$$X_i = \alpha_i + \boldsymbol{\beta_i}^T \mathrm{pa}(X_i) + \epsilon_i, \qquad i = 1, 2, \ldots, p;$$

where $\epsilon \sim \mathsf{N}(0, \sigma_i^2)$ is the random disturbance, $\boldsymbol{\beta}_i$ is the vector of regression coefficients, and α_i is the base level or an intercept. Assume that variables are topologically ordered with respect to G. Then, the matrix of regression coefficients $\mathbf{B} = \{\beta_{ij}\}_{i,j=1}^{p}$ will be strictly upper triangular. The matrix representation of the model thus is

$$\mathbf{X} = \boldsymbol{\alpha} + \mathbf{X}\mathbf{B} + \boldsymbol{\epsilon}, \tag{6.1}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)'$ and $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_p)'$. By substituting this equality recursively $p-1$ times on the right hand side of (6.1), we obtain

$$\mathbf{X} = \boldsymbol{\alpha}(\mathbf{I} + \mathbf{B} + \ldots + \mathbf{B}^{p-1}) + \mathbf{X}\mathbf{B}^{p} + \boldsymbol{\epsilon}(\mathbf{I} + \mathbf{B} + \ldots + \mathbf{B}^{p-1}).$$

Since matrix $\mathbf{B}$ is strictly upper triangular, the power matrix $\mathbf{B}^p$ will be a zero matrix. This further implies that the sum $\sum_{k=0}^{\infty} \mathbf{B}^k$ has finitely many non zero terms (and equals $\mathbf{I} + \mathbf{B} + \ldots + \mathbf{B}^{p-1}$). An established result in matrix algebra states that when such a sum converges, its sum equals $(\mathbf{I} - \mathbf{B})^{-1}$ (Harville, 2008). Let $\mathbf{L} = (\mathbf{I} - \mathbf{B})^{-1}$. We can then express (6.1) as

$$\mathbf{X} = \boldsymbol{\alpha}\mathbf{L} + \boldsymbol{\epsilon}\mathbf{L}.$$

and can specify the model as

$$\mathbf{X} \sim \mathsf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \qquad \text{where } \boldsymbol{\mu} = \boldsymbol{\alpha}\mathbf{L}, \quad \boldsymbol{\Sigma} = \mathbf{L} \times \operatorname{diag}\left\{\sigma_i^2\right\}_{i=1}^{p} \times \mathbf{L}'. \tag{6.2}$$

This model representation is useful when investigating effects of interventions in terms of effects on the mean and variance. To find parameters of the post-intervention distribution, it is sufficient to replace an equation specifying the distribution of the variable X_k with an intervention specification and then recompute the model parameters according to the preceding formula.

For us, it will be useful to summarize intervention effects by a set of univariate measures, the so-called silencing effects. The silencing effect δ_i gives the change in the mean of X_i given the unit decrease in the mean of the targeted X_k

$$\delta_i = \mathsf{E}(X_i \mid X_k = \alpha + 1) - \mathsf{E}(X_i \mid X_k = \alpha).$$

Since we assume a multivariate normal distribution, δ_i is independent of α. The vector of silencing effects can be easily found through matrix algebra. Matrix

$$\mathbf{L} = (\mathbf{I} - \mathbf{B})^{-1} \tag{6.3}$$

is upper triangular having 1s on the main diagonal. The mean of the variable X_i is now expressed as a linear combination of components of $\boldsymbol{\alpha}$

$$\begin{aligned} \mathrm{E}X_i &= \sum_{j=1}^{p} \alpha_j l_{ji} \\ &= \sum_{j=1}^{i-1} \alpha_j l_{ji} + \alpha_i \\ &= \alpha_k l_{ki} + \sum_{j=1, j \neq k}^{i-1} \alpha_j l_{ji} + \alpha_i. \end{aligned} \tag{6.4}$$

If we assume that the intervention affects only the base level of the targeted variable leaving all other αs unchanged, the silencing effect δ_i equals l_{ki}, where l_{ki} stands for the element of $\mathbf{L}$ in the kth row and ith position. The vector of silencing effects corresponding to an intervention on X_k is thus given by the kth row of the matrix $\mathbf{L}$.

6.2 Application to real data

In this section, we apply the proposed approach for the prediction of effects of gene silencing to the data from the *Drosophila Menalogaster* experiment. As already described in Chapter 3, the data consist of two sets of 14 observations of 12 genes, the first set corresponding to the treatment (knockdown) group and the second set corresponding to the control group. This experiment provides an excellent opportunity to access the performance of our approach, since we are able to compare model based predictions with observed effects of gene silencing. In order to that, we build a statistical model using only observations from the control group, and then compare our predictions with the actually observed changes seen in the knockdown group.

Guided structural learning. The first step consists of refining the pathway. We start from a DAG, containing 12 genes, constructed by hand by biologists on the basis of the WNT pathway shown in Figure 3.1. We call this a pathway DAG (see left panel in Figure 6.1). Next, we find a topological ordering of this DAG, using the `topological.sort` function of the `igraph` R package (Csardi and Nepusz, 2006). We pass the obtained ordering along with gene expression data of the control group to the CK2 algorithm (see Section 5.1.2). The resulting network is shown in the right panel of Figure 6.1.

Three issues are worth mentioning. First is the non-uniqueness of the topological ordering. The result of the refining clearly depends on the chosen order, and so the

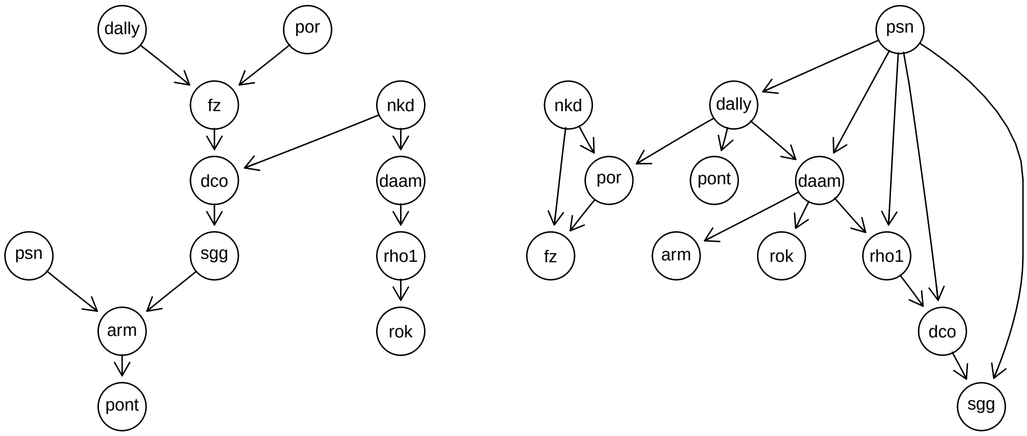

Fig. 6.1 The pathway DAG (left) and the refined DAG obtained by CK2 (right).

conclusions based on the refined DAG are conditional on the ordering. One possible remedy is to combine results from several orderings compatible with the pathway DAG. In our study this is not necessary, since the part of the network downstream from the silenced gene is stable across different orderings, and thus the predictions are not affected by the choice of the initial ordering. The second issue is related to the choice of the structure learning algorithm. We saw in Section 5.2 that approaches working with categorized variables showed more promise than CK2 algorithm. However, we believe that the biological intervention is more precisely reproduced on continuous measurements. This statement was verified by performing intervention analysis on pathways refined by different algorithms considered in Section 5.2. The results obtained by CK2, not reported here, were superior with respect to the results of approaches that used categorized measurements. Third, refinement can be used to find new hypothesis to be tested or as a guidance for future silencing experiments. For instance, the refined graph can signal a possible inaccuracy in the representation of molecular pathways. In this study, the levels of the *dally* gene were increased dramatically after silencing, an effect that could not be explained neither by the original pathway nor by refined DAG. This led us to look for a possible explanation in the literature. It happens that this gene, a participant of the WNT pathway, is itself regulated by the WNT pathway, so that there is a feedback loop not depicted in the KEGG representation. This explains its behaviour in the knockdown group, and inability of our models to predict its values.

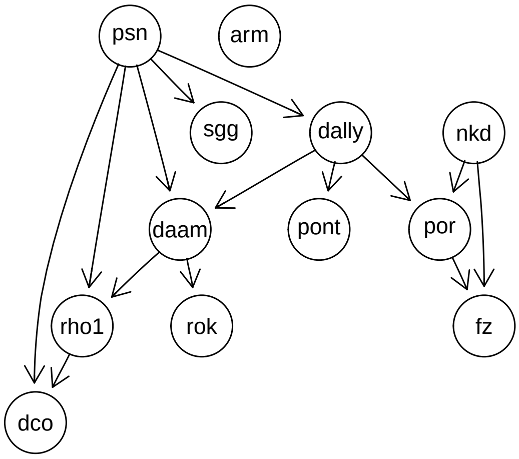

Fig. 6.2 The consensus DAG.

Model uncertainty - the consensus DAG. To account for the variability in the learned model structure, we resort to the bootstrap strategy as described in Section 5.3. We sample 2000 samples with replacement from the original data and then estimate the structure for every sample using the CK2 algorithm. This allows us to assign an empirical measure of uncertainty to every plausible edge (an edge is plausible when it is in line with the topological ordering) by counting how many times out of 2000 it is discovered by the algorithm. On the basis of this result, we construct a consensus DAG, which consists of all the edges that were discovered at least $c\%$ of times, where c is an appropriately chosen threshold level. Obviously, the threshold level controls the number of edges in the resulting DAG. Subject matter considerations tell us that networks of genes are expected to be sparse, and in this particular case the number of edges is expected to approximately match the number of genes. The choice $c = 0.5$ leads to a structure with 11 edges, shown in Figure 6.2.

Prediction of gene silencing effects. To predict the effects of silencing, we applied intervention calculus to the consensus DAG. To this aim, we assume that the consensus DAG is causal. We already mentioned that such DAG cannot fully represent the underlying biological system. Despite this considerations, discussions with biologists confirmed that orientation of edges is consistent with biological expectations, up to the feedback loops not representable in this structure.

After estimating the parameters of the consensus model, the vector of silencing effects was computed according to (6.3). We recall that the intervention is assumed to have a direct effect only on the mean of the targeted gene. Under this assumption, the

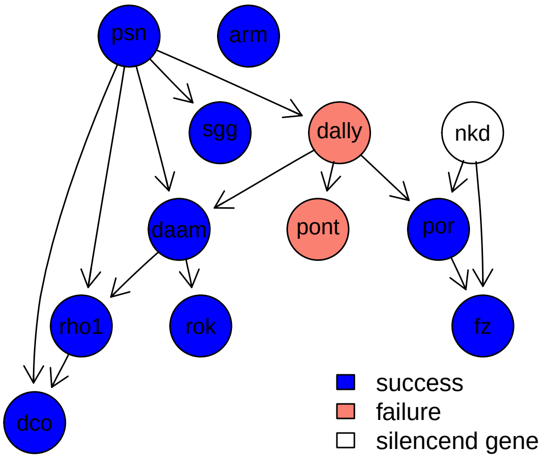

Fig. 6.3 The predictions based on the consensus DAG. We mark as successes the genes in which the hypothesis of equality of the predicted mean after intervention and the mean in the knockdown condition is not rejected.

distribution of genes after the knockdown remains multivariate normal with a shifted mean, and an unchanged diagonal of the covariance matrix. To compute the mean of each gene after the intervention, we plug in (6.4) for α_k the sample mean of the *nkd* gene in the knockdown group. To evaluate the goodness of prediction, we compare with a two sample test the mean of each variable after the intervention and the mean of the same variable in the knockdown condition. We assume the two populations to be heteroscedastic, and we estimate the variances taking into account technical variability as in Section 3.3. The asymptotic p-values of these tests are shown in Table 6.1. We see that predictions of *dally* and *pont* have very low p-values, implying that the predictions of silencing effects did not correspond to that observed in the knockdowns. The prediction of the remaining 9 genes is successful, in the sense that the hypothesis of equality of the means is not rejected. This is also shown in Figure 6.3, where we mark as successes the genes for which the hypothesis of equality of means was not rejected.

We also compute the effects of gene silencing for two other models: the pathway DAG and the refined DAG. The results are shown in Figure 6.4. We note that by performing pathway refining the number of successes went from 5 to 9. The important conclusion from this example is that both refined DAG and the consensus DAG show considerable improvement in the predictive accuracy with respect to the pathway DAG.

An alternative way to make use of the bootstrap strategy is the following. Instead

Table 6.1 *Drosophila melanogaster* experiment: p-values of tests of the equality of predicted means and means in the knockdown condition.

	pathway	average DAG
psn	0.34	0.34
dally	3.2×10^{-14}	3.2×10^{-14}
por	0.47	0.22
daam	1.3×10^{-5}	0.28
fz	6.4×10^{-3}	0.30
rho1	8.7×10^{-5}	0.45
dco	0.08	0.42
rok	0.06	0.07
sgg	0.17	0.27
arm	0.02	0.49
pont	3.6×10^{-11}	8.1×10^{-07}

of using a consensus DAG to predict effects of silencing, we can compute the silencing effects at each bootstrap replication. In this way, we obtain a bootstrap distribution of silencing effects. That distribution is a mixture of two components, one corresponding to a random variable degenerate at zero and the other to a random variable with a non zero mean. An example showing the bootstrap distribution of *daam* silencing effects is shown in Figure 6.5. The degenerate component of the distribution corresponds to the bootstrap replications in which no direct path between the *nkd* and *daam* is estimated (models in which *daam* is not a descendant of *nkd*). In these cases, the silencing has no effect on *daam* and the silencing effect δ will be zero. In Table 6.2, for each gene, we give the number of times (out of 2000 replications) in which the direct path was not present. In addition to that, Table 6.2 also shows the bootstrap percentile confidence intervals for the mean of each gene after silencing. In constructing this interval, we took into consideration only the non degenerate component of the bootstrap distribution. We note, that the *psn* gene is not included in this Table, since it is the first gene in the topological ordering, and therefore never a descendent of *nkd*. The conclusions as to the prediction of individual genes remain the same as the ones of the consensus DAG.

To conclude, the bootstrap strategy in this approach offers two distinct and equally interesting pieces of information: the percentage of cases when no effect was found, and the interval for the predicted mean when the effect was observed. Nevertheless, the check about plausibility of the causal interpretation of the DAGs resulting at each bootstrap iteration becomes impossible, so that the interpretation of results might be more delicate.

Table 6.2 Summary of the bootstrap distribution of silencing effects: the number of bootstrap samples in which the estimated model implied no effect of *nkd* silencing, the confidence interval for the mean after silencing, and for reference, estimated means of the genes in the knockdown group $\bar{Y}_k$.

	$\#\delta = 0$	confidence interval	$\bar{Y}_k$
dally	1590	(1.05, 1.60)	5.46
por	560	(0.68, 0.97)	0.79
daam	1393	(2.49, 4.26)	3.55
fz	82	(0.44, 0.75)	0.55
rho1	1578	(20.30, 63.79)	43.26
dco	1597	(3.09, 6.20)	4.58
rok	1401	(1.90, 3.38)	3.91
sgg	1531	(4.36, 6.37)	5.24
arm	1386	(24.45, 40.70)	37.34
pont	645	(1.21, 1.74)	2.49

6.3 Notes and observations

Causal inference. The language of causal inference has at least three formalisms: potentials outcomes (Rubin 1974, Holland 1986), functional equations (Pearl 2000) and graphical models (Dawid 2002, Lauritzen 2001 and Spirtes et al. 2000). None of the approaches dominates the others in terms of popularity or applicability. Depending on the particular question at hand, it is usually the case that one of them is more suitable than the others, providing easier and more elegant representation.

Functional causal models. When causal models were first used in genetics (Wright, 1934) or in econometrics (Haavelmo, 1943), they both relied on functional equations representation. Pearl (2000) formalised the notion of a functional causal model. In this specification, the causal model consists of a set of equations of the form

$$X_v = g_v\left[\text{pa}(x_v), U_v\right], \tag{6.5}$$

where $g_v, v \in V$ are deterministic functions associated to physical mechanisms relating X_v to its immediate causes (its parents) and U_v is a random disturbance, introducing a stochastic component to the model. When each equation represents an autonomous mechanism, the system of equations of the form 6.5 is called a system of structural

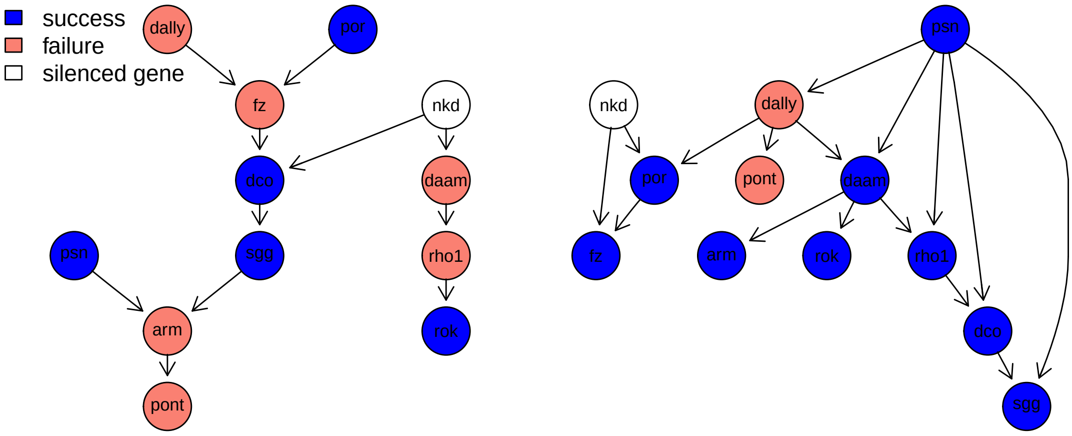

Fig. 6.4 The predictions of the mean expression values after the silencing of the *nkd* gene based on the pathway DAG (left) and the refined DAG obtained by CK2 (right). We mark as successes the genes in which the hypothesis of equality of the predicted mean after intervention and the mean in the knockout condition is not rejected.

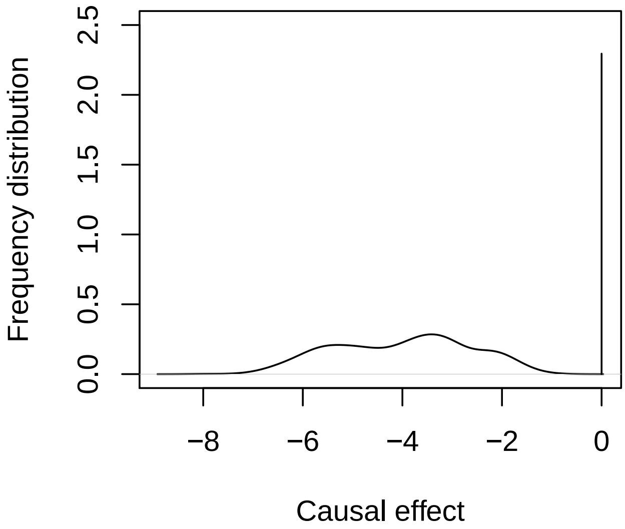

Fig. 6.5 The bootstrap distribution of silencing effects on *daam*.

equations. If, in addition, each mechanism determines the value of a single variable, the system is called a causal structural system. The model is a nonparametric, nonlinear generalisation of the linear structural equations model SEM

$$X_v = \sum_{k \neq v} \alpha_{kv} X_k + U_v, \quad v \in V,$$

that we use here, and that has become a standard tool in statistical analysis in the social sciences and econometrics. Here, non zero α_{kv} correspond to the set of parents of X_v in 6.5.

An important assumption of this model, specifying that individual mechanisms represented by functions g_v are independent or autonomous, implies that changing one mechanism is not affecting others. This can be seen as a modularity property which ensures that the behaviour of the system after an external intervention can be predicted.

In this framework, studying interventions is fairly straightforward: if a mechanism governing the distribution of a variable X_a is changed, the corresponding function $g_a(\cdot)$ is replaced by a $g^*_a(\cdot)$. Graphically, this can be represented as deleting arrows going into X_a, since its parents no longer affect its value. Pearl calls this post-intervention graph a *mutilated graph.* The new mechanism is not limited to being a constant function, so more complex interventions are easily implemented in this framework.

Chapter 7

Conclusions

The problem of prediction of the effects of silencing is notoriously complex, as probably is the majority of problems regarding biological systems. Here, we tackled some of the issues that we encountered in the past two years in an attempt to provide a suitable statistical framework to accommodate such problem. Even though not even a double amount of time would bring us to a closure and a complete solution, our first results, presented in this thesis, show promise.

In our approach, we first build a graphical model for a set of genes from an underlying pathway. We then apply the CK2 algorithm to refine the graphical structure. Next, we perform intervention analysis on the refined graph, taking into account the uncertainty in the refined model. Finally, we obtain confidence intervals for the mean expression values of genes after silencing. Many open problems await future research, and we mention some of them here.

The first step of the proposed approach consists of building a DAG for the set of genes on the basis of the information provided in the pathway. Although this might not seem like a difficult task, it is certainly not an automatic procedure. A close collaboration with domain experts is necessary so that the chosen DAG faithfully reflects the relations depicted in the pathway. One difficulty is presented by the cycles. Cycles and feedback loops are very typical of biological systems, and the assumption of acyclicity, typical of approaches based on directed graphs, is a simplifying one. One possible solution is to remove the weakest edge of the cycle based on expression data with minimum expression profile correlation between nodes (see Edwards et al., 2012). However, dropping edges usually has a biological cost. A different strategy would be to replace DAGs with more general graphs. We are currently looking into chain graphs, for which Lauritzen and Richardson (2002) proposed causal interpretation. An alternative strategy, when temporal data are available, is offered by dynamic Bayesian networks (Perrin et al., 2003).

Another difficulty in translation of pathways to graphs is associated with compounds and complexes. Compounds are not measured by microarray experiments and thus should be appropriately removed. Nodes composed by multiple elements can be protein complexes (proteins linked by protein-protein interactions) or groups containing alternative members (like gene families, genes with similar biochemical functions Sales et al., 2012). Two different strategies could be followed to solve this issue:

- selection of one variable representative of the complex (the mean of all gene products, the gene with the highest expression, the first principal component, etc.);
- expansion of the node into multiple nodes. In general, protein complexes should be expanded into cliques, while gene families should be expanded without connections among them.

The key ingredient in our approach is the topological ordering of the pathway DAG. All results drawn from the structure learned by CK2 are thus conditional on the chosen ordering. In our fruit fly experiment, this did not play a significant role, since the subgraph downstream from the silenced gene was small and stable across different orderings. In general, a possible approach could be to consider a sample of topological orderings consistent with the pathway DAG. In combination with the bootstrap approach to assess the uncertainty in the graphical structure, this would lead to a consensus graph that is independent of a particular choice of the topological ordering.

We mention one potential extension of this approach concerning experimental design. Careful experimental design could substantially alleviate the cost of silencing experiments. By jointly modelling intervention and wild type data, we can study adaptive experimental design, where gene silencing experiments are performed sequentially, so that after every step the gene whose silencing would be the most informative is chosen for the successive experiment.

One of the biggest challenges when dealing with gene expression data is separating the technical noise from the biological signal. The ideal way to assess the extent of the technical artefacts is to perform experiments that allow for estimation of the technical variability. When such an estimate is available, an open question remains as to how to take it into account and incorporate in the structure learning procedures.

References

Bader, G. D., Cary, M. P., and Sander, C. (2006). Pathguide: a pathway resource list. *Nucleic Acids Research*, 34(Database issue):D504–6.

Bansal, M., Belcastro, V., Ambesi-Impiombato, A., and Di Bernardo, D. (2007). How to infer gene networks from expression profiles. *Molecular systems biology*, 3(1).

Bondy, A. and Murty, U. (2010). *Graph Theory*. Graduate Texts in Mathematics. Springer.

Bonissone, P., Henrion, M., Kanal, L., and Lemmer, J. (1991). Equivalence and synthesis of causal models. In *Uncertainty in artificial intelligence*, volume 6, page 255.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Chickering, D. M. (2003). Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554.

Chickering, D. M. and Meek, C. (2002). Finding optimal bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 94–102. Morgan Kaufmann Publishers Inc.

Cooper, G. F. and Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347.

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, Complex Systems:1695.

Cussens, J. (2011). Bayesian network learning with cutting planes. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 153–160, Corvallis, Oregon. AUAI Press.

Dawid, A. P. (1979). Conditional independence in statistical theory. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–31.

Dawid, A. P. (2002). Influence diagrams for causal modelling and inference. *International Statistical Review*, 70(2):161–189.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, 28:157–175.

Djordjilović, V., Chiogna, M., Massa, S., and Romualdi, C. (2013). Graphical modeling for gene set analysis: a critical appraisal. *submitted to Biometrical Journal*.

Edwards, D., Wang, L., and Sorensen, P. (2012). Network-enabled gene expression analysis. *BMC Bioinformatics*, 13(1):167.

Flockhart, I. T., Booker, M., Hu, Y., McElvany, B., Gilly, Q., Mathey-Prevot, B., Perrimon, N., and Mohr, S. E. (2012). Flyrnai. org—the database of the drosophila rnai screening center: 2012 update. *Nucleic acids research*, 40(D1):D715–D719.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Fraley, C., Raftery, A. E., Murphy, T. B., and Scrucca, L. (2012). *mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation.*

Friedman, N., Goldszmidt, M., and Wyner, A. (1999). Data analysis with bayesian networks: A bootstrap approach. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 196–205. Morgan Kaufmann Publishers Inc.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620.

Glass, L. and Kauffman, S. A. (1973). The logical analysis of continuous, non-linear biochemical control networks. *Journal of theoretical Biology*, 39(1):103–129.

Haavelmo, T. (1943). The statistical implications of a system of simultaneous equations. *Econometrica, Journal of the Econometric Society*, pages 1–12.

Harville, D. A. (2008). *Matrix algebra from a statistician's perspective*. Springer.

Haughton, D. M. et al. (1988). On the choice of a model to fit data from an exponential family. *The Annals of Statistics*, 16(1):342–355.

Holland, P. W. (1986). Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960.

Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., and Miyano, S. (2004). Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. *Journal of Bioinformatics and Computational Biology*, 2(01):77–98.

Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., Jassal, B., Gopinath, G. R., Wu, G. R., Matthews, L., Lewis, S., Birney, E., and Stein, L. (2005). Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(Database issue):D428–32.

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., and Bühlmann, P. (2012). Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26.

Kanehisa, M. and Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30. PMID: 10592173.

Koivisto, M. and Sood, K. (2004). Exact bayesian structure discovery in bayesian networks. *The Journal of Machine Learning Research*, 5:549–573.

Lauritzen, S. L. (1996). *Graphical Models*. Clarendon Press, Oxford.

Lauritzen, S. L. (2001). Causal inference from graphical models. *Complex stochastic systems*, pages 63–107.

Lauritzen, S. L., Dawid, A. P., Larsen, B. N., and Leimer, H.-G. (1990). Independence properties of directed markov fields. *Networks*, 20(5):491–505.

Lauritzen, S. L. and Richardson, T. S. (2002). Chain graph models and their causal interpretations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):321–348.

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411.

Livak, K. J. and Schmittgen, T. D. (2001). Analysis of relative gene expression data using real-time quantitative pcr and the 2< sup>- $\delta\delta$ct</sup> method. *methods*, 25(4):402–408.

Massa, S. and Sales, G. (2013). *topologyGSA: Gene Set Analysis Exploiting Pathway Topology*. R package version 1.4.2.

Ni, J.-Q., Liu, L.-P., Binari, R., Hardy, R., Shim, H.-S., Cavallaro, A., Booker, M., Pfeiffer, B. D., Markstein, M., Wang, H., et al. (2009). A drosophila resource of transgenic rnai lines for neurogenetics. *Genetics*, 182(4):1089–1100.

Nishimura, D. (2001). Biocarta. *Biotech Software & Internet Report: The Computer Software Journal for Scient*, 2(3):117–120.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 27:29–34.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Pearl, J. (2000). *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press.

Pearl, J. and Paz, A. (1985). *Graphoids: A Graph-Based Logic for Reasoning: About Relevance Relations*. Computer Science Department, University of California.

Pearl, J., Verma, T., et al. (1991). *A theory of inferred causation*. Morgan Kaufmann San Mateo, CA.

Perrin, B.-E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., and d'Alche Buc, F. (2003). Gene networks inference using dynamic bayesian networks. *Bioinformatics*, 19(suppl 2):ii138–ii148.

Rau, A., Jaffrézic, F., and Nuel, G. (2013). Joint estimation of causal effects from observational and intervention gene expression data. *BMC systems biology*, 7(1):111.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and non-randomized studies. *Journal of educational Psychology*, 66(5):688.

Sales, G., Calura, E., Cavalieri, D., and Romualdi, C. (2012). graphite - a bioconductor package to convert pathway topology to gene network. *BMC Bioinformatics*, 13(1):20.

Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., and Buetow, K. H. (2009). Pid: the pathway interaction database. *Nucleic Acids Research*, 37(Database issue):D674–9.

Schäfer, J., Opgen-Rhein, R., Zuber, V., Ahdesmäki, M., Silva, A. P. D., and Strimmer., K. (2013). *corpcor: Efficient Estimation of Covariance and (Partial) Correlation*, r package version 1.6.6 edition.

Schneider, I. (1972). Cell lines derived from late embryonic stages of drosophila melanogaster. *Journal of embryology and experimental morphology*, 27(2):353–365.

Shafer, G. (1996). *The art of causal conjecture.* MIT press.

Shipley, B. (2002). *Cause and correlation in biology: a user's guide to path analysis, structural equations and causal inference.* Cambridge University Press.

Shojaie, A., Jauhiainen, A., Kallitsis, M., and Michailidis, G. (2014). Inferring regulatory networks by combining perturbation screens and steady state gene expression profiles. *PloS one*, 9(2):e82393.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, prediction, and search*, volume 81. MIT press.

Thomas, R. (1991). Regulatory networks seen as asynchronous automata: a logical description. *Journal of theoretical biology*, 153(1):1–23.

Vastrik, I., D'Eustachio, P., Schmidt, E., Joshi-Tope, G., Gopinath, G., Croft, D., de Bono, B., Gillespie, M., Jassal, B., Lewis, S., Matthews, L., Wu, G., Birney, E., and Stein, L. (2007). Reactome: a knowledge base of biologic pathways and processes. *Genome Biology*, 8(3):R39. PMID: 17367534.

Vogel, C. and Marcotte, E. M. (2012). Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nature Reviews Genetics*, 13(4):227–232.

Wright, S. (1934). The method of path coefficients. *The Annals of Mathematical Statistics*, 5(3):161–215.

Vera Djordjilović

CURRICULUM VITAE

Contact Information

University of Padova
Department of Statistical Sciences
via Cesare Battisti, 241-243
35121 Padova. Italy.

Tel. +39 049 827 4174
e-mail: djordjilovic@stat.unipd.it

Current Position

Since January 2012; (expected completion: March 2015)
PhD Student in Statistical Sciences, University of Padova.
Thesis title: Graphical modelling of biological pathways
Supervisor: Prof. Chiogna
Co-supervisor: Prof. Romualdi.

Research interests

- Biostatistics
- Graphical models
- Statistical modelling of infectious diseases

Education

2009 – 2011
Master degree in mathematical statistics.
Charles University in Prague, Faculty of mathematics and physics

Title of dissertation: “Statistical survival analysis and incidence function”
Supervisor: Prof. Petr Volf
Final mark: 1

2006 – 2009
Bachelor degree in mathematics.
Charles University in Prague, Faculty of mathematics and physics
Title of dissertation: “Poisson distribution and related models ”
Supervisor: Prof. Jiří Anděl
Final mark: 1

Visiting periods

July 2013
Department of statistics,
Oxford, England.
Supervisor: Sofia Massa

August – December 2013
Institute of Information theory and Automation
Prague, Czech Republic.
Supervisor: Milan Studený

October 2014
Institute of Information theory and Automation
Prague, Czech Republic.
Supervisor: Milan Studený

Further education

April 2013
Introduction to Graphical Models (16 hours) (*A. Roverato*)
Univesity of Bologna

July 2013
Highly Structured Stochastic Systems:
Bayesian Computation with INLA (*H. Rue*)
Graphical Models for High Dimensional Data (*V. Didelez*)
Newcastle University

June 2010
Summer Institute in Statistics and Modelling of Infectious Diseases
Department of Biostatstics
University of Washington

Work experience

September 2010 – May 2011
IBM, Prague, Czech Republic.
Internship.

Awards and Scholarship

2012-2014
Cariparo PhD Scholarhip for foreign students, University of Padova.
2012
Third prize in the competition for the best Master thesis in the Department of Probability and Statistics, Faculty of Mathematics and Physics, Charles University.
2008-2011
Merit Scholarhip, Faculty of Mathematics and Physics, Charles University.
2005-2011
Czech Government Scholarship.
2003-2005
Serbian Government Scholarship.

Computer skills

- R, SPSS

Language skills

Serbian native; English fluent; Italian fluent, Czech fluent.

Publications

Articles in journals
Djordjilović, V., Chiogna, M., Massa, M.S., Romualdi, C. (2013). Graphical modelling for gene set analysis: a critical appraisal. Submitted to *Biometrical Journal*

Conference presentations

Djordjilović, V., Chiogna, M., Massa, M.S., Romualdi, C. (2013). Graphical modelling for gene set analysis: a case-study in knockout experiments. *IX Conference of the International Biometric Society - Italian Region*, Bressanone, Italy, 27 - 28 June.

Djordjilović, V., Chiogna, M., Massa, M.S., Romualdi, C.(2014). Simulating gene silencing through intervention analysis. *International Biometric conference*, Florence, Italy, 6-11 July.

Djordjilović, V., Chiogna, M., Massa, M.S., Romualdi, C.(2014). Refining the structure of a pathway with a view to prediction of gene silencing effects. *International Workshop on Statistical Modelling*, Gottingen, Germany, 14-18 July.

Teaching experience

March – June 2014
Statistics
Degree in molecular biology
exercises, 25 hours
Instructor: Prof. Alessandra Bianchi

References

Prof. Monica Chiogna
Department of Statistical Sciences
University of Padova
via Cesare Battisti, 241-243
35121 Padova, Italy.
e-mail: monica@stat.unipd.it

Prof. Milan Studený
Institute of Information theory and Automation
Pod Vodárenskou věží, 4
CZ-182 08, Prague 8,
Czech Republic.
e-mail: studeny@utia.cas.cz