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 XXII

A COMPARISON OF PROCEDURES FOR STRUCTURAL LEARNING OF BIOLOGICAL NETWORKS

Direttore della Scuola: Ch.ma Prof.ssa Alessandra Salvan
Supervisore: Ch.mo Prof. Alberto Roverato
Co-supervisore: Dott.ssa Vanessa Didelez

Dottoranda: Vanna Albieri

1 Febbraio 2010

Abstract

Over the past years, microarray technologies have produced a tremendous amount of gene expression data. The availability of these data has motivated researchers to assess genes function and to gain a deeper understanding of the cellular processes, using network theory as tool for the analysis. An elegant framework for modeling and inferring network structures in biological systems is provided by graphical models. They allow the stochastic description of network associations and dependence structures in complex highly structured data. However, typically gene expression data set includes a large number of variables but only few samples making standard graphical model theories inapplicable. The issues presented by genetic data have led to further extend the theory of graphical models to allow their applications in this area. The main aim of this thesis is the comparison of recent procedures, which estimate sparse concentration matrices and learn the structure of biological networks, through the use of both simulated and real data. The compared procedures are: G-Lasso algorithm (Friedman et al., 2008), Shrinkage estimator with empirical Bayes approach for model selection (Schäfer and Strimmer, 2005a, 2005b), PC-algorithm (Kalisch and Bühlmann, 2007). When $n > p$, we consider also the simple frequentist approach based on MLE and t-test for model selection (see Lauritzen, 1996). Regarding the simulated data, for having a realistic simulation of the biological structures, the data have the peculiarity to reproduce few gene regulatory network structures of interest and they are generated by exploiting some properties of the Cholesky decomposition of a matrix. Concerning the real data, we consider the analysis of one of the best characterized system: *Escherichia coli.* A large part of its transcriptional regulatory network is known, hence it can be used as a gold-standard to assess the performance of different procedures in the comparative study.

Riassunto

Negli ultimi anni, le tecnologie dei microarray hanno prodotto una grande quantità di dati provenienti da processi di espressione genica. La disponibilità di questi dati ha permesso ai ricercatori di poter approfondire lo studio della funzione dei diversi geni e poter acquisire una più profonda conoscenza sui processi cellulari, utilizzando come strumento di ricerca la teoria dei network. I modelli grafici risultano essere un utile strumento per la modellazione e l'analisi delle strutture dei networks derivanti da dati biologici. Infatti, questi modelli consentono di rappresentare in modo stocastico le associazioni e le strutture di dipendenza tra gli elementi di data set con struttura complessa. Tuttavia, i dati derivanti da profili di espressione genica si presentano con un elevato numero di variabili ma solo poche osservazioni rendendo, perciò, la teoria classica dei modelli grafici inapplicabile. I problemi legati all'utilizzo di dati genetici hanno portato ad estendere la teoria dei modelli grafici per consentire l'impiego di questi modelli anche in questo campo di applicazione. Lo scopo principale di questa tesi è quello di confrontare, attraverso l'utilizzo di dati simulati e reali, recenti procedure sviluppate con lo scopo di stimare matrici di concentazione sparse e ricostruire i networks biologici. Le procedure considerate per il confronto sono: l'algoritmo G-Lasso (Friedman et al., 2008), lo stimatore Shrinkage associato con l' approccio Bayes empirico per la selezione del modello (Schäfer and Strimmer, 2005a, 2005b), l'algoritmo PC (Kalisch and Bühlmann, 2007). Quando $n > p$, consideriamo anche un semplice approccio frequentista basato sullo stimatore ML e l'utilizzo del test t per la selezione del modello (si veda Lauritzen, 1996). Per quanto riguarda i dati simulati, per avere strutture biologiche simili a quelle reali, i dati hanno la peculiarità di riprodurre alcune strutture dei network di regolazione genica e sono ottenuti sfruttando alcune proprietà della decomposizione di Cholesky di una matrice. Per il confronto con dati reali, sono stati utilizzati dati derivanti da uno dei sistemi maggiormente studiati: *Escherichia coli*. Infatti, grand parte del network di regolazione genica di questo battere è noto, quindi può essere utilizzato come riferimento per valutare il rendimento delle diverse procedure poste a confronto.

Contents

1 Introduction 1

1.1 Overview 1

1.2 Main contributions of the thesis 3

2 Biological networks 5

2.1 Biological background 5

2.2 Microarray technology 6

2.2.1 cDNA Microarrays 8

2.2.2 Synthetic Oligonucleotide Microarrays 8

2.3 Microarray data 9

2.4 Biological networks 10

3 Graphical model theory 13

3.1 Graph theory 13

3.2 Conditional independence graph 15

3.2.1 Conditional independence 15

3.2.2 Undirected independence graphs 16

3.2.3 Directed acyclic independence graphs 16

3.2.4 Markov equivalence 17

3.3 Gaussian Graphical models 17

3.3.1 Model definition 17

3.3.2 MLEs of a GGM 19

3.3.3 Learning the structure of a GGM 20

4 Construct a synthetic gene regulatory network 23

4.1 Synthetic gene regulatory networks 23

4.1.1 Graph motifs 23

4.1.2 The graph structures for the simulation study 24

4.1.3 Markov equivalence between graph structures 26

4.2 Generating synthetic gene regulatory networks 27
    4.2.1 Gene regulatory network: how to generate a specific structure . . 27
    4.2.2 Cholesky decomposition 28
    4.2.3 An example of the use of Cholesky decomposition to design a specific $\mathbf{\Sigma^{-1}}$ 31
4.3 Data generation 32
    4.3.1 Procedure for data set generation 32
    4.3.2 Construction of the concentration matrix 33
    4.3.3 Simulation of data 36

5 Procedures for learning GRN **39**
5.1 G-Lasso 39
5.2 Shrinkage estimator and empirical Bayes approach for model selection . . 42
5.3 PC-algorithm 44
5.4 Procedures under comparison 48

6 Comparative study with simulated data **49**
6.1 Performance measures for the simulated data 49
6.2 Results of the analysis with simulated data 55
    6.2.1 G-Lasso 55
    6.2.2 Shrinkage estimator with empirical Bayes approach and **t**-test approach 60
    6.2.3 Maximum likelihood estimator with the empirical Bayes approach and the **t**-test approach 66
    6.2.4 Pc-algorithm 71
    6.2.5 PR curves and box-plots of MSE 75

7 Comparative study with real data **83**
7.1 *Escherichia coli* data set 83
7.2 Performance measure for the *E. coli* data 84
7.3 Results of the analysis with *E. coli* data 87

8 Conclusion **91**
8.1 Summing-up 91
8.2 Discussion 93

A Precision-Recall curves **95**

Bibliography **99**

List of Figures

2.1 Gene expression process (figure source: The Internet Encyclopedia of Science). 7

3.1 Example of graphs 13

4.1 Common network motifs. 24
4.2 Example of graph motif. 25
4.3 Network motifs under study. 27
4.4 Perfect DAGs version of the graph motifs under study. 28
4.5 General example of graph. 31

6.1 Plot of the true adjacency matrices. In the first row, there are the structures with $p = 20$; in the second row, there are the structures with $p = 100$; in the third row, there are the structures with $p = 200$. 53
6.2 Plot of the true partial correlations matrices. In the first row, there are the structures with $p = 20$; in the second row, there are the structures with $p = 100$; in the third row, there are the structures with $p = 200$. 54
6.3 Plot of adjacency matrices estimated with G-Lasso ($\lambda = 0.1$). In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$. 58
6.4 Plot of partial correlation matrices estimated with G-Lasso ($\lambda = 0.1$). In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$. 59
6.5 Plot of adjacency matrices estimated with Shrinkage and empirical Bayes approach. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$. 63

6.6 Plot of adjacency matrices estimated with Shrinkage and t-test approach. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$. 64

6.7 Plot of partial correlation matrices estimated with Shrinkage. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$. 65

6.8 Plot of adjacency estimated with ML and empirical Bayes approach. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$. 68

6.9 Plot of adjacency estimated with ML and t-test approach. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$. 69

6.10 Plot of partial correlation matrices estimated with ML. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$. 70

6.11 Plot of adjacency matrices estimated with PC-algorithm. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$. 73

6.12 Plot of partial correlation matrices estimated by ipf-algorithm for PC-algorithm. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$. 74

6.13 Comparison of precision-recall curves for pairwise structure ($n = 150$); dashed for ML estimator, dotted for G-Lasso estimator, and dotdash for Shrinkage estimator. 77

6.14 Comparison of precision-recall curves for hub structure ($n = 150$); dashed for ML estimator, dotted for G-Lasso estimator, and dotdash for Shrinkage estimator. 78

6.15 Comparison of precision-recall curves for cascade structure ($n = 150$); dashed for ML estimator, dotted for G-Lasso estimator, and dotdash for Shrinkage estimator. 79

6.16 Box-plots of MSE for the different methods with $p = 20$ and $n = 150$ (SHR: shrinkage, ML: maximum likelihood, PC: PC-algorithm, GL 1: G-Lasso with $\lambda = 0.05$, GL 3: G-Lasso with $\lambda = 0.5$, GL 4: G-Lasso with $\lambda = 0.8$). 80
6.17 Box-plot of MSE for the different methods with $p = 100$ and $n = 150$ (SHR: shrinkage, ML: maximum likelihood, PC: PC-algorithm, GL 1: G-Lasso with $\lambda = 0.05$, GL 3: G-Lasso with $\lambda = 0.5$, GL 4: G-Lasso with $\lambda = 0.8$). 81
6.18 Box-plots of MSE for the different methods with $p = 200$ and $n = 150$ (SHR: shrinkage, PC: PC-algorithm, GL 1: G-Lasso with $\lambda = 0.05$, GL 3: G-Lasso with $\lambda = 0.5$, GL 4: G-Lasso with $\lambda = 0.8$). 82

7.1 Plot of the benchmark adjacency matrix for *E. coli* data. 85
7.2 Precision-recall curves for G-Lasso with *E. coli* data: dashed for $\lambda = 0.05$, dotted for $\lambda = 0.1$, dotdash for $\lambda = 0.5$, and longdash for $\lambda = 0.8$. (a) Complete representation of PR curves. (b)-(c) Zoom in on PR curves. . . 86
7.3 Plot of adjacency matrices for *E. coli* data. (a) G-Lasso ($\lambda = 0.1$), (b) Shrinkage estimator with empirical Bayes approach, and (c) PC-algorithm. 89
7.4 Comparison of precision-recall curves for *E. coli* data: dotted for G-Lasso ($\lambda = 0.1$) and twodash for Shrinkage estimator. (a) Complete representation of PR curves. (b)-(c) Zoom in on PR curves. 90

A.1 Precision-Recall curves for G-Lasso ($p = 20$, $n = 150$); dashed for $\lambda = 0.05$, dotted for $\lambda = 0.1$, dotdash for $\lambda = 0.5$, and longdash for $\lambda = 0.8$. 96
A.2 Precision-Recall curves for G-Lasso ($p = 100$, $n = 150$); dashed for $\lambda = 0.05$, dotted for $\lambda = 0.1$, dotdash for $\lambda = 0.5$, and longdash for $\lambda = 0.8$. . 97
A.3 Precision-Recall curves for G-Lasso ($p = 200$, $n = 150$); dashed for $\lambda = 0.05$, dotted for $\lambda = 0.1$, dotdash for $\lambda = 0.5$, and longdash for $\lambda = 0.8$. . 98

List of Tables

6.1 Matrix for performance measures. 49
6.2 G-Lasso algorithm. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$. 57
6.3 Shrinkage estimator. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$. 62
6.4 ML estimator. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$. 67
6.5 PC-algorithm. 72

7.1 Performance measures for *E. coli* data. (a) G-Lasso algorithm, (b) Shrinkage estimator with empirical Bayes approach, and (c) PC-algorithm. . . . 88

Chapter 1

Introduction

1.1 Overview

Recent progresses in molecular biology and advances in experimental methodology have led to an unprecedented growth in molecular data. The availability of these detailed data gave the impulse to originate a new field in Biology called *Systems Biology* (Kitano, 2001). Systems Biology aims to reconstruct the structure and dynamics of biological processes and systems as a whole; this is done by moving the analysis from molecular level (reductionist approach) to systems level (wholist approach). Reductionists look at one single element of the system to find its role in biological processes as well as the connection with other elements and the mechanisms of action. In contrast, the wholist approach aims to study the entire system rather than the single elements, taking a snapshot of all elements at a certain level (e.g. genes, transcripts, proteins, or metabolites).

In this regard, for the analysis of biological data, mathematical and computational techniques are used to handle the complexity of the systems and the wealth of data. In particular, network analysis allows to understand how the elements of the system interact between each other. An elegant framework for modeling and inferring network structures in biological system is provided by *probabilistic graphical models* (Whittaker, 1990). These models allow the stochastic description of network associations and dependences in complex highly structured data and, at the same time, they offer an advanced statistical framework for inference. The elements of a system are represented as vertices of a graph and the interactions between them are represented as edges.

Typical biological networks are gene regulatory networks, protein interaction networks, and metabolic networks. In this thesis we focus on gene regulatory networks (GRN), where vertices of the graph are genes and edges represent regulatory interactions. The data for the network analysis are produced by the microarray technologies. They allow to observe the amount of mRNA simultaneously for a large number of genes under different experimental conditions and produce the so-called *microarray data.* These data can be seen as a random sample of a multivariate distribution defined by a set of p random variable for the p genes of the system. For each of the p random variables, there is a vector of values that comes from n different measurements. A typical feature of microarray data is that the number of observations n, on the order of tens or at most hundreads, is smaller than the number of variables p, on the order of hundreds or even thousands. This is the so-called *large p-small n* issue that leads to new challenges in statistical inference and, more specifically, proscribes the application of most of the existing methods for structure learning of graphical models. In addition, there are not standard models for representing GRN or other biological networks, since every network has different properties and the graphical model has to be tailored to the specific situation.

In the literature, inference for GRN used both Gaussian graphical models (GGM) and directed acyclic graph (DAG) models for normal distributions. In this thesis we focus on GGM. More precisely, we consider the subclass of GGM which are Markov equivalent to DAG models and, therefore, the results of this thesis can be extended to this case. A GGM is a statistical model for Gaussian distribution associated with an undirected graph where missing edges correspond to zero partial correlations and, therefore, to conditional independence statements. Partial correlation is a measure of association between two variables (i.e. genes) conditioning on all the remaining observed variables and it is computed from the concentration matrix $\Omega = \Sigma^{-1}$ (Lauritzen, 1996). Learning a GGM from data is equivalent to learning the zero pattern of the concentration matrix Ω, but the *large p-small n* setting of microarray data makes standard structural learning procedures for GGM no longer readily applicable. In the specific, the application of these procedures requires that the sample covariance matrix S has full rank in order to obtain $\hat{\Omega} = S^{-1}$, but S has full rank, with probability one, if and only if $n > p$ (Dykstra, 1970). Moreover, even in the case S has full rank, direct application of traditional structural learning procedures would be unfeasible due to the large number of variables that lead to computational problems.

A way to face these challenges is exploiting background information on the network structure so as to develop tailored procedures. In particular, it is known that GRN are sparse and this means that the number of edges in the network is much smaller than the number of interactions of the complete network. In this thesis we consider recent procedures for learning sparse networks in the *large p-small n* issue and compare their performance both on simulated data and real data using the R language.

The outline of the thesis is as follows. Chapter 2 gives a brief introduction to molecular biology and microarray technology required for this thesis. In Chapter 3, there is a short introduction to graph terminology, conditional independent graph, and Gaussian graphical models. Chapter 4 regards to the construction of the simulated data that have the peculiarity to reflect and to reproduce some gene regulatory network structures of interest. Chapter 5 presents the different methodologies considered for the comparison. The results of the comparative study for the simulated data and real data are presented in Chapter 6 and Chapter 7, respectively. Finally, in Chapter 8 there are the conclusion and discussion with an exposure of possible directions for future research.

1.2 Main contributions of the thesis

An overview of the main contributions of the Ph.D. thesis presented in this work is listed below.

- The principal contribution of the Ph.D. thesis is itself the comparison study of different methodologies that have been proposed recently in the literature for learning a GGM for sparse networks in the *large p-small n* issue. In the literature there are other comparative studies of procedures for learning GRNs, as for example the papers by Werhli et al. (2006) and Soranzo et al. (2007). Anyway, the considered approaches, that are G-Lasso algorithm (Friedman et al., 2008), Shrinkage estimator with empirical Bayes approach for the model selection (Schäfer and Strimmer, 2005a, 2005b), PC-algorithm (Kalisch and Bühlmann, 2007), and , when $n > p$, a simple frequentist approach based on MLE and t-test for model selection (see Lauritzen, 1996) have never been compared to each other.

- The evaluation is firstly carried out on simulated data sets that aim to reflect some gene regulatory network structures in different scenarios. In order to have simulated data as close as possible to real data, we specify some network structures of interest taking in consideration the biological background on gene regulatory networks. This analysis returns two fundamental characteristics for these types of networks. First, they have some basic structures that are repeated often in the same network and are present in many different organism (network motifs); second, the networks are sparse. Consequently, we derive three structures that are represented singularly in three scenarios depending only on the number of variables.

- By exploiting some results on the Cholesky decomposition of a matrix, we have written some functions in the R language to generate data from the derived network structures.

- In order to investigate a more realistic setting, the different approaches are applied to an available data set of *Escherichia coli*. Its complete transcriptional regulatory network is unknown, but the database RegulonDB (Gama-Castro et al., 2008) provides a curated set of transcription factor and target gene relationships that it is possible to use as a gold-standard to assess the performance in the comparative study.

- In order to have a clear and complete comparison between methodologies, we use several measures to evaluate the accuracy of each approach in the structural learning of the networks. In particular, there are three main groups of measures used for the comparison. The first group of measures aims to evaluate generally the precision in the reconstruction of the networks; the second group of measure gives informations on the goodness in the estimation of the parameters. Finally, the third group of measures returns general information on the model selection.

Chapter 2

Biological networks

What follow is a brief introduction to basic concepts of molecular biology, microarray technology, and biological networks that provide the requisite concepts relevant to this thesis. For futher details refer to genetic and molecular biology textbooks (e.g. Alberts et al., 2008; Gibson and Muse, 2004; Griffiths et al., 2005).

2.1 Biological background

Cells are the fundamental working units of every living system. The nucleus of each cell contains the chromosomes that carry the instructions needed to direct the cell activities in the production of proteins via the *DNA* (deoxyribonucleic acid). DNA is a double-stranded, linear polymer consisting of a sugar-phosphate backbone attached to subunits called *nucleotides*. There are four nucleotide bases: adenine (A), thymine (T), cytosine (C), and guanine (G). The two strands of the double helix are held together by weak hydrogen bonds between complementary bases on the strands. Base pairing occurs as follows: A pairs with T, and G pairs with C. The particular order of the bases arranged along the sugar-phosphate backbone is called the DNA sequence. The *genome* is the complete DNA of an organism and encodes the *genetic code* required to create a particular organism with its own unique traits. The nucleotide bases A, T, C and G are the letters that spell out these genetic instructions by producing a three-letter word code (*codons*), where each specific sequence of three DNA bases encodes an *amino acid*. Amino acids are the basic units of *proteins*, which perform most life functions.

With few exceptions, all human cells contain the same DNA, but despite carrying the same set of instructions, cells are actually different. These differences are due to the fact that, stimulated by cell regulatory mechanisms or environmental factors, segments of DNA express the genetic code and provide instructions to the cells on when and in what quantity to produce specific proteins. These segments of DNA are the *genes* and the process by which they become active is called their *expression.* The *gene expression level* is an integer valued or continuous measure that provides a quantitative description of the gene expression by measuring the number of intermediary molecules produced during this process. These molecules are the *mRNA* (messenger ribonucleic acid) and the *tRNA* (transfer ribonucleic acid), and they are produced during the two steps of *transcription* and *translation* that lead to the synthesis of a protein. This two-step representation of the protein-synthesis process constitutes the *central dogma* of molecular biology (Fig. 2.1).

- *Transcription.* The first step of a gene expression is the creation of a complementary copy of the gene sequence stored in one of the two DNA complementary strands. The complementary copy of the gene DNA code transcribes U (uracil) for A, A for T, G for C and C for G into the mRNA.

- *Translation.* The mRNA transcript is moved from the nucleus to the cellular cytoplasm, where it serves as a template on which tRNA molecules, which carry amino acids, are lined up. The amino acids are then linked together to form a protein chain.

2.2 Microarray technology

The basic idea behind microarray technology is to simultaneously measure the relative expression level of thousands of genes within a particular cell population or tissue. Two key technical concepts behind this measurement process are *reverse transcription* and *hybridization.*

- *Reverse transcription.* The mRNA transcript of a gene can be experimentally isolated from a cell, and reverse-transcribed into a complementary DNA copy called cDNA. A collection of cDNAs transcribed from cellular mRNA constitutes the cDNA library of a cell. Similarly, double-stranded cDNA can be reverse-transcribed into a complementary copy called cRNA.

- *Hybridization.* Hybridization is the process of base pairing two single strands of DNA or RNA. DNA molecules are doublestranded and these two strands melt apart at a characteristic melting temperature, usually above 65°C. As the temperature is reduced and held below the melting temperature, single-stranded molecules bind back to their counterparts. In the same way, a mRNA molecule can hybridize to a melted cDNA molecule when the mRNA contains the complementary code of the cDNA strands.

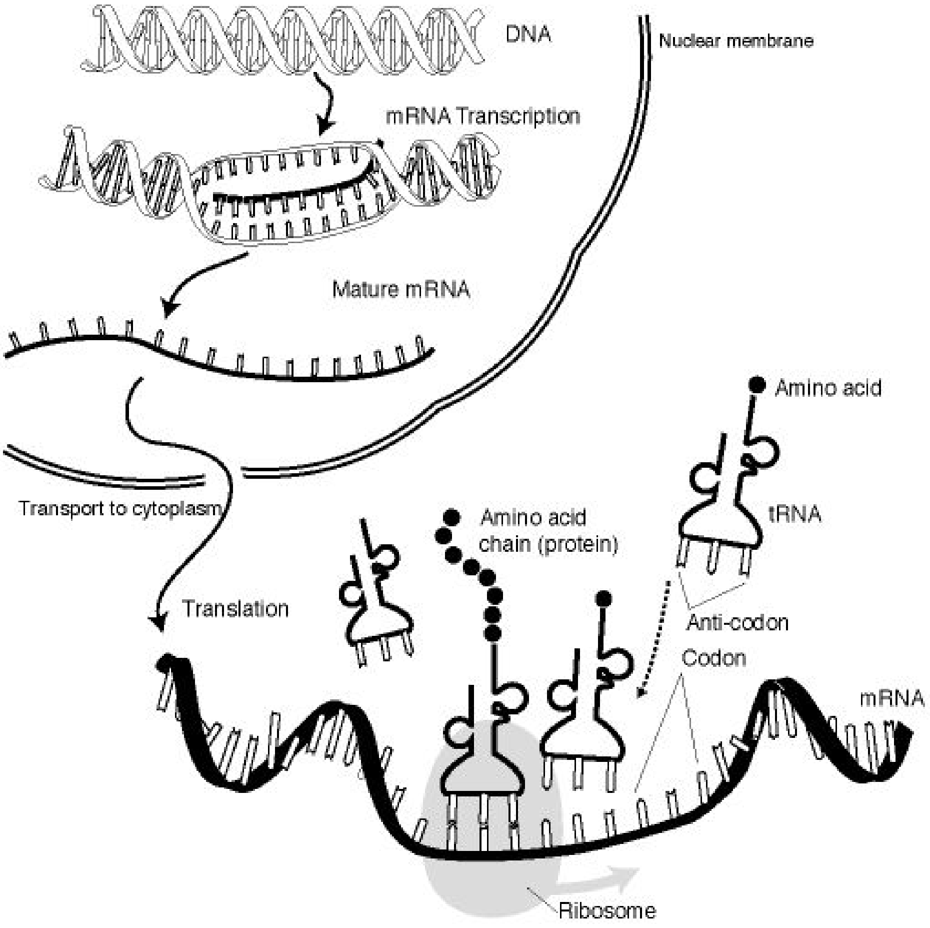


Figure 2.1: Gene expression process (figure source: The Internet Encyclopedia of Science).

Microarray technology is used to measure the relative level of expression of genes in a particular cell or tissue by hybridizing a labeled cDNA representation of the cellular mRNA to cDNA sequences (*cDNA microarrays*) or by hybridizing a labeled cRNA representation of the cellular mRNA to short specific segments known as synthetic oligonucleotides or oligos (*synthetic oligonucleotide microarrays*; Lockhart et al., 1996). The tethered cDNA sequences or oligos are called *probes*, while the cDNA or cRNA representation of cellular mRNA extracted from the cell is called the *target*. In both cases, the probes represent either genes of known identity or segments of functional DNA. The target is labeled with fluorescent dye and hybridized to the probes. The higher the amount of cDNA or cRNA hybridized to a probe, the more intense the fluorescent dye signal will be on that probe. The relative mRNA abundance of a gene in a particular cell or tissue is therefore measured by the emission intensity of the probes.

Synthetic oligonucleotide and cDNA microarrays are the two most popular microarray technologies and a simple description of both these processes for conducting microarray experiments is briefly outlined in the following.

2.2.1 cDNA Microarrays

The production of the microarray starts with the selection of the probes to be placed on the microarray and amplification of the corresponding cDNA clones by a technique known as *polymerase chain reaction* (PCR). The PCR allows multiple rounds of amplification of a minimal amount of DNA to produce sufficient quantities of a sample. The cDNA microarrays are produced by spotting PCR samples of cDNA strands in approximately equal amounts on a glass slide using a high-speed robot. Each strand of cDNA identifies uniquely with its code, a gene or a segment of functional DNA, so that each spot in the microarray corresponds to a gene or a segment of functional DNA. To prepare the target, investigators extract total RNA or mRNA produced from two types of cells, for example test and reference cells. Then, by using a single round of reverse transcription, the mRNA from the two samples is fluorescently labeled with Cy3 (green) and Cy5 (red), and the target mixture is hybridized to the probes on the glass slides. During the hybridization, if segments of the mRNA representation in the target find their complementary portion among the samples of cDNA on the glass slide, they will bind together. When the hybridization is complete, the glass slide is washed and laser excitement of the glass slide is used to yield a luminous emission that is then measured by a scanning microscope. Fluorescence measurements are made with a microscope that illuminates each spot and measures fluorescence for each dye separately, thus providing a measure of the relative mRNA abundance for each gene in the two cells. The intensity of the green spot measures the relative mRNA abundance of the gene in the cell that had reverse-transcribed mRNA labeled with Cy3, while the intensity of the red spot measures the relative mRNA abundance of the gene in the cell that had reverse-transcribed mRNA labeled with Cy5. Grey spots denote genes that were expressed in neither cell type.

2.2.2 Synthetic Oligonucleotide Microarrays

Synthetic oligonucleotide microarrays, also known by the trademark *Affymetrix GeneChip*, are fabricated by placing short cDNA sequences (oligonucleotides) on a small silicon chip by means of the same photolithographic techniques used in computer microprocessor fabrication. On the GeneChip platform, each probe is 25 bases long and each gene is represented by 16-20 pairs of oligonucleotides. A probe pair consists of a perfect match (PM) probe and a mismatch (MM) probe. Each PM probe is chosen on the basis of uniqueness criteria and proprietary, empirical rules designed to improve the odds that probes will hybridize with high specificity. The MM probe is identical to the corresponding PM probe except for the base in the central position, which is replaced with its complementary base. To prepare the target, investigators extract total RNA from a cell or tissue. The mRNA is reverse-transcribed into cDNA, which is made double-stranded and then converted into cRNA using a transcription reaction that fluorescently labels the target. Once hybridization has occurred, the microarray is washed and scanned with a standard laser scanner. The scanner generates an image of the microarray that is gridded to identify the cells that contain each probe and analyzed to extract the signal intensity of each probe cell.

2.3 Microarray data

In both cDNA and oligonucleotide microarrays, hybridization of the target to the probes determines a chemical reaction that is captured into a digital image by a scanning laser device. The next step is to translate the intensity of each hybridization signal into a table with numerical measures. The quality of the image analysis process is crucial for accurate interpretation of the data, and a variety of algorithms and software tools tailored to the different aspects of cDNA and oligonucleotide microarray images have been developed. A grid over the microarray is used to associate the signal from each spot with its location on the microarray, and thereby with its base-paring sequence identification. With good filtration of the scattered light, the detector will record only the light from the fluorescently labeled hybridized pairs, and greater intensity of the fluorecense will be detected at spots where more cDNA or cRNA have hybridized to the microarray. The ratio of the fluorescent light emissions between the two different walelengths, corresponding to the two different dyes used to label the unknown and control target samples, is the indirect measurement of the relative gene transcript expression levels.

Both microarray technologies provide a panoramic view of the activity of genes under particular experimental conditions. They let the experimenters observe the molecular profile of a cell in a particular condition. The simplest experiment is a comparative experiment to identify the genes differentially expressed in two conditions. More complex experimental questions involve molecular profiling of several conditions at a time to measure the gene expression levels in cells grown or observed in a particular condition, and different samples can be assumed to be stochastically independent. The data generated by microarray experiments can be viewed as a matrix of expression levels, organized by genes versus tissue (or cell population) samples. In the case where a tissue sample corresponds to a single microarray experiments, we can represent the output from n experiments in the form of a $n \times p$ matrix (array). Each row of the matrix contains the expression levels on the p genes monitored in the microarrays, while each column contains the expression levels of a genes as it varies over the n tissue sample.

Before data analysis, the quantitative measurements of gene expression data, produced by microarray experiments, require a preprocessing stage to reduce their noise. This noise is caused by gene transcripts that do not contribute information to the experiment outcome and those that do not change across experiments. A common strategy to reduce data variability and dimensionality is to perform two preprocessing operations known as normalization and filtering, on either the raw or transformed data. The goal of the normalization operation is to remove systematic distortions across microarrays to render comparable the experiments conducted under different conditions. The aims of the filtering operation are to reduce variability by removing those genes that have measurements that are not sufficiently accurate and to reduce the dimensionality of the data by removing genes that are not sufficiently differentiated.

2.4 Biological networks

The behavior of complex cellular and organism systems emerges from the concerted activities of many interacting components such as genes and gene products. At a highly abstract level, the cooperating components can be considered as a set of vertices that are connected to each other, with links (edges) representing pairwise interactions. Vertices and edges together form a network and, more formally spoken, a graph (see Chapter 3). Typical biological networks are: gene regulatory network, protein-protein network, and metabolic network. In this work, we focus on gene regulatory network and the description of its network structure is briefly given in Chapter 4.

Gene regulatory network (transcriptional regulatory network). A gene regulatory network is a collection of DNA segments in a cell which interact with each other (indirectly through their RNA and protein expression products) and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA. In this network entities as genes represent the vertices set and regulatory interactions represent the edges set.

Protein-protein network. Protein-protein network identifies and catalogs physical interactions between pairs or groups of proteins. Understanding protein-protein interactions is important for the investigation of intracellular signaling pathways and for gaining insights into various biochemical processes. In this network entities as proteins represent the vertices set and protein interactions represent the edges set.

Metabolic network. A metabolic network is the complete set of metabolic and physical processes that determine the physiological and biochemical properties of a cell. As such, these networks comprise the chemical reactions of metabolism as well as the regulatory interactions that guide these reactions. In this network entities as metabolites represent the vertices set and reactions represent the edges set.

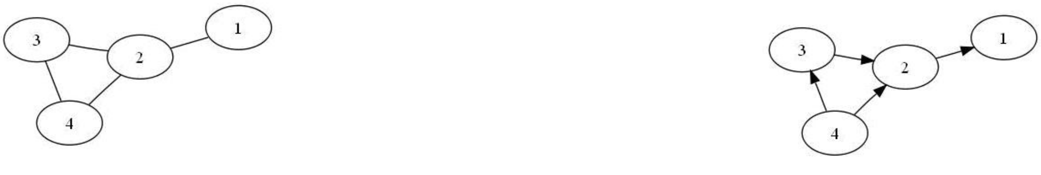
Chapter 3

Graphical model theory

3.1 Graph theory

In the following, we present the requisite definitions and notation on graph theory relevant to this thesis; for more details we suggest to consult the book of Whittaker (1990).
A *graph* $\mathcal{G} = (V, E)$ is a mathematical object that consists of a finite set of vertices $V = \{1, \ldots, p\}$ and a set of edges $E \subseteq V \times V$. We distinguish between undirected and directed graphs. An *undirected graph* has only *undirected edges*. An undirected edge between $i, j \in V$, that is $\{i, j\} \in E$, is represented as a line $i - j$ and an example of undirected graph is presented in figure 3.1a. A *directed graph* has only *directed edges* between $i, j \in V$, with $(i, j) \in E$ but $(j, i) \notin E$. A directed edge of the graph is represented as an arrow $i \rightarrow j$ (Fig. 3.1b). In the case a vertex l appears in a constellation $i \rightarrow l \leftarrow j$, then it is called a *collider* and, in addition, if both $(i, j), (j, i) \notin E$, then i, l, j constellation is called a *V-structure*.
The *adjacency matrix* $A = \{a_{ij}\}$ of a directed or an undirected graph $\mathcal{G}$ on p vertices is the $p \times p$ matrix where if $\{i, j\} \in E$ then $a_{ij} = a_{ji} = 1$ and if $(i, j) \in E$ $a_{ij} = 1$, and zero otherwise.

A sequence of distinct and ordered vertices $(j_0, \ldots, j_n)$ is called a *path of length n from*



(a) Undirected

(b) Directed

Figure 3.1: Example of graphs

j_0 to j_n if $\{j_{i-1}, j_i\} \in E$, or $(j_{i-1}, j_i) \in E$, or $(j_i, j_{i-1}) \in E$ for all $i = 1, \ldots, n$. It is an *undirected path* between j_0 and j_n if $\{j_{i-1}, j_i\} \in E$ for all $i = 1, \ldots, n$. It is a *directed path* between j_0 and j_n if $(j_{i-1}, j_i) \in E$ for all $i = 1, \ldots, n$. A *cycle* is defined as a directed path with the difference that $j_0 = j_n$.

Given $A \subseteq V$ a subset of vertices of the graph, the *induced subgraph* $\mathcal{G}_A$ is defined as $\mathcal{G}_A = (A, E_A)$, where $E_A = (A \times A) \cap E$. A graph, or a subgraph, is *complete* if there is an edge, directed or undirected, between any pair of vertices.

So far, we have introduced general graph theory that is common for undirected and directed graphs; from now on, we are going to present specific graph theory first for undirected graphs and then for directed graphs.

Undirected graphs

For an undirected graph $\mathcal{G}$ we have that a subset of vertices $A \subseteq V$ *separates* two vertices i and j if every path joining the two vertices contains at least one vertex from the separating set. The subset $A \subseteq V$ is said to separate two subsets $B, C \subseteq V$ if it separates every pair of vertices $i \in B$ and $j \in C$. A *clique* is given by a subset of vertices that induce a complete subgraph but for which the addition of a further vertex renders the induced subgraph incomplete, that is, a clique is a maximally complete subgraph. The *boundary*, or *neighbours*, of A, $bd(A)$, is the set of all the vertices in $V \backslash A$ that have an edge with a vertex in A and the *closure* of A is $cl(A) = bd(A) \cup A$. A partition (A, B, C) of V is called a *decomposition* of $\mathcal{G}$ if the following conditions hold: (i) C separates A and B and (ii) C is complete. The graph $\mathcal{G}$ is *decomposable* if one of the following conditions holds: (i) it is complete or (ii) there exists a proper decomposition $(A,\ B,\ C)$ such that both subgraphs $\mathcal{G}_{A \cup C}$ and $\mathcal{G}_{B \cup C}$ are decomposable. Let $B_1, \ldots, B_k$ be a sequence of subsets of the vertex set V; the *histories* of the sequence are defined as $H_j = B_1 \cup \cdots \cup B_j$ for $j = 1, \ldots, k$, whereas $R_j = B_j \backslash H_{j-1}$ and $S_j = H_{j-1} \cap B_j$, for $j = 2, \ldots, k$, are the *residuals* and *separators* of the sequence, respectively. The sequence is said to be *perfect* if the following conditions are fulfilled

(i) for all $i > 1$ there is a $j < i$ such that $S_i \subseteq B_j$;

(ii) the sets S_i are complete for all i.

Let $C_1, \ldots, C_k$ be a sequence of the cliques of $\mathcal{G}$. If this sequence is perfect, then the numbering of the vertices V obtaining by taking first the vertices in C_1 and then those in R_2, R_3 and so on is called *perfect* (Lauritzen, 1996; Lemma 2.12).

Directed graphs

In the family of directed graphs, a graph without cycles is called a *directed acyclic graph* (DAG) and it is indicated with $\mathcal{D}$; an example of DAG is presented in figure 3.1b, note that vertex 2 is a collider but there are no V-structures. For a DAG $\mathcal{D}$, given two vertices $i, j \in V$ if $(i, j) \in E$, i.e. $i \rightarrow j \in E$, then the vertex i is a *parent* of j and the vertex j is a *child* of i. Hence, we denote by $pa(A)$ the *set of parents* of A, i.e. the set of all those vertices in $V \backslash A$ that have a child in A. Similarly, $ch(A)$ is the *set of children* of A, i.e. the set of all those vertices in $V \backslash A$ that have a parent in A. Moreover, if there is a directed path from i to j, then i is called *ancestor* of j and j is called *descendant* of i. Hence, we refer with $de(A)$ the *set of descendants* of A, i.e. the set of all those vertices in $V \backslash A$ that have an ancestor in A. Analogously, $an(A)$ is the *set of ancestors* of A, i.e. the set of all those vertices in $V \backslash A$ that have a descendant in A. The *non-descendants* of A are $nd(A) = V \backslash (de(A) \cup A)$ and the *ancestral set* of A is $An(A) = an(A) \cup A$. Given the disjoint subsets A, B and C of a DAG $\mathcal{D}$, then A and B are *d-separated* by C if and only if A and B are separated by C in $\left(\mathcal{D}_{An(A \cup B \cup C)}\right)^m$ (Lauritzen, 1996; Proposition 3.25). Ultimately, we give three more useful definitions related to DAGs. First, the *skeleton* $\mathcal{D}^u$, or undirected version of $\mathcal{D}$, is the graph given by replacing the directed edges with undirected edges. Second, a *moral graph* $\mathcal{D}^m$, associated with the directed graph $\mathcal{D}$, is an undirected graph constructing by (i) adding an undirected edge between every pair of non-adjacent vertices that have a common child and (ii) turning all directed edges into undirected edges. Third, a *perfect DAG* is a graph for which the parents of every node form a complete set, i.e. there are no V-structures and $\mathcal{D}^m = \mathcal{D}^u$.

3.2 Conditional independence graph

After a summary of the essential concepts from graph theory, we introduce and define conditional independence graphs, undirected and directed, of a p-dimensional vector of random variables.

3.2.1 Conditional independence

Let $\mathbf{X}_V = (X_1, \ldots, X_p)^T$ a continuous random vector, indexed by $V = \{1, \ldots, p\}$, with joint density function $f_{X_V}(\cdot)$. The disjoint subset $A, B, C \subset V$ index the subvectors X_A, X_B and X_C, respectively. Let $f(x_A, x_B, x_C)$ be the density function of $X_{A \cup B \cup C}$, then X_A is *conditionally independent* of X_B given X_C, written $X_A \perp\!\!\!\perp X_B | X_C$, if and only if the density function of X_A and X_B conditional on X_C satisfies

$$f_{X_A X_B|X_C}(x_A, x_B|x_C) = f_{X_A|X_C}(x_A|x_C) f_{X_B|X_C}(x_B|x_C)$$

for all values of x_A and x_B and for all x_C such that $f_{X_C}(x_C) > 0$.
An equivalent characterization of $X_A \perp\!\!\!\perp X_B|X_C$ is given by the factorization criterion for conditional independence

Proposition 1 (factorization criterion) *The random subvectors X_A and X_B are conditionally independent given X_C if and only if there exist functions g and h such that*

$$f(x_A, x_B, x_C) = g(x_A, x_C) h(x_B, x_C) \tag{3.1}$$

for all x_C with $f_{X_C}(x_C) > 0$.

See proof in Whittaker (1990, Proposition 2.2.1).

3.2.2 Undirected independence graphs

Let $\mathcal{G} = (V, E)$ be an undirected graph. The random vector $\mathbf{X}_V$ satisfies the *pairwise Markov property* with respect to $\mathcal{G}$ if for every pair of non-adjacent vertices $i, j \in V$ it holds that $X_i \perp\!\!\!\perp X_j|X_{V\setminus\{i,j\}}$.
There exists others two Markov properties. With respect to $\mathcal{G}$, $\mathbf{X}_V$ is said to satisfy

- the *Local Markov property* if for any $i \in V$ it holds that $X_i \perp\!\!\!\perp X_{V\setminus cl(i)}|X_{bd(i)}$;
- the *Global Markov property* if for any triplet (A, B, C) of disjoint subsets of V such that C separates A and B in $\mathcal{G}$ it holds that $X_A \perp\!\!\!\perp X_B|X_C$.

It is a remarkable fact that, if $f_X > 0$, the three Markov properties are equivalent (see Whittaker, 1990, pg. 70). In this thesis we only consider random vectors with positive density and for this reason we shortly say that $\mathbf{X}_V$ is undirected Markov with respect to $\mathcal{G}$.

3.2.3 Directed acyclic independence graphs

Let $\mathcal{D} = (V, E)$ be a DAG. The random vector $\mathbf{X}_V$ satisfies the *pairwise directed Markov property* with respect to $\mathcal{D}$ if for every pair non-adjacent vertices $i, j \in V$ such that $j \in nd(i)$ it holds that $X_i \perp\!\!\!\perp X_j|X_{nd(i)\setminus\{j\}}$.
Also for the directed acyclic independence graphs there are other two Markov properties. With respect to $\mathcal{D}$, $\mathbf{X}_V$ is said to satisfy

- the *Local directed Markov property* if for any $i \in V$ it holds that $X_i \perp\!\!\!\perp X_{nd(i)} | X_{pa(i)}$;
- the *Global directed Markov property* if for any disjoint subsets $A, B, C \subset V$ such that C separates A and B in $[\mathcal{D}_{An(A \cup B \cup C)}]^m$ it holds that $X_A \perp\!\!\!\perp X_B | X_C$.

Moreover, the density of $f(x)$ admits a *recursive factorization* with respect to a DAG if

$$f(x) = \prod_{i=1}^{p} f(x_i | \mathbf{x}_{pa(i)}).$$

All directed Markov properties are equivalent without any positive requirement for the density; see proof in Cowell et al. (1999, pg. 74). For this reason in the following we simply say that $\mathbf{X}_V$ is directed Markov with respect to $\mathcal{D}$.

3.2.4 Markov equivalence

For an undirected graph $\mathcal{G} = (V, E)$, let $\mathcal{M}_U(\mathcal{G})$ denote the family of probability distributions that are undirected Markov with respect to $\mathcal{G}$. Similarly, we denote by $\mathcal{M}_D(\mathcal{D})$ the family of probability distributions that are directed Markov with respect to a DAG $\mathcal{D}$.

Two DAGs $\mathcal{D}_1 = (V, E_1)$ and $\mathcal{D}_2 = (V, E_2)$ on the same vertex set are said to be Markov equivalent if $\mathcal{M}_D(\mathcal{D}_1) = \mathcal{M}_D(\mathcal{D}_2)$. It can be shown that $\mathcal{D}_1$ and $\mathcal{D}_2$ are Markov equivalent if and only if they have the same skeleton and the same V-structure (see Cowell et al., 1999, pg. 79).

An undirected graph $\mathcal{G} = (V, E)$ and a DAG $\mathcal{D} = (V, E')$ are Markov equivalent if $\mathcal{M}_U(\mathcal{G}) = \mathcal{M}_D(\mathcal{D})$. It can be shown that for every decomposable graph $\mathcal{G} = (V, E)$ there exists a DAG $\mathcal{D} = (V, E')$ Markov equivalent to $\mathcal{G}$, and $\mathcal{D}$ can be constructed as a perfect directed version of $\mathcal{G}$ (see Roverato, 2005).

Furthermore, for any perfect DAG $\mathcal{D}$, the skeleton of $\mathcal{D}$ is an undirected decomposable graph Markov equivalent to $\mathcal{D}$. If $\mathcal{G}$ is non-decomposable then there does not exist any DAG $\mathcal{D}$ Markov equivalent to $\mathcal{G}$.

3.3 Gaussian Graphical models

3.3.1 Model definition

For data analysis, we assume that the variables of a continuous random vector $\mathbf{X}_V \equiv \mathbf{X}$, with $V = \{1, \ldots, p\}$, have a jointly Normal $N_p(\mu, \Sigma)$, with mean vector $\mu = (\mu_1, \ldots, \mu_p)^T$ and a positive definite covariance matrix $\Sigma = \{\sigma_{ij}\}$, where $1 \leq i,\, j \leq p$. Hence, given an undirected graph $\mathcal{G} = (V, E)$ and a random vector $\mathbf{X} \backsim N_p(\mu, \Sigma)$, a *Gaussian graphical model* (GGM), also known as *covariance selection* or *concentration graph model*, for $\mathbf{X}$ with graph $\mathcal{G}$ is the family of normal distributions for $\mathbf{X}$ that are undirected Markov with respect to $\mathcal{G}$.

The first paper that introduced the concept of GGM is the one of Dempster (1972) and further details can be found in the books by Whittaker (1990) and by Edwards (2000). In this section we describe the basic theory associated to Gaussian graphical model and the procedure to fit a GGM.

Under a GGM, the data $\mathbf{X}$ are assumed to be distributed as a $N_p(\mu, \Sigma)$ with a multivariate density function of the form

$$f(x) = (2\pi)^{-p/2}\det(\Sigma)^{-1/2}\exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}, \tag{3.2}$$

where μ and Σ are called the moment parameters. Using the exponential family representation, equation (3.2) can be rewritten in term of canonical parameters: $\Sigma^{-1} = \Omega = \{w_{ij}\}$, that is called *precision* or *concentration* matrix, and $\beta = \Sigma^{-1}\mu$. Then the density function of $\mathbf{X}$ becomes

$$\begin{aligned} f(x) &= \exp\left\{\alpha + \beta^T x - x^T\Omega x/2\right\} \\ &= \exp\left\{\alpha + \sum_{i=1}^{p}\beta_i x_i - \sum_{i=1}^{p}\sum_{j=1}^{p}\omega_{ij}x_i x_j/2\right\}, \end{aligned} \tag{3.3}$$

where α is the normalizing constant. Using the factorization criterion, presented in equation (3.1), it results that the relation between X_i and X_j given the remaining $p-2$ variables is explained by the element ω_{ij} of the concentration matrix Ω and it holds that

$$X_i \perp\!\!\!\perp X_j | X_{V\setminus\{i,j\}} \iff \omega_{ij} = 0.$$

Hence, we have that the pattern of zero entries in the concentration matrix corresponds to conditional independences restrictions between variables and if $\mathbf{X}$ is Markov with respect to $\mathcal{G} = (V, E)$ then

$$\{i, j\} \notin E \Rightarrow \omega_{ij} = 0.$$

In GGM framework, the correlation between any two nodes i and j conditional on all the remainder of the nodes is described by the *partial correlation coefficient* and denoted by $\rho_{ij\cdot V\setminus\{i,j\}}$. Standard graphical model theory , e.g. Edwards (2000), shows that the partial correlation can be expressed in terms of the elements of the concentration matrix Ω. This result leads the procedure to compute the partial correlation coefficients via the relation

$$\rho_{ij\cdot V\setminus\{i,j\}} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}, \tag{3.4}$$

see Lauritzen (1996, pg. 129-130) for more details.

3.3.2 MLEs of a GGM

The observations in a sample are directly related to the probability model under consideration by the likelihood. The standard results and techniques of maximum likelihood estimation and likelihood ratio test (Cox and Hinkley, 1974) are applicable to GGM, and in general to graphical models.

Under a GGM, the elements of interest are the independences between variables and, for a multivariate normal distribution, they are characterized by the covariance matrix Σ or its inverse Ω. As the correspondence between Σ and Ω is one to one, either parametrizations can be used. The mean vector μ, on the contrary, in this context is of less interest and, thus, it is allowed to be entirely arbitrary.

Let us assume that $X^1, \dots, X^n$ is an i.i.d sample of the random vector $\mathbf{X}$, where $X^i = (X_1^i, \dots, X_p^i)$ and its realization is $x^i = (x_1^i, \dots, x_p^i)^T$, that has a multivariate normal distribution with parameter $\Theta = (\mu, \Sigma)$, the *likelihood function L(θ)* is given by

$$\begin{aligned} L(\theta) &\propto \prod_{i=i}^{n} \det(\Sigma)^{-1/2} \exp\left\{-\frac{1}{2}(x^i-\mu)^T\Sigma^{-1}(x^i-\mu)\right\} \\ &\propto \det(\Sigma)^{-n/2} \exp\left\{-\frac{1}{2}\sum_{i=i}^{n}(x^i-\mu)^T\Sigma^{-1}(x^i-\mu)\right\}. \end{aligned} \tag{3.5}$$

Rewriting the likelihood function in the log form, we obtain the *log-likelihood function l(θ)*

$$l(\theta) = -\frac{n}{2}\log\det(\Sigma) - \frac{1}{2}\sum_{i=i}^{n}(x^i-\mu)^T\Sigma^{-1}(x^i-\mu). \tag{3.6}$$

Define the *sample mean* vector and the *sample covariance* matrix, that are sufficient statistics for μ and Σ, by

$$\bar{x} = \frac{1}{n}\sum_{i=i}^{n} x^i \quad \text{and} \quad S = \frac{1}{n}\sum_{i=i}^{n}(x^i-\bar{x})(x^i-\bar{x})^T,$$

the log-likelihood function becomes

$$l(\theta) = -\frac{n}{2}\log\det(\Sigma) - \frac{n}{2}\mathrm{tr}(\Sigma^{-1}S) - \frac{n}{2}(\bar{x}-\mu)^T\Sigma^{-1}(\bar{x}-\mu), \tag{3.7}$$

where tr$(\cdot)$ indicates the trace of a matrix (see Whittaker, 1990, pg. 171-172).
In the initial part of this section we underline the fact that in a GGM the mean vector μ is a nuisance parameter and the main interest is for Σ. So, we may set the mean vector equal to its maximum likelihood estimator, i.e. $\hat{\mu} = \bar{x}$, and take the profile log-likelihood function defined by $l(\Sigma) = l(\hat{\mu}, \Sigma)$. Then, the log-likelihood function of equation (3.7) in terms of the concentration matrix Ω becomes

$$l(\Omega) = \frac{n}{2} \log\det(\Omega) - \frac{n}{2} \mathrm{tr}(\Omega S), \tag{3.8}$$

and the maximum likelihood estimator for Ω is given by $\hat{\Omega} = \hat{\Sigma}^{-1} = S^{-1}$ (Whittaker, 1990, pg. 175), or considering the unbiased version of the maximum likelihood estimator of Σ, i.e. $S^u = \frac{n}{n-1}S$.
Finally, if we consider to have a GGM with a graph $\mathcal{G}$ the maximum likelihood estimator of Ω is given by the following theorem

Theorem 1 *The maximum likelihood estimator of a graphical model with graph $\mathcal{G}$, based on a random sample from the multivariate Normal distribution, satisfies the likelihood equations*

$$\hat{\omega}_{ij} = 0,$$

whenever vertices i and j are not adjacent in $\mathcal{G}$, and,

$$\hat{\Sigma}_{aa} = S_{aa}$$

whenever the subset a of vertices in $\mathcal{G}$ form a clique. The estimated parameters Ω and Σ are related by $\hat{\Omega} = \hat{\Sigma}^{-1}$, and are unique with probability one.

See proof in Whittaker (1990, Theorem 6.6.1).

3.3.3 Learning the structure of a GGM

There are several procedures for learning a GGM from data, but when the number of variables is large, most of these methods are computationally unfeasible. In this thesis we consider methods designed to deal with the *large p-small n* issue, typical on biological data, and compare such methods with a simple frequentist method.
Given a random sample of n observations from the vector of variables $\mathbf{X}$, a naive frequentist procedure for learning a GGM goes as follow.

1. Estimate the covariance matrix Σ by the sample covariance matrix $S = \hat{\Sigma}$.

2. Compute its inverse S^{-1} to obtain an estimate of the concentration matrix $\hat{\Omega}$. It should be noted that it is possible to use the sample correlation matrix $\hat{R}$ instead of S.

3. Compute the sample partial correlation coefficients applying the formula (3.4) to the elements of $\hat{\Omega}$.

4. Use the procedure described below to remove edges from the complete graph.

A basic operation to perform the structural learning of a GGM (stage 4) is given by the $p(p-1)/2$ statistical tests

$$H_0 : \rho_{ij \cdot V \setminus \{i,j\}} = 0 \text{ vs } H_1 : \rho_{ij \cdot V \setminus \{i,j\}} \neq 0. \tag{3.9}$$

Under the null hypothesis that the true partial correlation $\rho_{ij \cdot V \setminus \{i,j\}}$ is zero, we consider the test given by

$$t = \sqrt{n-p}\, \frac{\widehat{\rho}_{ij \cdot V \setminus \{i,j\}}}{\sqrt{1 - (\widehat{\rho}_{ij \cdot V \setminus \{i,j\}})^2}}, \tag{3.10}$$

which is distributed as a Student's t with $n-p$ degrees of freedom (Lauritzen, 1996, Section 5.3.3). This test is equivalen to the t-test for the hypothesis that the partial correlation regression coefficient $\hat{\beta}_{ij \cdot V \setminus \{i\}}$ is equal to zero in the model for linear regression of X_i on $X_{V \setminus \{i\}}$. In the rest of this thesis, we refer to this procedure with the name of t-test approach.

If in the true graph $\mathcal{G}$ there is an edge between vertices i and j, then hypothesis H_0 is false and the alternative H_1 is true. Consequently, if we have performed the $p(p-1)/2$ tests of the hypotheses in (3.9), then we draw an edge between i and j in the graph if and only if the hypothesis H_0 is rejected. Let $\alpha \in (0,1)$ be the significance level employed, and let π_{ij} be the p-value for H_0 in (3.9). Hence, the graph $\hat{\mathcal{G}}(\alpha)$ that is selected at level α has the adjacency matrix $\hat{A}(\alpha) = \{\hat{a}_{ij}(\alpha)\}$ with entries

$$\hat{a}_{ij}(\alpha) = \hat{a}_{ji}(\alpha) = \begin{cases} 1 & \text{if } \pi_{ij} < \alpha, \\ 0 & \text{if } \pi_{ij} \geq \alpha. \end{cases}$$

In this procedure of model selection $k = p(p-1)/2$ simultaneous tests are carried out, so a multiple testing procedure should be apply. Here, we consider to use the method introduced by Benjamini and Hochberg (1995), the *false discovery rate* (FDR), which controls the expected proportion of incorrectly rejected null hypotheses (type I errors) for independent test statistics. For a review on other multiple testing procedures in GGM context see the paper by Drton and Perlman (2007). Controlling the FDR at level

α allows us to select a graph $\hat{\mathcal{G}}^*(\alpha)$ such that the proportion of incorrect edges among all the edges of $\hat{\mathcal{G}}^*(\alpha)$ is smaller than α in expectation

$$E\left[\frac{\text{edges incorrectly included in } \hat{\mathcal{G}}^*(\alpha)}{\text{edges included in } \hat{\mathcal{G}}^*(\alpha)}\right] \leq \alpha.$$

Let $H_1, \dots, H_k$ be the null hypotheses, $\pi_1, \dots, \pi_k$ their corresponding p-values and η_0 the fraction of true zero partial correlations. The procedure to controls the FDR at level α is as follows:

1. Construct the set of ordered $p-$values $\pi_{(1)}, \pi_{(2)}, \dots, \pi_{(k)}$ with corresponding edges $e_{(1)}, e_{(2)}, \dots, e_{(k)}$.
2. Let i_α be the largest i for which $\pi_{(i)} \leq (\frac{i}{E})(\frac{\alpha}{\eta_0})$.
3. Reject the null hypothesis of zero partial correlation for all edges $e_{(1)}, e_{(2)}, \dots, e_{(i_\alpha)}$ that satisfies the constrain.

Note that the most conservative choice is to set $\eta_0 = 1$ (Benjamini and Hochberg, 1995); alternatively, η_0 may be estimated adaptively from the data (Benjamini and Hochberg, 2000; Schäfer and Strimmer, 2005a).

Chapter 4

Construct a synthetic gene regulatory network

4.1 Synthetic gene regulatory networks

4.1.1 Graph motifs

Recent works presented by Uri Alon and his group (Milo et al., 2002; Alon, 2007) indicate that transcriptional networks contain a small set of recurring regulation patterns, called *network motifs*. Network motifs are defined as patterns of interconnections that recur in many different parts of a network at frequencies much higher than those found in random networks. They were first discovered in the gene regulatory network of the bacteria *Escherichia coli* (Shen-Orr et al., 2002). The same motifs have since been found in the transcription networks of other bacteria (Eichenberger et al., 2004), as well as yeast (Lee et al., 2002; Milo et al., 2002), and higher organisms (Odom et al., 2004; Boyer et al., 2005; Iranfar et al., 2006). Moreover, every network motif is associated with a specific information-processing functions in gene regulatory networks. A brief overview of some of the most common network motifs and their associated function is given below; for further details consult Milo et al. (2002) and Alon (2007).

The first basic transcription interaction is the *simple interaction* represented by a single arrow in the network; simple regulation can serve as a reference for understanding the dynamic functions of network motifs. Simple regulation occurs when a transcription factor Y, activated by a signal S_Y, regulates gene X with no additional interaction (Fig. 4.1a). The second family of network motifs is the *feed-forward loop* (Fig. 4.1b); it appears in hundreds of gene systems and organisms. This motif consists of three genes: a regulator, X, which regulates Y, and a gene Z, which is regulated by both X and Y. Because each of the three regulatory interactions in the feed-forward loop can be either activation or repression, there are different possible structural types of it. The third family of network motifs is called *single-input modules* and they have a simple pattern in which a regulator X regulates a group of target genes and no other regulator regulates any of these genes (Fig. 4.1c). The main function of this motif is to allow coordinated expression of a group of genes with shared function. We remark that even though transcriptional networks are know to be sparse, single input modules may contain a large number of target genes $Z_1, \ldots, Z_m$ forming in this way a hub structure. The final family of network motifs consist of a set of regulators that combinatorially controls a set of output genes and they are called *multi-input motifs* (Fig. 4.1d). In order to better understand the function of this motif one has to obtain more information about the way the multiple inputs are integrated by the genes. Formally, given either a directed or an undirected graph $\mathcal{G} = (V, E)$ a *motif* is a proper subgraph $\mathcal{G}_A$ of $\mathcal{G}$ where $A \subset V$ is the set of vertices (genes) of the motif. The *size* of the motif is $|A|$, that is the cardinality of A. A *match* of a motif $\mathcal{G}_A$ is a subgraph $\mathcal{G}_B$ of $\mathcal{G}$ which is isomorph to $\mathcal{G}_A$. This means that $|A| = |B|$ and there exists a bijective function $h : A \rightarrow B$ such that any two vertices i and j of $\mathcal{G}_A$ are adjacent in $\mathcal{G}_A$ if and only if $h(i)$ and $h(j)$ are adjacent in $\mathcal{G}_B$. In figure 4.2, there is an illustration of a target graph $\mathcal{G}$ (a), a motif $\mathcal{G}'$ (b), and a highlighted match $\mathcal{G}''$ of the motif $\mathcal{G}'$ in the target graph $\mathcal{G}$ (c).

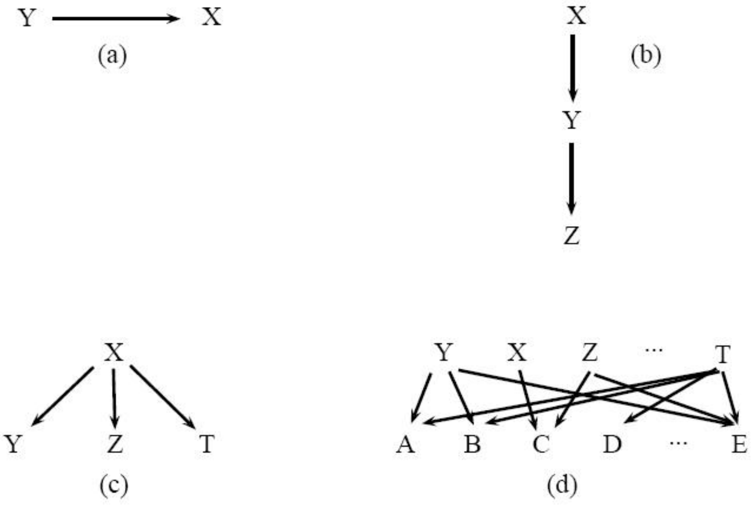


Figure 4.1: Common network motifs.

4.1.2 The graph structures for the simulation study

For the simulation study, we reproduce networks that have two characteristics: they represent as close as possible the real scenario of gene regulatory networks, i.e. (i) recurrent graph motifs and (ii) the sparseness.

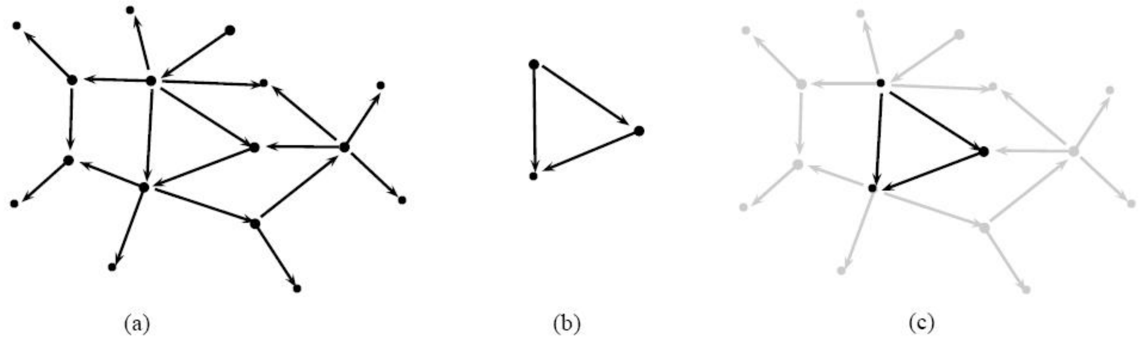


Figure 4.2: Example of graph motif.

The graph structure under study, based on the network motifs presented in the previous section, are

1. the hub structure;
2. the cascade structure;
3. the pairwise structure.

The hub frame (Fig. 4.3a) that represents the single-input motif, is a common type of structure in a gene regulatory network; in addition, it is also one of the most difficult structure to be discovered in structural learning. Indeed, it is very convenient to induce graph sparseness by assuming an upper bound to the number of neighbours (or parents) of every vertex and this excludes the presence of hubs from the graph. Furthermore, hub structures are difficult to identify when the sample size is small because the conditional distribution of the transcription factor given the target genes involves a number of variables that may exceed the sample size. The cascade structure (Fig. 4.3b) is similar to the feed-forward loop motif, but without the interaction between the first and last gene; hence, it is represented by a sequence of interaction between genes, in which every gene has at least one connection but at most two connections. Finally, the pairwise structure (Fig. 4.3c) refers to the simple interaction in which pair of genes are connected. It is important to underline a statistical particularity of this last structure: the marginal dependences coincide with the conditional dependences, i.e. zeros in Σ and Ω are the same. Starting from these three graph motifs, we generate data sets from a specific sparse network that includes only a single graph motif replicated a certain number of times (i.e., matches) and with a specific size (i.e., motif size). The network has also a fixed, but sparse, number of interactions out of all possible interactions (i.e., edges). We simulate three groups of network structures that have different number of variables p, that represent the genes. The relative values for every group are

GROUP 1 ($p = 20$)

- HUB (Fig. 4.3a): matches = 1, edges = 10, motif size = 11.
- CASCADE (Fig. 4.3b): matches = 1, edges = 10, motif size = 11.
- PAIRWISE (Fig. 4.3c): matches = 10, edges = 10, motif size = 2.

GROUP 2 ($p = 100$)

- HUB: matches = 9, edges = 91, motif size = 11.
- CASCADE: matches = 9, edges = 91, motif size = 11.
- PAIRWISE: matches = 50, edges = 50, motif size = 2.

GROUP 3 ($p = 200$)

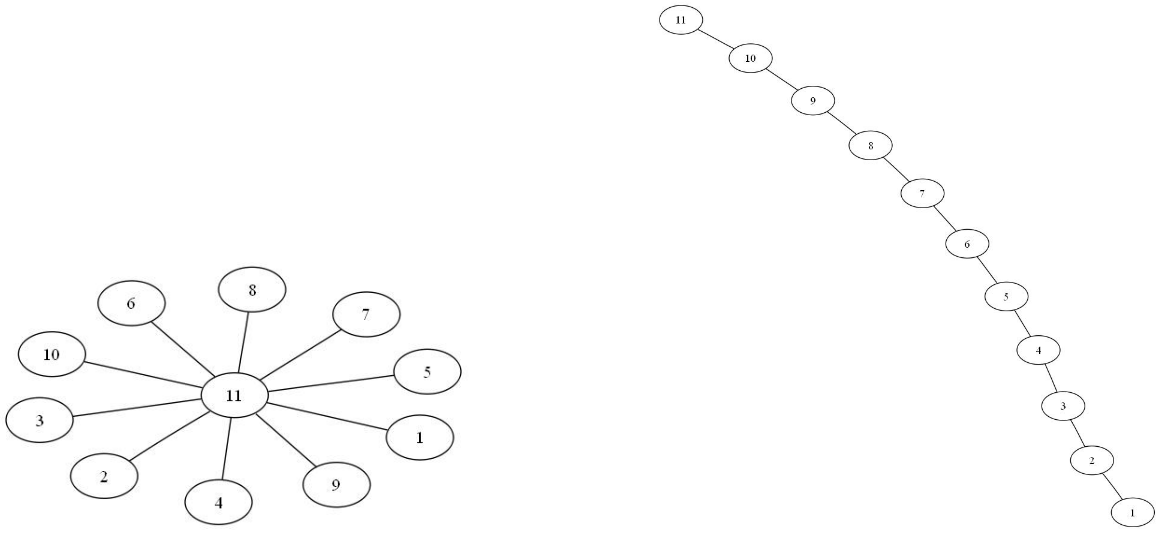
- HUB: matches = 18, edges = 182, motif size = 11.
- CASCADE: matches = 18, edges = 182, motif size = 11.
- PAIRWISE: matches = 100, edges = 100, motif size = 2.

Note that for both hub and cascade structure in the first group 9 variables are not part of the motif and they are totally independent. Moreover, in the second group one graph motif has "motif size = 12", and in the third group two graph motifs have "motif size = 12".

4.1.3 Markov equivalence between graph structures

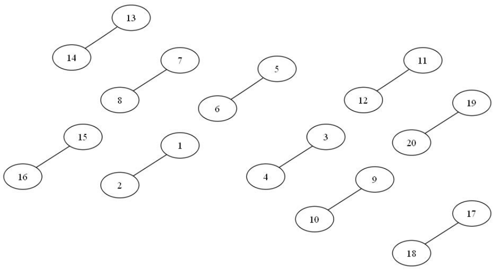
The three graph structures presented in the previous section have a peculiarity: they are all decomposable graphs. Hence, there exists for each of the three motifs in the structure a perfect DAG that is Markov equivalent to this motif. Figure 4.4 shows some Markov equivalent DAGs for the hub structure (a), the cascade structure (b), and the pairwise structure (c).

The use of these graph structures, that belong to the class of Markov equivalent models between undirected graph and DAG, allows us to consider both these types of models to represent the simulated networks. In practice, we can use both types of models to study the same data set. Consequently, we can compare the relative merits and shortcomings of the methods that have been proposed in the literature for both models in a fair manner.



(a) Hub

(b) Cascade



(c) Pairwise

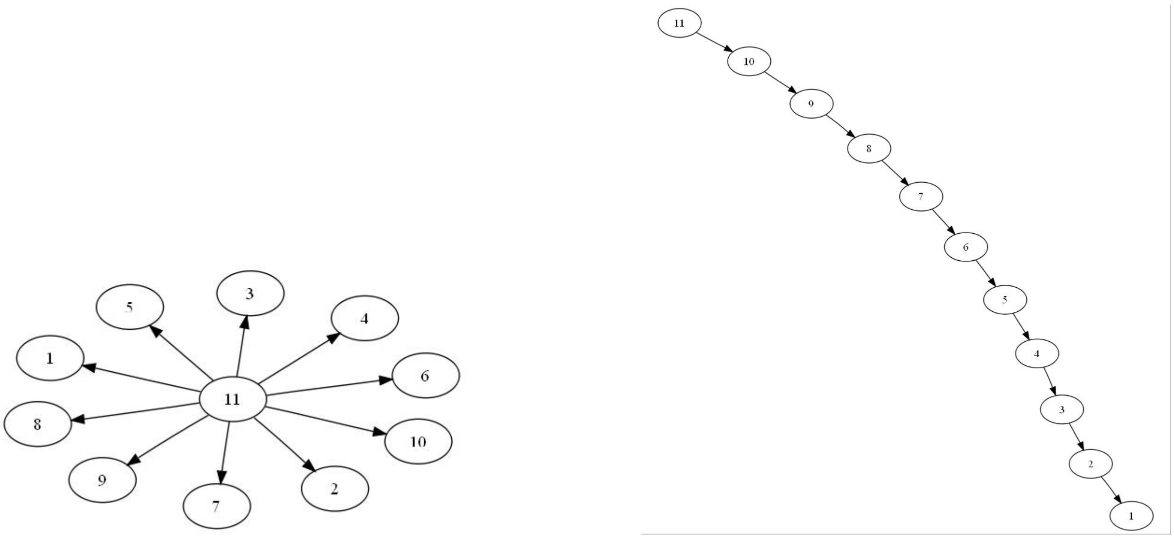
Figure 4.3: Network motifs under study.

4.2 Generating synthetic gene regulatory networks

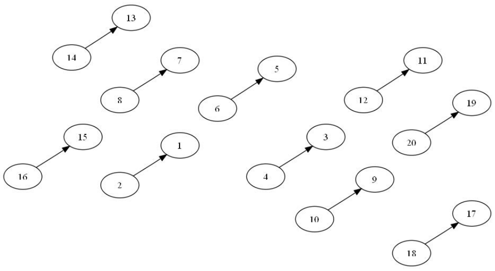
4.2.1 Gene regulatory network: how to generate a specific structure

In the previous section we emphasized the idea that in a simulation study the main goal is to imitate the real scenario of interest as close as possible. This means that in the process of data set generation it is important to consider the principal biological characteristics of the network: the presence of recurrent graph motifs and the sparseness in term of interactions between the elements of the network.

Considering the use of Gaussian graphical models as a tool for studying gene regulatory networks, for reproducing a graph motif we have to define a graph $\mathcal{G}$ with a definite set of vertices and edges based on the motif of interest. The sparseness, instead, is implemented by a small number of interactions out of all possible interactions between the elements of the network, i.e. a large number of zero entries in the concentration matrix. Hence, in



(a) Hub (b) Cascade



(c) Pairwise

Figure 4.4: Perfect DAGs version of the graph motifs under study.

order to simulate data from a GGM with a specific and sparse structure we have to define the concentration matrix $\Sigma^{-1} = \{\omega_{ij}\}$ of the model, where the elements ω_{ij} different from zero indicate the conditional dependency between the variables of the graph. In addition to these two biological peculiarities of the gene regulatory networks, we have also to take in consideration the mathematical aspect related to GGM, in particular to the concentration matrix. Σ^{-1} has to be a positive definite matrix and the alternative parametrization of the concentration matrix given by the Cholesky decomposition, for decomposable graphs, is a suitable solution to create a concentration matrix with a specific and sparse structure, but at the same time it guarantees the positive definiteness of the matrix.

4.2.2 Cholesky decomposition

Let us consider a decomposable graph $\mathcal{G} = (V, E)$ with vertex set $V = \{1, \ldots, p\}$ and the set of edges $E \subseteq V \times V$. We assume that the vertices in the set V are ordered according a perfect numbering presented in section (3.1), but taken in reverse order such that 1 is the last vertex and p is the first vertex in the ordered set V. In addition, we assume that $\mathbf{X} \equiv \mathbf{X}_V$ is a random vector with a multivariate normal distribution $N(0, \Sigma)$ and $\mathcal{D}$ is a perfect DAG associated with $\mathcal{G}$, obtained from the above vertex ordering.

From the results presented in the work of Paulsen et al. (1989), since the rows and columns of Σ are ordered according to a perfect vertex elimination scheme for $\mathcal{G}$, we can define the concentration matrix using the Cholesky decomposition $\Sigma^{-1} = \Phi^T \Phi$, where Φ is a upper triangular matrix. This decomposition is useful because the upper triangular matrix Φ has the same zero pattern as Σ^{-1} and the elements of Φ are variation independent (Barndorff-Nielsen, 1976; Lauritzen, 1996, Appendix 5). Moreover, the elements of Φ are interpretable as parameters of the conditional distribution involved in the recursive factorization of the density function of $\mathbf{X}$ according to $\mathcal{D}$ (Wermuth, 1980). In the following, it is presented how we can provide a decomposable covariance model, with a given structure in the concentration matrix, by means of a triangular matrix Φ and Cholesky decomposition.

For the partition of the vertex set V into the subset $A \subset V$ and $B = V \backslash A$, we have that $X_A \sim N(0, \Sigma_{AA})$ and $X_A | X_B \sim N(\Gamma_{A|B} x_B, \Sigma_{A|B})$ where

$$\Gamma_{A|B} = \Sigma_{AB} \Sigma_{BB}^{-1} \quad , \quad \Sigma_{A|B} = \Sigma_{AA} - \Sigma_{AB} \Sigma_{BB}^{-1} \Sigma_{BA}.$$

So, the concentration matrix Σ^{-1}, associated to $\mathcal{G}$, using the Cholesky decomposition can be written as

$$\mathbf{\Sigma^{-1}} = \begin{pmatrix} \Phi_{AA}^T & 0 \\ \Phi_{BA} & \Phi_{BB}^T \end{pmatrix} \begin{pmatrix} \Phi_{AA} & \Phi_{AB} \\ 0 & \Phi_{BB} \end{pmatrix}.$$

The upper row-block (Φ_{AA}, Φ_{AB}) of Φ gives an alternative parametrization of the conditional distribution $X_A | X_B$, whereas the lower row-block Φ_{BB} derives from the Cholesky decomposition of $(\Sigma_{BB})^{-1}$ and it indicates the marginal distribution of X_B. For obtaining the one-to-one transformation between the elements of the upper triangular matrix Φ and the parameters of the distribution of the variables associated to partitioned vertex set V, we apply the rules for the inverse of a partitioned matrix to the Cholesky decomposition $\Sigma^{-1} = \Phi^T \Phi$. Then, we have

$$\Phi_{AA}^T \Phi_{AA} = (\Sigma_{A|B})^{-1} \quad , \quad -\Phi_{AA}^{-1} \Phi_{AB} = (\Gamma_{A|B}) \quad , \quad \Phi_{BB}^T \Phi_{BB} = \Sigma_{BB}^{-1}$$

(Roverato, 2000).

In the specific, given the matrix product of the Cholesky decomposition

$$\mathbf{\Sigma^{-1}} = \begin{pmatrix} \Phi_{AA}^T & 0 \\ \Phi_{BA} & \Phi_{BB}^T \end{pmatrix} \begin{pmatrix} \Phi_{AA} & \Phi_{AB} \\ 0 & \Phi_{BB} \end{pmatrix} = \begin{pmatrix} \Phi_{AA}^T \Phi_{AA} & \Phi_{AA}^T \Phi_{AB} \\ \Phi_{BA} \Phi_{AA} & \Phi_{BA} \Phi_{AB} + \Phi_{BB}^T \Phi_{BB} \end{pmatrix},$$

the above results are obtaining as follow

1. $\left(\Sigma^{-1}\right)_{AA} = \left[\Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}\right]^{-1} = (\Sigma_{A|B})^{-1}$,
   $\left(\Sigma^{-1}\right)_{AA} = \Phi_{AA}^T\Phi_{AA}$
   $\Longrightarrow \Phi_{AA}^T\Phi_{AA} = (\Sigma_{A|B})^{-1}$.

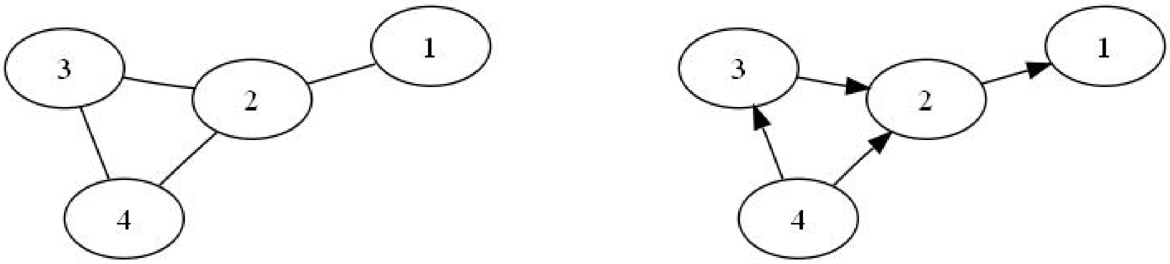
2. $\left(\Sigma^{-1}\right)_{AB} = -\left(\Sigma_{A|B}\right)^{-1}\Sigma_{AB}\Sigma_{BB}^{-1} = -\Phi_{AA}^T\Phi_{AA}\Gamma_{A|B}$,
   $\left(\Sigma^{-1}\right)_{AB} = \Phi_{AA}^T\Phi_{AB}$,
   $\Phi_{AA}^T\Phi_{AB} = -\Phi_{AA}^T\Phi_{AA}\Gamma_{A|B}$
   $\Longrightarrow -\Phi_{AA}^{-1}\Phi_{AB} = \Gamma_{A|B}$.

3. $\Sigma_{BB} = \left[\left(\Sigma^{-1}\right)_{BB} - \left(\Sigma^{-1}\right)_{BA}\left(\Sigma^{-1}\right)_{AA}^{-1}\left(\Sigma^{-1}\right)_{AB}\right]^{-1}$,
   $\left(\Sigma^{-1}\right)_{BB} = \Phi_{BA}\Phi_{AB} + \Phi_{BB}^T\Phi_{BB}$,
   $\left(\Sigma^{-1}\right)_{BA} = \Phi_{BA}\Phi_{AA}$,
   $\left(\Sigma^{-1}\right)_{AA}^{-1} = \Phi_{AA}^{-1}\Phi_{AA}^{-T}$,
   $\left(\Sigma^{-1}\right)_{AB} = \Phi_{AA}^T\Phi_{AB}$,
   $\Sigma_{BB} = \Phi_{BA}\Phi_{AB} + \Phi_{BB}^T\Phi_{BB} - \Phi_{BA}\Phi_{AA}\Phi_{AA}^{-1}\Phi_{AA}^{-T}\Phi_{AA}^T\Phi_{AB}$,
   $\quad = \Phi_{BA}\Phi_{AB} + \Phi_{BB}^T\Phi_{BB} - \Phi_{BA}\Phi_{AB}$
   $\Longrightarrow (\Sigma_{BB})^{-1} = \Sigma_{BB}^{-1} = \Phi_{BB}^T\Phi_{BB}$.

The Cholesky decomposition $\Phi^T\Phi$ generates a concentration matrix, with a given zero pattern, for a decomposable covariance model for two reasons: the Markov equivalence between a decomposable graph and its associated perfect DAG, and the relation between a perfect DAG and the upper triangular matrix Φ. In the specific, for decomposable covariance models the factorization on $\mathcal{G}$ and the recursive factorization on $\mathcal{D}$ of the distribution of $\mathbf{X}$ are equivalent (see Section 3.2.4). Consequently, we can base the parametrization of the decomposable covariance model on Φ, that is strictly related to the recursive factorization on $\mathcal{D}$. Indeed, given the perfect elimination scheme of V, in the upper triangle of Φ the ith row with respect to $\mathcal{D}$, $\left(\Phi_{\{i\}\{i\}}, \Phi_{\{i\}pa(i)}\right)$, has $v_i + 1$ non-zero elements, with $v_i = |pa(i)|$. Applying recursively this procedure one vertex at time, we obtain that the ith row of Φ provides an alternative parametrization of the conditional distribution of $X_i|X_{pa(i)}$. These conditional statements are described by a linear recursive system where the zero pattern for the regression coefficients is the same as in the concentration matrix (Wermuth, 1980).

4.2.3 An example of the use of Cholesky decomposition to design a specific Σ^{-1}

In order to describe the use of the Cholesky decomposition to obtain Σ^{-1} for a specific network, we use a simple graph with only 4 vertices and 4 edges.
Let us consider the decomposable graph $\mathcal{G}$ presented in figure 4.5a, since its vertices are enumerated according to a perfect vertex elimination scheme, there exists a perfect directed acyclic version $\mathcal{D}$ associated with it (Fig. 4.5b). The joint density function related to this graph, for the factorizations property, is given by



(a) Decomposable graph

(b) Perfect DAG

Figure 4.5: General example of graph.

$$f(x_1, x_2, x_3, x_4) = f(x_1|x_2)f(x_2|x_3, x_4)f(x_3|x_4)f(x_4).$$

Following Wermuth (1980) we have that the variables, indexed by the vertices of the graph, imply a system of recursive equations as follows

$$\begin{aligned}
&X_1 = \beta_{1,2}X_2 + \epsilon_1 \quad \text{with} \;\; \epsilon_1 \sim N(0, \sigma^2_{1|2});\\
&X_2 = \beta_{2,3}X_3 + \beta_{2,4}X_4 + \epsilon_2 \quad \text{with} \;\; \epsilon_2 \sim N(0, \sigma^2_{2|3,4});\\
&X_3 = \beta_{3,4}X_4 + \epsilon_3 \quad \text{with} \;\; \epsilon_3 \sim N(0, \sigma^2_{3|4});\\
&X_4 \sim N(0, \sigma^2_4).
\end{aligned}$$

The above equations suggest the upper triangular matrix Φ of the Cholesky decomposition of Σ^{-1} with the following form

$$\Phi = \begin{pmatrix} \phi_{1,1} & \phi_{1,2} & 0 & 0 \\ & \phi_{2,2} & \phi_{2,3} & \phi_{2,4} \\ & & \phi_{3,3} & \phi_{3,4} \\ & & & \phi_{4,4} \end{pmatrix},$$

where the elements of Φ are equal to

$$
\begin{aligned}
&\phi_{1,1} = \frac{1}{\sqrt{\sigma^2_{1|2}}}, \quad \phi_{1,2} = -\frac{\beta_{1,2}}{\sqrt{\sigma^2_{1|2}}};\\
&\phi_{2,2} = \frac{1}{\sqrt{\sigma^2_{2|3,4}}}, \quad \phi_{2,3} = -\frac{\beta_{2,3}}{\sqrt{\sigma^2_{2|3,4}}}, \quad \phi_{2,4} = -\frac{\beta_{2,4}}{\sqrt{\sigma^2_{1|3,4}}};\\
&\phi_{3,3} = \frac{1}{\sqrt{\sigma^2_{3|4}}}, \quad \phi_{3,4} = -\frac{\beta_{3,4}}{\sqrt{\sigma^2_{3|4}}};\\
&\phi_{4,4} = \frac{1}{\sqrt{\sigma^2_{4}}};
\end{aligned}
$$

(Cox and Wermuth, 1996, Chapter 3).
Finally, we can obtain the concentration matrix, with the desired zero pattern, as $\Sigma^{-1} = \Phi^T\Phi$

$$
\Sigma^{-1} = \Phi^T\Phi = \begin{pmatrix} \phi_{1,1} & & & \\ \phi_{2,1} & \phi_{2,2} & & \\ 0 & \phi_{3,2} & \phi_{3,3} & \\ 0 & \phi_{4,2} & \phi_{4,3} & \phi_{4,4} \end{pmatrix} \begin{pmatrix} \phi_{1,1} & \phi_{1,2} & 0 & 0 \\ & \phi_{2,2} & \phi_{2,3} & \phi_{2,4} \\ & & \phi_{3,3} & \phi_{3,4} \\ & & & \phi_{4,4} \end{pmatrix}
$$

$$
= \begin{pmatrix} \phi^2_{1,1} & \phi_{1,1}\,\phi_{1,2} & 0 & 0 \\ & \phi^2_{1,2} + \phi^2_{2,2} & \phi_{2,2}\,\phi_{2,3} & \phi_{2,2}\,\phi_{2,4} \\ & & \phi^2_{2,3} + \phi^2_{3,3} & \phi_{3,2}\,\phi_{2,4} + \phi_{3,3}\,\phi_{3,4} \\ & \text{symm.} & & \phi^2_{2,4} + \phi^2_{3,4} + \phi^2_{4,4} \end{pmatrix}.
$$

4.3 Data generation

4.3.1 Procedure for data set generation

To simulate a data set according to a gene regulatory network, with a specific structure and using a Gaussian graphical model, the principal steps are

1. Translate the identified graph structure into a zero pattern of Σ^{-1}.

2. Construct the concentration matrix Σ^{-1}, with the given zero pattern, using the Cholesky decomposition $\Sigma^{-1} = \Phi^T\Phi$. In Σ^{-1} the elements equal to zero indicate conditional independences.

3. Simulate data from a multivariate Normal distribution with mean vector μ equal

to zero and covariance matrix Σ that derives from the concentration matrix $\Sigma^{-1} = \Phi^T \Phi$.

4.3.2 Construction of the concentration matrix

According to the above procedure for the simulation of data, the first two stages concern the generation of Σ^{-1} with the given zero pattern. In the following, we present the generation of the three concentration matrices only for the first group with $p = 20$. The concentration matrices of the other two groups of network structures, with $p = 100$ and $p = 200$, differ from the first group only for the number of replications of the basic graph motifs (see Section 4.1.2 for the frequencies of basic graph motifs in the groups).

HUB structure

Starting from the decomposable undirected graph (Fig. 4.3a), we obtain the associated perfect DAG (Fig. 4.4a), with the following joint density function

$$f(x_1, x_2, \ldots, x_{20}) = f(x_1|x_{11})f(x_2|x_{11})\ldots f(x_{10}|x_{11})f(x_{11})f(x_{12})\ldots f(x_{20}).$$

The system of recursive equations is

$$\begin{aligned}
X_1 &= \beta_{1,11}X_{11} + \epsilon_1 \quad \text{with} \;\; \epsilon_1 \sim N(0, \sigma^2_{1|11});\\
X_2 &= \beta_{2,11}X_{11} + \epsilon_2 \quad \text{with} \;\; \epsilon_2 \sim N(0, \sigma^2_{2|11});\\
&\vdots\\
X_{10} &= \beta_{10,11}X_{11} + \epsilon_{10} \quad \text{with} \;\; \epsilon_{10} \sim N(0, \sigma^2_{10|11});\\
X_{11} &\sim N(0, \sigma^2_{11});\\
X_{12} &\sim N(0, \sigma^2_{12});\\
&\vdots\\
X_{20} &\sim N(0, \sigma^2_{20}).
\end{aligned}$$

Hence, we can construct the upper triangular matrix Φ that has the form

$$\Phi = \begin{pmatrix}
\phi_{1,1} & 0 & & \cdots & 0 & \phi_{1,11} & 0 & & \cdots & 0 \\
& \phi_{2,2} & 0 & \cdots & 0 & \phi_{2,11} & 0 & & \cdots & 0 \\
& & \ddots & & \vdots & \vdots & \vdots & & \cdots & \vdots \\
& & & \ddots & \vdots & \vdots & \vdots & & \cdots & \vdots \\
& & & & \phi_{10,10} & \phi_{10,11} & 0 & & \cdots & 0 \\
& & & & & \phi_{11,11} & 0 & & \cdots & 0 \\
& & & & & & \phi_{12,12} & & \cdots & 0 \\
& & & & & & & \ddots & & \vdots \\
& & & & & & & & \ddots & \vdots \\
& & & & & & & & & \phi_{20,20}
\end{pmatrix},$$

where

$$\begin{aligned}
\phi_{i,i} &= \frac{1}{\sqrt{\sigma^2_{i|11}}}, \quad \text{for} \quad i = 1, \ldots, 10; \\
\phi_{i,i} &= \frac{1}{\sqrt{\sigma^2_{i}}}, \quad \text{for} \quad i = 11, \ldots, 20; \\
\phi_{i,11} &= -\frac{\beta_{i,11}}{\sqrt{\sigma^2_{i|11}}}, \quad \text{for} \quad i = 1, \ldots, 10.
\end{aligned}$$

Then, using the Cholesky decomposition, the concentration matrix Σ^{-1} is

$$\Sigma^{-1} = \begin{pmatrix}
\phi^2_{1,1} & 0 & & \cdots & 0 & \phi_{1,1}\,\phi_{1,11} & 0 & & \cdots & 0 \\
& \phi^2_{2,2} & 0 & \cdots & 0 & \phi_{2,2}\,\phi_{2,11} & 0 & & \cdots & 0 \\
& & \ddots & & \vdots & \vdots & \vdots & & \cdots & \vdots \\
& & & \ddots & \vdots & \vdots & \vdots & & \cdots & \vdots \\
& & & & \phi^2_{10,10} & \phi_{10,10}\,\phi_{10,11} & 0 & & \cdots & 0 \\
& & & & & \sum_{i=1}^{11} \phi^2_{i,11} & 0 & & \cdots & 0 \\
& & & & & & \phi^2_{12,12} & & \cdots & 0 \\
& & & & & \text{symm.} & & \ddots & & \vdots \\
& & & & & & & & \ddots & \vdots \\
& & & & & & & & & \phi^2_{20,20}
\end{pmatrix}.$$

CASCADE structure

Starting from the decomposable undirected graph (Fig. 4.3b), we obtain the associated perfect DAG (Fig. 4.4b) and its joint density function has the form

$$f(x_1, x_2, \dots, x_{20}) = \left[\prod_{i=1}^{10} f(x_i|x_{i+1})\right] f(x_{11})f(x_{12})\dots f(x_{20}).$$

The system of recursive equations is

$$\begin{aligned}
X_i &= \beta_{i,i+1}X_{i+1} + \epsilon_i \quad \text{with} \;\; \epsilon_i \sim N(0, \sigma^2_{i|i+1}) \;\; \text{for} \quad i = 1, \dots, 10;\\
X_{11} &\sim N(0, \sigma^2_{11});\\
X_{12} &\sim N(0, \sigma^2_{12});\\
&\vdots\\
X_{20} &\sim N(0, \sigma^2_{20}).
\end{aligned}$$

Hence, we can construct the upper triangular matrix Φ which elements different from zero are

$$\begin{aligned}
\phi_{i,i} &= \frac{1}{\sqrt{\sigma^2_{i|i+1}}}, \quad \text{for} \;\; i = 1, \dots, 10;\\
\phi_{i,i} &= \frac{1}{\sqrt{\sigma^2_i}}, \quad \text{for} \;\; i = 11, \dots, 20;\\
\phi_{i,i+1} &= -\frac{\beta_{i,i+1}}{\sqrt{\sigma^2_{i|i+1}}}, \quad \text{for} \;\; i = 1, \dots, 10.
\end{aligned}$$

Then, using the Cholesky decomposition, we obtain the concentration matrix Σ^{-1} with elements different from zero given by

$$\begin{aligned}
\omega_{i,i} &= \phi^2_{i,i}, \quad \text{for} \;\; i = 1, 11 \dots, 20;\\
\omega_{i,i} &= \phi^2_{i,i} + \phi^2_{i-1,i}, \quad \text{for} \;\; i = 2, \dots, 10;\\
\omega_{i,i+1} &= \phi^2_{i,i} + \phi^2_{i,i+1}, \quad \text{for} \;\; i = 1, \dots, 10.
\end{aligned}$$

PAIRWISE structure

Starting from the decomposable undirected graph (Fig. 4.3c), we obtain the associated perfect DAG (Fig. 4.4c). Its joint density function has the form

$$f(x_1, x_2, \ldots, x_{20}) = \prod_{i \in T} f(x_i | x_{i+1}) f(x_{i+1}),$$

where $T = \{2j - 1,\ \text{with}\ j = 1, \ldots, 10\}$.
The system of recursive equations is

$$\begin{aligned} X_i &= \beta_{i,i+1} X_{i+1} + \epsilon_i \quad \text{with} \ \ \epsilon_i \sim N(0, \sigma^2_{i|i+1}) \quad \text{for} \quad i \in T; \\ X_{i+1} &\sim N(0, \sigma^2_{i+1}) \quad \text{for} \quad i \in T. \end{aligned}$$

Hence, we can construct the upper triangular matrix Φ which elements different from zero are

$$\begin{aligned} \phi_{i,i} &= \frac{1}{\sqrt{\sigma^2_{i|i+1}}} \quad \text{for} \quad i \in T; \\ \phi_{i,i} &= \frac{1}{\sqrt{\sigma^2_{i+1}}} \quad \text{for} \quad i \notin T; \\ \phi_{i,i+1} &= -\frac{\beta_{i,i+1}}{\sqrt{\sigma^2_{i|i+1}}} \quad \text{for} \quad i \in T. \end{aligned}$$

Then, using the Cholesky decomposition, we obtain the concentration matrix Σ^{-1} with elements different from zero given by

$$\begin{aligned} \omega_{i,i} &= \phi^2_{i,i} + \phi^2_{i,i+1}, \quad \text{for} \ \ i \in T; \\ \omega_{i,i} &= \phi^2_{i+1,i+1}, \quad \text{for} \ \ i \notin T; \\ \omega_{i,i+1} &= \phi_{i,i+1} \phi^2_{i+1,i+1}, \quad \text{for} \ \ i \in T. \end{aligned}$$

4.3.3 Simulation of data

The final stage in the generation of data from the given network structures is the simulation of the data sets from multivariate normal distributions $N_p(0, \Sigma)$.
The covariance matrix Σ for every structure, in the three groups, derives from the inverse of the associated concentration matrix $\Sigma^{-1} = \Phi^T \Phi$. The parameter values for obtaining the elements of Φ, i.e. β and σ^2, are considered to be equal among the same structure in all the three groups. Moreover, the choice of their values does not have any reference in literature, but it has the only aim to delineate the network structure in the simulated

data. The values of the parameters, that form the elements of Φ, and the consequent values of the partial correlation coefficients are

HUB

- $\sigma^2_{i|11} = 0.0025$, for $i = 1, \dots 10$;
- $\sigma^2_{11} = 1$;
- $\sigma^2_i = 4$, for $i = 12, \dots, 20$;
- $\beta_{i,11} = 1$, for $i = 1, \dots 10$;

$\Longrightarrow$ $\rho_{i\cdot 11|\text{rest}} = 0.316$, for $i = 1, \dots 10$.

CASCADE

- $\sigma^2_{i|i+1} = 0.0025$, for $i = 1, \dots 10$;
- $\sigma^2_{11} = 1$;
- $\sigma^2_i = 4$, for $i = 12, \dots 20$;
- $\beta_{i,i+1} = 1$, for $i = 1, \dots 10$;

$\Longrightarrow$ $\rho_{i\cdot i+1|\text{rest}} = 0.5$ and $\rho_{i\cdot i+1|\text{rest}} = 0.7$, for $i = 1, \dots 10$.

PAIRWISE

- $\sigma^2_{i|i+1} = 0.49$, for $i \in T$;
- $\sigma^2_i = 1$, for $i \notin T$;
- $\beta_{i,i+1} = 0.5$, for $i \in T$;

$\Longrightarrow$ $\rho_{i\cdot i+1|\text{rest}} = 0.581$, for $i \in T$.

In order to have different setting with respect to the number of variables p, we considered fixed the sample size in all the three groups with value equals to $n = 150$. Then, we have simulated several times the data with the *ad hoc* covariance matrix $\Sigma = (\Phi^T\Phi)^{-1}$ as follow

- Group 1: 2000 replications.
- Group 2: 100 replications.
- Group 3: 100 replications.

Chapter 5

Procedures for learning GRN

The aim of this chapter is to present the recently developed methodologies that we considered in the comparative study for learning GRNs. In the specific, they are: the G-Lasso algorithm (Friedman et al., 2008), the Shrinkage estimator with the empirical Bayes approach for the model selection (Schäfer and Strimmer, 2005a; Schäfer and Strimmer, 2005b), and the PC-algorithm (Kalisch and Bühlmann, 2007).

5.1 G-Lasso

The Graphical Lasso algorithm, called G-Lasso, proposed by Friedman et al. (2008) is a recent and promising approach to estimate sparse undirected graphical models using the idea behind the lasso method proposed by Tibshirani (1996). Tibshirani's lasso method imposes an L_1 penalty for the estimation of the regression coefficients in linear models that consequently sets many coefficients exactly equal to zero. Recently, many authors have proposed the estimation of sparse undirected graphical models through the use of lasso penalized regression approach of Tibshirani. An interesting method is the one presented by Meinshausen and Bühlmann (2006). This approach uses the lasso penalization for model selection in GGM to find the set of neighbors, of each node in the graph, by regressing the corresponding variable of the node against the remaining variables, but it only achieves an approximation to the estimation of Σ^{-1}. Other authors have proposed algorithms for the exact maximization of the L_1-penalized log-likelihood in GGMs context (Yuan and Lin, 2007; Banerjee et al., 2008).

The G-Lasso algorithm (Friedman et al., 2008) is based on the Meinshausen and Bühlmann's lasso and the blockwise coordinate descent algorithm introduced by Banerjee et al. (2008). It fits a lasso model to each variable for the estimation of the concentration matrix, as in Meinshausen and Bühlmann's lasso, but it uses the procedure of Banerjee et al. for solving the lasso problem. Hence, it achieves a sparse estimator of the concentration matrix Σ^{-1} which performs simultaneously parameter estimation and model selection. We suggest also to consult the recent work of Fan et al. (2009) for a vision on G-Lasso properties.

Given a sample with n observations of the random vector $\mathbf{X} = (X_1, \ldots, X_p)^T$ with a multivariate normal distribution $N_p(\mu, \Sigma)$, the penalized log-likelihood that has to be maximized over the concentration matrix $\Omega = \Sigma^{-1}$ is

$$\begin{aligned} l_1(\Omega, \lambda; \mathbf{X}) &= l(\Omega) - \lambda(\Omega) \\ &= \log \det \Omega - \mathrm{tr}(S\Omega) - \lambda \mid\mid \Omega \mid\mid_1, \end{aligned} \tag{5.1}$$

where $\mid\mid \Omega \mid\mid_1$ is the L_1 norm, i.e. the sum of the absolute values of the elements of Σ^{-1}, and $\lambda \geq 0$ is the penalty parameter.

Using the equation (5.1), Banerjee *et al.* (2008) consider the estimation of Σ, rather than Σ^{-1}, by optimizing each row and corresponding column of T, estimator of Σ, using a blockwise interior point procedure. Partitioning T and S as follows

$$\mathbf{T} = \begin{pmatrix} T_{11} & t_{12} \\ t_{12}^T & t_{22} \end{pmatrix}, \mathbf{S} = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix},$$

they show that the solution for t_{12} satisfies

$$t_{12} = \mathrm{argmin}_y \left\{ y^T T_{11}^{-1} y : ||y - s_{12}||_\infty \leq \lambda \right\}. \tag{5.2}$$

Then, they poin out that to solve (5.2) is equivalent to solve the following problem

$$min_\beta \left\{ \frac{1}{2} \mid\mid T_{11}^{1/2} \beta - b \mid\mid^2 + \lambda \mid\mid \beta \mid\mid_1 \right\}, \tag{5.3}$$

where $b = T_{11}^{-1/2} s_{12}$ (see Banerjee *et al.* (2008) for more details). Friedman et al. (2008) show that problem (5.3) is equivalent to a lasso problem where β is the coefficient for the pth variable on the others.

Banerjee *et al.* (2008) do not pursue effectively estimation of the concentration matrix and this is the part presented in Friedman et al. (2008) as G-Lasso algorithm. In details, from the relation $T\Omega = I$ in matrix terms

$$\begin{pmatrix} T_{11} & t_{12} \\ t_{12}^T & t_{22} \end{pmatrix} \begin{pmatrix} \Omega_{11} & \omega_{12} \\ \omega_{12}^T & \omega_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0^T & 1 \end{pmatrix},$$

they derive the two expressions to compute $\hat{\Omega}$

$$\begin{aligned}\omega_{12} &= -T_{11}^{-1} t_{12} \omega_{22};\\ \omega_{22} &= 1/(t_{22} - t_{12}^T T_{11}^{-1} t_{12}).\end{aligned}$$

Since, the solution of lasso problem in (5.3) returns that $\hat{\beta} = T_{11}^{-1} t_{12}$, hence

$$\hat{\omega}_{22} = 1/(t_{22} - t_{12}^T \hat{\beta}); \tag{5.4}$$

$$\hat{\omega}_{12} = -\hat{\beta}\hat{\omega}_{22}. \tag{5.5}$$

So, the G-lasso algorithm is constructed as follows

1. Start with $T = S + \lambda I$. The diagonal of T remains unchanged in what follows.
2. For each $j = 1, 2, \ldots p, 1, 2, \ldots p, \ldots$, solve the lasso problem (5.3), which takes as input the inner products T_{11} and s_{12}. This gives a $p - 1$ vector solution $\hat{\beta}$. Fill in the corresponding row and column of T using $t_{12} = T_{11}\hat{\beta}$.
3. Continue until convergence.
4. Compute $\hat{\Omega}$ using equations (5.4) and (5.5).

In the G-Lasso, the main point is the choice of the penalty parameter. The authors do not give any suggestion for the selection of the optimal λ, but they make two important remarks. First, setting $\lambda = 0$ then $T = S$ and the algorithm computes the maximum likelihood estimator S^{-1} using a linear regression at each stage. Second, the penalty term could be a scalar or a matrix. The first situation imposes the same amount of regularization for every variable; while, a penalty matrix allows to penalize differently each inverse covariance elements.

The use of G-Lasso to estimate the concentration matrix, and then to derive the partial correlation matrix of a Gaussian graphical model is justified by two important aspect related to this algorithm. First, when the number of samples n is smaller than the number of variables p, the empirical covariance matrix S is not invertible. In this cases, for $\lambda > 0$, the G-Lasso estimator performs some regularization so that the estimator $\hat{\Sigma}$ is always invertible regardless of the sample size (Banerjee et al., 2008). Second, even in cases where $n > p$ and S is invertible, the concentration matrix S^{-1} may not be sparse, even if there are conditional independences among the variables in the distribution. G-Lasso algorithm finds a very sparse solution that still explains the data. A larger value of λ corresponds to a sparser solution that fits the data less well; while, a smaller λ corresponds to a solution that fits the data well but is less sparse.

In our comparative study with regard to the penalty term, we evaluate the performance of the method considering four different scalar values that are $\lambda \in \{0.05, 0.1, 0.5, 0.8\}$.

5.2 Shrinkage estimator and empirical Bayes approach for model selection

A new covariance estimator and model selection procedure, that are suitable for data set with a large number of variables but only few observations, has been presented by Schäfer and Strimmer (2005a,b). The basic idea of the covariance estimator is to improve the empirical covariance matrix estimator S by a shrinkage regularization of it. The model selection procedure is based on the empirical Bayes approach suggested by Efron et al. (2001) and it presents a small-sample edge inclusion test. In the following, we first present the shrinkage estimator for the covariance matrix and then the empirical Bayes model selection procedure.

The general problem behind the use of the shrinkage regularization for large-dimensional estimation could be summarized as following. Let $\Psi = (\psi_1, \ldots, \psi_2)$ denote the parameters of the unrestricted high-dimensional model of interest and $\Theta = (\theta_i)$ the matching parameters of a lower dimensional restricted submodel. By fitting each of the two models to the observed data the estimators for the parameters are: $U = \hat{\Psi}$ and $T = \hat{\Theta}$. The two estimators will show different characteristics, in particular U will exhibit a high variance, whereas T could be a biased estimator of the true Ψ. Hence, instead of choosing between one of the two estimators, the linear shrinkage approach suggests to combine both estimators in a weighted average

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

where $\lambda \in [0, 1]$ indicates the shrinkage intensity. Expression (5.6) suggests that the crucial point in this procedure is the selection of the optimal value for the shrinkage parameter and Schäfer and Strimmer consider the theorem develops by Ledoit and Wolf (2003) to obtain a suitable λ. The theorem allows to choose λ that guarantees the minimization of a risk function and, in the specific, Schäfer and Strimmer consider to minimize the MSE of the shrinkage estimator. Hence, we have

$$R(\lambda) = E\left[L(\lambda)\right] = E\left[\sum_{i=1}^{p} (u_i^\star - \psi_i)^2\right]. \tag{5.7}$$

Assuming that the first two moments of the distribution of U and T exist and that U is an unbiased estimator of Ψ, equation (5.7) may be expended as follows

$$\begin{aligned} R(\lambda) &= E\left[L(\lambda)\right] \\ &= \sum_{i=1}^{p} \mathrm{Var}(u_i^\star) + \left[E(u_i^\star) - \psi_i\right]^2 \\ &= \sum_{i=1}^{p} \lambda^2 \mathrm{Var}(t_i) + (1-\lambda)^2 \mathrm{Var}(u_i) + 2\lambda(1-\lambda)\mathrm{Cov}(u_i, t_i) + \left[\lambda E(t_i - u_i) + \mathrm{Bias}(u_i)\right]^2 . \end{aligned} \tag{5.8}$$

Analytically minimizing the function (5.8), with respect to λ, gives the following expression for the optimal value

$$\lambda^\star = \frac{\sum_{i=1}^{p} Var(u_i) - Cov(t_i, u_i)}{\sum_{i=1}^{p} E\left[(t_i - u_i)^2\right]}. \tag{5.9}$$

The estimation of the unrestricted covariance matrix constitutes a special case of the general high-dimensional problem presented above. Schäfer and Strimmer (2005b) translate the application of shrinkage regularization to this specific problem and the weighted shrinkage estimator for the covariance matrix is given by

$$S^\star = \lambda T + (1-\lambda) S, \tag{5.10}$$

where the unconstrained unbiased empirical covariance matrix S replaces the unconstrained estimate U of equation (5.6). In this case, the optimal shrinkage parameter is obtained by minimizing the expected value of

$$\begin{aligned} L(\lambda) &= \|S^\star - \Sigma\|_F^2 \\ &= \sum_{i=1}^{p}\sum_{j=1}^{p} (\lambda t_{ij} + (1-\lambda) s_{ij} - \sigma_{ij})^2, \end{aligned} \tag{5.11}$$

where $\| \cdot \|_F^2$ is the squared Frobenius norm. The minimization of this function with respect to λ depends on the covariance target T. In their paper the authors present different covariance targets $T = (t_{ij})$, but for genomic problems they recommend the one called "diagonal-unequal variance" where

$$t_{ij} = \begin{cases} s_{ii} & \text{if } i = j; \\ 0 & \text{if } i \neq j. \end{cases}$$

With this choice, the optimal shrinkage intensity becomes

$$\hat{\lambda}^{\star} = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij})}{\sum_{i \neq j} s_{ij}^2};$$

(see Schäfer and Strimmer, 2005b, Table 2).
Unlike the G-Lasso, the above shrinkage estimation of Σ does not lead to zero elements in Σ^{-1}, therefore it needs to be supplemented by tests for zero partial correlation coefficients. Schäfer and Strimmer (2005a) suggested a statistical test procedure based on large-scale multiple testing of edges using the assumption that genetic networks are typically sparse (Yeung et al., 2002). Indeed, it is reasonable that in a network only a small fraction of all possible edges will correspond to true edges, whereas for the remaining majority the corresponding true partial correlation coefficients will vanish. Therefore, in the approach of Schäfer and Strimmer (2005a) the distribution of the partial correlations $\rho_{ij \cdot V \setminus \{i,j\}} \equiv \rho$ across edges is taken as the mixture

$$f(\rho) = \eta_0 f_0(\rho; k) + (1 - \eta_0) f_A(\rho).$$

Here, f_0 is the null distribution and is given in Hotelling (1953), $f_A \sim \mathcal{U}(-1, 1)$ is assumed to be the distribution of observed partial correlations, k is the degree of freedom, and η_0 is the (unknown) proportion of missing edges. Fitting this mixture distribution to the observed partial correlation coefficients allows to infer the parameters $\hat{\eta}_0$ and $\hat{k}$. It is then straightforward to compute two-sided p-values for each possible edge in the corresponding network, using the exact null distribution f_0 with $\hat{k}$ as plug-in estimate.
In our comparative study, we use the shrinkage estimator for obtaining the concentration matrix and then we apply the empirical Bayes approach for model selection with the FDR correction.

5.3 PC-algorithm

Differently from the previous methods, the PC-algorithm is an approach for estimating the structure of DAGs rather that undirected graphs. In this thesis, we deal with the version of PC-algorithm proposed by Kalisch and Bühlmann (2007) that is a modification of the previous algorithm presented in Spirtes and Glymour (1991) and Spirtes et al. (2000) and it aims to estimate the skeleton and the equivalence class of a very high-dimensional and sparse DAG. The PC-algorithm starts from a complete undirected graph and deletes successively edges based on conditional independence decisions. This yields an undirected graph which can then be partially directed and further extended to represent

the underlying DAG.
Let us consider a DAG $\mathcal{D} = (V, E)$, with nodes corresponding to the component of a random vector $\mathbf{X} \in \mathbb{R}^p$, and P as the probability distribution generated from $\mathcal{D}$. Hence, there exists a whole equivalent class of DAGs that corresponds to the distribution P, based on the characterization of equivalent class given in Section 3.2.4. A common tool for visualizing equivalence classes of DAGs are *completed partially directed acyclic graphs* (CPDAG). A partially directed acyclic graph (PDAG) is a graph where some edges are directed and some are undirected. In addition, one cannot trace a cycle by following the direction of directed edges and any direction for undirected edges. A PDAG is complete if (i) every directed edge exists also in every DAG belonging to the equivalence class of the DAG and (ii) for every undirected edge $i - j$ there exists a DAG with $i \rightarrow j$ and a DAG with $i \leftarrow j$ in the equivalence class.
The main goal of the PC-algorithm is the estimation of CPDAG that consists of two main parts

1. Estimation of the skeleton (Algorithm 1).
2. Partial orientation of edges (Algorithm 2).

The authors point out that all statistical inference is done in the first part, while the second part is only an application of deterministic rules. Hence, if the first part is done correctly, the second will never fail. In a high-dimensional setting, the CPDAG is harder to recover than the skeleton. Moreover, the interpretation of the CPDAG depends much more on the global correctness of the graph; while, the interpretation of the skeleton depends only on a local region and is thus more reliable. Kalisch and Bühlmann (2007) conclude that if the true underlying probability mechanisms are generated from a DAG, the main goal of PC-algorithm is to find the CPDAG. But if in a high-dimensional setting an approximation of the CPDAG seems hopeless, the use of undirected skeleton could be an interesting alternative to the CPDAG.

First part: finding the skeleton

In the *population version* of the PC-algorithm (Spirtes et al., 2000), presented in Algorithm 1, it is assumed that perfect knowledge about all true conditional independence relations is available. The Algorithm 1 constructs the true skeleton of the DAG and the maximum reached value of ℓ in Algorithm 1 is given by $m_{reach} \in \{q-1, q\}$, where q indicates the maximum number of neighbors. The proof of these results are presented in Spirtes et al. (2000, Theorem 5.1) and Kalisch and Bühlmann (2007, Appendix A.1).
In their work, Kalisch and Bühlmann (2007) modified the population version of PC-algorithm (Algorithm 1) for obtaining the PC-algorithm for finite samples. Furthermore,

Algorithm 1

```
INPUT: Vertex set $V$, Conditional Independence Information
OUTPUT: Estimated skeleton $\mathcal{D}^u$, separation sets $S$ (only needed when directing the skeleton afterwards)
Form the complete undirected graph $\tilde{\mathcal{D}}^u$ on the vertex set $V$.
$\ell = -1$; $\mathcal{D}^u = \tilde{\mathcal{D}}^u$
repeat
  $\ell = \ell + 1$
  repeat
    Select a (new) ordered pair of nodes $i$,$j$ that are adjacent in $\mathcal{D}^u$ such that $|\text{adj}(\mathcal{D}^u, i)\setminus \{j\}\,| \geq \ell$
    repeat
      Choose (new) $\mathbf{k} \subseteq \text{adj}(\mathcal{D}^u, i)\setminus \{j\}$ with $|\mathbf{k}| = \ell$.
      if $i$ and $j$ are conditionally independent given $\mathbf{k}$ then
        Delete edge $i$, $j$
        Denote this new graph by $\mathcal{D}^u$
        Save $\mathbf{k}$ in $S(i, j)$ and $S(j, i)$
      end if
    until edge $i$, $j$ is deleted or all $\mathbf{k} \subseteq \text{adj}(\mathcal{D}^u, i)\setminus \{j\}$ with $|\mathbf{k}| = \ell$ have been chosen
  until all ordered pairs of adjacent variables $i$ and $j$ such that $|\text{adj}(\mathcal{D}^u, i)\setminus \{j\}\,| \geq \ell$ and
  $\mathbf{k} \subseteq \text{adj}(\mathcal{D}^u, i)\setminus \{j\}$ with $|k| = \ell$ have been tested for conditional independence
until for each ordered pair of adjacent nodes $i, j$ : $|\text{adj}(\mathcal{D}^u, i)\setminus \{j\}\,| \geq \ell$.
```

they limit to the case of random variables with a multivariate normal distribution and assume faithful models, i.e. the conditional independence relations correspond to d-separations and vice versa.

For finite samples, it is necessary to estimate conditional independences and, in the Gaussian case, these conditional independences can be inferred from partial correlations (see Section 3.3). For estimating the sample partial correlations, Kalisch and Bühlmann (2007) consider to use recursively the following identity

$$\rho_{i,j|\mathbf{k}} = \frac{\rho_{i,j|\mathbf{k}\setminus h} - \rho_{i,h|\mathbf{k}\setminus h}\, \rho_{j,h|\mathbf{k}\setminus h}}{\sqrt{(1 - \rho^2_{i,h|\mathbf{k}\setminus h})(1 - \rho^2_{j,h|\mathbf{k}\setminus h})}}, \tag{5.12}$$

for some $h \in \mathbf{k}$, with $\mathbf{k} \subseteq V\setminus \{i, j\}$.

For testing whether a partial correlation is zero or not, they apply Fisher's z-transform (Pace and Salvan, 1997, pg. 326)

$$Z(i, j|\mathbf{k}) = \frac{1}{2} \log \left(\frac{1 + \widehat{\rho}_{i,j|\mathbf{k}}}{1 - \widehat{\rho}_{i,j|\mathbf{k}}} \right). \tag{5.13}$$

Using the significance level α, the null hypothesis $H_0(i, j|\mathbf{k}) : \rho_{i,j|\mathbf{k}} = 0$ against the two-sided alternative $H_1(i, j|\mathbf{k}) : \rho_{i,j|\mathbf{k}} \neq 0$ is rejected if $\sqrt{n - |\mathbf{k}| - 3}|Z(i, j|\mathbf{k})| > \Phi^{-1}(1 - \alpha/2)$, where $\Phi(\cdot)$ denotes the cdf of a $N(0, 1)$.

The PC-algorithm presented by Kalisch and Bühlmann (2007) is almost identical to the population version shown in Algorithm 1 with the only difference in the if-statement of line 11 that should be replace by

$$\textbf{if} \sqrt{n - |\mathbf{k}| - 3}|Z(i, j|\mathbf{k})| \le \Phi^{-1}(1 - \alpha/2) \textbf{ then}.$$

In the new PC-algorithm there are two important aspects. First, the algorithm yields a data-dependent value $\hat{m}_{reach,n}$, which is the sample version of m_{reach}. Second, the only tuning parameter is the significance level α.

Second part: extending the skeleton to the equivalence class

The second part of the algorithm turns the skeleton into a CPDAG. It applies simple rules to the results of the first part and the output of Algorithm 2 is a CPDAG. Finally, it is important to underline one of the main question related to PC-algorithm: the problem of consistency. This problem has been treated in Spirtes et al. (2000) and Robins et al. (2003) in the context of causal inference for a class of methods containing the PC-algorithm. They show that assuming only faithfulness achieves pointwise consistency, but not the uniform consistency. For this reason, Kalisch and Bühlmann (2007, Section 3) provide additional assumptions under which the PC-algorithm is uniformly consistent; moreover, they show that the consistency holds even in the case of a high dimensional setting, but with a sparse structure.

Algorithm 2

```
INPUT: Skeleton $\mathcal{D}^u$, separation sets $S$
OUTPUT: CPDAG $\mathcal{D}$
for all pairs of nonadjacent variables $i$, $j$ with common neighbor $k$ do
  if $k \notin S(i,j)$ then
    Replace $i - k - j$ in $\mathcal{D}^u$ by $i \to k \leftarrow j$
  end if
end for
In the resulting PDAG, try to orient as many undirected edges as possible by repeated application of the following three rules:
R1 Orient $j - k$ into $j \to k$ whenever there is an arrow $i \to j$ such that $i$ and $k$ are nonadjacent.
R2 Orient $i - j$ into $i \to j$ whenever there is a chain $i \to k \to j$.
R3 Orient $i - j$ into $i \to j$ whenever there are two chains $i - k \to j$ and $i - l \to j$ such that $k$ and $l$ are nonadjacent.
R4 Orient $i - j$ into $i \to j$ whenever there are two chains $i - k \to l$ and $k \to l \to j$ such that $k$ and $l$ are nonadjacent.
```

In our comparative study, we decide to use $\alpha = 0.05$. To ensure a fair comparison between this method and the others under study (i.e. G-Lasso, Shrinkage, and MLE) we construct the moral graph $\mathcal{D}^u$ of every DAG $\mathcal{D}$ learned by PC-algorithm. Hence, all the performance measures are generated using moral graphs and the partial correlation matrix of the sparse selected model are calculated using the ipf-algorithm (Whittaker, 1990, p. 182).

5.4 Procedures under comparison

For the comparative study with both simulated and real data, finally, we consider the following methodologies:

- G-Lasso with different penalty values.
- Shrinkage estimator and statistical test based on both:
    - empirical Bayes approach;
    - t-test approach (if $n > p$).
- MLE (if $n > p$) and statistical test based on both:
    - empirical Bayes approach;
    - t-test approach.
- PC-algorithm with moralized graphs.

In the two procedures combine with statistical tests, i.e. Shrinkage estimator and MLE, we use the FDR correction at level $\alpha = 0.05$ to correct the multiple testing problem (see Section 3.3.3). The FDR decision rule requires also the specification of η_0, the fraction of true null hypotheses, that in this context is the fraction of true zero partial correlations. In the comparative study with simulated data, we set η_0 equals to the known value of zero partial correlation coefficients with respect to each structure and group. While for the comparative study with real data, we consider η_0 equals to the value of zero partial correlations derived from the data that we use as benchmark transcriptional network of *E.coli*.

Chapter 6

Comparative study with simulated data

6.1 Performance measures for the simulated data

For the simulated data of each structure, obtained using the procedure presented in Chapter 4, we have the "true adjacency matrix" and the "true partial correlation matrix" that we can use as a benchmark. For every method considered for the comparative study, after its application on the simulated data, we obtain the "estimated adjacency matrix" and the "estimated partial correlation matrix". Comparing the true matrices with estimated ones, we constructed the different measures to compare each others the procedures under study.

Measures based on the adjacency matrix

Comparing the true adjacency matrix and the estimated adjacency matrix, we have a table as (6.1). In the specific, the elements of the this table are

	True edges	True no-edges
Identified edges	Tp	Fp
Missing edges	Fn	Tn
	P	N

Table 6.1: Matrix for performance measures.

- True positive (Tp): true edge correctly identified as edge.
- True negative (Tn): true no-edge correctly identified as missing edge.
- False positive (Fp): true no-edge incorrectly identified as edge.

- False negative (Fn): true edge incorrectly identified as missing edge.
- Total number of true edges (P).
- Total number of true no-edges (N).

Related to the above classification of edges, there are several statistical measures to evaluate the performance of different approaches for structural learning of GRNs. For our comparative study, we considered the performance measures listed in the following

- **Precision rate** $Prec = \frac{Tp}{Tp+Fp}$.
- **True Positive rate (sensitivity/recall)** $Tpr = \frac{Tp}{P}$.
- **Accuracy** $Acc = \frac{Tp+Tn}{P+N}$.
- **Error rate** $Err = \frac{Fp+Fn}{P+N}$.
- **False Positive rate** $Fpr = \frac{Fp}{N}$.
- **False Negative rate** $Fnr = \frac{Fn}{P}$.
- **True Negative rate (specificity)** $Tnr = \frac{Tn}{N}$.

During the implementation of the procedures, we obtain a table as (6.1) at every replication of the simulated data; then, for the analysis we compute an average with respect to all replications of every measure.

Moreover, we compare the true adjacency matrix and the estimated adjacency matrix for every structure using a graphical representation of these matrices. The benchmark plots, i.e. plots of the true adjacent matrices, are presented in figure 6.1, where black points indicate true edges or, equivalently, an entry of the adjacency matrix equal to one. For all the methods, we plot the summary adjacency matrix, i.e. the sum of the adjacency matrices computed at every replication. The graphical representation of the summary adjacency matrices gives a general image of the overall classification of edges by each method. In the matrix, each entry is the proportion of times the procedure has found an edge over all the replications. Thus, every entry of the matrix is a number between 0 and 1; this number is represented as a grey intensity with black associated with the value 1 and white with the value 0.

Measures based on the partial correlation matrix

For every structure, we compare the true partial correlation matrix with the estimated partial correlation matrix using the box-plot of the *mean square error* (MSE). This measure evaluates the biasness in the covariance selection and it is defined as

$$\text{MSE} = \frac{1}{p \times (p-1)/2} \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} (\rho_{ij \cdot V \setminus \{ij\}} - \widehat{\rho}_{ij \cdot V \setminus \{ij\}})^2.$$

In order to have a criterion of reference to interpret the results, we consider the G-Lasso with $\lambda = 0.1$ as the reference method. Hence, box-plots around the value 1 indicate that the two approaches estimate similar values of partial correlation coefficients; box-plots over/under the values 1 indicate a best performance of G-Lasso with $\lambda = 0.1$/other considered method in the estimation of partial correlation coefficients.

Moreover, we compare the true partial correlation matrix and the estimated partial correlation matrix for every structure using a graphical representation of these matrices. The benchmark plots, i.e. plots of the true partial correlation matrices, are presented in figure 6.2. For all the methods, we plot the average of partial correlation matrices with respect to all replications. In the plots the meaning of the colors are

- black color indicates value of partial correlation equals to 1;
- white color indicates value of partial correlation equals to 0;
- colors in the scale of grey indicate value of partial correlation between 0 and 1.

Precision-Recall curves

An interesting method to have a cross-comparison among the considered procedures (except for PC-algorithm) is given by the *Precision-Recall curve* (PR). This curve illustrates the quality in the reconstruction of a network based only on comparing true edges and inferred edges, that are obtained from the estimated partial correlation coefficients.

Every considered method, a part from the PC-algorithm, returns an estimation of the partial correlation matrix. From this matrix we can derive the adjacency matrix by ordering the estimated partial correlation coefficients, in absolute value, from the highest value to the lowest one, and then applying a threshold. All the estimated partial correlation coefficients, in absolute value, greater of the threshold correspond to identified edges. Since every estimated partial correlation coefficient is associated to a precision rate and a recall value, varying the threshold from 1 to 0 allows one to compute the Precision-Recall curve.

We have used the PR curves instead of the well known ROC curves since the latter curves are based on sensitivity and 1- specificity, i.e. false negative, and with sparse data the value of false negatives is irrelevant. Hence, the ROC curves would not been suitable measure for the comparison among the procedures in the structural learning of the networks. Moreover, with the PR curves, we obtain an assessment of procedures which does not depend on the threshold, but only on the estimated partial correlation matrices. This is particularly useful for the different approaches that we use for the comparative study. In the case of the G-Lasso, the values of precision and recall are obtained from the elements exactly equal to zero in the estimated partial correlation matrix, that is computed using the concentration matrix estimated by this algorithm. These measures, however, could return bad results because of the large number of partial correlation coefficients that are very close to zero but not exactly equal to zero. So, the PR-curve overcomes the problem associated to the values close to zero and gives a clearer image of the performance of G-Lasso. Concerning MLE and Shrinkage estimator, the computation of the PR curves solves the problem related to the choice of a threshold, in the model selection part, for inferring the partial correlation matrix of the final graph.

For the analysis with simulated data, we remark some aspects related to the use of PR curves.

- We compute the PR curves using the “ROCR” package of R and we consider a vertical average of the PR curves. This means that the curves show an average, respect all the replications, of the precision rate with respect to every recall value.

- The results presented in the precision-recall curves are not comparable with the single values of precision rate and recall that are shown in the tables of all the performance measures. The average values of these performance measures in the tables are marginal, whereas the average values in the curves show the precision rate conditioned to the recall values.

- PR curves are not computed for the PC-algorithm because it does not estimate the partial correlation matrix.

- For the G-Lasso, during the cross-comparison among the procedures we insert only the PR curves associated to $\lambda = 0.1$. The PR curves for all the values of λ are presented in Appendix A.

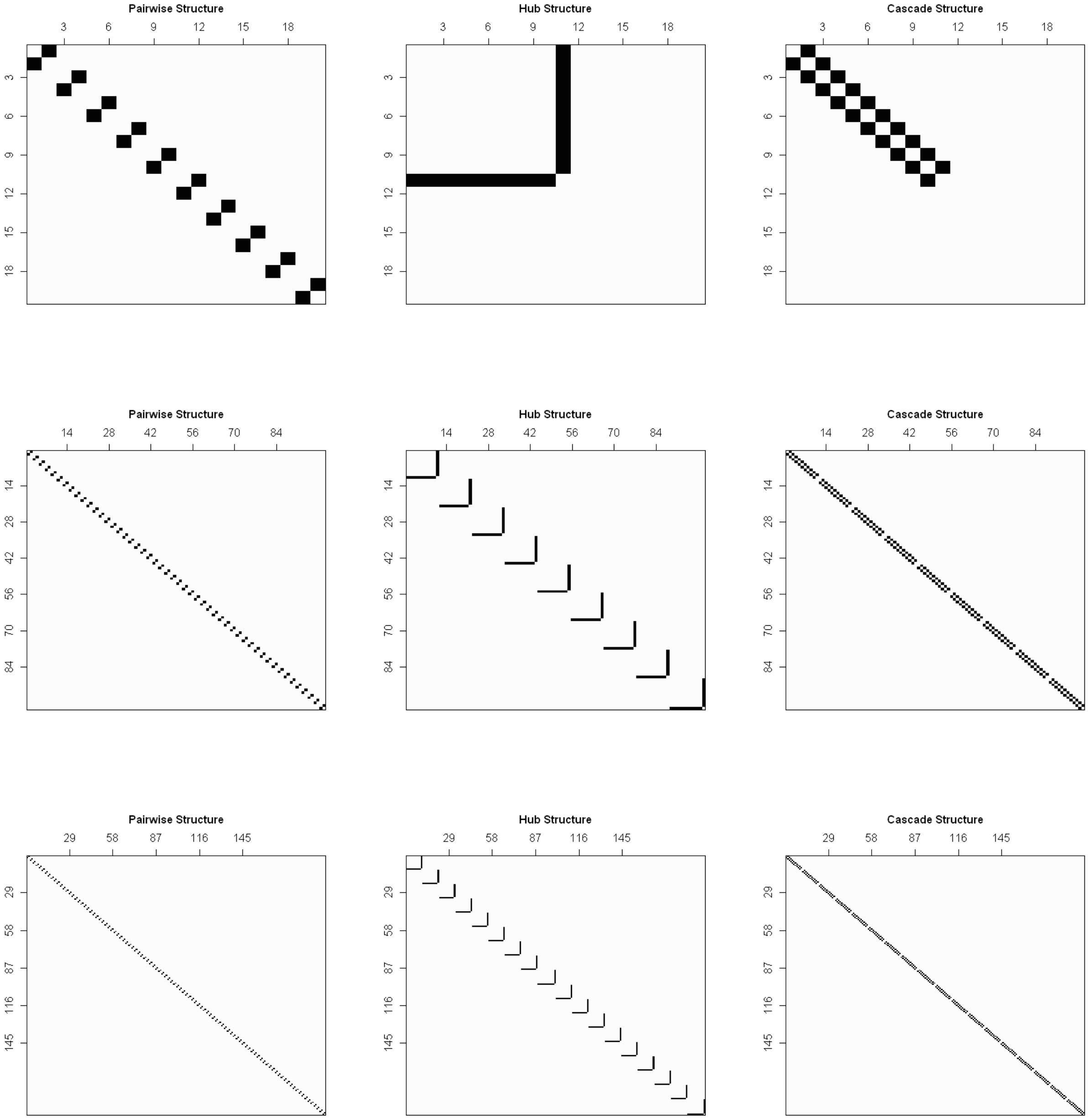


Figure 6.1: Plot of the true adjacency matrices. In the first row, there are the structures with $p = 20$; in the second row, there are the structures with $p = 100$; in the third row, there are the structures with $p = 200$.

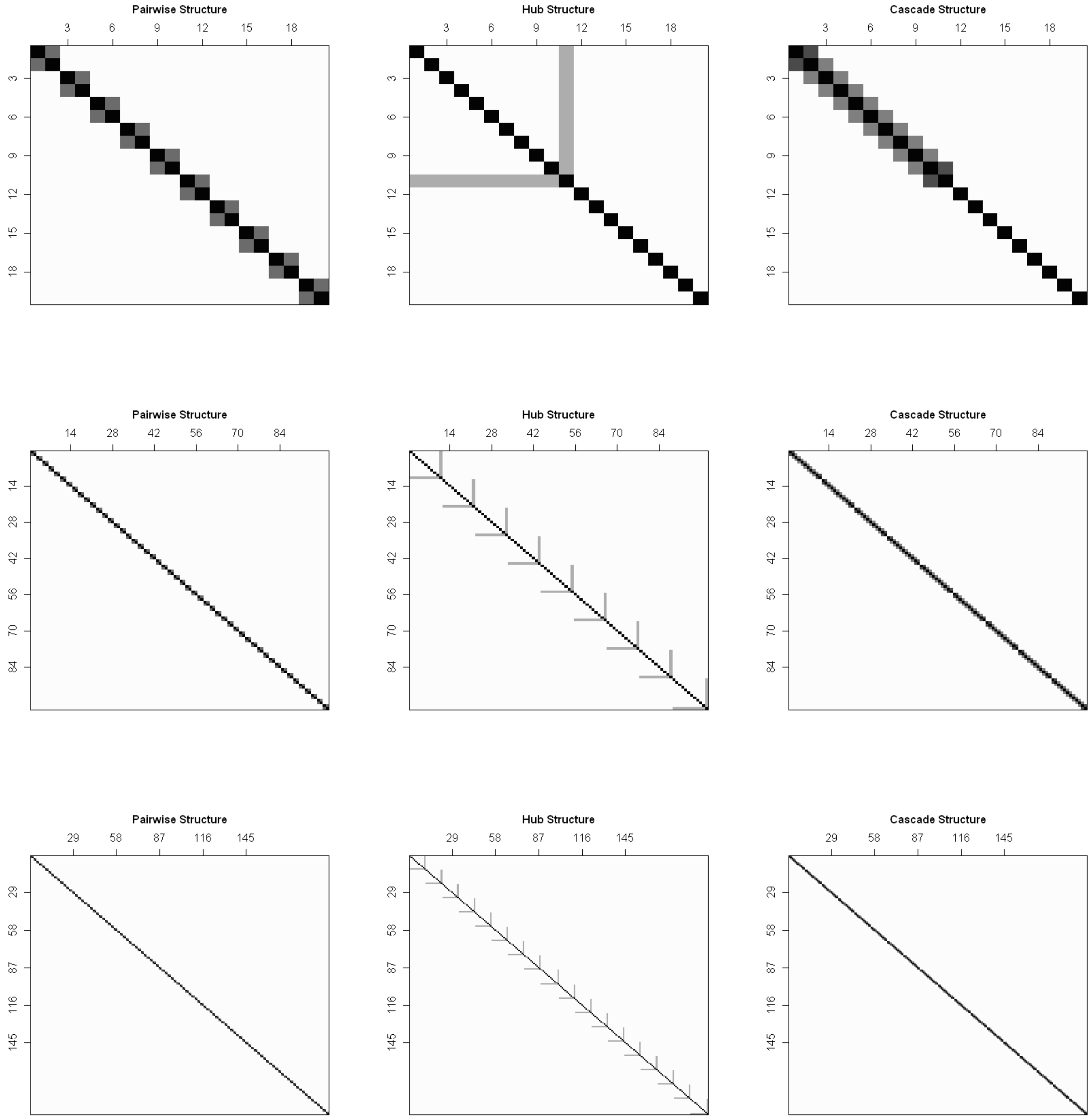


Figure 6.2: Plot of the true partial correlations matrices. In the first row, there are the structures with $p = 20$; in the second row, there are the structures with $p = 100$; in the third row, there are the structures with $p = 200$.

6.2 Results of the analysis with simulated data

6.2.1 G-Lasso

Table 6.2 presents the results of the performance measures. We notice that in general there is not a best penalty term, in particular the different penalty terms influence changes in term of Tpr and Fpr. In the specific, the main findings are

- For the **pairwise structure**, in all the groups, there is a difference in the results based on the value of λ. Among the penalty term $\lambda = \{0.1, 0.5, 0.8\}$, there are remarkable differences between Tpr and Tnr, and consequently in all the others measures. The trend of these differences is similar for all the values of p. Nevertheless, preferring a high value of Tpr respect to low value of Fpr, $\lambda = 0.1$ seems best.

- The **hub structure** and **cascade structure** have a similar behavior in the three groups. With an increment of the penalty term, there is a slight improvement in the performance measures, in particular for the Fpr. Anyway, a too high penalty, i.e. $\lambda = 0.8$, produces a decrement of the value of Tpr, but among the others three values there is not a better λ.

From the above results, we decide to restrict all the remaining analysis to the penalty term $\lambda = 0.1$.
Figure 6.3 represents the summary adjacency matrices estimated by G-Lasso with $\lambda = 0.1$. It is visible that these graphical representations return the same results show in the previous tables. Comparing these plots with the ones that reproduce the adjacency matrices of the true graphs (Fig.6.1), we observe that

- For the **pairwise structure**, the false positives do not show any pattern through replications.

- For the **hub structure** and **cascade structure**, there are systematically the same false positives, that means a pattern through replications.

The figure 6.4 shows the average of estimated partial correlations by G-Lasso with $\lambda = 0.1$. The comparison of these plots with the plots of adjacency matrices underlines the problem anticipated during the description of PR curve. Some values of the estimated partial correlation matrices are close to zero but not set exactly to zero, so they are considered as identified edges in the adjacency matrices and they produce a huge number of false positives. From this figure, in particular we observe that

- For the **pairwise structure**, in all the three groups, the white area means that the values of partial correlations, associated with the uncorrectly identified edges, are almost zero.

- For the **hub structure** and **cascade structure**, with $p = 20$, the white area in the lower right block indicates that the partial correlations of these uncorrectly identified edges are close to zero. In contrast, the grey areas of the upper left block and around the main diagonal, in the groups with $p = 100, 200$, indicate values of partial correlation coefficients similar between correctly identified edges and uncorrectly identified edges. This means that there is not a distinction between true edge and true no-edge.

	$\lambda_1 = 0.05$	$\lambda_2 = 0.1$	$\lambda_3 = 0.5$	$\lambda_4 = 0.8$
Tpr	1.000000	1.000000	0.805200	0.013850
Precision	0.114625	0.241057	1.000000	0.129000
Accuracy	0.591039	0.830679	0.989747	0.948097
Error Rate	0.408961	0.169321	0.010253	0.051903
Fpr	0.431681	0.178728	0.000000	0.000000
Fnr	0.000000	0.000000	0.194800	0.986150
Tnr	0.568319	0.821272	1.000000	1.000000

(a) **Pairwise structure**

	$\lambda_1 = 0.05$	$\lambda_2 = 0.1$	$\lambda_3 = 0.5$	$\lambda_4 = 0.8$
Tpr	1.000000	1.000000	1.000000	0.965050
Precision	0.112966	0.142149	0.181818	0.176464
Accuracy	0.584411	0.679650	0.763158	0.769868
Error Rate	0.415589	0.320350	0.236842	0.230132
Fpr	0.438678	0.338147	0.250000	0.240975
Fnr	0.000000	0.000000	0.000000	0.034950
Tnr	0.561322	0.661853	0.750000	0.759025

(b) **Hub structure**

	$\lambda_1 = 0.05$	$\lambda_2 = 0.1$	$\lambda_3 = 0.5$	$\lambda_4 = 0.8$
Tpr	1.000000	1.000000	1.000000	0.958150
Precision	0.113392	0.142930	0.181818	0.179702
Accuracy	0.586397	0.682058	0.763158	0.771695
Error Rate	0.413603	0.317942	0.236842	0.228305
Fpr	0.436581	0.335606	0.250000	0.238664
Fnr	0.000000	0.000000	0.000000	0.041850
Tnr	0.563419	0.664394	0.750000	0.761336

(c) **Cascade structure**

	$\lambda_1 = 0.05$	$\lambda_2 = 0.1$	$\lambda_3 = 0.5$	$\lambda_4 = 0.8$
Tpr	1.000000	1.000000	0.797600	0.013800
Precision	0.024599	0.056234	1.000000	0.450000
Accuracy	0.599360	0.830257	0.997956	0.990038
Error Rate	0.400640	0.169743	0.002044	0.009962
Fpr	0.404729	0.171476	0.000000	0.000000
Fnr	0.000000	0.000000	0.202400	0.986200
Tnr	0.595271	0.828524	1.000000	1.000000

(d) **Pairwise structure**

	$\lambda_1 = 0.05$	$\lambda_2 = 0.1$	$\lambda_3 = 0.5$	$\lambda_4 = 0.8$
Tpr	1.000000	1.000000	1.000000	0.960549
Precision	0.144036	0.152789	0.179842	0.179961
Accuracy	0.890552	0.897612	0.916162	0.918810
Error Rate	0.109448	0.102388	0.083838	0.081190
Fpr	0.111498	0.104305	0.085409	0.081972
Fnr	0.000000	0.000000	0.000000	0.039451
Tnr	0.888502	0.895695	0.914591	0.918028

(e) **Hub structure**

	$\lambda_1 = 0.05$	$\lambda_2 = 0.1$	$\lambda_3 = 0.5$	$\lambda_4 = 0.8$
Tpr	1.000000	1.000000	1.000000	0.972418
Precision	0.142792	0.152573	0.179842	0.180225
Accuracy	0.889465	0.897535	0.916162	0.918149
Error Rate	0.110535	0.102465	0.083838	0.081851
Fpr	0.112605	0.104384	0.085409	0.082867
Fnr	0.000000	0.000000	0.000000	0.027582
Tnr	0.887395	0.895616	0.914591	0.917133

(f) **Cascade structure**

	$\lambda_1 = 0.05$	$\lambda_2 = 0.1$	$\lambda_3 = 0.5$	$\lambda_4 = 0.8$
Tpr	1.000000	1.000000	0.796900	0.012500
Precision	0.013314	0.029900	1.000000	0.730000
Accuracy	0.627551	0.836906	0.998979	0.995038
Error Rate	0.372449	0.163094	0.001021	0.004962
Fpr	0.374330	0.163918	0.000000	0.000000
Fnr	0.000000	0.000000	0.203100	0.987500
Tnr	0.625670	0.836082	1.000000	1.000000

(g) **Pairwise structure**

	$\lambda_1 = 0.05$	$\lambda_2 = 0.1$	$\lambda_3 = 0.5$	$\lambda_4 = 0.8$
Tpr	1.000000	1.000000	1.000000	0.973462
Precision	0.115904	0.126754	0.179842	0.180010
Accuracy	0.930109	0.936704	0.958291	0.959202
Error Rate	0.069891	0.063296	0.041709	0.040798
Fpr	0.070536	0.063880	0.042094	0.040930
Fnr	0.000000	0.000000	0.000000	0.026538
Tnr	0.929464	0.936120	0.957906	0.959070

(h) **Hub structure**

	$\lambda_1 = 0.05$	$\lambda_2 = 0.1$	$\lambda_3 = 0.5$	$\lambda_4 = 0.8$
Tpr	1.000000	1.000000	1.000000	0.966593
Precision	0.115852	0.130235	0.179842	0.180305
Accuracy	0.930093	0.938746	0.958291	0.959502
Error Rate	0.069907	0.061254	0.041709	0.040498
Fpr	0.070552	0.061820	0.042094	0.040564
Fnr	0.000000	0.000000	0.000000	0.033407
Tnr	0.929448	0.938180	0.957906	0.959436

(i) **Cascade structure**

Table 6.2: G-Lasso algorithm. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$.

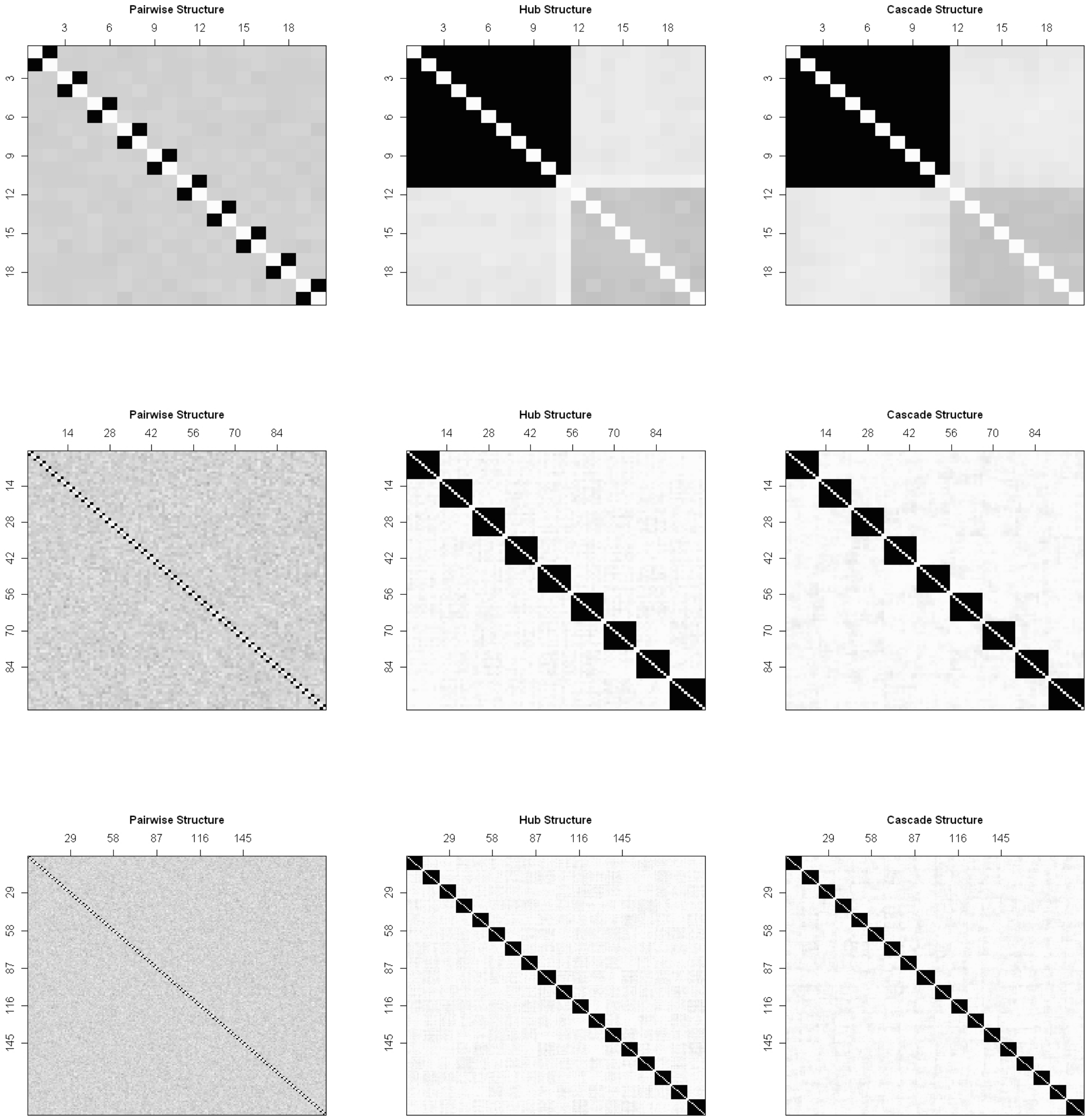


Figure 6.3: Plot of adjacency matrices estimated with G-Lasso ($\lambda = 0.1$). In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$.

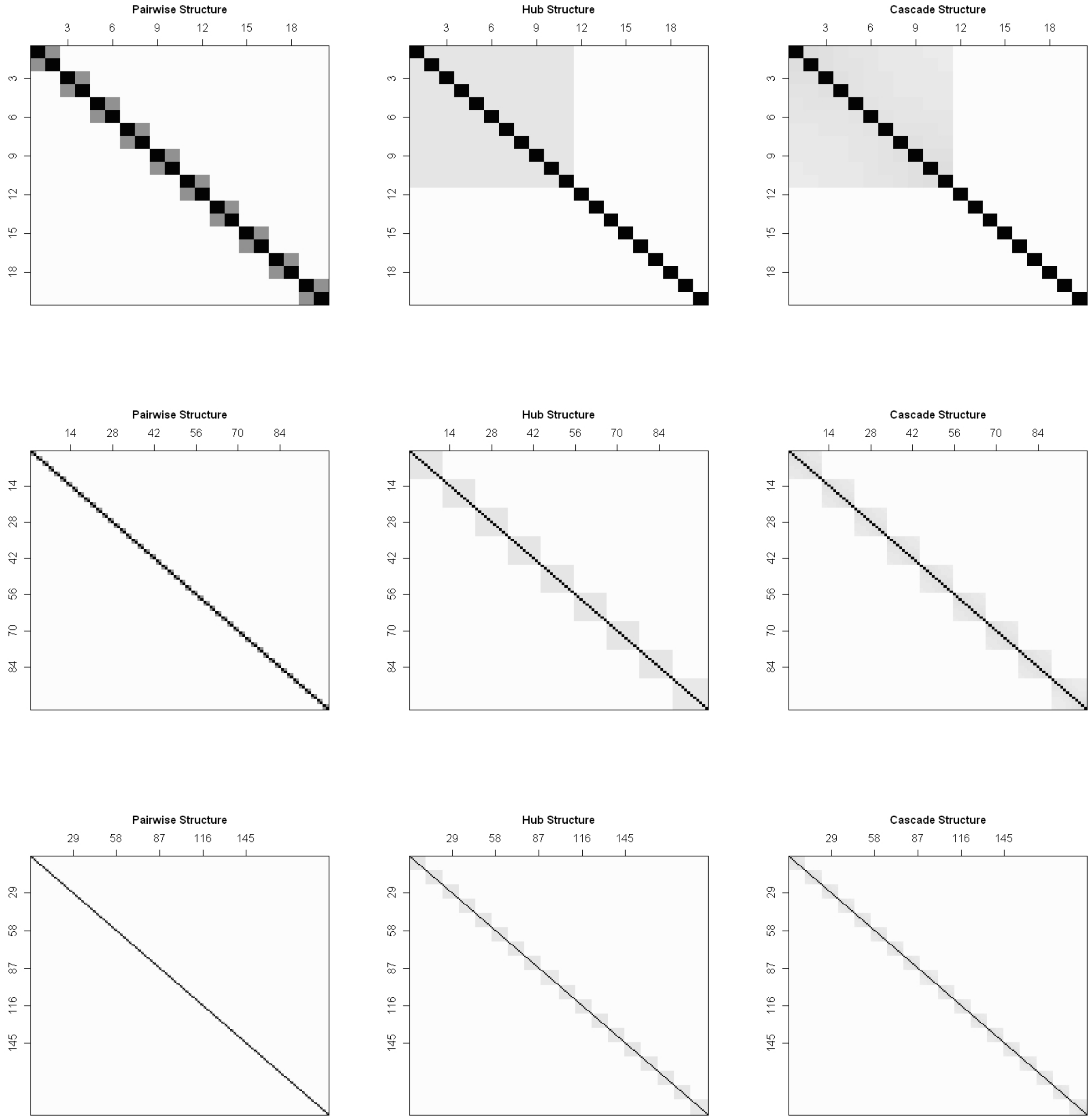


Figure 6.4: Plot of partial correlation matrices estimated with G-Lasso ($\lambda = 0.1$). In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$.

6.2.2 Shrinkage estimator with empirical Bayes approach and t-test approach

Tables 6.3 gives the performance measures computed considering the Shrinkage estimator with either the empirical Bayes approach and the t-test approach for the model selection part. The main and unexpected result is related to the use of the Shrinkage estimator matched with the t-test approach for the model selection that returns almost all measures equal to zero. A plausible explanation could be related to use of false discovery rate as correction method for the multiple hypothesis tests. The FDR imposes a too low theoretical significance level respect to the estimated p-values produced by the t-test approach and, consequently, there is not the identification of the networks. In contrast, the main results for the Shrinkage estimator associated with the empirical Bayes approach are

- For the **pairwise structure**, the results in table 6.3a-c-e show almost a perfect learning of the true structure without any relevant problem of false positives.
- For the **hub structure** and **cascade structure**, with $p = 20$, there is a high value for Fnr and low value of Tpr, in particular for the hub structure. Then, with the increment of p, there is a noticeable improvement of both measures.

In the figure 6.5 are presented the summary adjacency matrices estimated by the Shrinkage estimator and both model selection procedures. Also in this case, it is visible the bad result of the combination between Shrinkage estimator and t-test approach for inferring the edges of the final graph. Hence, we decide to focus only on the analysis of the combination of Shrinkage estimator and empirical Bayes approach. Comparing these plots with the ones that reproduce the adjacency matrices of the true graphs (Fig.6.1), we have that

- For the **pairwise structure**, in all the three groups, there is a perfect learning of the structure.
- For the **hub structure**, with $p = 20$, there are two main concentration of false positives edges. One, that shows a pattern through replications, in the lower right block and one, that does not follow any pattern, in the upper left block. Both cases produce high values of false positives. In addition, in the upper left block the frequency of identified edges, that correspond to true edges, is higher than the remaining part of the block, but this produces the high value of Fnr.
- For the **cascade structure**, with $p = 20$, there is a slight lower right block of false positives. In addition, there is not any pattern in the identification of the true edges

that implies a high value of false negatives.

- For the **hub structure** and **cascade structure**, with $p = 100, 200$, there is a concentration of false positives only in the area close to the true structure.

Figure 6.7 shows the graphical representations of the average of estimated partial correlation matrices by the Shrinkage estimator. From these plots, we notice that there is not a distinction of partial correlation coefficients between true and no-true edges and in particular we observe that

- For the **hub structure** and **cascade structure**, with $p = 20$, the grey area in the lower right block and in the upper left block indicate values of partial correlation coefficients similar between correctly identified edges and uncorrectly identified edges. This means that there is not a distinction between true edge and true no-edge. There is the same problem also with $p = 100, 200$ only in the blocks around the main diagonal.
- For the **pairwise structure**, in contrast, in all the three groups there is a perfect learning of true partial correlation coefficients.

	PAIRWISE	HUB	CASCADE
Tpr	0.990900	0.046700	0.434500
Precision	0.987252	0.013590	0.659329
Accuracy	0.999053	0.926358	0.955618
Error Rate	0.000947	0.073642	0.044382
Fpr	0.000494	0.024772	0.015431
Fnr	0.009100	0.953300	0.565500
Tnr	0.999506	0.975228	0.984569

(a) **Empirical Bayes approach**

	PAIRWISE	HUB	CASCADE
Tpr	0.982150	0.000000	0.000000
Precision	1.000000	0.000000	0.000000
Accuracy	0.999061	0.947345	0.947355
Error Rate	0.000939	0.052655	0.052645
Fpr	0.000000	0.000025	0.000014
Fnr	0.017850	1.000000	1.000000
Tnr	1.000000	0.999975	0.999986

(b) **t-test approach**

	PAIRWISE	HUB	CASCADE
Tpr	0.999800	1.000000	1.000000
Precision	0.999412	0.179831	0.196204
Accuracy	0.999992	0.916156	0.924028
Error Rate	0.000008	0.083844	0.075972
Fpr	0.000006	0.085415	0.077395
Fnr	0.000200	0.000000	0.000000
Tnr	0.999994	0.914585	0.922605

(c) **Empirical Bayes approach**

	PAIRWISE	HUB	CASCADE
Tpr	0.000000	0.000000	0.000000
Precision	0.000000	0.000000	0.000000
Accuracy	0.989899	0.981616	0.981616
Error Rate	0.010101	0.018384	0.018384
Fpr	0.000000	0.000000	0.000000
Fnr	1.000000	1.000000	1.000000
Tnr	1.000000	1.000000	1.000000

(d) **t-test approach**

	PAIRWISE	HUB	CASCADE
Tpr	0.999600	1.000000	1.000000
Precision	0.999703	0.179530	0.179527
Accuracy	0.999996	0.958203	0.958202
Error Rate	0.000004	0.041797	0.041798
Fpr	0.000002	0.042183	0.042184
Fnr	0.000400	0.000000	0.000000
Tnr	0.999998	0.957817	0.957816

(e) **Empirical Bayes approach**

Table 6.3: Shrinkage estimator. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$.

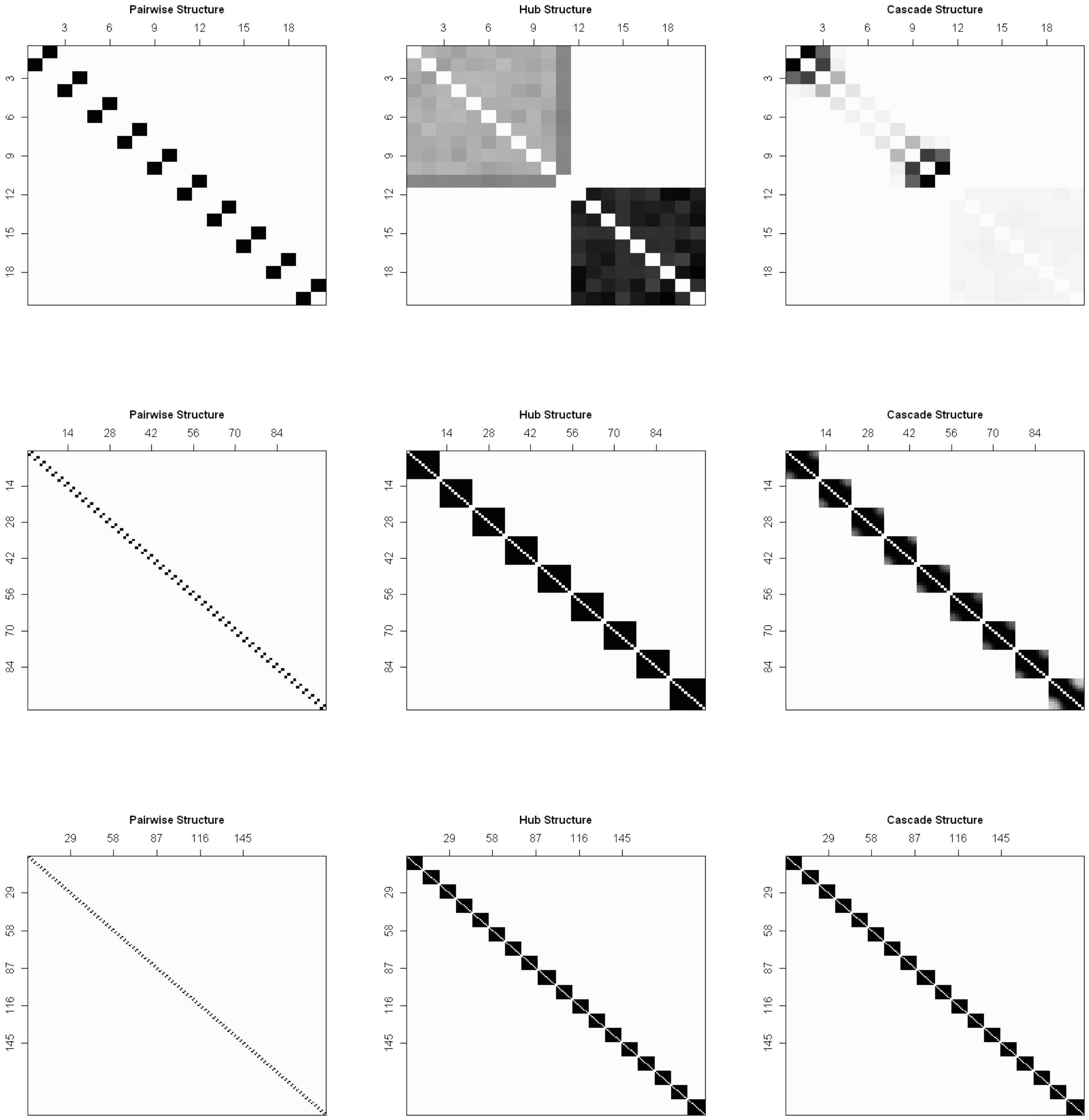


Figure 6.5: Plot of adjacency matrices estimated with Shrinkage and empirical Bayes approach. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$.

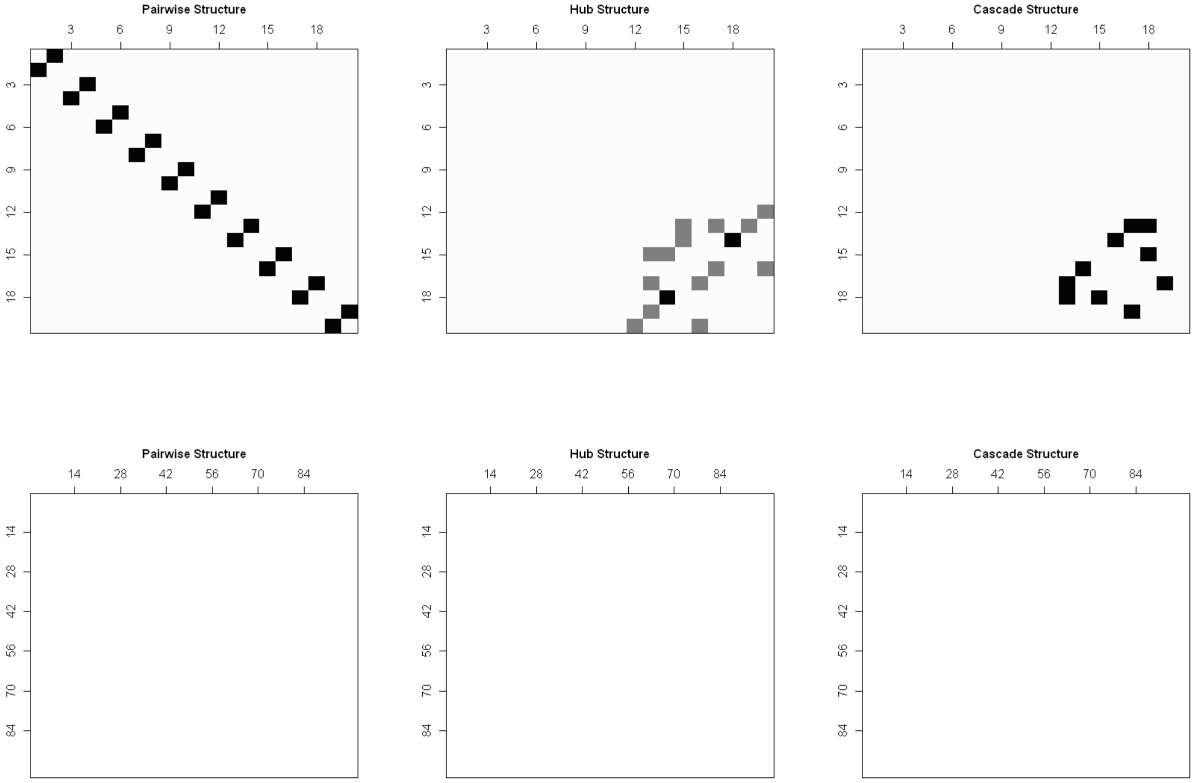


Figure 6.6: Plot of adjacency matrices estimated with Shrinkage and t-test approach. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$.

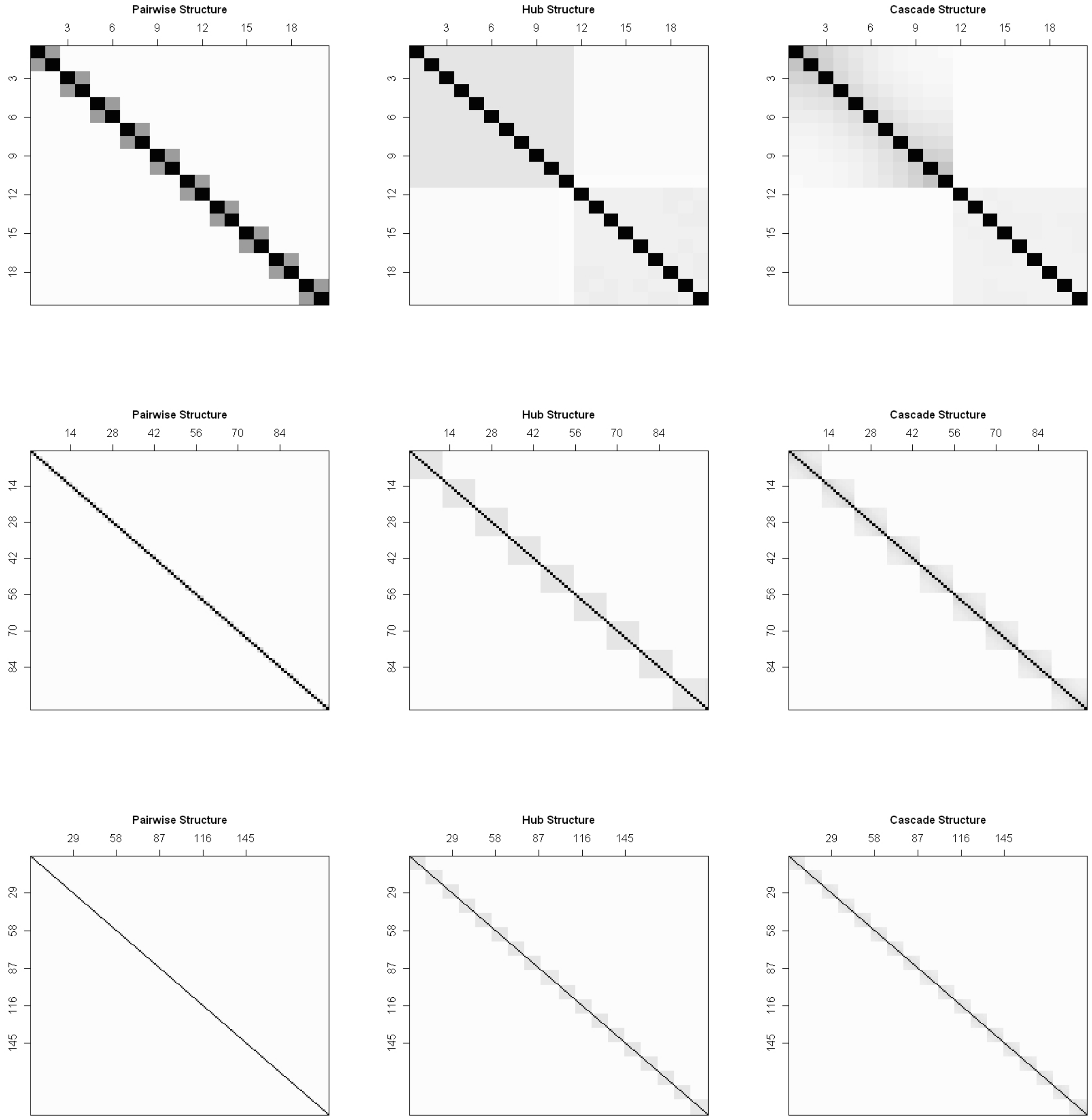


Figure 6.7: Plot of partial correlation matrices estimated with Shrinkage. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$.

6.2.3 Maximum likelihood estimator with the empirical Bayes approach and the t-test approach

Table 6.4 presents the performance measures for the combination of the ML estimator and model selection with either the empirical Bayes approach and the t-test approach. The results do not underline noticeable differences between the two combinations and in both case the results are very similar. In the specific, we observe that

- In the group with $p < n$, the learning process for the networks with the **pairwise structure** and **cascade structure** is almost perfect. In contrast, for the **hub structure**, there is the discovery of only about half of the true edges that suggests an under-estimation of the edges of the true structure.
- When the difference between p and n decreases, in **all the structures** there is a remarkable decrement of the Tpr, but the Fpr is still almost equal to zero.

Figures 6.8 and figure 6.9 represent the summary adjacency matrices estimated by the MLE and model selection with either the empirical Bayes approach and the t-test approach. The figures show that the two combinations of ML estimator with both model selection approaches have very similar and good results. Indeed, comparing these plots with the ones that reproduce the true adjacency matrices (Fig.6.1) the main results are

- For **pairwise structure** and **cascade structure**, there is a perfect match between the true and estimated plots.
- For the **hub structure**, with $p = 100$, even if there is a good match between the plots, the estimation of true edges does not show any pattern.

Figure 6.10 shows the average of estimated partial correlations by the ML estimator. These plots underline that there is a perfect learning of true partial correlation coefficients for all the structures.

	PAIRWISE	HUB	CASCADE
Tpr	0.981750	0.503900	0.955300
Precision	0.985890	0.965431	0.988153
Accuracy	0.998495	0.973358	0.997171
Error Rate	0.001505	0.026642	0.002829
Fpr	0.000575	0.000561	0.000503
Fnr	0.018250	0.496100	0.044700
Tnr	0.999425	0.999439	0.999497

(a) **Empirical Bayes approach**

	PAIRWISE	HUB	CASCADE
Tpr	0.999950	0.527750	0.996450
Precision	0.995680	0.991746	0.995931
Accuracy	0.999745	0.974913	0.999576
Error Rate	0.000255	0.025087	0.000424
Fpr	0.000267	0.000244	0.000250
Fnr	0.000050	0.472250	0.003550
Tnr	0.999733	0.999756	0.999750

(b) **t-test approach**

	PAIRWISE	HUB	CASCADE
Tpr	0.585000	0.013297	0.387582
Precision	0.999012	0.586250	1.000000
Accuracy	0.995802	0.981853	0.988741
Error Rate	0.004198	0.018147	0.011259
Fpr	0.000006	0.000008	0.000000
Fnr	0.415000	0.986703	0.612418
Tnr	0.999994	0.999992	1.000000

(c) **Empirical Bayes approach**

	PAIRWISE	HUB	CASCADE
Tpr	0.584400	0.011868	0.385934
Precision	0.998634	0.650000	0.998679
Accuracy	0.995794	0.981822	0.988701
Error Rate	0.004206	0.018178	0.011299
Fpr	0.000008	0.000012	0.000010
Fnr	0.415600	0.988132	0.614066
Tnr	0.999992	0.999988	0.999990

(d) **t-test approach**

Table 6.4: ML estimator. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$.

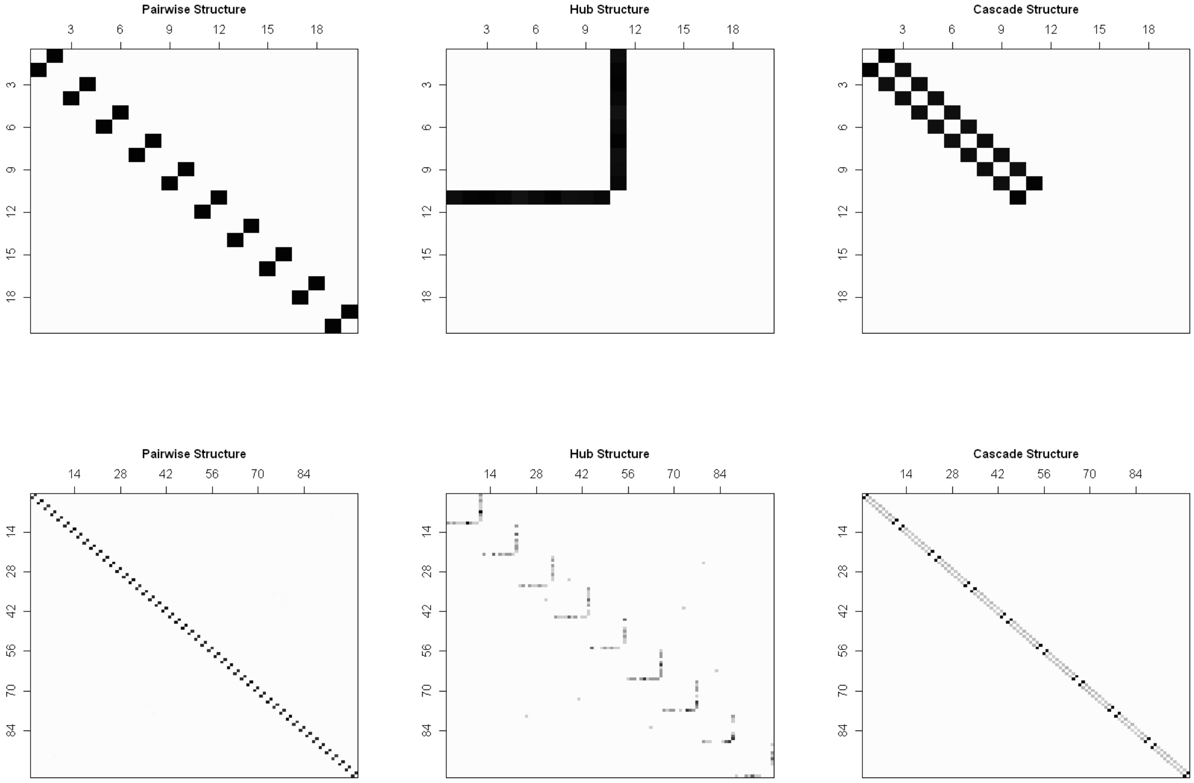


Figure 6.8: Plot of adjacency estimated with ML and empirical Bayes approach. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$.

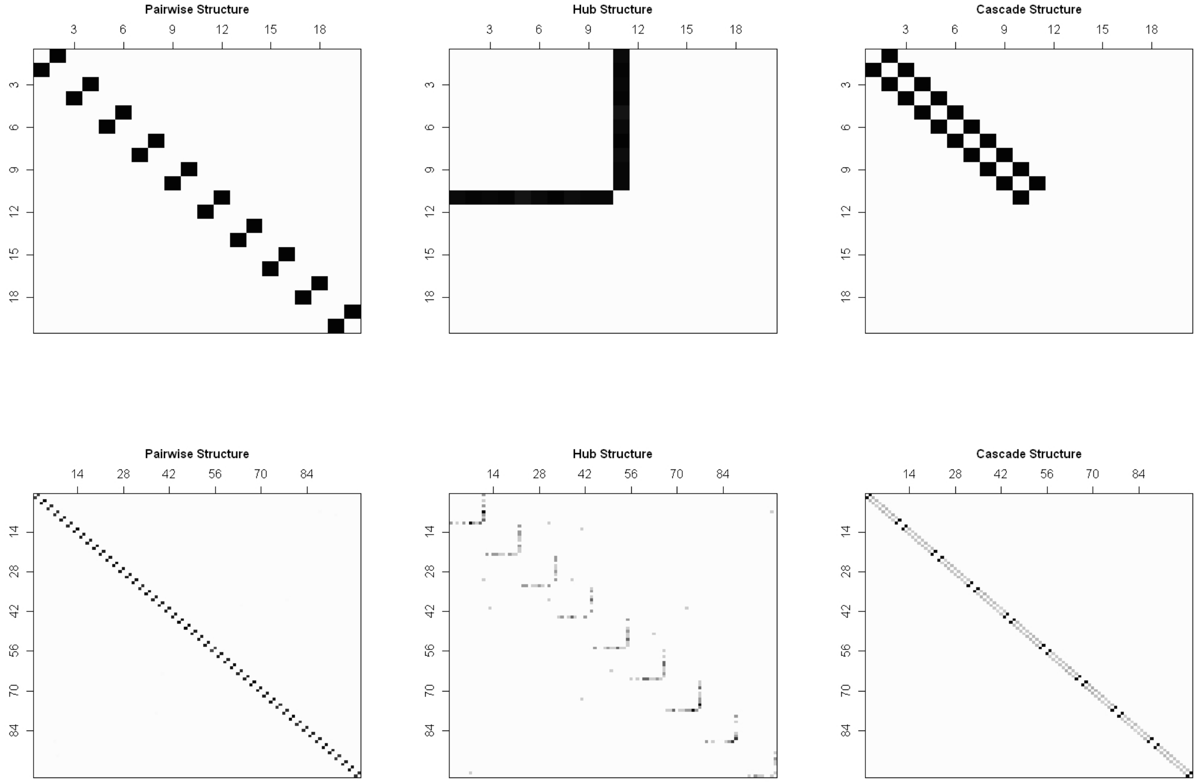


Figure 6.9: Plot of adjacency estimated with ML and t-test approach. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$.

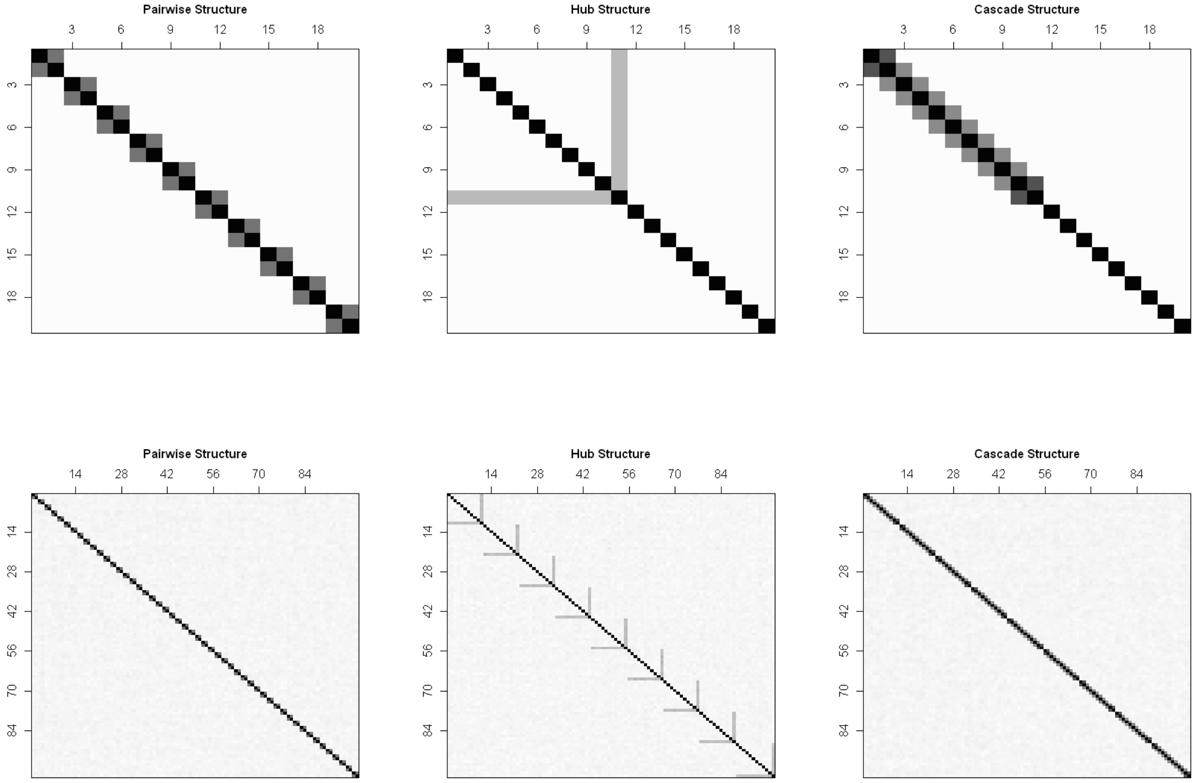


Figure 6.10: Plot of partial correlation matrices estimated with ML. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$.

6.2.4 Pc-algorithm

Table 6.5 refers to the performance measures associated with the PC-algorithm. For all the structures, the algorithm looks quite good in learning the true edges. In the specific, we notice that

- Among the structure, the **hub structure** with $p = 20$ has the higher value of Fpr, but this value decreases with the increment of p with respect to n.
- For the **pairwise structure**, there is a substantive decrement of the precision rate that with $p = 200$ results to be the lowest one.

Figure 6.11 shows the summary adjacency matrices obtained by means of PC-algorithm. Except for the hub structure, the results of the other two structures are very similar to the benchmark plots (Fig.6.1). The main findings are

- For the **pairwise structure**, there is a slight presence of random false positives.
- For the **hub structure**, with $p = 20$, there are two block of false positives. A light block in the lower right part of the plot, and a noticeable block in the upper left part. Anyway, in the latter block there is a distinction between the correctly identified edges, always identified, and the incorrectly identified edges that are learned no in all the replications.
- For the **cascade structure**, with only $p = 20$, there is a concentration of false positives in the lower right block.

Figure 6.12 presents the average of estimated partial correlations for moralized graphs of PC-algorithm. Every matrices have been estimated by means of ipf-algorithm on the adjacency matrix of the moral graph. The results are quite satisfactory, but they should be read with a special care because they are not obtained under the saturated model, as for the other procedures, but under the sparse model selected by the learning procedure. Hence, we consider these plot only to have a complete representation of PC-algorithm with respect to the other procedures.

	PAIRWISE	HUB	CASCADE
Tpr	1.000000	0.968350	1.000000
Precision	0.745960	0.223642	0.485368
Accuracy	0.979753	0.798637	0.943287
Error Rate	0.020247	0.201363	0.056713
Fpr	0.021372	0.210792	0.059864
Fnr	0.000000	0.031650	0.000000
Tnr	0.978628	0.789208	0.940136

(a) **Three structures with** $p = 20$ **and** $n = 150$

	PAIRWISE	HUB	CASCADE
Tpr	1.000000	0.962418	1.000000
Precision	0.392280	0.246132	0.537069
Accuracy	0.984152	0.938998	0.984024
Error Rate	0.015848	0.061002	0.015976
Fpr	0.016010	0.061441	0.016275
Fnr	0.000000	0.037582	0.000000
Tnr	0.983990	0.938559	0.983725

(b) **Three structures with** $p = 100$ **and** $n = 150$

	PAIRWISE	HUB	CASCADE
Tpr	1.000000	0.961538	1.000000
Precision	0.272318	0.309960	0.544648
Accuracy	0.986514	0.976279	0.992233
Error Rate	0.013486	0.023721	0.007767
Fpr	0.013555	0.023585	0.007839
Fnr	0.000000	0.038462	0.000000
Tnr	0.986445	0.976415	0.992161

(c) **Three structures with** $p = 200$ **and** $n = 150$

Table 6.5: PC-algorithm.

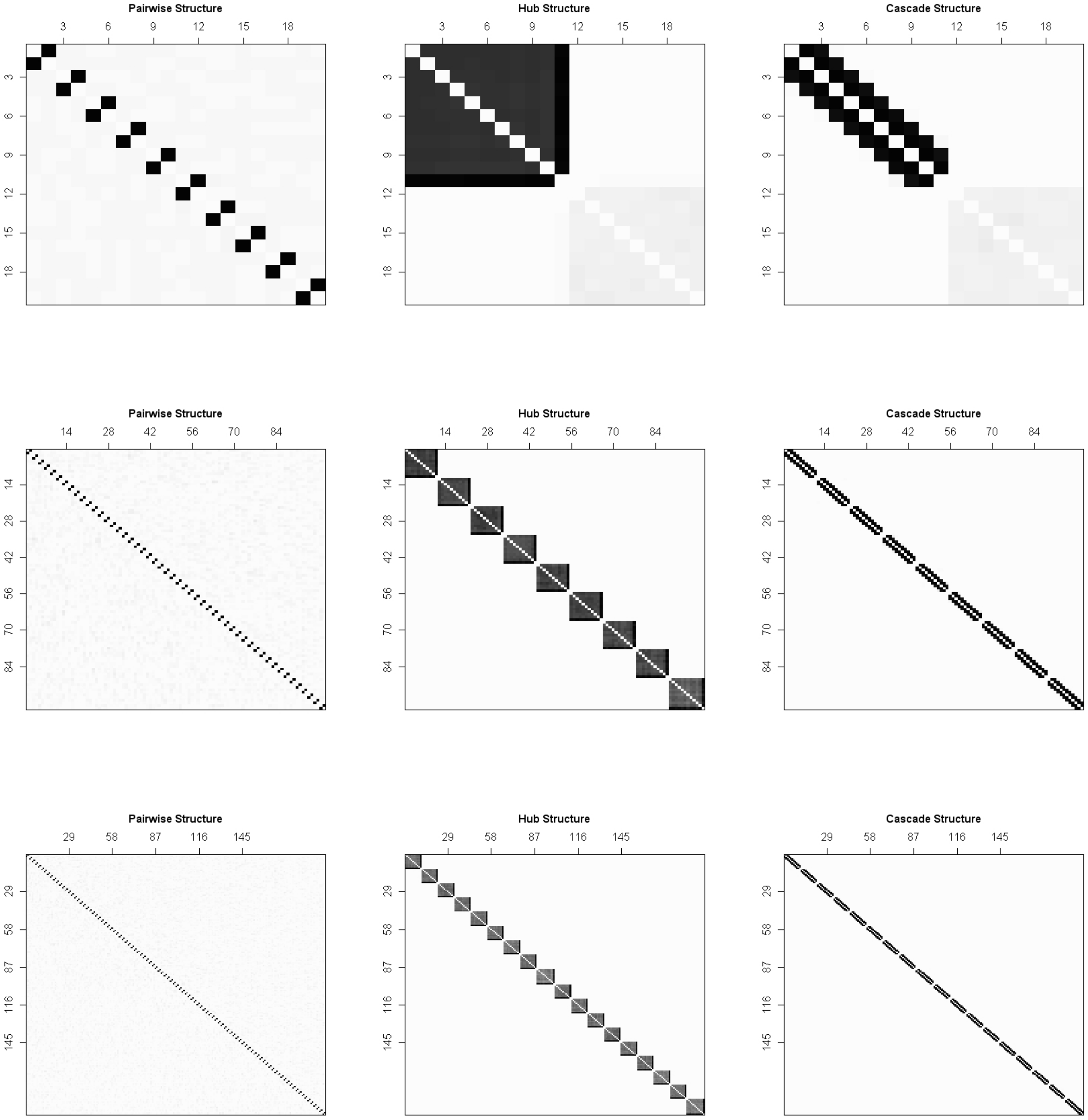
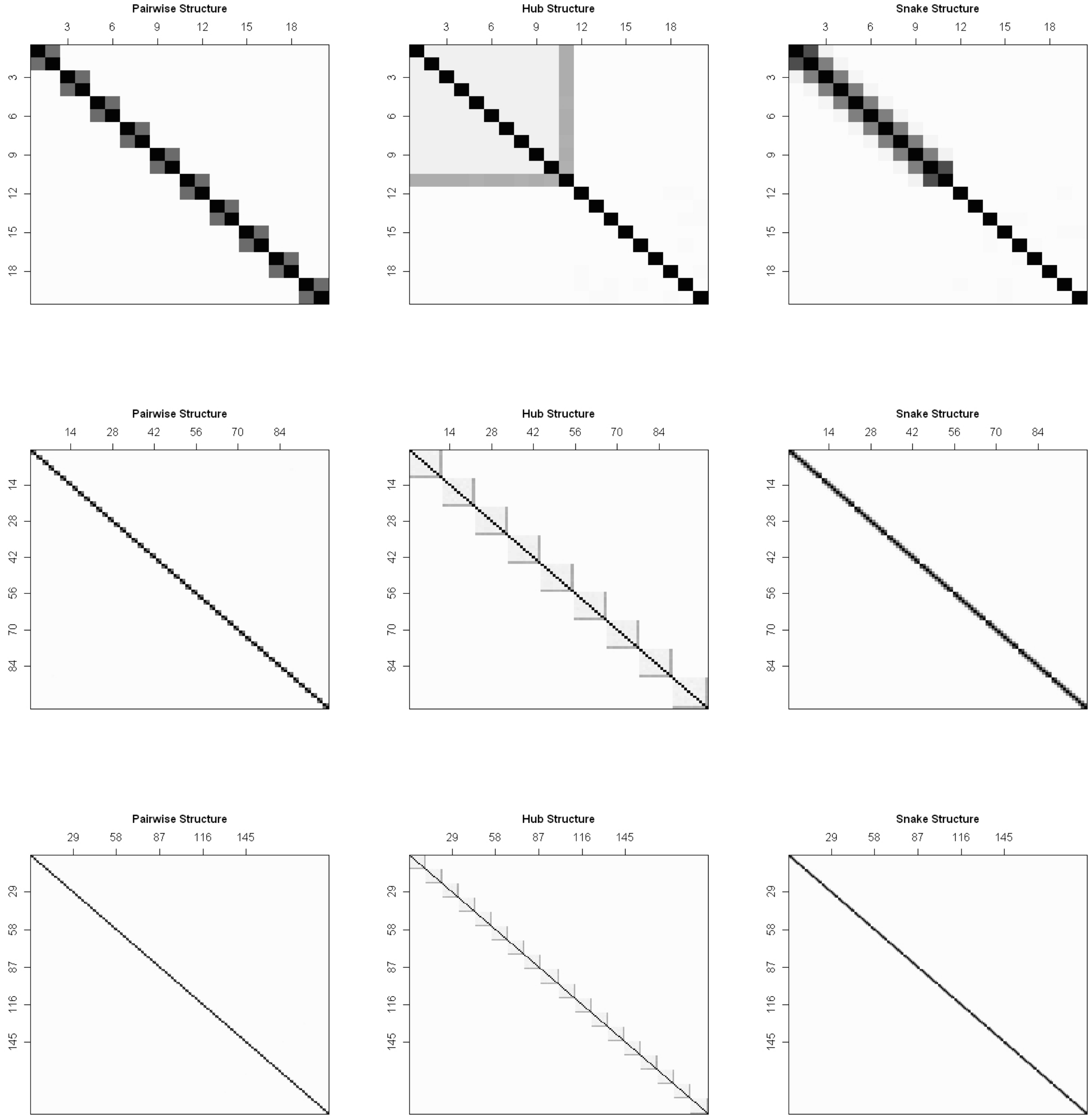


Figure 6.11: Plot of adjacency matrices estimated with PC-algorithm. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$.


Pairwise Structure

Hub Structure

Snake Structure

Figure 6.12: Plot of partial correlation matrices estimated by ipf-algorithm for PC-algorithm. In the first row, there are the structures with $p = 20$ and $n = 150$; in the second row, there are the structures with $p = 100$ and $n = 150$; in the third row, there are the structures with $p = 200$ and $n = 150$.

6.2.5 PR curves and box-plots of MSE

So far, we have studied individually each method for collecting information on their strengths and on their drawbacks for every structure among the three groups. In order to compare the performance of each procedure with the others, we use the PR curves and the box-plots of the MSE.
Regarding the PR curves for each structure, the main results are listed below.

- For the **pairwise structure**, the PR curves are illustrated in figure 6.13. With $p = 20$, we note that the curves of MLE and Shrinkage coincide, and with $p = 100, 200$, the curves of G-Lasso and Shrinkage estimator coincide. In general, all the methods seem good in discovering at least 90% of the edges.

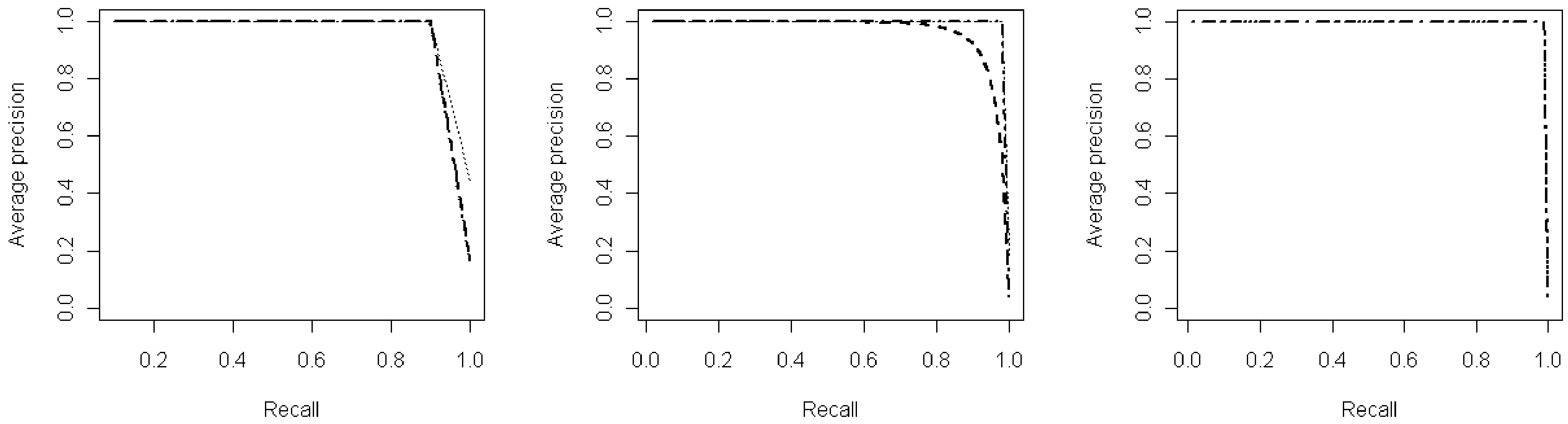
- For the **hub structure**, the PR curves are presented in figure 6.14. With $p = 20$, the plot shows a good performance of MLE compare to the others methods, and between G-Lasso and Shrinkage estimator the latter is slightly better. With $p = 100$, the curve of MLE indicates a decreasing inadequacy of this estimator for the hub structure. Anyway, until the value of recall equals to 0.5, the MLE still out-performs the others methods that show a similar and weak performance in term of precision. In the group with $p = 200$, G-Lasso and Shrinkage estimator still have a similar and quite bad trend.

- For the **cascade structure**, the PR curves are shown in figure 6.15. They indicate MLE, when applicable, as the more suitable method for this structure. The PR curves of G-Lasso and Shrinkage estimator have a similar and constant trend within the three groups. They give a reasonable high value of precision, around 0.6, until the discovery of 90% of edges.

Figures 6.16, 6.17, and 6.18 present the box-plots of MSE of the three structures within the groups with $p = 20$, $p = 100$, and $p = 200$, respectively, where G-Lasso with $\lambda = 0.1$ is the reference method. We notice that the PC-algorithm seems to have a much better behavior than the other methods. However, a direct comparison of the PC-algorithm with other procedures makes no sense because every partial correlation matrix of PC-algorithm has been fitted under the sparse selected model. The good performance of the MSE is therefore a consequence of sparsity. For completeness sake, we decided to include also this case in the box-plots but we are aware that this is of little usefulness. From these box-plots, we see that

- For the **pairwise structure**, there are remarkable differences among the procedures since the ranges of y-axes are quite large (in particular for $p = 20, 100$). The reference method is the more suitable estimator of partial correlation coefficients with respect to the others approaches, in particular with the increment of p. In contrast, with the increment of p, the box-plots of MLE indicate a biasness of this estimator.

- For the **hub structure**, the MLE still shows its biasness in the estimation of partial correlation coefficients with respect to the G-Lasso with $\lambda = 0.1$. With $p = 20$, the Shrinkage estimator performs slightly worse than the other procedures, but with the increment of p its values of MSE are similar to the values of the reference method. In contrast, for the G-Lasso an higher value of penalty seems more suitable in term of parameter estimation.
- For the **cascade structure**, with $p = 20$ Shrinkage estimator, MLE, and G-Lasso, with $\lambda = 0.05$, result to be slightly better estimators compare to the reference method. With the increment of p the methods show similar values of MSE, except for MLE that becomes the worst estimator.

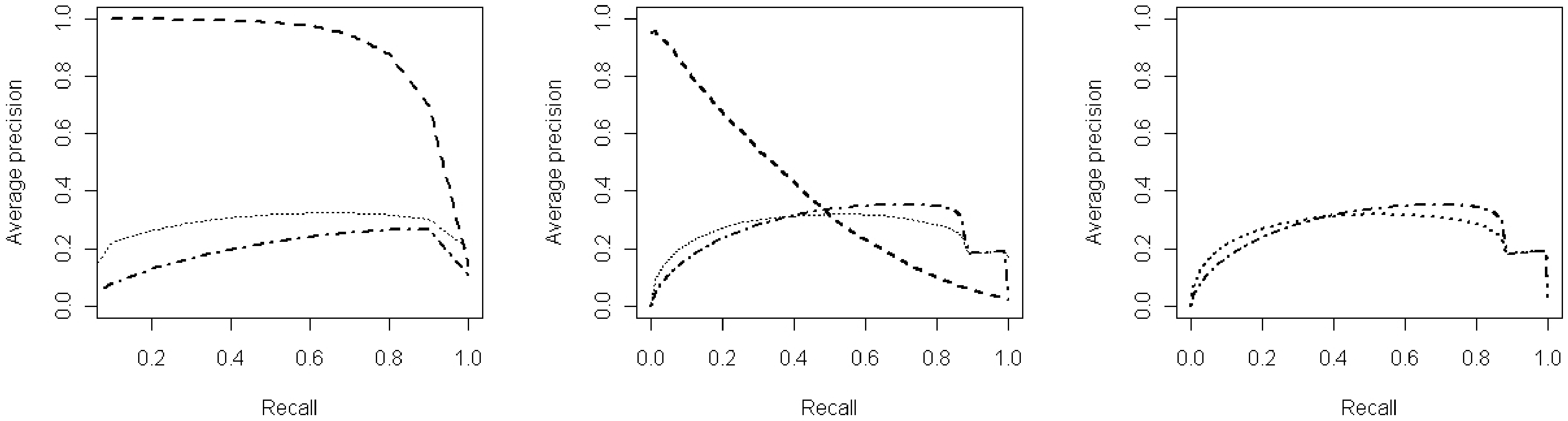


(a) 20 variables

(b) 100 variables

(c) 200 variables

Figure 6.13: Comparison of precision-recall curves for pairwise structure ($n = 150$); dashed for ML estimator, dotted for G-Lasso estimator, and dotdash for Shrinkage estimator.

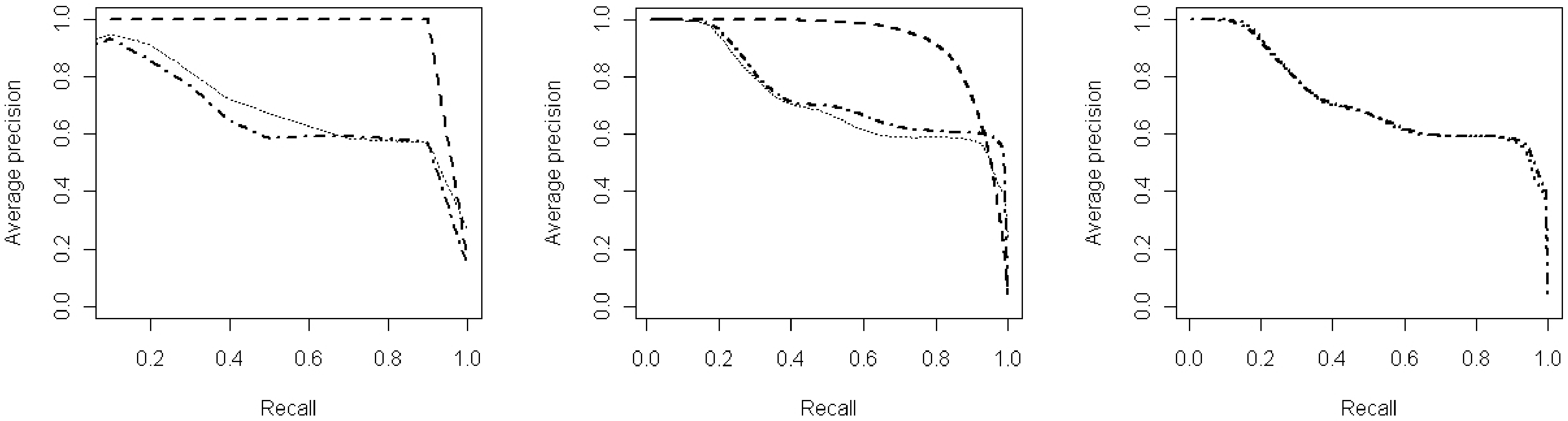


(a) 20 variables

(b) 100 variables

(c) 200 variables

Figure 6.14: Comparison of precision-recall curves for hub structure ($n = 150$); dashed for ML estimator, dotted for G-Lasso estimator, and dotdash for Shrinkage estimator.



(a) 20 variables

(b) 100 variables

(c) 200 variables

Figure 6.15: Comparison of precision-recall curves for cascade structure ($n = 150$); dashed for ML estimator, dotted for G-Lasso estimator, and dotdash for Shrinkage estimator.

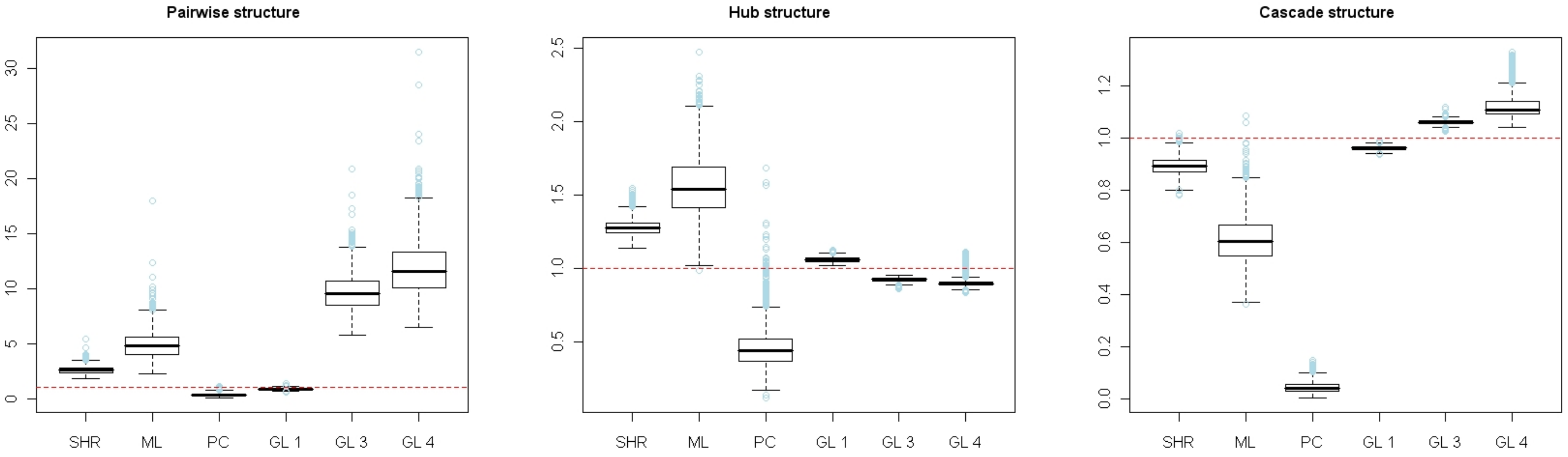


Figure 6.16: Box-plots of MSE for the different methods with $p = 20$ and $n = 150$ (SHR: shrinkage, ML: maximum likelihood, PC: PC-algorithm, GL 1: G-Lasso with $\lambda = 0.05$, GL 3: G-Lasso with $\lambda = 0.5$, GL 4: G-Lasso with $\lambda = 0.8$).

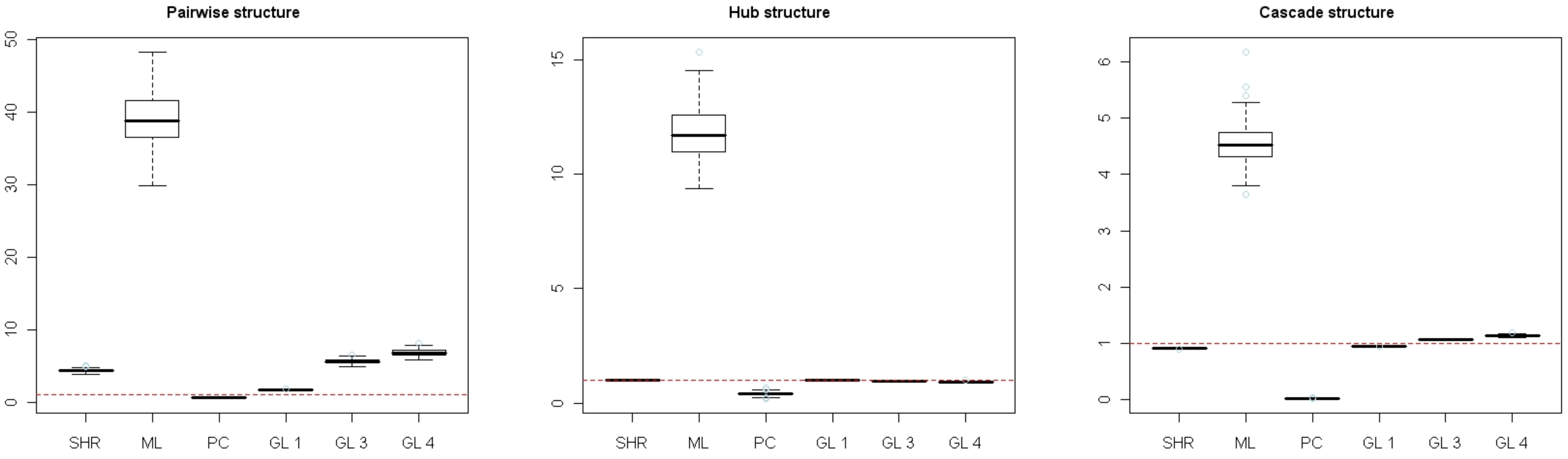


Figure 6.17: Box-plot of MSE for the different methods with $p = 100$ and $n = 150$ (SHR: shrinkage, ML: maximum likelihood, PC: PC-algorithm, GL 1: G-Lasso with $\lambda = 0.05$, GL 3: G-Lasso with $\lambda = 0.5$, GL 4: G-Lasso with $\lambda = 0.8$).

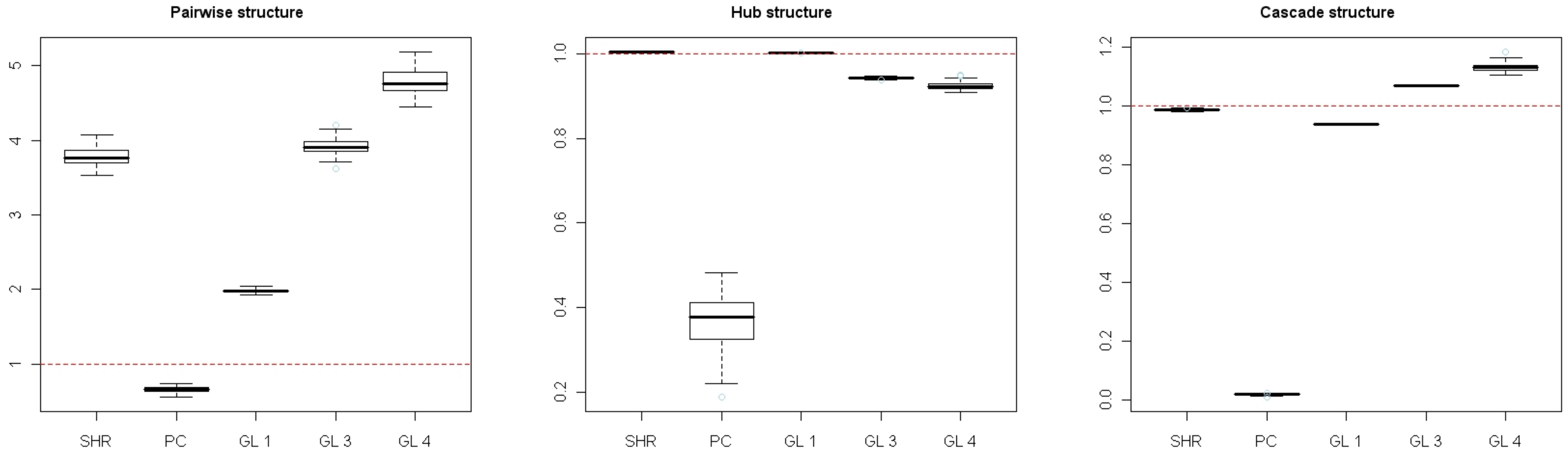


Figure 6.18: Box-plots of MSE for the different methods with $p = 200$ and $n = 150$ (SHR: shrinkage, PC: PC-algorithm, GL 1: G-Lasso with $\lambda = 0.05$, GL 3: G-Lasso with $\lambda = 0.5$, GL 4: G-Lasso with $\lambda = 0.8$).

Chapter 7

Comparative study with real data

7.1 *Escherichia coli* data set

For a comparative study with real data, we consider the gene regulatory network of *Escherichia coli* (*E. coli*): a bacterium that commonly lives in the lower intestine of warm-blooded organisms. We use the *E. coli* transcriptional network since it is the most complete experimentally characterized network of a single cell. The RegulonDB (Gama-Castro et al., 2008) is a specialized database that collects the available experimental data on regulatory interactions between transcription factor (TF) and their target genes (TG) in *E. coli*. However, the information in this database is still far from complete and, as reported in the latest release of RegulonDB (Gama-Castro et al., 2008), there is currently knowledge on transcriptional regulation for only about one third of the genes.

We consider the microarray data contained in *EcoliOxygen* data file available in the R package "qpgraph", that refers to the paper of Castelo and Roverato (2009). The microarray data are based on $n = 43$ experiments of various mutants under oxygen deprivation presented by Covert et al. (2004). The mutants were designed to monitor the response from *E. coli* during an oxygen shift in order to target the *a priori* most relevant part of the transcriptional network. The measurement has been done by using six strains with knockouts of the following key transcriptional regulators in the oxygen response: $\Delta arcA$, $\Delta appY$, Δfnr, $\Delta oxyR$, $\Delta soxS$, and the double knockout $\Delta arcA \Delta fnr$. The *EcoliOxygen* data file is formed by two object: one contains the *E. coli* transcriptional network from RegulonDB and one contains the expression profiles of $p = 4205$ genes under the $n = 43$ experiments of Covert et al. (2004), downloaded from the Gene Expression Omnibus (Barrett et al., 2007) with accession GDS680. These two data sets contain a subset of the original data and they are obtained through the filtering steps described in Castelo and Roverato (2009) to the original data set.

In order to have a gold-standard network to assess the performance of different methods in the comparative study, we filtered the expression profile data in *EcoliOxygen* considering only those genes forming part in RegulonDB of the regulatory modules of the five knocked-out transcription factors (Castelo and Roverato, 2009). In this way, the network of the expression profile data is restricted to $p = 378$ genes. From RegulonDB, we learn that the these 378 genes are involved only in 681 interactions out of 71253 interactions (complete network); hence, for simplicity, we have decided to restrict the comparative study to the 100 genes that have the largest variability measured by the interquartile range. Therefore, the final *E. coli* data set used for the comparative study has $p = 100$ and $n = 43$.

7.2 Performance measure for the *E. coli* data

For the *E. coli* data, using the functions implemented in the reference manual of the R package "qp-graph", we derive from RegulonDB the "true adjacency matrix" that we can use as the benchmark. For every method considered for the comparative study, after its implementation using the *E. coli* data, we obtain the "estimated adjacency matrix" and the "estimated partial correlation matrix". Comparing the true matrices to the estimated ones, we constructed the different measures to compare procedures under study.
It is important to remark two aspects related to the benchmark network. As anticipated in Section 7.1, the data obtained from RegulonDB for the transcriptional network of *E. coli* are not complete. Moreover, this database collects information on regulatory interactions under several experimental conditions and, sometimes, different experimental conditions are measured at the same time. Consequently, either of these aspects can influence negatively the performance measurements.

Measures based on adjacency matrix

Comparing the true adjacency matrix and the estimated adjacency matrix, we obtain a table as the one presented in Section 6.1 from which we can derive the same statistic measures presented in that section.
Moreover, we compare the true adjacency matrix and the estimated adjacency matrix of real data using a graphical representation of these matrices. The benchmark plot, i.e. plot of the true adjacent matrix, is presented in figure 7.1, where black points indicate true edges, i.e. an entry equals to one in the adjacency matrix. For all the methods, we plot the adjacency matrices, to give a general image of the identified/missing edges by each method, where black points indicate identified edges.

Precision-Recall curves

Also for the analysis with real data, we use the PR curves (Section 6.2) in order to have a cross-comparison among the considered procedures (except for PC-algorithm). With this data set, we remark that

- The PR curves are computed using the "ROCR" package of R.
- The PR curve is not compute for the PC-algorithm because it does not estimate the partial correlation matrix.
- Figure 7.2 shows PR curves of the G-Lasso for all the values of λ. Plots 7.2b-c are two zoom in on the part that returns more information on the behavior of G-Lasso with different values of λ. For the cross-comparison among the procedures, we have decided to insert only the PR curves of the G-Lasso associated with $\lambda = 0.1$.

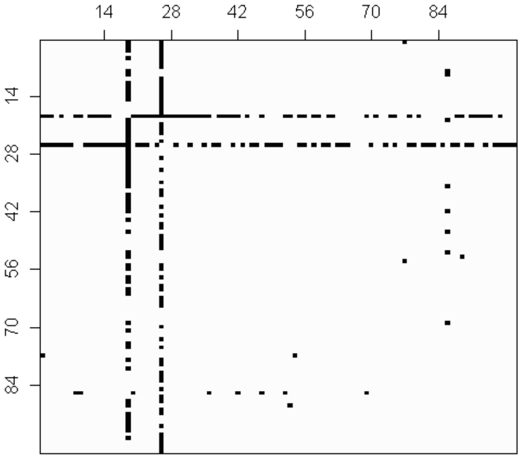
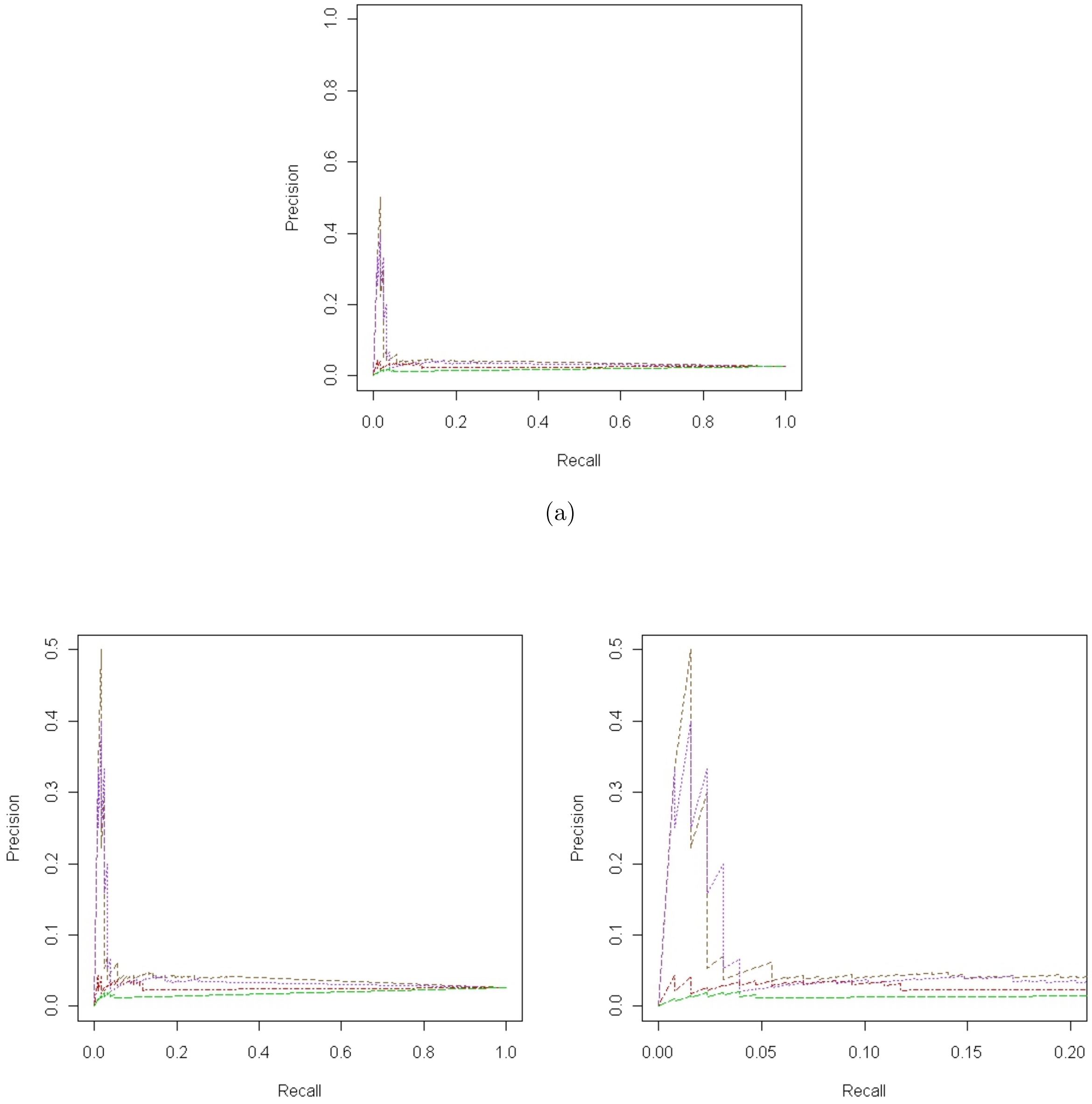


Figure 7.1: Plot of the benchmark adjacency matrix for *E. coli* data.



(a)

(b)

(c)

Figure 7.2: Precision-recall curves for G-Lasso with *E. coli* data: dashed for $\lambda = 0.05$, dotted for $\lambda = 0.1$, dotdash for $\lambda = 0.5$, and longdash for $\lambda = 0.8$. (a) Complete representation of PR curves. (b)-(c) Zoom in on PR curves.

7.3 Results of the analysis with *E. coli* data

Table 7.1 presents the performance measures for the compared methods with the *E. coli* data; the methods are: G-Lasso (Tab. 7.1a), Shrinkage estimator with empirical Bayes approach for model selection (Tab. 7.1b), and PC-algorithm (Tab. 7.1c). From this table, we notice that all approaches perform poorly and the slightly better one, in term of both Tpr and Tnr, is the G-Lasso. In details the main findings are

- For the **G-Lasso**, with increasing value of λ there is visible decrease of the Tpr and, in contrast, an increment of Tnr; consequently, there is an improvement or a worsening of the other performance measures. However, regarding the penalty term, a compromise seems be achieved with $\lambda = 0.1$
- For the **Shrinkage estimator with empirical Bayes approach**, we see a very good value of Tnr. This high value influences precision rate which is weak but higher than for the other methods. In contrast, the Tpr is quite bad.
- For the **PC-algorithm**, the performance is are very similar to the Shrinkage with empirical Bayesian approach, with only a noticeable worsening in the result of precision rate.

Figure 7.3 shows the plots of the adjacency matrices computed by the G-Lasso with $\lambda = 0.1$ (a), the Shrinkage estimator with empirical Bayes approach (b), and PC-algorithm (c). The comparison of these graphical representations of the adjacency matrices with the true adjacency matrix (Fig. 7.1) suggests that all the methods do not at all learn the real structure of the network. From the plot of the true adjacency matrix we see that the network is essentially formed by two hub structures, with a high motif size, and few pairwise structures. This particular network structure could be the explanation of the poor performance of the three methods. Indeed, from the simulation study we have noticed that the hub structure was the more complicated structure to be learned.

Figure 7.4 presents PR curves for G-Lasso with $\lambda = 0.1$ and Shrinkage estimator. Plots 7.4b-c are two zoom in on the part of these curves that returns more information on their behavior. From the curves, we see that at the very beginning the curves of both methods are similar, with only a weak better performance of G-Lasso; then the curve of Shrinkage indicates a better performance of this estimator. Both approaches have very low value of precision and its maximum value is around 0.4 with a recall of about 0.03.

	λ=0.05	λ=0.1	λ=0.5	λ=0.8
Tpr	0.328125	0.257812	0.125000	0.046875
Precision	0.041217	0.035408	0.023495	0.012448
Accuracy	0.785253	0.799192	0.843030	0.879192
Error Rate	0.214747	0.200808	0.156970	0.120808
Fpr	0.202613	0.186437	0.137910	0.098714
Fnr	0.671875	0.742188	0.875000	0.953125
Tnr	0.797387	0.813563	0.862090	0.901286

(a)

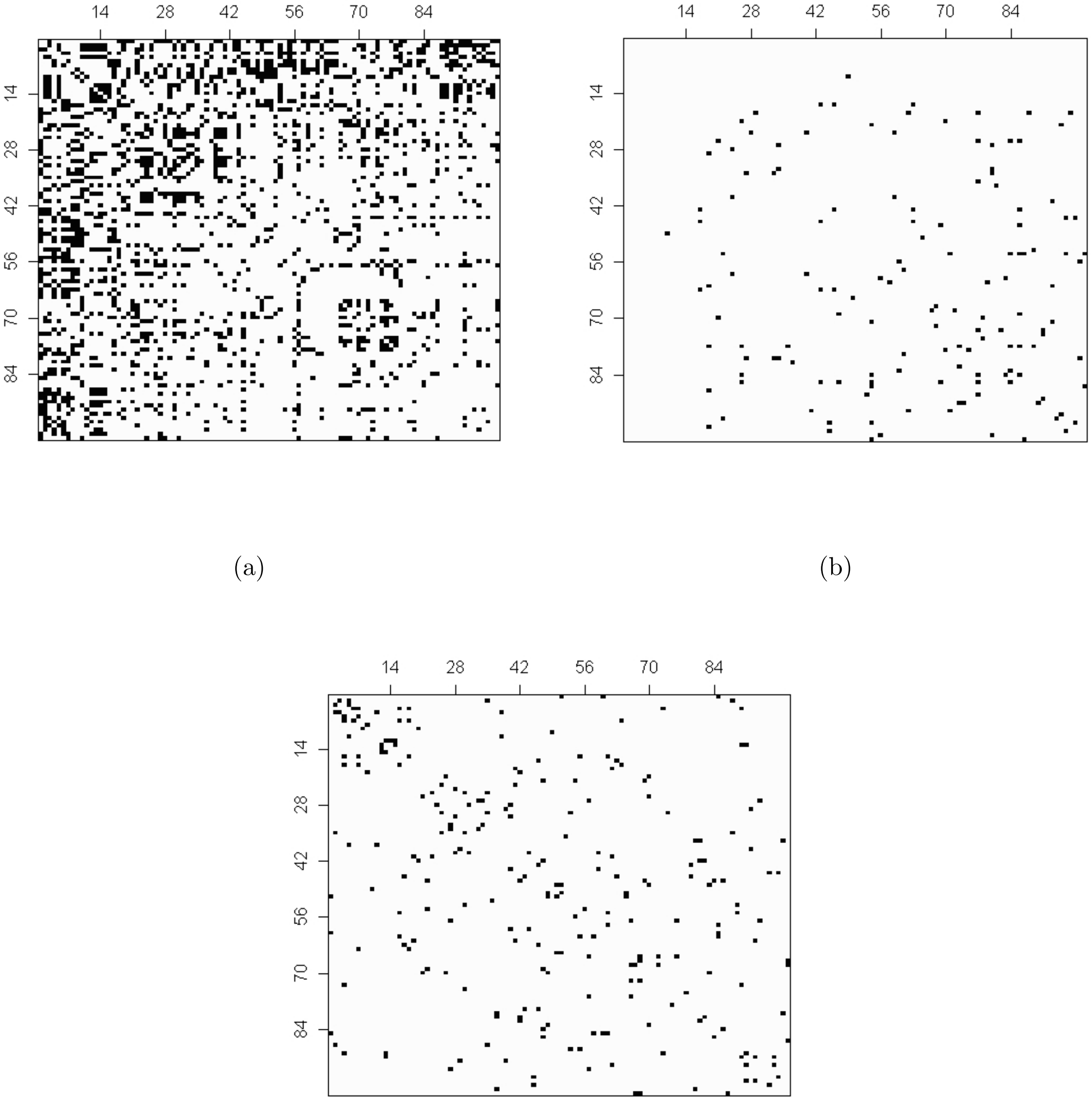
Tpr	0.062500
Precision	0.114286
Accuracy	0.963232
Error Rate	0.036768
Fpr	0.012858
Fnr	0.937500
Tnr	0.987142

(b)

Tpr	0.046875
Precision	0.048780
Accuracy	0.951717
Error Rate	0.048283
Fpr	0.024264
Fnr	0.953125
Tnr	0.975736

(c)

Table 7.1: Performance measures for *E. coli* data. (a) G-Lasso algorithm, (b) Shrinkage estimator with empirical Bayes approach, and (c) PC-algorithm.

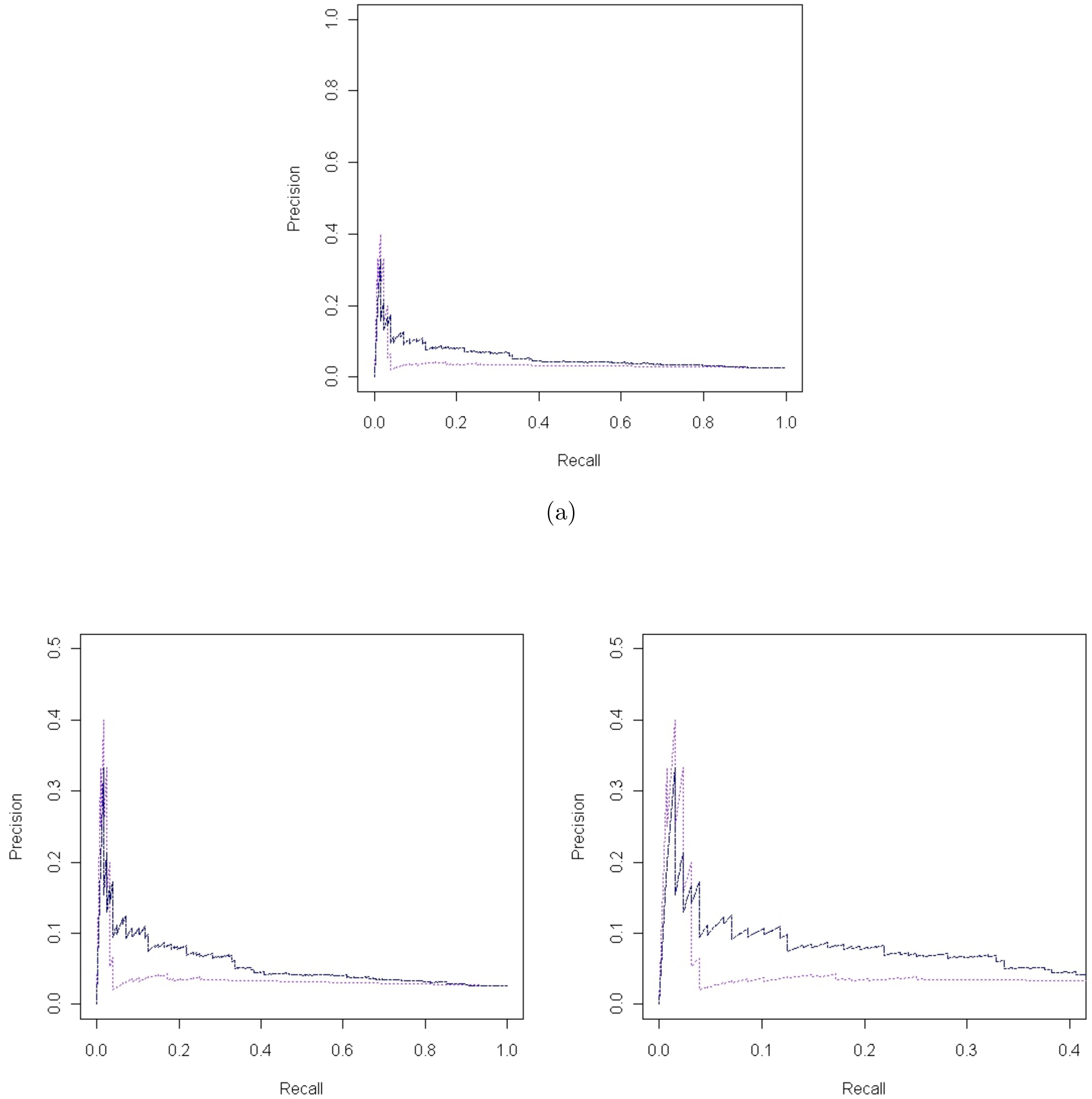


(a)

(b)

(c)

Figure 7.3: Plot of adjacency matrices for *E. coli* data. (a) G-Lasso ($\lambda = 0.1$), (b) Shrinkage estimator with empirical Bayes approach, and (c) PC-algorithm.



(a)

(b)

(c)

Figure 7.4: Comparison of precision-recall curves for *E. coli* data: dotted for G-Lasso ($\lambda = 0.1$) and twodash for Shrinkage estimator. (a) Complete representation of PR curves. (b)-(c) Zoom in on PR curves.

Chapter 8

Conclusion

8.1 Summing-up

Gaussian graphical models have become a common tool for structural learning of gene regulatory networks by means of microarray data. However, their application to genetic data is quite challenging, as the number of genes p is usually much larger than the number of available samples n, so that classical GGM theory is not applicable. Several solutions have been proposed in the literature to extend the theory of GGM and to enable their use in this area. In particular, there are two main ingredients that should be take in consideration for structure learning of gene regulatory networks using GGM: the estimation of partial correlation matrix and the identification of the set of edges that form the final graphical model.

In this thesis, we compared some recent procedures that aim to learn sparse networks in the *large p-small n* setting, through the use of both simulated and real data. We used different measures to evaluate the performance of these procedures. We considered different statistical performance measures and a graphical representation of the estimated adjacency matrices to evaluate the overall performance of each methodology in learning the network structures. We used the box-plots of MSE and a graphical representation of the estimated partial correlation coefficients for evaluating the accuracy in the estimation of partial correlation matrix. Finally, we computed the Precision-Recall curves to have general information on the model selection part of the structural learning process.

Overall performance. Concerning the analysis with simulated data, we noticed that the G-Lasso showed a high rate of false positives. In particular this problem was mainly present with the setting $n > p$ and, for the hub structure and cascade structure, it seemed to follow a pattern through replications. The results for the Shrinkage estimator have been considered only with the empirical Bayes approach, since the use of the Shrinkage estimator and the t-test for model selection did not identify any structure. We observed that this procedure had good performance for the pairwise structure. In contrast, it showed a worse performance for the hub structure and cascade structure, especially in term of false positives. In contrast, the MLE presented very similar results with either the empirical Bayes approach and the t-test approach. With the setting $n > p$, it was a very good structural learning procedure, also for hub structure, but with $n \to p$ its performance decreased, in particular for the high value of false negatives. The PC-algorithm presented a satisfactory performance for pairwise structure and cascade structure, but a fairly good performance for the hub structure because of the presence of false positives. With *E. coli* data, for the G-Lasso the results were very similar to the ones described above, except for a higher rate of false negative. In contrast, for the Shrinkage estimator with the empirical Bayes approach and the PC-algorithm, their performance have remarkably decreased. They had high values of false negatives that indicated an inability to learn the real network.

Partial correlation estimation. From this study, available only for the study with simulated data, we observed that the G-Lasso and the Shrinkage estimator had a similar behavior. They both did not distinguish between the values of partial correlation coefficients, but their estimation did not seem as biased as the ones of MLE. Indeed, the MLE performed poorly in term of covariance selection, in particular with $n \to p$. The PC-algorithm had a much better behavior than the other methods, but a direct comparison of the PC-algorithm with other procedures did not make sense because partial correlation matrices of PC-algorithm have been fitted under the sparse selected models. The good performance in terms of estimated partial correlation coefficients was therefore a consequence of sparsity and, only for completeness sake, we decided to include this case.

Precision-Recall curves. Regarding the comparison with simulated data, we noticed that the MLE, when applicable, was the more suitable method for all the structures, even if it worsened with $n \to p$. In contrast, the G-Lasso and the Shrinkage estimator were constant with the increment of p among the structures, and they showed a pretty bad performance only for the hub structure. The behavior of these latter methods were similar and quite bad with *E. coli* data.

Therefore, our main findings observed in this comparative study can be summarized as follow.

- For the G-Lasso the crucial point is the choice of the penalty parameter. A scalar penalty term, that is equal for every elements of Ω, does not learn in a accurate way the network structures.

- The Shrinkage estimator in the realistic setting $p > n$ does not out-perform the other procedures.

- The use of the empirical Bayes approach seems an interesting alternative to the t-test, in particular when the data set has $p > n$.

- The MLE, when it can be computed, out-performs other procedures as far as model selection is considered.

- The PC-algorithm performs well in the structural learning of the network, but it cannot be evaluate in terms of parameter estimation since it does not return information on the partial correlation coefficients.

- The choice of a suitable threshold for evaluating which partial correlation coefficients should be set to zero is a difficult point in the structural learning process.

- With simulated data, all the current methods seem able to extract a certain amount of structural information from the networks, although the precision is very low and the number of false positives very high. In addition, they all exhibit considerable difficulties if the network is formed by hub structures, which is the more interesting structure.

- With *E. coli* data, the methods perform very bad over all. They all show limitations in handling a network structure with a high sparsity rate that is based on hub structure with big motif size.

8.2 Discussion

There are many directions that can be considered for further research. Against the background of the present work three points appear particularly important.

- From background information on the structures of the gene regulatory network, it is known that these networks are typically sparse. This means that the number of edges in the network is much smaller than the number of possible edges in the complete network. In the structural learning procedures, the sparsity assumption is typically implemented by assuming that vertices have a small number of neighbors. Consequently, a structure as the hub one, where one vertex has a huge number of neighbors, is omitted from the analysis even if it is one of the most common structures in biological networks of many organisms. Hence, it is important that procedures and algorithms allow to identify structures that are biologically more realistic.

- More research needs to be done in the field of model selection for gene regulatory networks, and in general for biological networks. In particular, it is essential to avoid the threshold problem for identifying the set of edges in the final model. The use of Precision-Recall curves seems a suitable solution because it overcomes the problem of choosing a threshold and, in addition, provides a simple way for cross-comparison among several procedures. However, in order to compute these curves a benchmark network structure is necessary and this is not alway available. For the identification of edge sets, a new launching point could be developed by using the idea of "empirical posterior probabilities" of edges being present in the network (Schäfer and Strimmer, 2005a).

- The G-Lasso procedure is a very attractive algorithm since it performs simultaneously parameter and model selection. The latter part is strictly related to the penalization imposes on the elements of the concentration matrix, so an appropriate choice of the penalty parameter is a very important stage in the use of this procedure. However, it is reasonable to conclude that a scalar penalty term is useless because it does not emphasize the small number of no missing edges in a sparse network. So, the unique term could be substituted with a penalty matrix in which there are different penalizations for each variable. The choice of these different penalty terms could arrive from *a priori* knowledge of the biological system under study, or even from a pre-analysis of the data. For instance, the qp-graphs procedure (Castelo and Roverato, 2006) can be used as an explorative tool to assess the submodels of the complete graph. In this way, we could have an idea of the missing edges and then define the penalty terms.

Appendix A

Precision-Recall curves

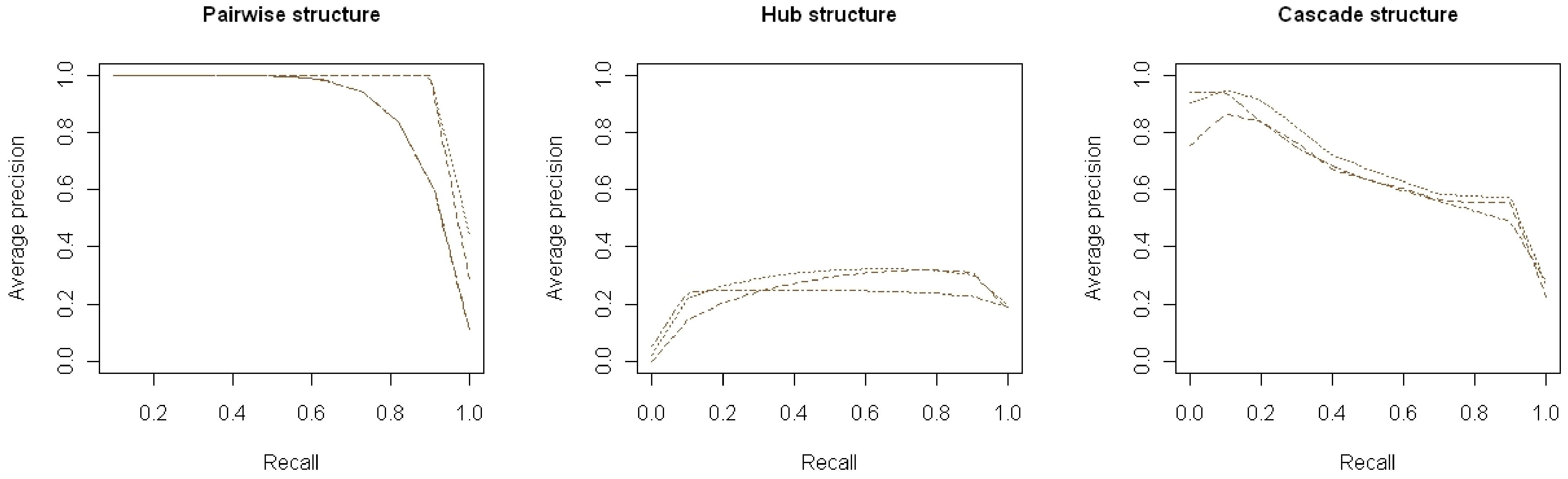


Figure A.1: Precision-Recall curves for G-Lasso ($p = 20$, $n = 150$); dashed for $\lambda = 0.05$, dotted for $\lambda = 0.1$, dotdash for $\lambda = 0.5$, and longdash for $\lambda = 0.8$.

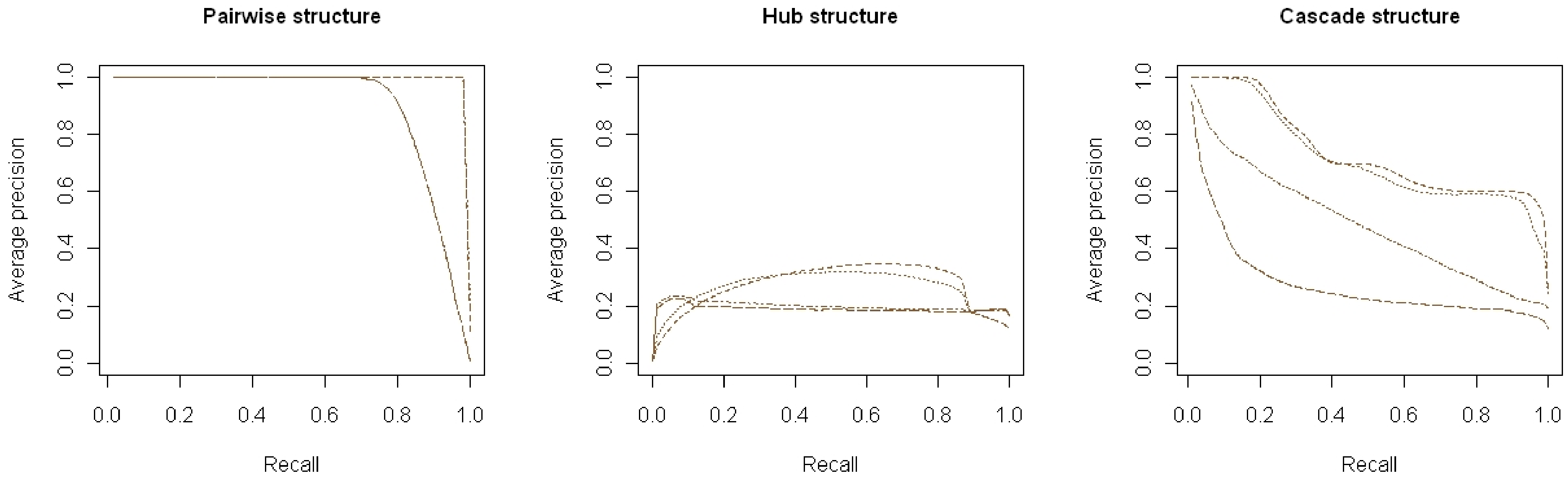


Figure A.2: Precision-Recall curves for G-Lasso ($p = 100$, $n = 150$); dashed for $\lambda = 0.05$, dotted for $\lambda = 0.1$, dotdash for $\lambda = 0.5$, and longdash for $\lambda = 0.8$.

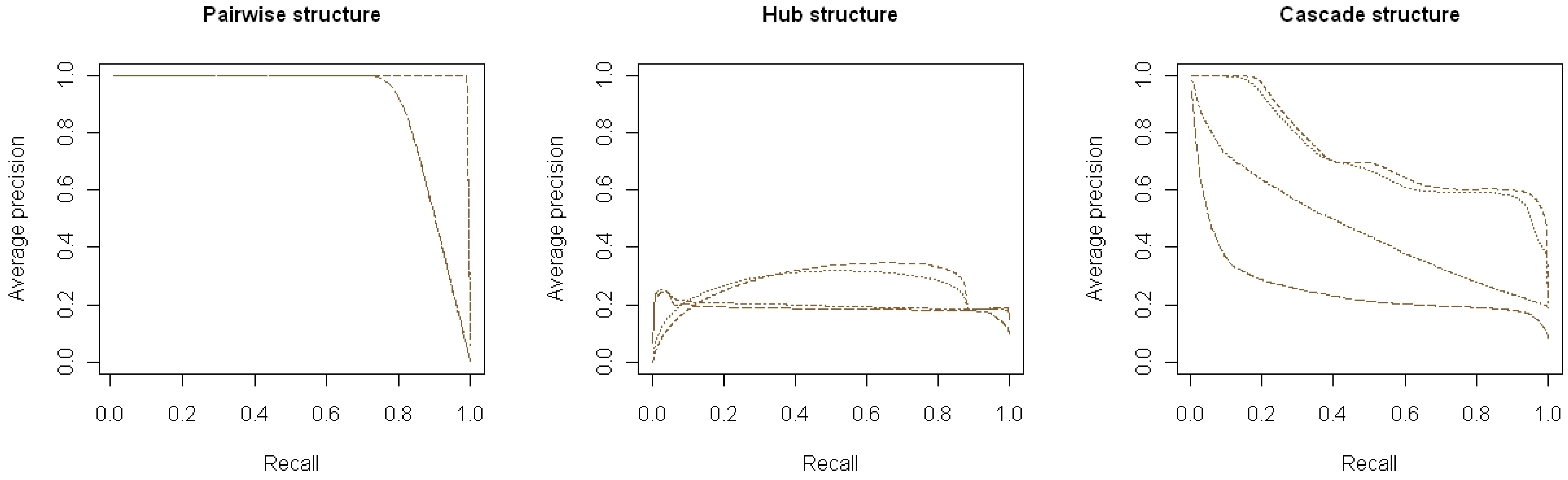


Figure A.3: Precision-Recall curves for G-Lasso ($p = 200$, $n = 150$); dashed for $\lambda = 0.05$, dotted for $\lambda = 0.1$, dotdash for $\lambda = 0.5$, and longdash for $\lambda = 0.8$.

Bibliography

Alberts, B., A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter (2008). *Molecular Biology of the Cell.* New York: Garland Science.

Alon, U. (2007). Network motifs: theory and experimental approaches. *Nat. Rev. Genet. 8*(6), 450–461.

Banerjee, O., L. E. Ghaoui, and A. d'Aspremont (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. Mach. Learn. Res. 9*, 485–516.

Barndorff-Nielsen, O. (1976). Factorization of likelihood function for full exponential families. *J. R. Stat. Soc. Ser. B 1*, 37–44.

Barrett, T., T. O. Suzek, D. B. Troup, S. E. Wilhite, W. C. Ngau, P. Ledoux, D. Rudnev, A. E. Lash, W. Fujibuchi, and R. Edgar (2007). Ncbi geo: mining millions of expression profilesŮdatabase and tools. *Nucl. Acids Res. 35*, D562–D566.

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc. B 57*(1), 289–300.

Benjamini, Y. and Y. Hochberg (2000). The adaptive control of false discovery rate in multiple hypotheses testing. *J. Behav. Educ. Statist. 25*(1), 60–83.

Boyer, L., T. I. Lee, M. F. Cole, S. E. Johnstone, S. S. Levine, J. P. Zucker, M. G. Guenther, R. M. Kumar, H. L. Murray, R. G. Jenner, D. K. Gifford, D. A. Melton, R. Jaenisch, and R. A. Young (2005). Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell 122*(6), 947–956.

Castelo, R. and A. Roverato (2006). A robust procedure for gaussian graphical model search from microarray data with p larger than n. *J. Mach. Learn. Res. 7*, 2621–2650.

Castelo, R. and A. Roverato (2009). Reverse engineering molecular regulatory networks from microarray data with qp-graphs. *J. Comput. Biol. 16*(2), 2621–2650.

Covert, M. W., E. M. Knight, J. L. Reed, M. J. Herrgard, and B. O. Palsson (2004). Integrating high-throughput and computational data elucidates bacterial networks. *Nature 429*(6987), 92–96.

Cowell, R. G., S. L. Lauritzen, A. P. Dawid, and D. J. Spiegelhalter (1999). *Probabilistic Networks and Expert Systems.* New York: Springer-Verlag.

Cox, D. and D. V. Hinkley (1974). *Theoretical Statistics.* London: Chapman and Hall.

Cox, D. and N. Wermuth (1996). *Multivariate dependencies: Models, analysis and interpretation.* London: Chapman and Hall.

Dempster, A. P. (1972). Covariance selection. *Biometrics 3*, 157–175.

Drton, M. and M. D. Perlman (2007). Multiple testing and error control in gaussian grafical model selection. *Statist. Sci. 22*(3), 430–449.

Dykstra, R. L. (1970). Establishing the positive definiteness of the sample covariance matrix. *Ann. Math. Statist. 41*(6), 2153–2154.

Edwards, D. (2000). *Introduction to Graphical Modelling.* New York: Springer.

Efron, B., R. Tibshirani, J. D. Storey, and V. Tusher (2001). Empirical bayes analysis of a microarray experiment. *J. Am. Statist. Assoc. 96*, 1151–1160.

Eichenberger, P., M. Fujita, S. T. Jensen, E. M. Conclo, D. Z. Rudner, S. T. Wang, C. Ferguson, K. Haga, T. Sato, J. S. Liu, and R. Losick (2004). The program of gene trascription for a single differentiating cell type during sporulation in bacillus subtilis. *PLoS Biol. 2*(10), e328.

Fan, J., Y. Feng, and Y. Wu (2009). Network exploration via the adaptive lasso and scad penalties. *Ann. Appl. Stat. 3*(2), 521–541.

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

Gama-Castro, S., V. Jimenez-Jacinto, M. Peralta-Gil, A. Santos-Zavaleta, M. I. Peñaloza-Spinola, B. Contreras-Moreira, J. Segura-Salazar, L. Muniz-Rascado, I. Martínez-Flores, H. Salgado, C. Bonavides-Martínez, C. Abreu-Goodger, C. Rodríguez-Penagos, J. Miranda-Ríos, E. Morett, E. Merino, A. M. Huerta, L. Trevino-Quintanilla, and J. Collado-Vides (2008). Regulondb (version 6.0): gene regulation model of escherichia coli k-12 beyond transcription, active (experimental) annotated promoters and textpresso navigation. *Nucl. Acids Res. 36*, D120–D124.

Gibson, G. and S. V. Muse (2004). *A primer of genome science.* Sunderland: Sinauer Associates.

Griffiths, A. J. F., S. R. Wessler, R. C. Lewontin, W. M. Gelbart, D. T. Suzuki, and J. H. Miller (2005). *Introduction to genetic analysis.* New York: WH Freeman.

Hotelling, H. (1953). New light on the correlation coefficient and its transforms. *J. R. Statist. Soc. B 15*, 193Ű–232.

Iranfar, N., D. Fuller, and W. F. Loomis (2006). Transcriptional regulation of post-aggregation genes in dictyostelium by a feed-forward loop involving gbf and lagc. *Dev. Biol. 290*(2), 460–469.

Kalisch, M. and P. Bühlmann (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *J. Mach. Learn. Res. 8*, 613–636.

Kitano, H. (2001). *Foundations of system biology.* Cambridge (MA): MIT Press; 2Rev edition.

Lauritzen, S. (1996). *Graphical models.* Oxford: Oxford University Press.

Ledoit, O. and M. Wolf (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *J. Empir. Finance 10*, 603–621.

Lee, T. I., N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young (2002). Transcriptional regulatory networks in *Saccharomyce cerevisiae*. *Science 298*, 799–804.

Lockhart, D., H. Dong, M. C. Bryne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat. Biotechnol. 14*, 1675–1680.

Meinshausen, N. and P. Bühlmann (2006). High-dimensional graphs and variable selection with the lasso. *Ann. Statist. 34*(3), 1436–1462.

Milo, R., S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon (2002). Network motifs: simple building blocks of complex networks. *Science 298*(5594), 824–827.

Odom, D. T., N. Zizlsperger, D. B. Gordon, G. W. Bell, N. J. Rinaldi, H. L. Murray, T. L. Volkert, J. Schreiber, P. A. Rolfe, D. K. Gifford, E. Fraenkel, G. I. Bell, and R. A. Young (2004). Control of pancreas and liver gene expression by hnf transcription factors. *Science 303*(5662), 1378–1381.

Pace, L. and A. Salvan (1997). *Principles of Statistical Inference from a Neo-Fisherian Perspective*. Singapore: World Scientific.

Paulsen, V. I., S. C. Power, and R. R. Smith (1989). Schur products and matrix completions. *J. Funct. Anal. 85*, 151–178.

Robins, J. M., R. Scheines, P. Spirtes, and L. Wasserman (2003). Uniform consistency in causal inference. *Biometrika 90*, 491–515.

Roverato, A. (2000). Cholesky decomposition of a hyper inverse wishart matrix. *Biometrika 87*(1), 99–112.

Roverato, A. (2005). A unified approach to the characterisation of markov equivalence classes of dags, chain graphs with no flags and chain graphs. *Scand. J. Statist. 32*, 295–312.

Schäfer, J. and K. Strimmer (2005a). An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics 21*(6), 754–764.

Schäfer, J. and K. Strimmer (2005b). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Stat. Appl. Genet. Mol. Biol. 4*(1), 1–32.

Shen-Orr, S. S., R. Milo, and U. Alon (2002). Network motifs in the transcriptional regulation network of escherichia coli. *Nat. Genet. 31*(1), 64–68.

Soranzo, N., G. Bianconi, and C. Altafini (2007). Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data. *Bioinformatics 23*(13), 1640–1647.

Spirtes, P. and C. Glymour (1991). An algorithm for fast recovery of sparse causal graphs. *Soc. Sci. Comput. Rev. 9*(1), 62–72.

Spirtes, P., C. Glymour, and R. Scheines (2000). *Causation, prediction, and search.* Cambridge (MA): MIT Press; 2Rev edition.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B 50*(1), 267–288.

Werhli, A. V., M. Grzegorczyk, and D. Husmeier (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics 22*(20), 2523–2531.

Wermuth, N. (1980). Linear recursive equations, covariance selection and path analysis. *J. Am. Statist. Assoc. 75*, 963–972.

Whittaker, J. (1990). *Graphical models in applied multivariate statistics.* New York: Wiley.

Yeung, M. K. S., J. Tegnér, and J. J. Collins (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. Natl. Acad. Sci. USA 99*, 6163–6168.

Yuan, M. and Y. Lin (2007). Model selection and estimation in the gaussian graphical model. *Biometrika 94*(1), 19–35.

Vanna Albieri

CURRICULUM VITAE

Contact Information

Department of Statistics, University of Padova
via Cesare Battisti 241-243, 35121 Padova (Italy)

e-mail: vanna.albieri@stat.unipd.it

Position

Starting from February 15, 2010
Department of Biostatistics and Epidemiology, Institute of Cancer Epidemiology, Danish Cancer Society, Copenhagen, Denmark.
Statistician.
Head of department: Dr. Joachim Schüz.

Education

January 2007-January 2010
PhD student in Statistical Sciences, University of Padova, Italy.
Thesis title: "A comparison of procedures for structural learning of biological networks".
Supervisors: Prof. Alberto Roverato, Dr. Vanessa Didelez.
Expected completion date: Spring 2010.

October 2003-July 2005
Second-level Laurea degree in Biostatistics and Experimental Statistics, University of Milano-Bicocca, Italy.
Thesis title: "Effect of concurrent radiation therapy and adjuvant chemotherapy CMF in women with early breast cancer".
Supervisors: Dr. Antonella Zambon, Prof. Giovanni Corrao.

October 2000-July 2003
First-level Laurea degree in Statistics, Population and Society, University of Padova, Italy.
Thesis title: "Analysis of school path and performance on students born in 1983 and resident in Rovigo".

Supervisor: Dr. Giovanna Boccuzzo.

Further education

September 2005-July 2006
University of California Santa Barbara (UCSB), USA.
Attended English courses and PhD classes in Statistics at UCSB as student of the "Extension Language Program".

Visiting periods

April 2008-June 2008/September 2008-December 2008
Department of Mathematics, University of Bristol, UK.
Researcher visiting under the supervision of Dr. Vanessa Didelez for developing part of the PhD thesis.

Teaching experiences

February 2009-May 2009
Faculty of Economics, University of Venezia, Italy.
Teaching assistant for the courses of Statistics I and Statistics II (15 hours).
Supervisor: Prof. Mario Romanazzi.

Conferences (general audience member)

December 17-18, 2007
Institut Henri Poincaré (IHP), Paris, France.
Mathematics for Biological Networks.

August 16, 2008
Department of Biostatistics, University of Copenhagen, Denmark.
Causal Inference and Marginal Structural Models.

August 17-21, 2008
The Royal School of Architecture, Copenhagen, Denmark.
International Society for Clinical Biostatistics 2008.

October 6-8, 2008
EURANDOM, Eindhoven, Netherlands.

Young European Statistician Workshop (YES-II) “High dimensional statistics”.

Seminar presentations

November 9, 2009
Department of Natural Sciences, University of Copenhagen, Denmark.
Seminar title: “A comparison of procedures for structural learning of biological networks”.

Work experiences

January 2005-June 2005
Institute for Cancer Research and Treatment (IRCC), Candiolo (TO), Italy.
Data manager at Department of Medical Oncology.
Director: M.D. Massimo Aglietta.

February 2003-May 2003
District of Rovigo, Rovigo, Italy.
Stager at Departmet of Statistics.
Director: Cinzia Viale.

Publications

Montemurro F., Gatti M., Redana S., Jacomuzzi M.E., Nanni D., Durando A., Puopolo M., Ponzone R., Rossi A., Albieri V., Valabrega G., Sismondi P., Gabriele P., Aglietta M. (2006). “Concurrent radiotherapy does not affect adjuvant CMF delivery but is associated with increased toxicity in women with early breast cancer”. *Journal of Chemotherapy*, 18, 90-97.