ULB UNIVERSITÉ LIBRE DE BRUXELLES, UNIVERSITÉ D'EUROPE

UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences
Département d'Informatique

Gaussian Graphical Model Selection for Gene Regulatory Network Reverse Engineering and Function Prediction

Kevin Kontos

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

Année académique 2008–2009

Gaussian Graphical Model Selection for Gene Regulatory Network Reverse Engineering and Function Prediction

Kevin Kontos

“We are drowning in information, but we are starved for knowledge.”

John Naisbitt

“Science may be described as the art of systematic over-simplification.”

Karl Popper

Abstract

One of the most important and challenging "knowledge extraction" tasks in bioinformatics is the reverse engineering of gene regulatory networks (GRNs) from DNA microarray gene expression data. Indeed, as a result of the development of high-throughput data-collection techniques, biology is experiencing a data flood phenomenon that pushes biologists toward a new view of biology–systems biology–that aims at system-level understanding of biological systems.

Unfortunately, even for small model organisms such as the yeast *Saccharomyces cerevisiae*, the number p of genes is much larger than the number n of expression data samples. The dimensionality issue induced by this "small n, large p" data setting renders standard statistical learning methods inadequate. Restricting the complexity of the models enables to deal with this serious impediment. Indeed, by introducing (a priori undesirable) bias in the model selection procedure, one reduces the variance of the selected model thereby increasing its accuracy.

Gaussian graphical models (GGMs) have proven to be a very powerful formalism to infer GRNs from expression data. Standard GGM selection techniques can unfortunately not be used in the "small n, large p" data setting. One way to overcome this issue is to resort to regularization. In particular, shrinkage estimators of the covariance matrix–required to infer GGMs–have proven to be very effective. Our first contribution consists in a new shrinkage estimator that improves upon existing ones through the use of a Monte Carlo (parametric bootstrap) procedure.

Another approach to GGM selection in the "small n, large p" data setting consists in reverse engineering limited-order partial correlation graphs (q-partial correlation graphs) to approximate GGMs. Our second contribution consists in an inference algorithm, the q-nested procedure, that builds a sequence of nested q-partial correlation graphs to take advantage of the smaller order graphs' topology to infer higher order graphs. This allows us to significantly speed up the inference of such graphs and to avoid problems related to multiple testing. Consequently, we are able to consider higher order graphs, thereby increasing the accuracy of the inferred graphs.

Another important challenge in bioinformatics is the prediction of gene function. An example of such a prediction task is the identification of genes that are targets of the nitrogen catabolite repression (NCR) selection mechanism in the yeast *Saccharomyces cerevisiae*. The study of model organisms such as *Saccharomyces cerevisiae* is indispensable for the understanding of more complex organisms. Our third contribution consists in extending the standard two-class classification approach by enriching the set of variables and comparing several feature selection techniques and classification algorithms.

Finally, our fourth contribution formulates the prediction of NCR target genes as a network inference task. We use GGM selection to infer multivariate dependencies between genes, and, starting from a set of genes known to be sensitive to NCR, we classify the

remaining genes. We hence avoid problems related to the choice of a negative training set and take advantage of the robustness of GGM selection techniques in the “small n, large p” data setting.

Résumé

L'un des principaux problèmes d'extraction de connaissance en bioinformatique est l'inférence de réseaux de régulation génique (GRNs ou *gene regulatory networks*) à partir de données provenant de puces à ADN. En effet, suite à l'essor des techniques de criblage à haut débit, la biologie est confrontée à un afflux massif de données qui incitent les biologistes à intégrer différents niveaux d'informations pour comprendre le fonctionnement global des systèmes biologiques.

Malheureusement, le nombre p de gènes est nettement supérieur au nombre n de données d'expression, y compris pour les petits organismes modèles tels la levure *Saccharomyces cerevisiae.* Ce problème de dimensionnalité rend les techniques classiques d'apprentissage statistique inappropriées. Une manière d'aborder ce problème consiste à restreindre la complexité des modèles. En effet, en introduisant du biais (a priori indésirable) dans la procédure de sélection de modèles, nous réduisons la variance du modèle sélectionné, accroissant ainsi sa précision.

Les modèles graphiques gaussien (GGMs ou Gaussian graphical models) se sont révélés être un puissant formalisme pour l'inférence des GRNs à partir de données d'expression. Les techniques classiques de sélection de GGM ne peuvent malheureusement pas être utilisées lorsque le nombre de données n est inférieur au nombre de gènes p. Afin de pallier cet inconvénient, une possibilité consiste à utiliser des approches de régularisation. En particulier, les estimateurs de type *shrinkage* de la matrice de covariance (qui détermine le GGM) se sont avérés être très robustes. Notre première contribution concerne la proposition d'un nouvel estimateur de type *shrinkage* utilisant une technique de Monte-Carlo (*bootstrap* paramétrique) dont les performances sont supérieures aux estimateurs existants.

Alternativement, la sélection de GGM peut s'effectuer via l'inférence de graphes de corrélation partielle d'ordre limité (ou *q-partial correlation graphs*) comme approximation de GGMs. Notre seconde contribution consiste en un algorithme, dénommé *q-nested procedure*, qui infère successivement des *q-partial correlation graphs* d'ordres croissants afin de tirer profit de la topologie des graphes d'ordres inférieurs. Cette approche nous permet de diminuer considérablement le temps requis à l'inférence de tels graphes. Par conséquent, il nous est possible de considérer des graphes d'ordres supérieurs, augmentant ainsi la précision des graphes inférés.

Un autre défi fondamental en bioinformatique est la prédiction de la fonction des gènes comme, par exemple, l'identification des gènes soumis à la répression catabolique azotée (NCR ou *nitrogen catabolite repression*) dans la levure *Saccharomyces cerevisiae.* L'étude d'organismes modèles tels *Saccharomyces cerevisiae* est un prérequis indispensable à la compréhension d'organismes plus complexes. Notre troisième contribution consiste en l'extension de l'approche classique de classification à deux classes pour l'identification de gènes soumis à la NCR par l'ajout de variables et la comparaison de plusieurs techniques de sélection de variables et de divers classificateurs.

Notre quatrième et dernière contribution consiste à formuler le problème de la prédiction de gènes soumis à la NCR comme un problème d'inférence de réseaux. Nous utilisons tout d'abord la sélection de GGMs pour inférer des dépendances multivariées entre gènes. Ensuite, étant donné un ensemble de gènes dont nous savons qu'ils sont impliqués dans la NCR, nous prédisons, parmi les gènes restants, ceux également soumis à la NCR. Nous évitons ainsi les problèmes liés au choix d'un ensemble d'exemples négatifs et tirons profit de la robustesse des techniques de sélection de GGMs.

Preface

Acknowledgments

First and foremost I wish to thank my supervisor, Prof. Gianluca Bontempi, who has guided my work. His constructive comments and criticism always inspired me to think more carefully and write more cogently. He has been a key mentor in my formation, and I am fortunate to have met him so early on during my time at the ULB.

Next, I would like to thank the members of my "steering committee," Prof. Hugues Bersini, Prof. Marcelle Kaufman and Prof. Guy Latouche, for their invaluable guidance.

I also wish to thank the members of the ARC project in which I took part, in particular Prof. Bruno André, Dr. Patrice Godard, Prof. Marcelle Kaufman and Prof. Jacques van Helden, for their constructive comments.

I am grateful to Prof. Raymond Devillers, Prof. Tom Lenaerts, Benjamin Haibe-Kains and Catharina Olsen, who took the time to read (parts of) a preliminary version of the thesis, for their invaluable feedback.

I also wish to thank the members of my jury, Prof. Bruno André, Prof. Hugues Bersini, Dr. Pierre Geurts, Prof. Tom Lenaerts and Prof. Jacques van Helden, for accepting to review my thesis.

I express my thanks to my colleagues of the Computer Science Department, in particular to the members of the Machine Learning Group (MLG), with whom I have passed great moments during the last four years.

Finally, I wish to convey special thanks to my relatives, in particular my parents and my brother, and friends, especially Alexandre (aka Alex, aka Μέγας Αλέξανδρος) and Daniel (aka El Gringo), for their support. Last but not least, I wish to express my sincere gratitude to Ludivine (aka Ludi) for her constant support, patience and encouragement, and for the unstinting reminder that there are more important things in life than a PhD thesis.

Financial support

The work presented in this thesis was supported by the Communauté Française de Belgique under Action de Recherche Concertée (ARC) Project no. 04/09-307. This project is entitled "Integrating experimental and theoretical approaches to decipher the molecular networks of nitrogen utilisation in yeast," and is steered by Prof. Bruno André (Département de Biologie Moléculaire, ULB), Prof. Gianluca Bontempi (Département d'Informatique, ULB), Prof. Jacques van Helden (Département de Biologie Moléculaire, ULB) and Prof. Marcelle Kaufman (Département de Chimie, ULB).

Members of the jury

- Prof. Bruno André, Université Libre de Bruxelles, Belgium
- Prof. Hugues Bersini, Université Libre de Bruxelles, Belgium
- Prof. Gianluca Bontempi, Université Libre de Bruxelles, Belgium (Thesis Supervisor)
- Dr. Pierre Geurts, Université de Liège, Belgium
- Prof. Tom Lenaerts, Université Libre de Bruxelles, Belgium (President)
- Prof. Jacques van Helden, Université Libre de Bruxelles, Belgium (Secretary)

Declaration

This thesis has been composed by the author himself and contains original work of his own execution. Some of the reported work has been done in cooperation with members of the aforementioned ARC Project, whose contributions are acknowledged in the relevant sections.

Publications

Parts of this thesis have been published in the following international peer-reviewed conference proceedings and journals:

- Kontos, K., André, B., van Helden, J., and Bontempi, G. (2009). Gaussian graphical models to infer putative genes involved in nitrogen catabolite repression in *S. cerevisiae*. In Pizzuti, C., Ritchie, M. D., and Giacobini, M., editors, *Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBIO 2009)*, volume 5483 of *Lecture Notes in Computer Science (LNCS)*, pages 13–24. Springer.
- Kontos, K. and Bontempi, G. (2008a). An improved shrinkage estimator to infer regulatory networks with Gaussian graphical models. In *Proceedings of the 24th Annual ACM Symposium on Applied Computing (ACM SAC 2009)*.
- Kontos, K. and Bontempi, G. (2008b). Nested q-partial graphs for genetic network inference from "small n, large p" microarray data. In Elloumi, M., Küng, J., Linial, M., Murphy, R., Schneider, K., and Toma, C., editors, *Proceedings of the 2nd International Conference on Bioinformatics Research and Development (BIRD 2008)*, number 13 in Communications in Computer and Information Science (CCIS), pages 273–287, Heidelberg. Springer.
- Kontos, K. and Bontempi, G. (2008c). Nested q-partial graphs for genetic network inference from "small n, large p" microarray data. In *Proceedings of Journées Ouvertes Biologie Informatique Mathématiques (JOBIM 2008)*.
- Kontos, K., Godard, P., André, B., van Helden, J., and Bontempi, G. (2008). Machine learning techniques to identify putative genes involved in nitrogen catabolite repression in the yeast *Saccharomyces cerevisiae*. *BMC Proceedings*, 2(Suppl 4):S5.
- Kontos, K., Godard, P., André, B., van Helden, J., and Bontempi, G. (2007). Machine learning techniques to identify putative genes involved in nitrogen catabolite repres-

sion in the yeast *Saccharomyces cerevisiae.* In *Proceedings of the First International Workshop on Machine Learning in Systems Biology (MLSB 2007)*, pages 21–26.

- Godard, P., Urrestarazu, A., Vissers, S., Kontos, K., Bontempi, G., van Helden, J., and André, B. (2007). Effect of 21 different nitrogen sources on global gene expression in the yeast *Saccharomyces cerevisiae. Molecular and Cellular Biology*, 27(8):3065–3086.

Other parts appeared in:

- Kontos, K. (2005). *Machine Learning Methods for Network Inference from Microarray Data.* Master's thesis, Université Libre de Bruxelles, Belgium.

Finally, some of the work performed during my PhD research, which was published in the following international peer-reviewed conference proceedings and journals, is not included in the thesis:

- Meyer, P. E., Kontos, K., and Bontempi, G. (2007a). Biological network inference using redundancy analysis. In Hochreiter, S. and Wagner, R., editors, *Proceedings of the 1st International Conference on Bioinformatics Research and Development (BIRD 2007), Lecture Notes in Bioinformatics*, volume 4414, pages 916–927. Springer.
- Meyer, P. E., Kontos, K., Lafitte, F., and Bontempi, G. (2007b). Information-theoretic inference of large transcriptional regulatory networks. *EURASIP Journal on Bioinformatics and Systems Biology*, Article ID 79879, 9 pages.
- Kontos, K. and Bontempi, G. (2006). Scale-free paradigm in yeast genetic regulatory network inferred from microarray data. In Kovacs, T. and Marshall, J., editors, *Proceedings of AISB'06: Adaptation in Artificial and Biological Systems*, volume 3, pages 139–144.

Contents

Background

Contributions to Gene Regulatory Network Reverse Engineering from Gene Expression Data

Contributions to Nitrogen Catabolite Repression Target Gene Prediction

Appendices

1

Introduction

We address the problem of reverse engineering GRNs from "small n, large p" DNA microarray data with multivariate probabilistic models, known as graphical models, in which conditional independence constraints between genes are specified by graphs. In particular, we focus on an undirected graphical model, which assumes multivariate normality of the data, known as the Gaussian graphical model (GGM). We then tackle the problem of nitrogen catabolite repression (NCR) target genes prediction in Saccharomyces cerevisae using two-class classification and the GGM.

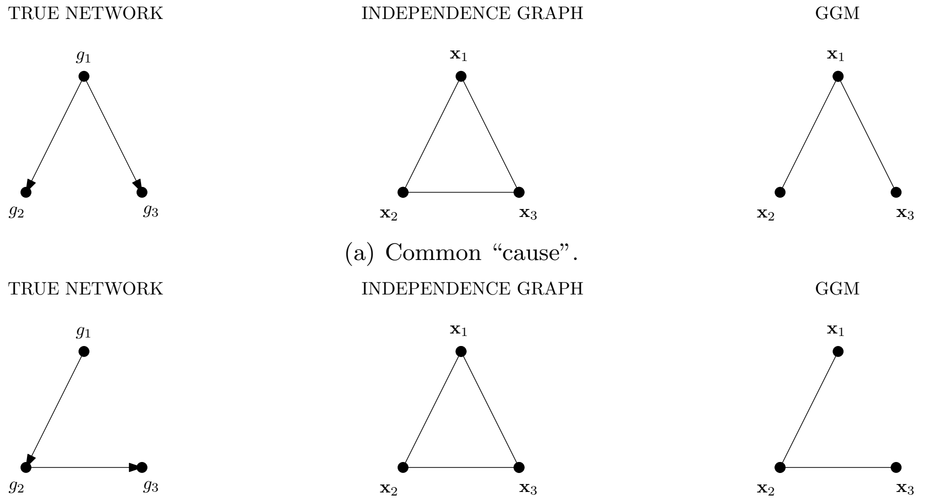
One of the most important and challenging tasks in biology consists in the identification of interactions between genetic components, like genes and proteins, within complex living organisms. Indeed, the data flood that biology is experiencing [198] is pushing scientists toward a new view of biology: *systems biology*, as it is called, aims at system-level understanding of biological systems. This field investigates the functional behavior and relationships of all of the components in a particular biological system [127, 252].

The availability of genome-wide gene expression technologies has enabled scientists to make considerable progress towards achieving this goal through the identification of the interactions between genes in living systems, or *gene regulatory networks*[1] (GRNs). In particular, *DNA microarrays* enable to monitor the whole *transcriptome* on a single chip so that researchers can have a picture of the interactions among thousands of genes simultaneously. As such this technology has attracted tremendous interest among biologists and has become one of the most widely used sources of genome-scale data [24].

The process of building GRNs from DNA microarray gene expression data, known as *network inference* or *reverse engineering*, is the first main topic of this thesis (Section 1.2).

This process is far from being trivial because of the poor information content of DNA microarray data, which are corrupted by substantial amounts of measurement noise [252], and the combinatorial nature of the problem. Indeed, gene expression levels are regulated by the combined action of multiple gene products [23, 122]. Moreover, the number n of measurements is relatively small (on the order of tens or hundreds) compared to the number p of measured objects (on the order of hundreds or thousands). This *"small n,*

[1] We use the terms "network" and "graph" interchangeably.



(a) Common "cause".

(b) Sequential pathway.

Figure 1.1: (a) A simple gene regulatory network (GRN; Section 2.3) consisting of 3 genes (left). Note that genes are denoted by g, while the random variables of interest (for example the expression levels of the genes) are denoted by $\mathbf{x}$. The arrow pointing from gene g_1 to gene g_2 (resp. g_3) means that g_1 regulates g_2 (resp. g_3). The expression levels $\mathbf{x}_2$ and $\mathbf{x}_3$ of, respectively, genes g_2 and g_3 are highly correlated with each other because g_2 and g_3 are both regulated by gene g_1. The spurious relation between the expression levels $\mathbf{x}_2$ and $\mathbf{x}_3$ will therefore be inferred in the independence graph (middle), which is a simple model that does not take into account the effect of the remaining variables ($\mathbf{x}_1$ in this example). In the GGM (right), however, the spurious relation between $\mathbf{x}_2$ and $\mathbf{x}_3$ will not be inferred because the effect of $\mathbf{x}_1$ is taken into account. Note that the directions of the identified connections are not inferred in the independence graph and the GGM. (b) This example, which is similar to (a), illustrates a sequential pathway.

large p" data setting renders learning tasks in molecular biology more challenging.

We tackle the problem of reverse engineering GRNs from DNA microarray data with multivariate probabilistic models, known as *graphical models*, in which conditional independence constraints between genes are specified by graphs. In particular, we focus on undirected graphical models which assume multivariate normality of the data, known as *Gaussian graphical models* (GGMs). GGMs have become very popular in bioinformatics as they enable to *distinguish between direct and indirect interactions* by taken into account the *effect of all remaining observed genes* (Figure 1.1).

Of course, for certain inference tasks, such as the reverse engineering of co-expression networks, independence graphs, which are simpler models that do not take into account the effect of the remaining variables, still play an important role [27].

Unfortunately, *GGM selection* is an ill-posed problem in the "small n, large p" data setting that characterizes many bioinformatics problems in general, and expression data in particular. Indeed, the usual sample concentration matrix—the maximum likelihood estimate of the (population) concentration matrix—requires the sample covariance matrix to be positive definite and this holds, with probability one, if and only if $n > p$ [69]. To cope with this dimensionality issue, two approaches have been proposed in the literature. The first one uses *regularization* and the second one uses limited-order partial correlation graphs, or *q-partial correlation graphs*. However, issues arise in both cases (Section 1.1). It is the aim of our two first contributions, which consist in a new *shrinkage estimator* and an algorithm–the *q-nested procedure*–to efficiently infer q-partial correlation graphs (Section 1.2) to tackle these problems.

Note that some authors try to mitigate the dimensionality issue by collecting as much data as possible (in order to increase the number n of samples) and by building large databases of experimental data. However, robust procedures, such as the ones developed in this thesis, are still crucial for inference tasks because simply increasing n is not sufficient. Indeed, even when $n > p$, standard techniques might still perform poorly, unless n is much larger than p (quantitatively determining how much larger n must be depends on the underlying inference task). Even if DNA microarray data can nowadays be more easily and more cheaply collected than a decade ago, new technologies, such as microRNA microarrays, are constantly being introduced. By the time these new technologies become the mainstream, robust methods able to cope with the dimensionality issue incurred by the "small n, large p" data setting are indispensable to take advantage of these new technologies.

Another important and challenging task in biology is *gene function prediction*. Often, biologists know the function of some (but not all) genes with respect to a specific process and their goal is to infer other genes involved in this process. In particular, the second main problem we will tackle in this thesis (Section 1.3) is the inference of *nitrogen catabolite repression* (NCR) target genes in the yeast *Saccharomyces cerevisae* (*S. cerevisae*). Yeast is a relatively simple unicellular organism for which the entire genomic sequence and the functional roles of approximately 60% of the genes are known [36, 37, 77]. Therefore, it has been widely used in genomics as a *model organism*. The study of such an organism is indispensable for the understanding of more complex ones [84].

NCR is the process studied in the ARC project that supported the work presented in this thesis (see the Preface). It is an important biological process in *S. cerevisae* which involves an essential nutrient for all life forms: *nitrogen*. The emergence of cells able to transport, catabolize and synthesize a wide variety of nitrogenous compounds has thus been favored by evolutionary selective pressure [107]. As a consequence, *S. cerevisiae* can use 27 distinct nitrogen-containing compounds, including amino acids, urea, ammonium, nitrogen bases, and purine derivatives [107]. Like most unicellular organisms, yeast transports and catabolizes good nitrogen sources in preference to poor ones. NCR is this selection mechanism. All known nitrogen catabolite pathways are regulated by four regulators (Gln3, Gat1, Dal80, and Deh1). Moreover, approximatively 40 genes have been annotated as NCR-sensitive. The ultimate goal is to identify all genes involved in NCR.

We first tackle this problem of inferring NCR target genes by adopting a "standard" two-class classification approach. We will then make a connection with the first part of the thesis by proposing a new approach for predicting NCR genes based on a network inference paradigm.

The thesis consists of four main contributions which are split into two parts. The first one concerns the reverse engineering of GRNs in the "small n, large p" data setting and is introduced in Section 1.2. The second part is devoted to the prediction of NCR target genes and is presented in Section 1.3. A summary of the four contributions and references to the relevant publications are provided in Section 1.4.

1.1 Gaussian graphical model selection in the "small n, large p" data setting

Graphical models are representations of multivariate probabilistic models in which conditional independence (Section 4.1) constraints are specified by graphs (Sections 4.4 and 4.5). The vertices of the graph represent the variables, i.e., the genes (more specifically, a variable represents a particular characteristic of a gene, such as its expression level).

These models have gained much attention as they encode full conditional relationships between variables, i.e., genes. Hence, they enable to distinguish between direct and indirect interactions. Although they have the disadvantage of leaving dynamical aspects of gene regulation implicit, these models are becoming increasingly important as recent studies indicate that "the topology of networks is a determining factor in both re-engineering the network as well as understanding network and organism evolution" [85].

The particularity of GGMs is that they assume multivariate normality of the data. The independence relationships between variables can hence be inferred through partial correlations which are intimately linked to linear regression and to the corresponding problem of estimating the covariance matrix.

Hence, GGMs only capture linear forms of dependencies. Although the normality of DNA microarray data is a disputed question [103, 269], it seems that this limitation is not very stringent given the poor information content of the data [103].

Unfortunately, GGM selection is an ill-posed problem in the "small n, large p" data setting (Section 4.7.1) that characterizes many bioinformatics problems in general, and DNA microarray data in particular (Section 2.1.2). Standard GGM selection approaches can therefore not be used. To cope with this dimensionality issue, two alternatives have been proposed in the literature.

The first one uses *regularization* which consists in constraining the parameters of an estimator to prevent overfitting by imposing a "simpler" estimator, hence reducing its variance (Section 3.5). Indeed, GGM selection reduces to estimating a covariance matrix. This avenue of research has been explored by many authors. In particular, Ledoit and Wolf [153] proposed a shrinkage estimator (see also Schäfer and Strimmer [212]). They showed that the estimation of the covariance matrix could be improved by finding an optimal convex combination of the sample covariance matrix and a constrained covariance matrix, for which they provide an analytical solution (Section 5.2). Intuitively, their approach reduces to balancing bias and variance to reduce the mean squared error (MSE). Unfortunately, the parameter defining shrinkage depends on unknown quantities and needs to be estimated consistently (Section 1.2).

The second approach to cope with the dimensionality issue is to use limited-order partial correlation graphs, or *q-partial correlation graphs* (Section 4.7.1). It has been shown both theoretically and experimentally that such graphs provide accurate approximations of the full conditional independence structure between the variables thanks to the sparsity of genetic networks. Despite the promising results obtained in the literature, the computational burden of the existing reverse engineering algorithms limits the applicability of this approach (Section 1.2).

We now present our contributions to tackle the problems arising with the existing solutions to GGM selection in the "small n, large p" setting within the context of GRNs inference from DNA microarray data (Section 1.2). Note that we will also use GGM selection for the identification of NCR target genes (Section 1.3).

1.2 Reverse engineering gene regulatory networks from DNA microarray data

One of the most important and challenging tasks in bioinformatics is thus the reverse engineering of biological networks. In particular, we focus on the inference of gene regulatory networks (GRNs, Section 2.3) from DNA microarray gene expression data (Section 2.1.2). We consider *observational data* (so-called passive observations) and not active interventions (such as gene knockouts). Moreover, we assume the observations to be *independent and identically distributed* (i.i.d) and will therefore not consider time-series data.

The inference of GRNs from microarray data makes two simplifying assumptions: the protein synthesis depends directly on the amount of mRNA (Section 2.1) and genes directly affect each other. These networks therefore constitute a simplification of the complete cellular system. However, they are a logical way of describing phenomena observed with transcription profiling.

Among the many existing modeling formalisms to reverse engineer GRNs from DNA microarray data (Section 2.4), we focus on graphical models (Section 2.4.4), and, more specifically, on the Gaussian graphical model (GGM; Section 2.4.4.1) which has gained much attention recently.

Concerning the regularization approach to GGM selection in the "small n, large p" data setting, we show that the optimal shrinkage intensity estimator of the shrinkage estimator proposed by Ledoit and Wolf [153] (see also Schäfer and Strimmer [212]) is biased (Section 5.3). Consequently, we propose a parametric bootstrap approach [34, 116] to estimate this bias (Section 5.4) and derive a "bias-corrected" shrinkage estimator (Section 5.5), which marks our first main contribution. The applicability and usefulness of our estimator are demonstrated on both simulated and real expression data (Sections 5.6 and 5.7, respectively).

As a second main contribution, we propose an efficient algorithm to considerably speed up the inference of q-partial correlation graphs (Chapter 6) within the context of the recently proposed q-partial correlation graph theory [30] that takes advantage of GRNs' *sparseness* (note that our procedure does not assume sparseness but exploits it when present). By adopting a screening procedure, we iteratively build nested graphs by discarding the less relevant edges. Moreover, by conditioning only on relevant variables, we diminish the problems related to multiple testing. This procedure allows us to faster infer limited-order partial correlation graphs and therefore to consider higher order values, which increases the accuracy of the inferred graph. The effectiveness of the proposed procedure is shown on simulated data.

1.3 Predicting nitrogen catabolite repression target genes

Nitrogen is an essential nutrient for all life forms. The emergence of cells able to transport, catabolize and synthesize a wide variety of nitrogenous compounds has thus been favored by evolutionary selective pressure [107]. As a consequence, the yeast *S. cerevisiae* can use 27 distinct nitrogen-containing compounds (Section 2.2). Like most unicellular organisms, yeast transports and catabolizes good nitrogen sources in preference to poor ones. Nitrogen catabolite repression (NCR) refers to this selection mechanism.

The ultimate goal is to identify all genes involved in NCR. This challenge has mainly been tackled by three genome-wide experimental studies [11, 107, 214], one of which [107] stems from the ARC project. In this contribution [107], we also proposed a bioinformatics approach, which we refer to as Godard et al. [107]'s approach, to complement the experimental study. Indeed bioinformatics methods offer the possibility to identify putative NCR genes and to discard uninteresting genes, hence strengthening the results of the experimental study.

A first approach to infer putative NCR genes is to adopt a classification approach [121, 219]. In Godard et al. [107], we formulated the identification of putative NCR genes in the yeast *S. cerevisiae* as a supervised two-class classification problem (Section 7.1). The (trained) classifiers predict whether genes are NCR-sensitive or not based on the number of occurrences of NCR-related motifs in their upstream noncoding sequences.

The third main contribution of the thesis consists in extending this two-class classification approach (Section 7.2). Instead of focusing on NCR-related motifs in the upstream noncoding sequences of the genes, we concentrate solely on the `GATA` motif. Indeed, the promoter regions of NCR target genes typically contain several `5'-GATA-3'` core sequences, which we will refer to as GATA boxes, recognized by the GATA family transcription factors (Godard et al. [107] and references therein). We specify a large number of variables related to this motif (Section 7.2.2). These variables define characteristics that biologists (who took part in the aforementioned ARC project) hypothesize to be relevant to NCR. Our goal mainly consists in determining new properties that could be determinant in NCR.

We also define a negative training set of manually-selected genes known to be insensitive to NCR (Section 7.2.1), thus avoiding the computational expensive undersampling approach (Section 3.7.3) adopted previously [107]. Besides, different classifiers (Section 7.2.3) and variable selection methods (Section 7.2.4) are compared.

We then show the effectiveness of our approach (Section 7.3). In particular, we show that all classifiers make significant and biologically valid predictions by comparing these predictions to annotated and putative NCR genes (Section 7.3.1), and by performing several negative controls. Moreover, the inferred NCR genes significantly overlap with putative NCR genes identified in three aforementioned genome-wide experimental studies

(Section 7.3.2). These results suggest that our approach can successfully identify potential NCR genes. Hence, the dimensionality of the problem of identifying all genes involved in NCR is drastically reduced. Finally, we identify previously uncharacterized variables. Further experimental analysis is however required to determine whether these variables indeed play a role in NCR.

Despite delivering promising results, these two-class classification approaches suffer from a major drawback: they require a negative training set. Indeed, four genes have been identified as NCR regulators and a few tens of genes have been annotated as NCR genes, but *a priori* there are no known "non-NCR" genes. Obviously, the faced problem corresponds more to *one-class classification* [233] than to two-class classification. One-class classification tries to discriminate one class of objects from all other possible objects by learning from a training set containing only the objects of that class. This observation leads us to the fourth main contribution of the thesis, which consists in a network inference approach to one-class classification (Section 8.1). In a nutshell, our approach consists in inferring a Gaussian graphical model (GGM) based on the number of occurrences of NCR-related motifs in the upstream noncoding sequences of the genes. To circumvent the dimensionality issue, we use Ledoit and Wolf [153]'s shrinkage estimator (Section 5.2). Given a set of NCR related genes, we then exploit the topology of the inferred network for functional information. More specifically, the neighbors of the genes of interest (the NCR regulators or/and the annotated NCR genes) are identified as putative NCR genes.

This approach does not require a negative training set[2] and thus avoids the problems encountered with the methods introduced in Chapter 7.

Furthermore, the network structure can give further insight into the considered problem. Indeed, "in real world applications, graphical [...] models are not only a tool for operations such as classification or prediction, but usually the network structures of the models themselves are also of great interest" [160]. This approach provides a more subtle and rich picture of the considered problem. Although we ultimately look at the neighbors of the genes of interest, the network topology offers the possibility to biologists to conduct a more detailed and refined analysis. While a standard classification approach only predicts NCR genes, a network approach also gives information on the interactions between the inferred NCR genes as well as on their interactions with the remaining genes. We deem that a network inference approach is more adequate to deal with such a problem.

Finally, this procedure is by far less computationally expensive that the two two-class classification approaches introduced previously. The feature selection and training phases are replaced by the inference of a regularized covariance matrix.

1.4 Contributions' summary

We summarize the four contributions of the thesis and provide references to the relevant publications.

1. In Chapter 5, we introduce an improved regularized estimator of the covariance matrix. We show that the optimal shrinkage intensity estimator proposed by Ledoit and Wolf

[2] Nevertheless, we will use negative validation sets for comparison purposes.

[153] (see also Schäfer and Strimmer [212]) is biased (Section 5.3). Consequently, we propose a parametric bootstrap approach [34, 116] to estimate this bias (Section 5.4) and derive a "bias-corrected" shrinkage estimator (Algorithm 5.1; Section 5.5). The applicability and usefulness of our estimator are demonstrated on both simulated and real expression data (Sections 5.6 and 5.7, respectively). This contribution appeared in:

– Kontos, K. and Bontempi, G. (2009a). An improved shrinkage estimator to infer regulatory networks with Gaussian graphical models. In *Proceedings of the 24th Annual ACM Symposium on Applied Computing (ACM SAC 2009)*.

2. In Chapter 6, we propose the q-nested procedure to infer limited-order partial correlation graphs or q-partial correlation graphs (Sections 6.1 and 6.2). It has been shown both theoretically and experimentally that such graphs provide accurate approximations of the full conditional independence structure between the variables thanks to the sparsity of genetic networks (Section 6.3). Alas, computing limited-order partial correlation coefficients for large networks, even for small order values, is computationally expensive, and often even intractable (Section 6.4). Moreover, problems deriving from multiple statistical testing arise, and one should expect that most of the edges are removed. Our procedure tackles both problems by reducing the dimensionality of the inference task (Algorithms 6.1 and 6.2; Section 6.5). By adopting a screening procedure, we iteratively build nested graphs by discarding the less relevant edges. Moreover, by conditioning only on relevant variables, we diminish the problems related to multiple testing. This procedure allows us to faster infer limited-order partial correlation graphs and therefore to consider higher order values, which increases the accuracy of the inferred graph. The effectiveness of the proposed procedure is shown on simulated data (Section 6.7). This contribution appeared in:

– Kontos, K. and Bontempi, G. (2008b). Nested q-partial graphs for genetic network inference from "small n, large p" microarray data. In Elloumi, M., Küng, J., Linial, M., Murphy, R., Schneider, K., and Toma, C., editors, *Proceedings of the 2nd International Conference on Bioinformatics Research and Development (BIRD 2008)*, number 13 in Communications in Computer and Information Science (CCIS), pages 273–287, Heidelberg. Springer.

– Kontos, K. and Bontempi, G. (2008c). Nested q-partial graphs for genetic network inference from "small n, large p" microarray data. In *Proceedings of Journées Ouvertes Biologie Informatique Mathématiques (JOBIM 2008)*.

3. In Chapter 7, we present a machine learning approach where the identification of putative NCR genes in the yeast *S. cerevisiae* is formulated as a supervised two-class classification problem. Classifiers predict whether genes are NCR-sensitive or not from a large number of variables related to the `GATA` motif in the upstream noncoding sequences of the genes (Section 7.2.2). The positive and negative training sets are composed of annotated NCR genes and manually-selected genes known to be insensitive to NCR, respectively (Section 7.2.1). Different classifiers (Section 7.2.3) and variable selection methods (Section 7.2.4) are compared. We then show the effectiveness of our approach (Section 7.3). In particular, we show that all classifiers make significant and biologically valid predictions by comparing these predictions to annotated and putative NCR genes (Section 7.3.1), and by performing several negative controls. Moreover, the inferred NCR genes significantly overlap with putative NCR genes identified in three genome-wide experimental and bioinformatics studies (Section 7.3.2). This contribution appeared in:

- Kontos, K., Godard, P., André, B., van Helden, J., and Bontempi, G. (2008). Machine learning techniques to identify putative genes involved in nitrogen catabolite repression in the yeast *Saccharomyces cerevisiae*. *BMC Proceedings*, 2(Suppl 4):S5.
- Kontos, K., Godard, P., André, B., van Helden, J., and Bontempi, G. (2007). Machine learning techniques to identify putative genes involved in nitrogen catabolite repression in the yeast *Saccharomyces cerevisiae*. In *Proceedings of the First International Workshop on Machine Learning in Systems Biology (MLSB 2007)*, pages 21–26.
- Godard, P., Urrestarazu, A., Vissers, S., Kontos, K., Bontempi, G., van Helden, J., and André, B. (2007). Effect of 21 different nitrogen sources on global gene expression in the yeast *Saccharomyces cerevisiae*. *Molecular and Cellular Biology*, 27(8):3065–3086.

4. In Chapter 8, we propose an approach based on Gaussian graphical models (GGMs), which enable to distinguish between direct and indirect interactions between genes, to identify putative NCR genes from putative NCR regulatory motifs and over-represented motifs in the upstream noncoding sequences of annotated NCR genes (Algorithm 8.1). Because of the high-dimensionality of the data, we use a shrinkage estimator of the covariance matrix to infer the GGMs. We show that our approach makes significant and biologically valid predictions. We also show that GGMs are more effective than models that rely on measures of direct interactions between genes. This contribution appeared in:

- Kontos, K., André, B., van Helden, J., and Bontempi, G. (2009). Gaussian graphical models to infer putative genes involved in nitrogen catabolite repression in *S. cerevisiae*. In Pizzuti, C., Ritchie, M. D., and Giacobini, M., editors, *Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBIO 2009)*, volume 5483 of *Lecture Notes in Computer Science (LNCS)*, pages 13–24. Springer.

1.5 Outline

In the *Background* part we introduce the biological setting (Chapter 2), the basic machine learning concepts (Chapter 3) and the Gaussian graphical model (GGM, Chapter 4) used throughout the thesis.

Next, we move to the *Contributions to Gene Regulatory Network Reverse Engineering from Gene Expression Data* part where we present our contributions to GGM selection in the “small n, large p” setting for inferring GRNs. They consist in an improved shrinkage estimator (Chapter 5) and a nested procedure for inferring q-partial (correlation) graphs (Chapter 6).

Subsequently, we proceed with the *Contributions to Nitrogen Catabolite Repression Target Gene Prediction* part where we present our contributions to NCR target gene prediction. We extend a “standard” two-class classification approach (Chapter 7) and propose a GGM selection method (Chapter 8) for this prediction task.

Chapter 9 concludes the thesis.

Appendix A summarizes the notation used throughout the thesis. Appendix B lists the commonly used abbreviations and acronyms. Appendix C provides a glossary of biological terms. The remaining appendices give background information that has been kept out of the main text for the sake of readability and are referenced in the relevant sections of the thesis. A subject index is provided after the bibliography.

Background

2

Biological Setting

For the proper understanding of the remainder of the thesis, we first overview some basic concepts of molecular biology (Section 2.1). In particular, we introduce DNA microarrays (Section 2.1.2) which provide the expression data from which gene regulatory networks (GRNs) are inferred. We also introduce an important biological process–nitrogen catabolite repression (NCR; Section 2.2)–occurring in the yeast *Saccharomyces cerevisae*. The study of such a model organism is indispensable for the understanding of more complex organisms. By addressing the problem of NCR target genes identification, we tackle another important and challenging task in bioinformatics, namely gene function prediction.

2.1 Basics of molecular biology

This section presents the biological processes involved in gene regulation. By necessity, it gives a simplified view of current biological knowledge. However, it provides a sufficient level of detail to allow an appreciation of what is included and omitted by the different formalisms presented in this thesis.

2.1.1 The DNA-protein relationship[1]

According to the current biological paradigm, information in a biological organism is stored in its *genome*. The genome constitutes the complete set of genes in the chromosomes of each cell of a particular organism. It consists of long molecules of DNA made up of chains of nucleotides in a double-helix structure.

Proteins are the fundamental structural and functional units in cells, which are the basic functional units of the genome.

The *central dogma* of molecular biology states that information stored in the DNA of a given gene is transcribed into RNA, which is then translated into proteins as illustrated in Figure 2.1.

The central dogma is represented by four major stages:

1. the DNA replicates its information in a process known as *replication* that involves many enzymes;
2. the DNA codes for the production of messenger RNA (mRNA) during *transcription*;

[1] This section is mainly based on Lodish et al. [163]

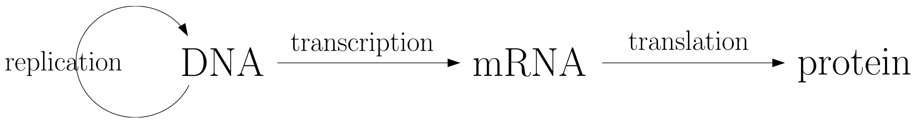


Figure 2.1: The central dogma of biology: DNA-protein relationship.

3. in eukaryotic cells, the mRNA is processed (essentially by splicing) and migrates from the nucleus to the cytoplasm (not shown in Figure 2.1) in a process called *RNA processing*;
4. messenger RNA carries coded information to ribosomes who "read" this information and use it for protein synthesis in a process known as *translation*.

Each protein is specialized to carry on a variety of important roles, such as a structural element, enzyme catalyst or antibody. A large subset of proteins known as *transcription factors* (TFs) also play a regulatory role, determining when, where and how much a particular gene is expressed into proteins. Because regulatory proteins are themselves the products of expressed genes, they are themselves under regulatory control, giving rise to complex networks of interacting genes (Section 2.3).

The following two sections describe the processes of transcription and translation that mediate the path from DNA to protein in eukaryotic and prokaryotic cells. The gene expression mechanism in both types of cells is essentially similar (there are some differences [230] which we will, however, not cover). All complex multicellular organisms are eukaryotic (their DNA is stored in the nucleus) and their cells tend to have a considerably higher level of regulatory complexity than single-celled prokaryotic organisms (that have no nucleus) such as bacteria.

A gene consists of a *regulatory region*, which controls when the gene will be activated, and a *coding region*, which specifies the protein that will be produced when the gene is activated as illustrated in Figure 2.2.

In prokaryotes, the regulatory region is generally located directly upstream of the coding region, whereas in eukaryotes elements of the regulatory region may be located at a considerable distance both upstream and downstream from the coding region. A regulatory region contains *binding sites*, i.e., specific sequences (or *motifs*), such as `GATA` for example, where specific TFs can bind to (Figure 2.2). Individual TFs may exercise either positive or negative control on the activation of a gene, increasing or decreasing its rate of *transcription*. When the activation conditions for a given gene are fulfilled, a large molecule called RNA polymerase binds to the TF complex and the DNA in the gene's coding region is unwound. The sequence of nucleotides on the coding strand of the DNA is then used as a template to create a single-stranded messenger RNA (mRNA) molecule [185].

While in prokaryotes, the coding region is contiguous, in eukaryotes, the coding region is broken up into a series of coding exons and noncoding introns, which must be spliced out of the initial RNA transcript. A number of other processing mechanisms are also possible at this stage. In many cases, a single eukaryotic gene can be spliced and edited in multiple

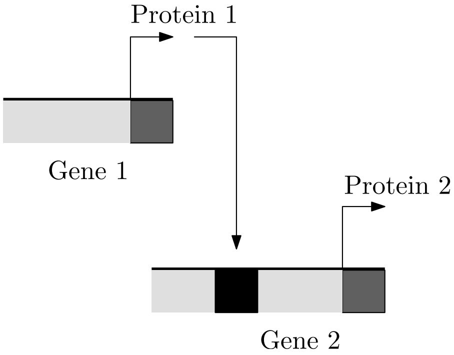


Figure 2.2: The regulatory regions (in light gray) are shown next to the coding regions (in dark gray), which start at the bent arrows. Gene 1 produces Protein 1 which binds to a specific binding site (in black) and induces the activation of Gene 2 which produces Protein 2.

ways to produce a variety of different protein products [167, 209, 220]. As the next step of gene expression, *translation*, occurs in the cytoplasm of the cell, mRNA molecules in eukaryotes must also be transported outside of the cell nucleus.

Once in the cytoplasm, mRNA molecules bind to another large molecule called a *ribosome*. A ribosome reads an mRNA molecule in triplet known as codons. Each codon maps to one of twenty possible amino acids, that are chained together in the order specified by the mRNA. The newly created amino acid chain then folds into a complex three-dimensional protein structure. Whereas DNA is a stable molecule, mRNA and proteins have only limited lifespan before they are broken down and their constituent nucleotides and amino acids are reused. Both mRNA and proteins may be degraded at different rates depending on their conformation and the presence or absence of other chemicals in the cell.

While the most well understood form of regulation occurs at the transcriptional level, the control of *gene expression*, which covers the entire process from transcription through the protein synthesis, may be carried out at almost any stage of protein synthesis. Regulation is also known to occur at the level of RNA processing, mRNA transport and translation, protein modification and mRNA and protein degradation.

The final measure of whether or not a gene is "expressed" is if the protein is produced, because it is the protein that will ultimately carry out the function specified by the gene (Figure 2.3).

Regulation can occur at any point in the pathway shown in Figure 2.4. Specifically, it occurs at the levels of transcription, RNA processing (only for eukaryotes; not shown in Figure 2.4), mRNA lifetime (longevity) and translation [163]. All these levels of gene regulation are important in determining the levels of gene expression. When using DNA microarrays (Section 2.1.2), it is the level of mRNA that is measured, thus only regulation at the level of transcription and (for eukaryotes) of RNA processing is considered.

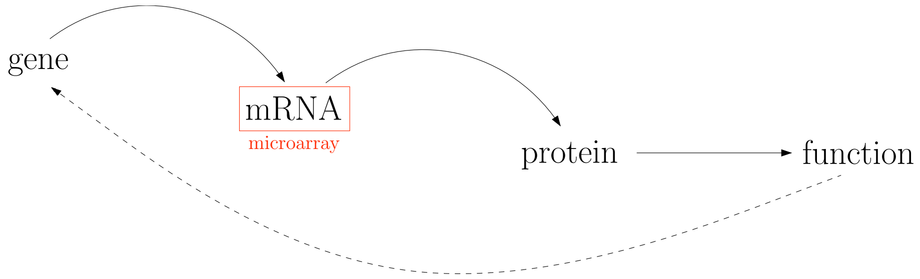


Figure 2.3: Gene activity is partially reflected in mRNA concentration, measured by DNA microarrays (Section 2.1.2).

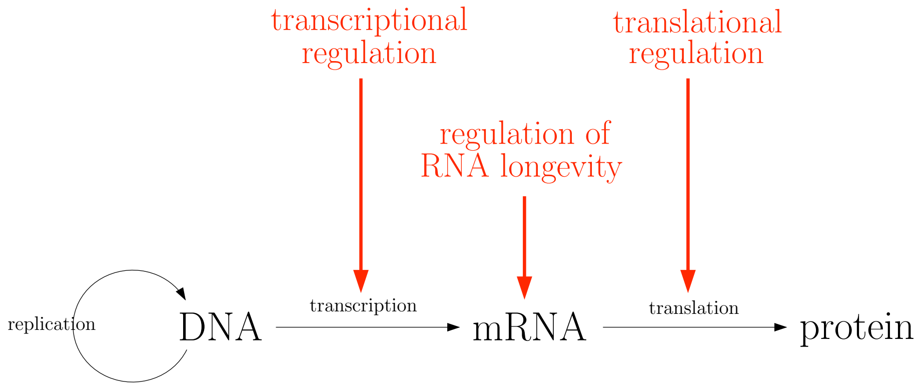


Figure 2.4: Regulation can occur at the levels of transcription, mRNA lifetime and translation.

2.1.2 Gene expression and DNA microarrays

DNA microarrays, simply referred to as microarrays,[2] use *nucleic acid hybridization* techniques to evaluate the mRNA expression profile of thousands of genes within a single experiment. This technology enables to monitor the whole *transcriptome* on a single chip so that researchers can have a picture of the interactions among thousands of genes simultaneously. As such it has attracted tremendous interest among biologists. After genome sequencing, DNA microarray analysis has become the most widely used source of genome-scale data [24].

More specifically, DNA microarrays are solid substrates hosting hundreds of single stranded DNAs with a specific sequence, representing the genes of an organism, which are found on localized features, the *spots*, arranged in grids [213]. These molecules, called *probes*, will hybridize with single stranded DNA molecules, named *targets*, that have been labeled during a *reverse transcription* procedure. The targets reflect the amount of mRNA isolated from a sample obtained under a particular influence factor. Thus, the amount of fluorescence emitted by each spot will be proportional with the amount of mRNA produced from the gene having the corresponding DNA sequence. The microarray is scanned and the resulting image (an example of which is given in Figure 2.5) is analyzed such that the signal from each feature or probe can be quantified into some numerical value which represents the expression level of a given gene in a given condition [65]. These values are typically represented by an $n \times p$ matrix, where n and p represent the number of samples and the number of genes, respectively.

Through the use of highly accurate robotic spotters, over $30,000$ spots can be placed on one slide, allowing molecular biologists to analyze virtually every gene present in a genome [213]. Microarray data sets hence typically describe a large number p of variables (on the order of hundreds or thousands) but only contain comparatively few samples n (on the order of tens or hundreds). This "small n, large p" data setting renders learning tasks in molecular biology even more challenging.

The emergence of DNA microarrays is largely due to the necessity to understand the networks of bio-molecular interactions at a global scale [65]. Indeed, it is widely believed that genes and their products (proteins) are processed in complex networks (Section 2.3). With the sequencing of the genomes of many organisms, the need for a quick snapshot of all or a large set of genes was thus pressing. A traditional approach in molecular biology was to use some method to render a gene inactive (knock out) and then study the effects of this knock out in other genes and processes in a given organism. Unfortunately, this approach enabled to study only few genes at a time, and was thus slow, expensive, and inefficient for a large scale screening of many genes.[3]

The main drawback of DNA microarrays is that they tend to be very noisy (sometimes,

[2] There is no ambiguity in using this term: although other types of microarrays exist, such as protein or tissue microarrays for example, we will only consider DNA microarrays.

[3] Although microarrays are invaluable as screening tools able to interrogate simultaneously thousands of genes, gene knockouts are still crucial for a focused research once interesting genes have been located. Indeed, knocking out a gene allows the study of the more complex effects of the gene, well beyond the mRNA abundance level [65].

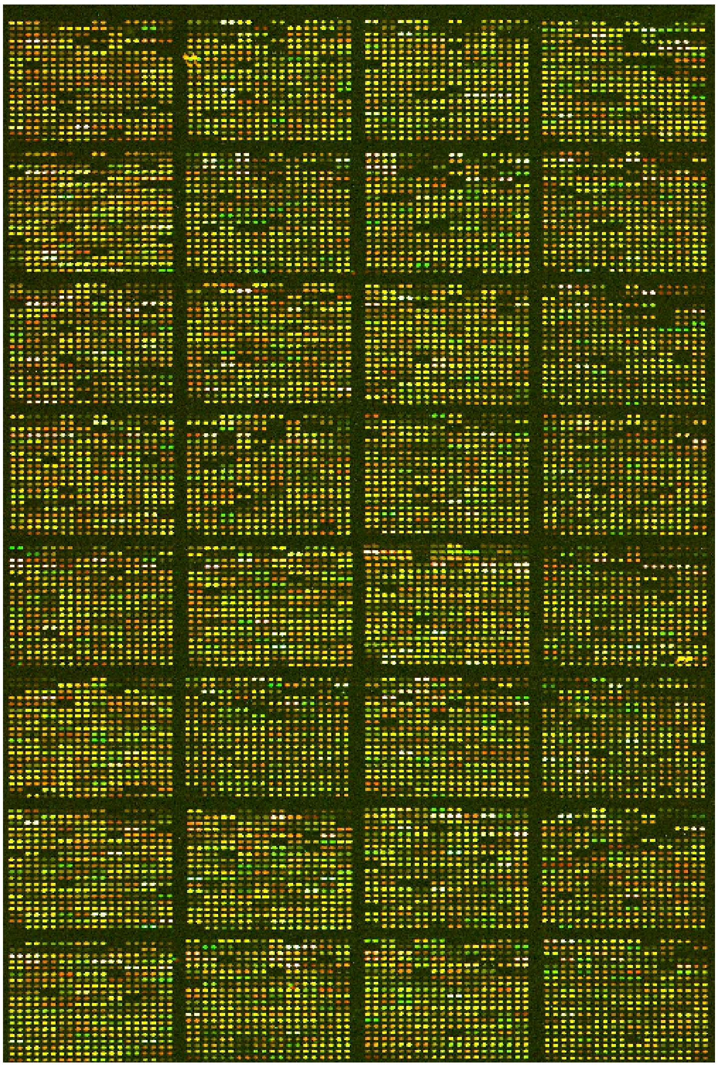


Figure 2.5: Image (reproduced from Godard [105]) resulting from the scanning of a microarray.

some values might even not be available) [65], despite the development of several tools such as statistical experimental design and data normalization to obtain high quality results [225, 269]. This noise is introduced at various steps of their production, such as sample preparation, RNA amplification/purification, chip hybridization and scanning [217].

2.1.3 The control tasks of the genome

The genome is responsible for controlling cellular tasks such as response to environmental conditions, the cell division cycle and cell differentiation. Each of these requires the regulation of gene expression in both space and time.

Throughout its lifetime, a cell must respond to many different types of environmental signals, an example of which is given by *nitrogen catabolite repression* (NCR) in the yeast *Saccharomyces cerevisiae* (Section 2.2). Single-celled bacteria are able to detect and move towards nutrient sources, they also react to changes in temperature and acidity. Multicellular cells must also respond to chemical signals emitted by neighboring cells in the organism. These external signals are transmitted to the genome via a series of chemical reactions known as *signal transduction pathways.*

As well as responding to external signals, the genome is also subject to internal control. The cell cycle plays the role of a cell's internal clock. In order for an organism to develop, each embryogenic cell goes through a process of growth, replication and division. After its growth, its entire genome is replicated to produce two identical copies. When the cell divides, each of its daughter cells contains one complete copy of the genome. The signals that tell a cell when to switch from growth to replication and from replication to division are controlled by a subset of genes that regulate timing.

Each cell of a multicellular organism contains identical genetic information (with some rare exceptions). The feature that distinguishes cells of different types is the set of genes that are active in a particular cell. This pattern activation determines which proteins are produced, and hence the functional properties of the cell. When an egg cell is initially fertilized, it is fully undifferentiated and has the potential to become any type of cell. As an organisms developmental program unfolds, its cells divide and undergo physical and chemical changes that result in their final state (for example, as blood or bone cells) becoming more differentiated. The role of the *gene regulatory network* in this process is to integrate the internal dynamics of the cell and external signals from the environment and other cells to control the differentiation process.

2.2 Nitrogen catabolite repression in the yeast *Saccharomyces cerevisiae*

The yeast *Saccharomyces cerevisae* (*S. cerevisae*) is a relative simple unicellular organisms for which the entire genomic sequence and the functional roles of approximately 60% of the genes are known [36, 37, 77]. Many results have already been obtained, in particular concerning its different cellular states [226] and its growth conditions [99, 183]. Therefore, it has been widely used in genomics as a *model organism* (another example of model organism is *Escherichia coli*; Section 5.7). The study of such organisms is indispensable for the understanding of more complex ones [84].

In particular, we focus on an important biological process known as *nitrogen catabolite repression* (NCR), which involves an essential nutrient for all life forms: *nitrogen.* The emergence of cells able to transport, catabolize and synthesize a wide variety of nitrogenous compounds has thus been favored by evolutionary selective pressure [107]. As a consequence, *S. cerevisiae* can use 27 distinct nitrogen-containing compounds, including amino acids, urea, ammonium, nitrogen bases, and purine derivatives [107].

Like most unicellular organisms, yeast transports and catabolizes good nitrogen sources (e.g., ammonium, glutamine, and asparagine) which support rapid growth, i.e., generation time (defined as the time required for a cell to complete one full growth cycle) of approximatively 2 hours, in preference to poor ones (e.g., isoleucine, methionine and threonine) which support slow growth, i.e., generation time larger than 3 hours [106, 107]. NCR refers to this selection mechanism [106, 107, 214]. It consists in the specific inhibition of transcriptional activation of genes encoding the permeases and catabolic enzymes needed to degrade poor nitrogen sources [214], as illustrated in Figure 2.6.

More specifically, “NCR acts through the inhibition of two transcription factors of the GATA family (Gln3 and Gat1/Nil1) which typically bind to upstream 5’-GATA-3’ core sequences and activate gene transcription [. . .]. The Gln3 and Gat1 factors are thus most active under limiting nitrogen supply conditions (e.g., when cells grow on poor nitrogen sources like urea and proline) and [. . .] upon the addition of rapamycin to nitrogen-rich media” [107].

“Rapamycin inhibits the Tor proteins, which are proposed to govern the inhibition of Gln3 and Gat1 under good nitrogen supply conditions. The Tor-dependent inhibition of Gln3 involves the Ure2 protein, whereas the repression of Gat1-dependent expression under good nitrogen supply conditions is also dependent on Gzf3/Deh1/Nil2, another GATA family transcription factor” [107].

Finally, a “fourth GATA factor encoded by the DAL80/UGA43 gene also acts as an inhibitor of Gat1 [. . .] under poor nitrogen supply conditions. Transcription of the GAT1, GZF3, and DAL80 genes is under the control of all four GATA factors.” [107].

These four key transcriptional regulators of NCR target genes are linked through a network of auto- and cross-regulations [107].

The ultimate goal is to identify all genes involved in NCR. “Several studies have focused on identifying in the complete yeast genome the genes subject to NCR or regulated by the GATA factors” [107]. In particular, the challenge of inferring putative NCR genes has been tackled by three experimental studies: Bar-Joseph et al. [11], Godard et al. [107], Scherens et al. [214]. Among these, Godard et al. [107] also proposed a bioinformatics approach (Section 7.1).

2.3 Gene regulatory networks

The data flood phenomenon biology is experiencing has propelled biologists toward the view that biological systems are fundamentally composed of two types of information: genes, encoding the molecular machines that execute the functions of life, and networks of regulatory interactions, specifying how genes are expressed [126].

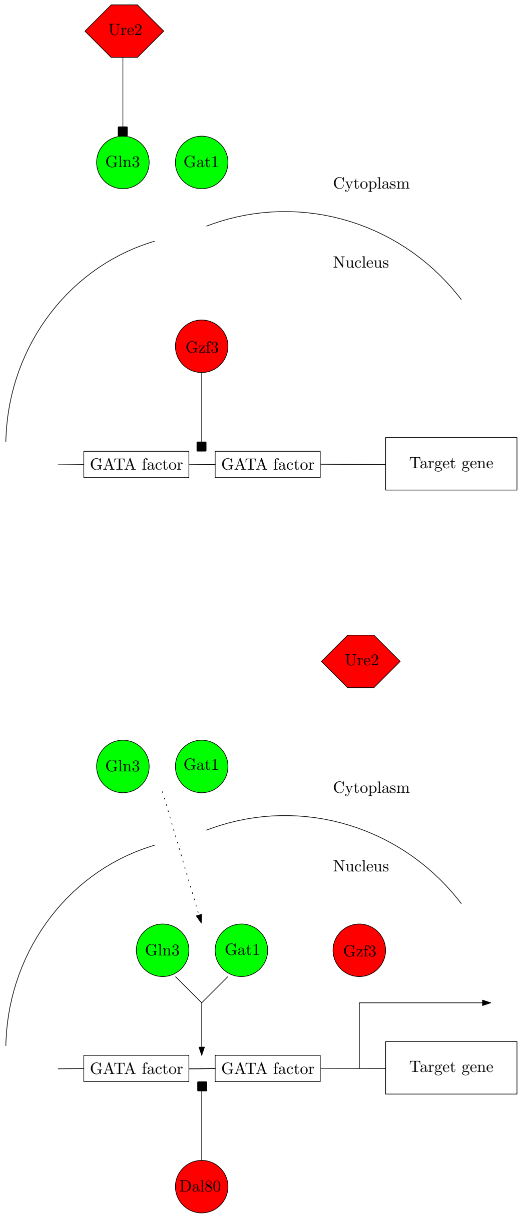


Figure 2.6: Illustration of nitrogen catabolite repression (NCR) reproduced from Godard [106]. Circles represent transcription factors and the hexagon represents the regulatory protein Ure2. The elements that activate the transcription of NCR target genes are shown in green and those that repress it are shown in red. See text for details.

The development of high-throughput data-collection techniques, as epitomized by the widespread use of DNA microarrays (Section 2.1.2), allows for the simultaneous interrogation of the status of a cell's components at any given time. Various types of interaction networks, including protein-protein interaction, metabolic, signaling and transcriptional regulatory networks, emerge from the sum of these interactions. None of these networks are independent, instead they form a "network of networks" that is responsible for the behavior of the cell [12]. Consequently, biological information has the following two important features: it operates on multiple hierarchical levels and it is processed in complex networks [126]. These information networks are typically *sparse* and robust, such that many single perturbations will not greatly affect them. However, there are key nodes or hubs in these networks where perturbation may have profound effects, which represent powerful targets for the understanding and manipulation of the system.

Two of the most important challenges in systems biology are the extent to which it is possible to model these genetic interactions as large networks of interacting elements and the way that these interactions can be effectively learned from measured expression data [252].

The inference or reverse engineering of genetic networks from expression data alone is far from being trivial because of the combinatorial nature of the problem and the poor information content of the data [252]. Indeed, gene expression levels are regulated by the combined action of multiple gene products [23, 122] and the number n of measurements (arrays) is relatively small compared to the number p of measured objects (genes). This "small n, large p" data setting induces the so-called *curse of dimensionality* (Section 3.3). Furthermore, the data is corrupted by a substantial amount of measurement noise (Section 2.1.2).

To cope with the combinatorial nature of the problem, simplifying hypotheses are made. Notably, we focus on *gene regulatory networks* (GRNs)–also referred to as *transcriptional regulatory networks* in the literature. These networks constitute a simplification of the complete cellular system, given that they are represented as if genes directly affect each other.

Indeed, networks of interactions between molecules can be constructed at various levels and can represent different types of interactions. Several biochemical networks have traditionally been considered [23]:

- *metabolic networks* that represent the chemical transformations between metabolites;
- *protein networks* that represent protein-protein interactions, such as formation of complexes and protein modification by signaling enzymes (also known as signaling networks);
- *gene regulatory networks* that represent relationships that can be established between genes, when observing how the expression level of each one affects the expression level of the others.

Of course, each of these networks is a simplification of the complete cellular system, which is referred to as the *global biochemical network* to emphasize that it explicitly includes all three types of molecule, i.e., metabolites, proteins and mRNA.

Hence, networks that are represented as if genes directly affect each other are phenomenological because they do not explicitly represent the proteins and metabolites that mediate cell interactions. However, they are a logical way of describing phenomena observed with transcription profiling, such as those that occur with DNA microarrays [23]. When exclusively monitoring gene expression to study some phenomenon, one is limited to constructing such a gene network to explain the data [23].

A model of a global biochemical network in which the three levels are shown explicitly as planes is illustrated in Figure 2.7. In any global biochemical network, genes do not interact directly with other genes (neither do the corresponding mRNAs); instead, gene induction or repression occurs through the action of specific proteins, which are, in turn, products of certain genes.

Gene expression can also be affected directly by metabolites, or through protein-metabolite complexes. However, it is often useful to abstract these actions of proteins and metabolites, and represent genes acting on other genes in a gene network (also called genetic regulatory, transcription or expression networks). This simplification of going from the global biochemical network to a gene network is akin to a projection of all interactions to the "gene space" (Figures 2.7 and 2.8).

The idea that genes dictate all that goes on inside a cell, materialized in the central dogma of molecular biology (Figure 2.1), which emphasizes that proteins, and consequently metabolites, are only synthesized when genes are activated, fails to acknowledge that gene expression is also influenced by the levels of protein and metabolite. It is now well established that regulation is distributed over all levels, and accordingly such systems are referred to as democratic, contrary to systems in which there is no feedback from proteins or metabolites to genes that are called dictatorial [265], but are currently only used conceptually. Although this indicates that future studies need to make more effort to monitor all three levels of regulation [169], it is still useful to study gene networks alone. The ability to create gene networks from experimental data and use them to reason about their dynamics and design principles will increase our understanding of cellular function [23].

The inference of GRNs from microarray data thus makes two simplifying assumptions: the protein synthesis depends directly on the amount of mRNA (Section 2.1) and genes directly affect each other.

2.4 Reverse engineering gene regulatory networks

Inferring GRNs from expression data is one of the most important and challenging tasks in bioinformatics (Section 2.3). Not surprisingly, a plethora of reverse engineering approaches have been proposed to model GRNs, as epitomized by the numerous literature reviews that have been published [50, 68, 79, 85, 93, 98, 100, 117, 169, 171, 221, 252], the first ones of which appeared in the 1960's and 1970's [104, 201, 237] (see also the extensive bibliography by Markowetz [168]). Although the computational study of gene regulation is a subject that already has a long history, the number of papers on the topic published in the last few years seems to be growing exponentially.

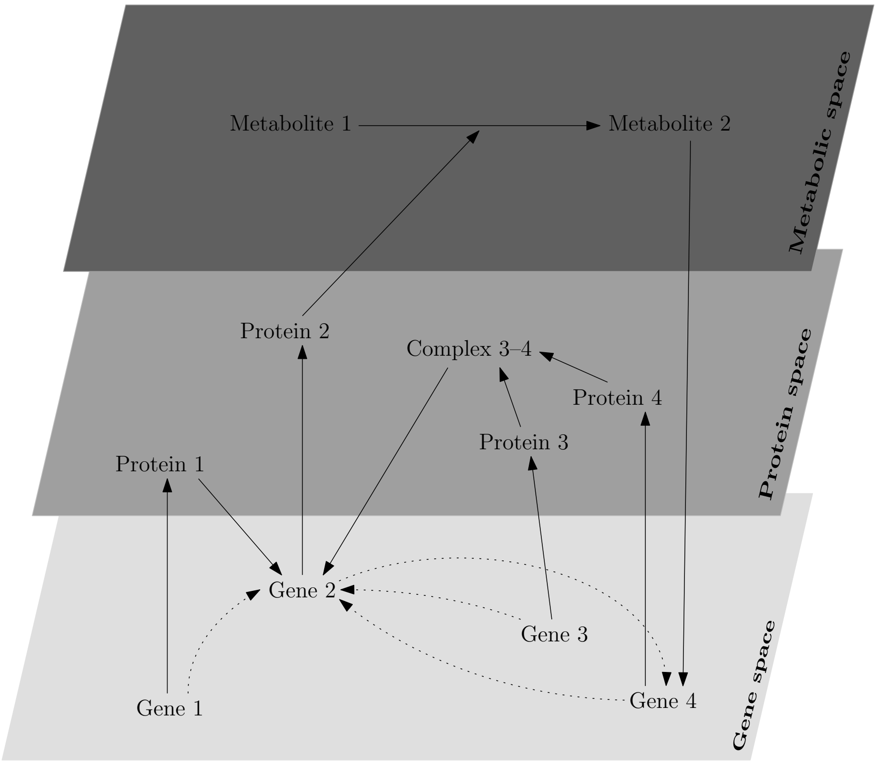


Figure 2.7: A hypothetical biochemical network (redrawn from Brazhnik et al. [23]). Molecular constituents (nodes of the network) are organized in three levels (spaces): mRNAs, proteins, and metabolites. Solid arrows indicate interactions, the signs of which (activation or repression) are not specified in this diagram. Three different mechanisms of gene-gene interactions are shown: regulation of gene 2 by the protein product of the gene 1; regulation of the gene 2 by the complex 3-4 formed by the products of gene 3 and gene 4; and regulation of gene 4 by the metabolite 2, which in turn is produced by protein 2. Projections of these interactions into the gene space, indicated by dashed lines, constitute a corresponding gene network.

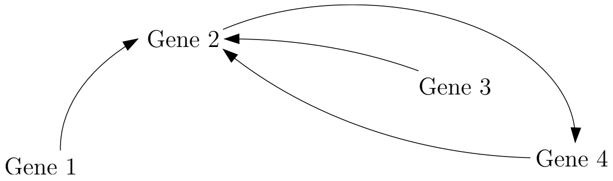


Figure 2.8: The genetic regulatory network (redrawn from Brazhnik et al. [23]) resulting from the biochemical network depicted in Figure 2.7.

For obvious reasons, it is out of the thesis's scope to review all existing methods (if at all possible). Instead, we introduce the general properties of existing modeling formalisms (Section 2.4.1) and we review the most important models: Boolean and generalized logical networks (Section 2.4.2), ordinary differential equations (Section 2.4.3), and graphical models (Section 2.4.4). This last family of models, and in particular the Gaussian graphical model (GGM) that we use throughout the thesis, will be studied in greater details in Chapter 4.

The other existing modeling formalisms, e.g., neural networks [54, 252] and additive regulation models [33, 55, 56, 179, 260], have had a more limited impact.

2.4.1 General properties of modeling formalisms

Despite the large number of existing approaches to reverse engineer GRNs, it is not clear what the advantages and disadvantages of each of the different approaches are and how they can be compared. There exists no common "hierarchy" in the literature for the numerous proposed formalisms (for example the reviews by de Jong [50], Dutilh and Hogeweg [68], Geard [100], Smolen et al. [221], van Someren et al. [252]). The hierarchies proposed in the literature usually depend on the modeling aspects their authors want to emphasize (see Kontos [141] for further details).

This is not really surprising since many aspects, which we now review, characterize the different existing formalisms [85, 252]:

- *Physical vs. combinatorial models*: Physical models, such as those based on differential equations, describe the quantitative relationships between the state variables in the system. Such models can be used to run simulations and predict the future behavior of the system. Unfortunately, they lack inference methods and any higher level organization is very difficult to obtain from the equations. Furthermore, the large number of parameters that need to be fitted requires many experiments to fit them to the data. On the other hand, combinatorial models focus on higher level of modeling and are most often qualitative. These models are typically represented as a graph of nodes and edges between them from which many important high-level questions can be readily answered (Section 2.4.4);
- *Static vs. dynamic models*: A principal difference between models is whether static or dynamic relations are modeled. Dynamic models assume that the gene expression levels at past time instants determine the current (changes in) gene expression levels. Dynamic models generally define a parametric model of interactions and try to estimate the parameters from time course gene expression data. Thus, dynamic models depict dependencies between microarray measurements taken at different time instances. Static models search for causal interactions within microarray measurements that are consistent across multiple microarrays. A nice feature of static models is that they can be applied to time course gene expression data as well as to static data;
- *Discrete vs. continuous models*: Gene expression measurements are continuous values that represent the relative amount of mRNA copies in a biological sample. Gene expression levels in the model can thus be represented by continuous values or by

discrete values when the data are quantized into a suitable number of discrete levels;

- *Deterministic vs. stochastic models*: A deterministic model always predicts the same outcome when the initial conditions are the same. A stochastic model models the probability distribution of possible outcomes. Both methods can be used depending on whether one wants to explicitly model the uncertainty or not;
- *Complexity of dependency relationships*: Especially for continuous models, the functional form of the interaction between genes provides a natural way to restrict the complexity of the model. A common choice is to restrict the model to allow only linear relationships (as is the case with Gaussian graphical models; see Section 2.4.4). Linear relationships may greatly simplify parameter estimations and in many cases allow analytical (closed form) solutions. Furthermore, the parameters are relatively easy to understand. However, the linear representation may limit the expressive power of the model;
- *Number of inputs*: A distinction between pairwise models and combinatorial models can be made. Pairwise models determine relationships between pairs of genes and thus only consider single-gene influences. Combinatorial models allow the combined effect of multiple genes to influence a target gene.

2.4.2 **Boolean and generalized logical networks**

One of the earliest approaches to modeling GRNs was to view them as networks of logical elements, known as *Boolean networks* [1–3, 125, 127, 134–137, 162, 165, 182, 223, 232]. This approach makes three simplifying assumptions [223]. First, it assumes that the state of a single gene can be represented by a Boolean variable expressing that it is either expressed or not. Second, interactions between elements are represented by Boolean functions which calculate the state of a gene from the activation of other genes. Transitions between states are therefore deterministic with a single output state for a given input. Third, timing is synchronous: at each time step the states of all genes are updated simultaneously (i.e., in parallel) by applying the Boolean function of each element to its inputs.

The main strengths of Boolean networks are their analytical tractability and the ease and efficiency with which they can be simulated. However, their validity to model GRNs has been questioned due to their perceived lack of applicability to biological systems [79] and the validity of the Boolean assumption (many examples exist where genes are regulated in a continuous manner [21, 22, 46, 221]) and the synchronicity assumption.

Generalized logic networks [236–239, 241] try to cope with the shortcomings of Boolean networks by allowing state variables to assume more than two levels [238, 249], by enabling more sophisticated forms of logical updating (such as weighted gene interactions [222]) and by allowing for asynchronous updating of elements [238].

The generalized logic formalism is a powerful method for analyzing limited-scale networks whose interactions are well known [176, 177, 207, 208, 234, 239, 240]. Unfortunately, the method's scalability is limited. It has been designed for the detailed analysis of relatively small systems consisting of well characterized interactions. It is therefore less suited to the exploration of large and less well-known systems.

2.4.3 Ordinary differential equations

Ordinary differential equations (ODEs) [7] are arguably the most widespread formalism to model dynamical systems in science and engineering. As such, they have been widely used to analyze GRNs [50]. ODEs model gene regulation by rate equations expressing the rate of production of a component of the system as a function of the concentrations of other components [50]:

$$\frac{d\mathbf{x}_i}{dt} = f_i\left(\mathbf{x}\right) \ , \quad i \in \{1, \ldots, p\} \ , \tag{2.1}$$

where $\mathbf{x} = \left(\mathbf{x}_1, \ldots, \mathbf{x}_p\right)^T \geq 0$ is the vector of gene product concentrations and $f_i : \mathbb{R}^p \longrightarrow \mathbb{R}$ are update functions.

ODEs provide an accurate representation of the physical system under investigation thanks to the detailed representation of regulatory interactions. Unfortunately, analytical solution of the rate equation (2.1) is normally not possible because the update functions are usually nonlinear. Therefore, one has to simplify the model (e.g., by using piecewise-linear differential equations [50]) or take recourse to numerical simulations [149, 175]. Nevertheless, this can be computationally very demanding for realistically sized systems.

2.4.4 Graphical models

Graphical models are representations of multivariate probabilistic models in which conditional independence (Section 4.1) constraints are specified by graphs (Sections 4.4 and 4.5). The vertices of the graph represent the variables, i.e., the genes (more specifically, a particular characteristic of the gene, e.g., its expression level). They are composed of *undirected graphical models* (Section 2.4.4.1), also known as *Markov random fields*, in which the links have no directional significance, and *directed graphical models* (Section 2.4.4.2), also known as *Bayesian networks*, in which the links have a particular directionality (indicated by arrows). Undirected and directed models mainly differ in how the independence relationships between the variables implied by the edges are read off the graph.

These models have gained much attention as they encode full conditional relationships between variables, i.e., genes. Hence, they enable to distinguish between direct and indirect interactions. Moreover, the availability of effective methods for their inference in the "small n, large p" setting contributes to their success as epitomized by the growing literature devoted to them.

Although they have the disadvantage of leaving dynamical aspects of gene regulation implicit, these models are becoming increasingly important as recent studies indicate that "the topology of networks is a determining factor in both re-engineering the network as well as understanding network and organism evolution" [85].

2.4.4.1 Gaussian graphical models

Gaussian graphical models (GGMs) are undirected graphical models that assume multivariate normality of the data. The (in)dependence relationships between variables can hence be inferred through partial correlations which are intimately linked to linear regression and to the corresponding problem of estimating the covariance matrix.

On observational data (so-called passive observations), GGMs' performance is similar to Bayesian networks', albeit at a significantly smaller computational cost [264]. The higher computational costs of inference with Bayesian networks over GGMs is only justified with active interventions (such as gene knockouts) when Bayesian networks typically outperform GGMs [264]. However, as we will only deal with observational data, we focus on GGMs in the present thesis (Chapter 4). More specifically, we use GGMs for the inference of GRNs (Chapters 5 and 6) and for the prediction of NCR target genes (Chapter 8).

2.4.4.2 **Bayesian networks**

Learning Bayesian networks from data is typically cast as an optimization problem, where the computational task is to find a structure that maximizes a statistically motivated score [118]. As this optimization problem is NP-hard [35], finding the optimal Bayesian network is only possible for networks that contain only a few tens of genes [186]. Most existing learning tools therefore address this problem using standard heuristic search techniques which are, however, not guaranteed to lead to a globally optimal solution [95, 96, 128, 190, 218]. As mentioned above, these methods are much more computationally demanding than GGMs.

As an additional problem, currently available expression data underdetermine the network, since at best a few hundreds of experiments provide information on the transcription level of thousands of genes. To tackle this dimensionality problem, several alternatives have been proposed in the literature [95, 96, 128, 190, 218]. Among these, Friedman and Pe'er [96] introduced an iterative algorithm that achieves faster learning by restricting the search space (where dependence is measured by mutual information). Another successful approach was proposed by Friedman et al. [95] (see also Pe'er et al. [190]) who presented a heuristic algorithm that focuses on features that are common to high-scoring networks instead of looking for a single network.

A Bayesian network approach toward modeling regulatory networks is attractive because of its solid basis in statistics, which enables it to deal with the stochastic aspects of gene expression and noisy measurements in a natural way. Moreover, Bayesian networks can be used when only incomplete knowledge about the system is available.

3

Overview of Supervised Learning

Throughout the thesis, we use concepts related to supervised learning which we introduce in Section 3.1. In particular, we will use the resampling method of cross-validation for model validation (Section 3.1.2). Other resampling methods include the bootstrap and the jackknife (Section 3.2) which will be used for parametric bias estimation (Chapter 5) and for nonparametric variance estimation (Chapter 8), respectively. The "small n, large p" data setting induces the so-called "curse of dimensionality" (Section 3.3) which will be dealt with by resorting to variable selection (Section 3.4) and regularization (Section 3.5). Subsequently, we introduce the linear regression model (Section 3.6) which is a particular supervised learning machine intimately linked to the Gaussian graphical model (Chapter 4) used throughout the thesis. Thereafter, we review some basic facts on classification (Section 3.7) that will be useful when we introduce the two-class classification approach to NCR target gene prediction (Chapter 7).

3.1 Supervised learning

Supervised learning [116, 178, 253] denotes the set of techniques for building a model of dependency between a set of input variables and a set of output variables from a training data set. This training set consists of pairs of *inputs* and *outputs*. The goal of the supervised learner is to predict the value of the output for any valid input after having seen only a finite number of training examples. To achieve this, the learner has to generalize from the training data to unseen situations.

According to the type of output, one can distinguish between two types of prediction tasks: *regression* where quantitative outputs (e.g., real or integer numbers) are predicted, and *classification* (or *pattern recognition*) where qualitative (or categorical) outputs are predicted. Qualitative outputs assume values in a finite set of classes where there is no explicit ordering.

In statistical terms, a supervised learning problem (Figure 3.1) can be described by the following elements [253, 254]:

- a data *generator* of input vectors $\mathbf{x} \in \mathscr{X} \subseteq \mathbb{R}^p$ independent and identically distributed (i.i.d.) according to some unknown (but fixed) probability distribution function[1] $F_{\mathbf{x}}(x)$ (probabilistic mapping);

[1] Appendix D reviews some basic probability concepts.

- a *target* operator which transforms the input $\mathbf{x}$ into the output value $\mathbf{y} \in \mathscr{Y}$ according to some unknown (but fixed) conditional distribution $F_{\mathbf{y}|\mathbf{x}}(y \mid x)$. In regression we typically have that $\mathscr{Y} \subseteq \mathbb{R}$, while in classification $\mathscr{Y} = \{c_1, \ldots, c_K\}$, where c_k, $k = 1, \ldots, K$, are the class labels and K is the (finite) number of classes;
- a *training set* $D_n = \{(x_{i\cdot}, y_i), i = 1, \ldots, n\} \in (\mathscr{X} \times \mathscr{Y})^n$ consisting of n pairs $(x_{i\cdot}, y_i) \in \mathscr{X} \times \mathscr{Y}$ i.i.d. according to the joint distribution function $F_{\mathbf{y}|\mathbf{x}}(y \mid x) F_{\mathbf{x}}(x)$ (note that the observed training set is considered as the realization of the random variable $\mathbf{D_n}$);
- a *learning machine* which, on the basis of the training set, returns a predictor of the target, called *hypothesis* or *model* (Section 3.1.1).

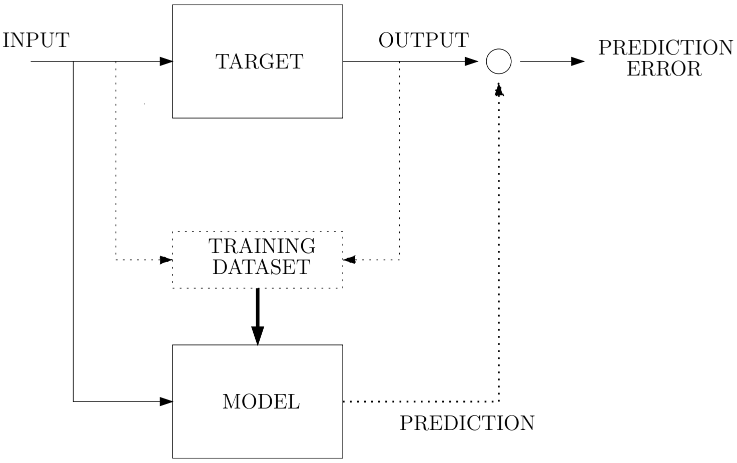


Figure 3.1: Supervised learning setting [19].

3.1.1 Learning machine

A learning machine has three components:

1. A class of *hypothesis* functions $h(\cdot, \alpha)$ where $\alpha \in \Lambda$ is a vector of parameters and Λ is the parameter space. We only consider single valued mappings for these functions.
2. A *cost* function $C(\cdot, \cdot)$ which, given a particular pair $(h(x), h'(x))$, measures the discrepancy $C(h(x), h'(x))$ between the output of $h(\cdot)$ and $h'(\cdot)$ given x. In regression, the cost function is usually quadratic:

$$C\big(h(x), h'(x)\big) = \big(h(x) - h'(x)\big)^2 , \tag{3.1}$$

while in classification one typically considers zero-one loss (3.12);

3. An *algorithm* of parametric identification which takes as input the training set D_n and returns as output a hypothesis function $h\left(\cdot,\alpha_n\right)$ with $\alpha_n \in \Lambda$. We only consider deterministic and symmetric algorithms. This means that the algorithms always return the same $h\left(\cdot,\alpha_n\right)$ for the same data set D_n and that they are insensitive to the ordering of the examples in D_n, respectively. The parametric identification of the hypothesis is typically performed according to the *empirical risk minimization* (ERM) principle [253, 254] where

$$\alpha_n = \alpha\left(D_n\right) = \arg\min_{\alpha\in\Lambda} R_{\text{emp}}\left(\alpha\right)$$

minimizes the *empirical risk*

$$R_{\text{emp}}\left(\alpha\right) = \frac{1}{n}\sum_{i=1}^{n} C\left(y_i, h\left(x_{i\cdot},\alpha\right)\right) \tag{3.2}$$

constructed on the basis of the data set D_n. Note that some learning machines use a constrained version of the ERM procedure for parametric identification, such as support vector machines (Section 3.7.2.4) for example.

The goal of a learning machine is to return a hypothesis with low *prediction error*, i.e., a hypothesis which computes an accurate estimate of the output of the target when the same value is an input to the target and the predictor. The prediction error is usually called *generalization error* since it measures the capacity of the hypothesis to generalize, that is to return a good prediction of the output for input values not contained in the training set.

A typical way of representing the unknown input/output relation is the *regression plus noise form*,

$$\mathbf{y} = o\left(x\right) + \boldsymbol{\epsilon}\ , \tag{3.3}$$

where $o\left(\cdot\right) : \mathscr{X} \to \mathscr{Y}$ is a deterministic function, also known as the *regression function*, and the term $\boldsymbol{\epsilon}$ represents the noise or random error. It is typically assumed that $\boldsymbol{\epsilon}$ is independent of $\mathbf{x}$ and $\mathbb{E}\left(\boldsymbol{\epsilon}\right) = 0$.

Hence, the learning machine aims at finding a model $h\left(x,\cdot\right)$ which is able to give a good approximation, i.e., having low generalization error, of the unknown function $o\left(x\right)$.

Suppose that a learning algorithm is available, that given a data set D_n, returns the set of parameters α_n of the model $h\left(x,\alpha_n\right)$. Recall that D_n is the realization of the random vector $\mathbf{D_n}$. For a given x, the *mean squared error* (MSE) is defined as the quadratic cost (3.1) averaged over the ensemble of training sets with n samples for a given input value x:

$$\begin{aligned}
\text{MSE}\left(x\right) &= \mathbb{E}_{\mathbf{D_n},\mathbf{y}}\left(C\left(\mathbf{y},\alpha\left(\mathbf{D_n}\right)\right) \mid x\right) \\
&= \mathbb{E}_{\mathbf{D_n},\mathbf{y}}\left(\left(\mathbf{y} - h\left(x,\alpha\left(\mathbf{D_n}\right)\right)\right)^2 |x\right) \\
&= \int_{\mathscr{X}^n\times\mathscr{Y}^n}\int_{\mathscr{Y}} \left(y - h\left(x,\alpha\left(D_n\right)\right)\right)^2 dF_{\mathbf{y}|\mathbf{x}}\left(y \mid x\right) dF_{\mathbf{D_n}}\left(D_n\right)\ ,
\end{aligned}$$

where y and D_n denote the realizations of the random variables $\mathbf{y}$ and $\mathbf{D_n}$, respectively. Since we are interested in the accuracy on the whole domain $\mathscr{X}$ (and not only on a specific point x), we consider the *mean integrated squared error* (MISE):

$$\begin{aligned}\text{MISE}\left(\mathbf{x}\right) &= \mathbb{E}_{\mathbf{x}}\left(\text{MSE}\left(\mathbf{x}\right)\right) \\ &= \int_{\mathscr{X}} \text{MSE}\left(x\right) dF_{\mathbf{x}}(x) \ .\end{aligned}$$

The MISE can be computed analytically only if the distribution generating the data is known. Since this is not the case, an estimate of the MISE needs to be returned.

3.1.2 Validation techniques

The empirical risk (3.2) is arguably the most obvious estimate of the MISE. However, it is well known that the empirical risk is a biased estimate of the MISE and that it tends to be smaller than the MISE, because the same data have been used both to construct and to evaluate $h\left(\cdot, \alpha_n\right)$ [116].

A way to obtain unbiased estimates of the MISE without making assumptions on the distribution underlying the data is to use *resampling* methods. Here, we consider *cross-validation* [116]. The basic idea of cross-validation is that ones builds a model from one part of the data and then uses that model to predict the rest of the data.

The training set D_n is divided into k mutually exclusive test partitions of approximately equal size (referred to as *k-fold cross-validation*). The samples not found in each test partition are independently used for selecting the hypothesis which will be tested on the partition itself. The average error over the k partitions is called the *cross-validated error rate.* In classification, the folds are sometimes stratified so that they contain approximately the same proportions of labels as the original data set. This is referred to as *stratified cross-validation.*

The k-fold cross-validation algorithm where k equals n is known as *leave-one-out (loo) cross validation.* This means that for the i-th sample (x_i, y_i), $i = 1, \ldots, n$, of the training set D_n, the parametric identification is carried out leaving that observation out of the training set, and the predicted value for the i-th observation, denoted by y_i^{-i}, is computed.

The corresponding estimate of the MISE prediction error is:

$$\begin{aligned}\widehat{\textbf{MISE}}_{\text{loo}} &= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{-i}\right)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h\left(x_{i\cdot}, \alpha^{-i}\right)\right)^2 \ ,\end{aligned}$$

where α^{-i} is the set of parameters returned by the parametric identification performed on the training set with the i-th sample set aside.

3.1.3 Model selection and the bias-variance tradeoff

Models have different levels of complexity. For instance, a quadratic polynomial with two parameters is less powerful than a $10,000$-dimensional linear classifier. Model selection is a step of the statistical modeling procedure which consists in selecting the best model given a data sample [116]. This is typically performed by minimizing the MSE [116].

Since the MSE can be decomposed in bias and variance terms [101] (Appendix E), model selection hence consists in selecting a model whose bias-variance tradeoff is optimal in the sense that it minimizes the MSE. This optimal tradeoff is achieved by choosing the optimal level of complexity of the model, as illustrated in Figure 3.2.

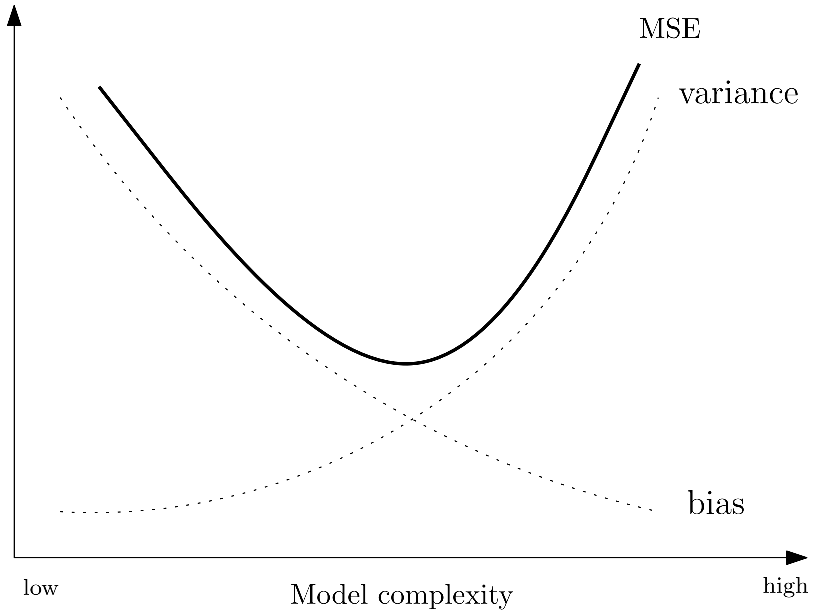


Figure 3.2: Schematic illustration of the bias-variance tradeoff (details in text).

Of course, the appropriate level of complexity depends on the true underlying function's complexity but also on the sample size. The smaller the latter becomes relative to the number of variables, the more important becomes the variance term. Simpler models can then achieve better predictive performance (Section 3.5) than more complex ones.

Model selection algorithms thus have a crucial role and "in interplay with subject-matter considerations [...] they may make a useful contribution to many analyses" [70]. However, "it is essential to regard model selection techniques as explorative tools rather than as truth-algorithms" [70].

3.2 The bootstrap and the jackknife

Resampling methods, which can be used for model validation (Section 3.1.2), can also be used to estimate the bias and variance of an estimator. In this thesis, we will use the bootstrap (Section 3.2.1) to estimate the bias (Section 5) and the jackknife (Section 3.2.2) to estimate standard errors (Chapter 8).

3.2.1 Bootstrap and bootstrap aggregating

The *bootstrap* is a method to estimate the distribution of an estimator [34, 76, 259]. From this distribution, several quantities of interest can be derived such as the estimator's variance and bias.

In the *nonparametric* case, one first draws a sample from the empirical distribution $\hat{F}_n$. Actually, since $\hat{F}_n$ gives probability $1/n$ to each data point, drawing n points at random from $\hat{F}_n$ is equivalent to drawing a sample of size n with replacement from the original data [259]. Second, the estimator of interest $\hat{\boldsymbol{\alpha}}_b$ is computed on this sample. This procedures is then repeated B times. Finally, the bias can be estimated as follows:

$$\hat{\boldsymbol{\alpha}} - \frac{1}{B}\sum_{b=1}^{B}\hat{\boldsymbol{\alpha}}_b ,$$

where $\hat{\boldsymbol{\alpha}}$ is the estimator computed on the original sample.

Of course, there is also a *parametric bootstrap.* If the (parametric) distribution F_θ depends on a parameter θ, we can simply sample from the parametric distribution $F_{\hat{\boldsymbol{\theta}}}$ where $\hat{\boldsymbol{\theta}}$ is an estimate of θ, instead of using the empirical distribution function $\hat{F}_n$.

Bootstrap aggregating or *bagging* [25] is a method to reduce the variance of an estimator. It generates bootstrap replicates of the training data set and uses these as new training data sets. The estimator of interested is then applied to each of these data sets and the multiple versions of this estimator are finally used to get an aggregated estimator. In regression, the aggregation consists in averaging the estimators. In classification, it consists in a plurality vote. Bagging particularly improves the accuracy of an estimator when perturbing the training data set causes significant changes in the estimator [25].

3.2.2 Jackknife

The *jackknife* [194, 195, 247] is a nonparametric method for estimating the bias and variance of an estimator. Here we focus only on the estimate of the variance since we will use it to estimate standard errors (Chapter 8).

Let $\hat{\boldsymbol{\alpha}}$ be an estimator of some quantity α. Let $\hat{\boldsymbol{\alpha}}_{(-i)}$ denote the estimator with the i-th observation removed. The jackknife estimate of the variance of $\hat{\boldsymbol{\alpha}}$ is [71, 259]:

$$\widehat{\mathbf{Var}}_{\text{jack}}\left(\hat{\boldsymbol{\alpha}}\right) = \frac{\tilde{\mathbf{s}}_{\text{jack}}}{n} ,$$

where

$$\tilde{\mathbf{s}}_{\text{jack}} = \frac{\sum_{i=1}^{n}\left(\tilde{\boldsymbol{\alpha}}_i - \frac{1}{n}\sum_{i=1}^{n}\tilde{\boldsymbol{\alpha}}_i\right)^2}{n-1}$$

is the sample variance of the so-called pseudo-values:

$$\tilde{\boldsymbol{\alpha}}_i = n\hat{\boldsymbol{\alpha}} - (n-1)\,\hat{\boldsymbol{\alpha}}_{(-i)} .$$

Under suitable conditions on $\hat{\boldsymbol{\alpha}}$, the jackknife estimate $\widehat{\mathbf{Var}}_{\text{jack}}\left(\hat{\boldsymbol{\alpha}}\right)$ consistently estimates (i.e., converges in probability to) $\mathrm{Var}\left(\alpha\right)$ [259].

3.3 Curse of dimensionality

The *curse of dimensionality* refers to the increasing difficulty of estimation as the dimension of the observations increases [13, 116, 259].

There are two versions of this "curse" [259]. The first is the computational curse of dimensionality which refers to the fact that the computational requirements of some methods can increase exponentially with dimension. The second is the statistical curse of dimensionality: in a d-dimensional setting, some methods require a sample size n that grows exponentially with d. This is the one we refer to by the curse of dimensionality.

For example [259], suppose we have n data points uniformly distributed on the interval $[-1,1]$. The expected number of points in the interval $[-0.1, 0.1]$ is $n/10$. Now suppose we have n data points on the 10-dimensional unit cube $[-1,1]^{10} = [-1,1] \times \cdots \times [-1,1]$. The expected number of points in the cube $[-0.1, 0.1]^{10}$ is

$$n \times \left(\frac{0.2}{2}\right)^{10} = \frac{n}{10,000,000,000} \, .$$

Hence, for small neighborhoods to have any data in them, n has to be extremely large.

This problem is particularly acute in bioinformatics where the number p of variables is much larger than the number n of samples. In statistical problems, this particular data setting is often summarized in the "small n, large p" catch phrase.

To circumvent this "curse," two possible solutions are the application of variable selection (Section 3.4) and regularization (Section 3.5).

3.4 Variable selection

Variable or *feature*[2] *selection* [109, 110] consists in selecting variables for a given prediction task. It has become the focus of much research [18, 109, 140], in particular in bioinformatics [205].

Indeed, the analysis of biological data, in particular microarray data, generally involves many irrelevant and redundant variables [109, 132] and often comparably few training examples. Microarray data also often contain noise. Moreover, the expression levels of many probes may be highly correlated. Such a characteristic is explained by the co-regulation of many genes: it is assumed that similar patterns in gene expression profiles usually suggest relationships between the genes or, equivalently, the genes targeted by the same transcription factors tend to show similar expression patterns [274]. Therefore, standard methods of supervised learning cannot be applied directly to obtain the parameter estimates. Including all the genes in the predictive model increases its variance and leads to poor predictive performance. Additionally, from a biological point of view, one should expect that only a small subset of the genes is relevant to predict the phenotypes.

Feature selection has therefore many potential benefits such as improving the prediction performance of the predictors, providing faster and more cost-effective predictors

[2] We use the terms "feature" and "variable" interchangeably. Note that a distinction is sometimes made in the literature [109].

(i.e., reducing training and utilization times, and reducing the measurement and storage requirements), and providing a better understanding of the underlying process that generated the data and thus facilitating data understanding (as well as data visualization) [109].

3.4.1 Variable ranking

Many variable selection algorithms include *variable ranking*, i.e., ranking variables according to their individual predictive power, as a principal or auxiliary selection mechanism because of its simplicity, scalability, and good empirical success [109]. To use variable ranking to build predictors, nested subsets incorporating progressively more and more variables of decreasing relevance are defined.

Selecting the most relevant variables is usually suboptimal for building a predictor, particularly if the variables are redundant [109]. However, even if variable ranking is not optimal, it may be preferable to variable subset selection methods (Section 3.4.2) because of its computational and statistical scalability. Computationally, it is efficient since it requires only the computation and sorting of p scores (where p is the number of input variables). Statistically, it is robust against overfitting because it introduces bias but it may have considerably less variance [116].

A common way to tackle the limitations of variable ranking is to consider subsets of variables that together have good predictive power, as opposed to ranking variables according to their individual predictive power.

3.4.2 Subset selection

Feature subset selection methods are essentially divided into filters, wrappers and embedded methods [109].

Filter methods select subsets of variables as a preprocessing step, independently of the chosen predictor. They assess the merits of features from the data ignoring the effects of the selected feature subset on the performance of the learning algorithm. Examples are methods that select variables by ranking them, by compression techniques (e.g., principal components analysis (PCA) and singular value decomposition (SVD)) [4, 115] or by computing univariate correlations with the output [112].

Wrapper methods assess subsets of variable according to their usefulness to a given predictor [140]. The learning algorithm is part of the evaluation function. The problem reduces to one of stochastic state space search. Examples are the forward and backward methods proposed in classical regression analysis.

Embedded methods perform variable selection as part of the learning procedure and are usually specific to given learning machines. Examples are classification trees and regularization methods which will be discussed in the following section.

3.5 Regularization and Stein's phenomenon

Regularization (or *shrinkage*) consists in constraining the parameters of an estimator (e.g., by restricting their number or imposing bounds on their values). The method's rationale is to prevent overfitting by imposing a "simpler" estimator, hence reducing its variance. By doing so, its bias will obviously increase (shrinkage is therefore sometimes referred to as *biased estimation*). In high-dimensional settings, however, the (often drastic) variance reduction generally compensates for the bias's increase, hence reducing the squared error (the bias-variance decomposition of the squared error introduced in Section 3.1.3).

Trading off bias for variance in order to reduce the squared error stems back to Stein [228]. Stein showed that the maximum likelihood estimator (MLE) of the mean of the multivariate normal distribution, i.e., the sample mean, is inadmissible under squared-error loss for dimensions higher or equal to 3. An estimator $\hat{\boldsymbol{\alpha}}$ of $\alpha \in \Lambda$ is *inadmissible* [231] if it is possible to construct another estimator $\hat{\boldsymbol{\alpha}}^*$ with smaller MSE on the entire parameter space,

$$\mathrm{MSE}_\alpha\left(\hat{\boldsymbol{\alpha}}^*\right) \leq \mathrm{MSE}_\alpha\left(\hat{\boldsymbol{\alpha}}\right) \ , \quad \forall\, \alpha \in \Lambda \ ,$$

and with strictly smaller MSE for at least one value,

$$\exists\, \alpha' \in \Lambda : \mathrm{MSE}_{\alpha'}\left(\hat{\boldsymbol{\alpha}}^*\right) < \mathrm{MSE}_{\alpha'}\left(\hat{\boldsymbol{\alpha}}\right) \ .$$

In other words, he proved that it is possible to construct an estimator with risk uniformly (i.e., over the entire parameter space) smaller than that of the MLE. This result is known as "*Stein's phenomenon*." Subsequently, James and Stein [130] proposed such an estimator known as the James-Stein estimator (see also Lehmann and Casella [155]).

Shrinkage has since then played an important role in mathematical statistics. In particular, it represents the cornerstone of Ledoit and Wolf [153]'s covariance matrix estimator which is the starting point of Chapter 5's contribution. It has also been successfully applied in regression (e.g., ridge regression, lasso, elastic net; Section 3.6.3) and in classification (e.g., regularized discriminant analysis; Section 3.7.2.3). In other fields, it has been used for solving integral equations (where it is referred to as Tikhonov regularization [244]) and nonlinear optimization problems (e.g., the Levenberg-Marquardt algorithm [170]).

3.6 Linear regression

The Gaussian graphical model (Chapter 4) which is studied in the present thesis is intimately related to linear regression (Section 4.3.3) which we now introduce.

3.6.1 The model

Let the input $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p) \in \mathbb{R}^p$ be a p-variate random variable and the output $\mathbf{y} \in \mathbb{R}$ be a real-valued random variable. In the *linear regression model* [5, 61, 197, 231], the relationship between $\mathbf{x}$ and $\mathbf{y}$, which is given (3.3) by

$$\mathbf{y} = o\left(\mathbf{x}\right) + \boldsymbol{\epsilon} \ , \quad \mathbb{E}\left(\boldsymbol{\epsilon}\right) = 0 \ ,$$

is modeled by a linear function:[3]

$$o(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} \beta_j \mathbf{x}_j \ , \tag{3.4}$$

where the β_j's are the unknown parameters or coefficients (how they are determined is explained below). This model hence assumes that the *regression function*,

$$\mathbb{E}(\mathbf{y} \mid \mathbf{x} = x) = o(x) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j \ ,$$

is linear.

We now present different techniques to estimate the parameters $\beta = (\beta_0, \ldots, \beta_p)^T$ of (3.4) from data. Without loss of generality, we assume the data to be centered. Hence, there is no need for an intercept term (under squared loss) [61] and thus $\beta_0 = 0$ in the sequel.

Suppose we have n i.i.d. observations. Let $x_{i\cdot} = (x_{i1}, \ldots, x_{ip})^T$ and y_i, $i = 1, \ldots, n$, denote the measurements for the i-th sample of variables $\mathbf{x}$ and $\mathbf{y}$, respectively. Let $X = (x_{1\cdot}, \ldots, x_{n\cdot})^T$ denote the $n \times p$ *data matrix* with i-th row given by $x_{i\cdot}$ and let $y = (y_1, \ldots, y_n)^T$ denote the n-dimensional *response vector*. Finally, let D_n denote the *data set* of available observations:

$$D_n = \{(x_{i\cdot}, y_i), i = 1, \ldots, n\} \ .$$

We start with the standard technique of least squares (Section 3.6.2) and we then present some regularization techniques (Section 3.6.3) that are used when the least squares approach is inappropriate.

3.6.2 Ordinary least squares

Ordinary least squares (OLS) consists in estimating the parameters by minimizing the *residual sum of squares*,

$$\mathrm{RSS}(\beta) = (y - X\beta)^T (y - X\beta) \ . \tag{3.5}$$

If the matrix X has full rank, it can easily be shown [231] that the (unique) solution to this minimization problem,

$$\hat{\beta}_{LS} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left((y - X\beta)^T (y - X\beta) \right) \ ,$$

is given by:

$$\hat{\beta}_{LS} = \left(X^T X\right)^{-1} X^T y \ , \tag{3.6}$$

by differentiating (3.5) with respect to β and by setting the first derivative to zero. The solution (3.6) corresponds to a minimum since the second derivative is a positive-definite matrix.

Under additional constraints, the least squares estimator can be shown to be the "best linear unbiased estimator" (BLUE) of β [231], where "best" means with minimum variance.

[3] The adjective "linear" refers solely to the parameters structure and not to the regression variables.

3.6.3 Regularization

When the matrix X is rank deficient, OLS becomes unusable. Further, OLS estimates often have low bias but large variance [116]. This implies that prediction accuracy (Section 3.1.3) can be improved by means of variable subset selection (Section 3.4) or regularization methods which trade off decreased variance for increased bias (Section 3.5). We now review some state-of-the-art regularization techniques for linear models.

3.6.3.1 Ridge regression

Ridge regression [120] minimizes the residual sum of squares subject to a bound on the L_2-norm of the coefficients:

$$\hat{\beta}_{ridge} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left((y - X\beta)^T (y - X\beta) \right) ,$$
$$\text{subject to} \quad \|\beta\|_2 \leq s , \tag{3.7}$$

where $s > 0$ and

$$\|\beta\|_2 = \beta^T \beta = \sum_{i=1}^{p} \beta_i^2 .$$

The ridge regression estimator can be equivalently written as

$$\hat{\beta}_{ridge} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left((y - X\beta)^T (y - X\beta) + \lambda \|\beta\|_2 \right) , \tag{3.8}$$

where $\lambda > 0$. There is a one-to-one correspondence between the parameters s in (3.7) and λ in (3.8) [116].

The ridge regression solution [116] is easily seen to be

$$\hat{\beta}_{ridge} = \left(X^T X + \lambda I \right)^{-1} X^T y , \tag{3.9}$$

where I is the identity matrix. It adds a positive constant to the diagonal of $X^T X$ before inversion to make the problem nonsingular. This was the main motivation for ridge regression [120] and traditional descriptions of this method start with (3.9). However, starting from (3.7) provides insight into how it works [116] and provides a general framework in which the other regularization methods we subsequently introduce can be integrated.

Note that in the "small n, large p" setting, it is possible to reduce the number of operations required to compute the ridge solution (3.9) from $O\left(p^3\right)$ (required to invert a $p \times p$ matrix) to $O\left(pn^2\right)$ using singular value decomposition (SVD) [114].

3.6.3.2 Lasso

Subset selection (Section 3.4.2) provides interpretable models but its prediction accuracy can sometimes be low because it is highly variable. This is due to its discrete nature: variables are either retained or dropped from the model [26]. Ridge regression, which is a continuous process, is reputed more stable. However, it does not provide any interpretable model because it does not set any coefficients to zero. To circumvent these issues, Tibshirani [242] introduced the least absolute shrinkage and selection operator, better known as the *lasso*, which minimizes the residual sum of squares subject to a bound on the L_1-norm of the coefficients. This can be equivalently written as:

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left((Y - X\beta)^T (Y - X\beta) + \lambda \|\beta\|_1 \right) , \tag{3.10}$$

where $\lambda > 0$ and

$$\|\beta\|_1 = \sum_{i=1}^{p} |\beta_i| .$$

By shrinking some coefficients and setting others to zero, the lasso tries to retain both the interpretability of subset selection and the stability of ridge regression.

Equation (3.10) can be efficiently solved using the least angle regression (LARS) algorithm which requires $O\left(n^3\right)$ operations (when $p > n$) [74].

3.6.3.3 Bridge regression

Bridge regression [89, 97] constrains the coefficients with an L_q-norm for $q \geq 0$:

$$\hat{\beta}_q = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left((y - X\beta)^T (y - X\beta) + \lambda \|\beta\|_q \right) ,$$

where $\lambda > 0$ and

$$\|\beta\|_q = \sum_{i=1}^{p} |\beta_i|^q .$$

It generalizes subset selection ($q = 0$), the lasso ($q = 1$) and ridge regression ($q = 2$). Note that only subset selection ($q = 0$) and the lasso penalty ($q = 1$) produce sparse solutions. Bridge regression with $1 < q < 2$ always keeps all predictors in the model, as does ridge regression [80].

3.6.3.4 Elastic net

Despite its success, the lasso has certain limitations. In particular, if there is a group of highly correlated variables, then the lasso tends to select only one variable from the group and does not care which one is selected [279]. To tackle this problem, Zou and Hastie [279] proposed the *elastic net* which combines ridge regression and the lasso by minimizing the residual sum of squares subject to bounds on the L_1-norm and the L_2-norm of the coefficients. This can be equivalently written as:

$$\hat{\beta}_{enet} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left((y - X\beta)^T (y - X\beta) + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2 \right) , \tag{3.11}$$

where $\lambda_1, \lambda_2 > 0$.

The elastic net simultaneously does automatic variable selection and continuous shrinkage as the lasso, while it selects groups of correlated variables.

As for the lasso, (3.11) can be solved by the LARS algorithm [279].

Tibshirani [242] and Fu [97] compared the prediction performance of the lasso, ridge and bridge regression and found that none of them uniformly dominates the other two.

Experiments on simulation and real data suggest that the elastic net outperforms the lasso in terms of prediction accuracy [279].

We have reviewed the most popular regularization methods for linear regression. Note, however, that other regularization techniques exist, such as partial least squares [270] or the adaptive lasso [278].

3.7 Classification

The problem of inferring putative nitrogen catabolite repression (NCR) genes which will be discussed in Chapter 7 can be cast as a two-class classification problem. We therefore need to clarify some aspects related to classification.

First, we start by introducing basic notions on classification (Section 3.7.1). Subsequently, we introduce four state-of-the-art two-class classifiers (Section 3.7.2) that we will use in Chapter 7. Finally, we present some important issues arising from imbalanced classes and detail how these can be dealt with (Section 3.7.3). In particular, we explain how to correct the a posteriori probabilities returned by a classifier (Section 3.7.3.3) when the classes' a priori probabilities estimated from the training data do not reflect the "true" a priori probabilities.

3.7.1 Bayes classifier

A *classifier* is a learning machine that predicts categorical outputs (Section 3.1). It is defined as a function $g(x) : \mathscr{X} \rightarrow \mathscr{Y}$ predicting a class $y \in \mathscr{Y} = \{c_1, \ldots, c_K\}$, where c_k, $k = 1, \ldots, K$, are the class labels and K is the number of classes, for each observed example $x \in \mathscr{X}$. The "goodness" of a classifier g is measured by the *risk* (or overall error) which under *zero-one loss* (misclassification rate) is given [181] by:

$$R(g) = \mathbb{P}(g(\mathbf{x}) \neq \mathbf{y}) = \sum_{y \in \mathscr{Y}} m_{\mathbf{y}}(y) \int_{\mathscr{X}} \mathbb{1}_{\{g(x) \neq y\}} f_{\mathbf{x}|\mathbf{y}}(x \mid y)\, dx \ , \tag{3.12}$$

where $m_{\mathbf{y}}(y)$ is the probability mass function of $\mathbf{y}$, $f_{\mathbf{x}|\mathbf{y}}(x \mid y)$ is the conditional density function of $\mathbf{x}$ given $\mathbf{y}$, and $\mathbb{1}_{\{\cdot\}}$ is the set indicator function.

For a given distribution, the optimal classifier, i.e., the one that minimizes $R(g)$, is called the *Bayes classifier*. Its risk is referred to as the *Bayes risk*.

It can be shown that the classifier g^* that maximizes the posterior probability,

$$g^*(\mathbf{x}) = \arg\max_y \mathbb{P}(y \mid \mathbf{x}) \ , \tag{3.13}$$

is optimal [53]. For strictly positive distributions, $g^*(\mathbf{x})$ is also unique, except on zero-measure subsets of $\mathscr{X}$. From the definition of conditional probability, i.e.,

$$\mathbb{P}(y \mid \mathbf{x}) = \frac{\mathbb{P}(y, \mathbf{x})}{\mathbb{P}(\mathbf{x})} \ ,$$

we note that maximizing the posterior probability is equivalent to maximizing the joint probability:

$$\arg\max_y \mathbb{P}(y \mid \mathbf{x}) = \arg\max_y \mathbb{P}(y, \mathbf{x}) \ . \tag{3.14}$$

3.7.2 Two-class classifiers

We describe four state-of-the-art classifiers, namely k-nearest neighbors, naive Bayes, linear discriminant analysis and support vector machines, that we will use in Chapter 7. We assume we have available a data set D_n of n samples:

$$D_n = \{(x_{i\cdot}, y_i), i = 1, \dots, n\} \ .$$

3.7.2.1 k-Nearest neighbors classifiers

k-Nearest neighbors (KNN) [66] assigns to a given sample $x_{j\cdot} = (x_{1j}, \dots, x_{pj})$ (absent from the training set D_n) the label most frequently represented (i.e., through a majority vote) among the $k \in \mathbb{N}^*$ nearest samples in D_n. The number k of neighbors is typically chosen by cross-validation (Section 3.1.2) [66]. KNN classifiers require a metric between samples which we assume is Euclidian distance (the usual distance with real-valued data). The Euclidian distance between two samples $x_{j\cdot}$ and $x_{k\cdot}$ is given by:

$$\|x_{j\cdot} - x_{k\cdot}\| = \left(\sum_{l=1}^{p} (x_{lj} - x_{lk})^2 \right)^{1/2} \ .$$

KNN classifiers are local instance-based classifiers: the training samples are kept in "memory" and the computation, which consists in fitting the data locally, is deferred until classification.

Despite its simplicity, KNN classifiers have been often rather successful in a large number of classification problems [116, 200]. Asymptotically, the error rate of the 1-NN classifier is bounded by twice the Bayes rate (Section 3.7.1) [40].

In high-dimensional settings, however, the bias-variance trade-off (Section 3.1.3) associated with estimation error is generally driven by the bias, which can be important even for the largest variance [92].

3.7.2.2 Naive Bayes

Naive Bayes (NB) [66] is probably the simplest classifier. It assumes that the variables are independent conditionally on the class y, i.e.,

$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid y \ , \quad \forall i, j \ . \tag{3.15}$$

From the chain rule of probability, we have:

$$\begin{aligned} \mathbb{P}(\mathbf{x} \mid y) &= \mathbb{P}(\mathbf{x}_1 \mid y)\, \mathbb{P}(\mathbf{x}_2 \mid y, \mathbf{x}_1)\, \mathbb{P}(\mathbf{x}_3 \mid y, \mathbf{x}_1, \mathbf{x}_2) \dots \mathbb{P}(\mathbf{x}_p \mid y, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{p-1}) \\ &= \prod_{j=1}^{p} \mathbb{P}(\mathbf{x}_j \mid y, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{j-1}) \end{aligned} \tag{3.16}$$

Under the "naive" assumptions (3.15), (3.16) reduces to

$$\mathbb{P}(\mathbf{x} \mid y) = \prod_{j=1}^{p} \mathbb{P}(\mathbf{x}_j \mid y) \ .$$

By (3.14) and by application of the product rule, we see that NB classifies by selecting

$$\arg\max_{y}\left(\hat{\mathbb{P}}\left(y\right)\prod_{j=1}^{p}\hat{\mathbb{P}}\left(\mathbf{x}_j \mid y\right)\right)$$

where $\hat{\mathbb{P}}\left(y\right)$ and $\hat{\mathbb{P}}\left(\mathbf{x}_j \mid y\right)$ are estimates of the respective probabilities derived from the frequency of their respective arguments in the training sample (with possible corrections such as the Laplace estimate) in the discrete case. In the continuous case, one has to make an assumption concerning the underlying distribution of the data, typically multivariate normality.

Under zero-one loss (misclassification rate), NB is optimal when attributes are independent given the class, i.e., when the assumptions (3.15) hold true. Moreover, NB is sometimes accurate even when these assumptions are violated [60]. Indeed, it appears that some violations of these assumptions do not matter [60]. This (partially) explains why NB has repeatedly performed better than expected in empirical trials in domains containing clear attribute dependences [60, 248].

As some violations do matter, there is an increasing body of work developing so-called semi-naive Bayes classifiers that attempt to alleviate the problems of the attribute independence assumption [94, 262, 277]. It seems, however, that detecting attribute dependence is not necessarily the best way to extend the Bayesian classifier and (at least the first) attempts to build on NB's success by relaxing the independence assumption have had mixed results [60].

3.7.2.3 **Discriminant analysis**

In discriminant analysis, each class density is modeled as a multivariate Gaussian:

$$f_{\mathbf{x},k}\left(x\right) = \left(2\pi\right)^{-p/2}\left(\det\Sigma_k\right)^{-1/2}\exp\left\{-\frac{1}{2}\left(x-\mu_k\right)^T\Sigma_k^{-1}\left(x-\mu_k\right)\right\}\ , \qquad -\infty < x < \infty\ , \quad k = 1,\ldots,K\ , \tag{3.17}$$

where μ_k and Σ_k are the mean vector and covariance matrix of class c_k, respectively. Depending on how Σ_k is estimated, one distinguishes between linear discriminant analysis and the alternative approaches of quadratic and regularized discriminant analyses.

Linear discriminant analysis

Linear discriminant analysis (LDA) [116] arises in the special case when all the classes are constrained to have a common covariance matrix:

$$\Sigma_k = \Sigma\ , \quad \forall k\ . \tag{3.18}$$

To compare two classes c_k and c_l, it is sufficient to look at the log-ratio

$$\log\frac{\mathbb{P}\left(y = c_k \mid \mathbf{x}\right)}{\mathbb{P}\left(y = c_l \mid \mathbf{x}\right)}\ .$$

Given Bayes's theorem, i.e.,

$$\mathbb{P}\left(y = c_k \mid \mathbf{x}\right) = \frac{f_{\mathbf{x},k}\left(x\right)\mathbb{P}\left(y = c_k\right)}{\sum_{l=1}^{K} f_{\mathbf{x},l}\left(x\right)\mathbb{P}\left(y = c_l\right)} \ ,$$

we have that:

$$\begin{aligned}\log\frac{\mathbb{P}\left(y = c_k \mid \mathbf{x}\right)}{\mathbb{P}\left(y = c_l \mid \mathbf{x}\right)} &= \log\frac{f_{\mathbf{x},k}\left(x\right)}{f_{\mathbf{x},l}\left(x\right)} + \log\frac{\mathbb{P}\left(y = c_k\right)}{\mathbb{P}\left(y = c_l\right)} \\ &= \log\frac{\mathbb{P}\left(y = c_k\right)}{\mathbb{P}\left(y = c_l\right)} - \frac{1}{2}\left(\mu_k + \mu_l\right)^T \Sigma^{-1}\left(\mu_k - \mu_l\right) + x^T\Sigma^{-1}\left(\mu_k - \mu_l\right) \ ,\end{aligned} \tag{3.19}$$

which is linear in x. The assumption (3.18) of equal covariance causes the normalization factors and the quadratic part in the exponents to cancel each other out, respectively.

From (3.19), we see that the decision rule (3.13) can be equivalently expressed as

$$\arg\max_k \delta_k\left(x\right) \ ,$$

where the δ_k's are the *linear discriminant functions* given by

$$\delta_k\left(x\right) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log\mathbb{P}\left(y = c_k\right) \ .$$

If the classes are equally likely a priori then LDA classifies the sample x to the class k for which the *Mahalanobis distance* from x to its center μ_k (appearing in the exponent of the density function (3.17)),

$$M\left(x,k\right) = \left(x - \mu_k\right)^T \Sigma^{-1}\left(x - \mu_k\right) \ , \tag{3.20}$$

is smallest [200].

The a priori probabilities of the classes in the training set are estimated by their frequencies:

$$\hat{\mathbb{P}}\left(y = c_k\right) = \frac{n_k}{n} \ ,$$

where n_k is the number of observations in class c_k.

The parameters of the Gaussian distributions are estimated by their sample counterparts:

$$\hat{\boldsymbol{\mu}}_k = \sum_{i:y_i=c_k}\frac{x_{i\cdot}}{n_k} \ ,$$

and

$$\hat{\mathbf{S}}_k = \frac{1}{n_k - 1}\sum_{i:y_i=c_k}\left(x_{i\cdot} - \hat{\boldsymbol{\mu}}_k\right)\left(x_{i\cdot} - \hat{\boldsymbol{\mu}}_k\right)^T \ . \tag{3.21}$$

Quadratic and regularized discriminant analyses

If the constraints (3.18) are dropped, i.e., if the Σ_k are not assumed to be equal, the cancellations in (3.19) do not occur and the resulting decision boundary is quadratic in x. The corresponding classifier is known as *Quadratic discriminant analysis* (QDA). Although these constraints are rather strong, QDA exhibits higher variance and requires generally larger samples than does LDA [257], which is problematic in the "small n, large p" setting prevalent in bioinformatics. Furthermore, classification rules based on QDA seem to be more sensitive to violations of the assumption underlying discriminant analysis [91], i.e., when the class conditional densities are not approximately normal.

Whether using LDA or QDA, robust estimators of the covariance matrix (Chapter 5) should however be preferred to the sample covariance matrix (3.21). Discriminant analysis with regularized estimators of the covariance matrix/matrices is referred to as regularized discriminant analysis (RDA). It was first introduced by Friedman [91] who used a shrinkage estimator similar to the one presented in Section 5.2, albeit a more computationally expensive one (the shrinkage parameter being chosen through cross-validation).

3.7.2.4 **Support vector machines**

Support vector machines (SVMs) [20, 116, 216, 261] try to separate the two classes of points using a linear function of the form

$$h\left(x\right) = w^T x + b \ , \tag{3.22}$$

with $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$. Such a function assigns the label $+1$ and -1 to the points $x \in \mathscr{X}$ with, respectively, $h\left(x\right) \geq 0$, and $h\left(x\right) < 0$. Hence, an observation $(x_{i\cdot}, y_i)$ is correctly classified (3.22) if $y_i h\left(x_{i\cdot}\right) \geq 0$. The classification problem reduces to learning such a function from the training data.

Learning (3.22) can be accomplished through empirical risk minimization (Section 3.1.1), i.e., by minimizing the number of classification errors on the training data set. The resulting classifiers are known as perceptrons [203]. Unfortunately, several problems arise with these classifiers [200]. In particular, when the data are linearly separable there is no unique solution (Figure 3.3), and when the data are not linearly separable the algorithm does not converge.

Optimal separating hyperplane when the two classes are linearly separable

The peculiarity of SVMs is that they do not exclusively focus on the number of misclassifications, but also on the confidence of the classifications. More specifically, they try to find a hyperplane, known as the *optimal separating hyperplane*, that separates the two classes and that maximizes the distance to the closest point from either class. Hence, they provide a unique solution to the problem of finding a separating hyperplane. Furthermore, by maximizing the margin between the two classes on the training data, SVMs lead to better classification performance on test data (as a consequence of results in learning theory [253, 254]).

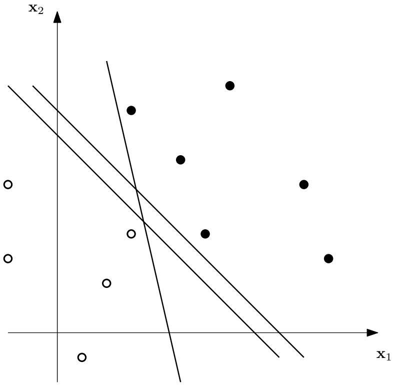


Figure 3.3: The empirical risk minimization principal does not define a unique solution, even when the training data are linearly separable (figure redrawn from Schölkopf et al. [216]).

As shown in Figure 3.4, when the data are linearly separable the function (3.22) defines two half-spaces of points classified positively and negatively "with large confidence," namely the sets

$$h^+ = \{x : h(x) \geq 1\} \quad \text{and} \quad h^- = \{x : h(x) \leq -1\} \ ,$$

respectively.

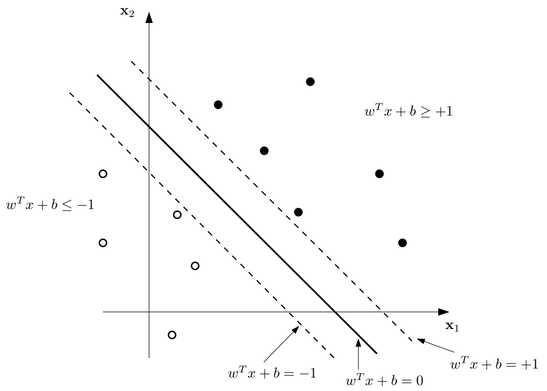


Figure 3.4: The affine function $h(x) = w^Tx + b$ defines two half-spaces where points are classified with large confidence (text) for the positive examples (black circles) and for the negative examples (white circles) (figure redrawn from Schölkopf et al. [216]).

The distance separating the two half-spaces, called the *margin*, is equal to $2/\|w\|$. SVMs try to correctly classify all points in the training set with strong confidence. Hence, they maximize $2/\|w\|$ under the constraints

$$y_i \left(w^T x_{i\cdot} + b\right) \geq 1 \ , \quad i = 1, \ldots, n \ . \tag{3.23}$$

Extension to the nonseparable case

Since a training set is not necessarily separable by a linear hyperplane, SVMs modify the constraints (3.23) using the continuous hinge loss function

$$C_{\mathrm{h}}(x, y) = \max\left(0, 1 - yh(x)\right) \ . \tag{3.24}$$

If a point (x, y) is correctly classified with large confidence, i.e., $yh(x) \geq 1$, then $C_{\mathrm{h}}(x, y) = 0$. When $yh(x) < 1$, x is either correctly classified with small confidence $(0 \leq yh(x) < 1)$ or misclassified $(yh(x) < 0)$. In these cases, the hinge loss is positive and increases with the distance from x to the correct half-space of large confidence.

SVMs require both a large margin and few misclassifications or classifications with little confidence on the training set, by solving the problem:

$$\underset{h(x)=w^T x+b}{\arg\min} \ \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{n} C_{\mathrm{h}}(x_{i\cdot}, y_i) \ , \tag{3.25}$$

where $c \in \mathbb{R}$ is the parameter that controls the tradeoff between the two requirements. This parameter is typically determined by cross-validation [116].

Since the hinge loss function (3.24) is not differentiable, the direct minimization of (3.25) is not straightforward. To tackle this problem, (3.25) is transformed (by introducing so-called slack variables) into an equivalent convex optimization problem which can be resolved using Lagrange multipliers [216].

Interestingly, the solution to (3.25) only involves the points in the training set through their dot products, $x_k x_l^T$, for $k, l = 1, \ldots, p$. Hence, SVM classifiers can be readily adapted to kernels.

Kernel

A *kernel* is defined as a real-valued function $k : \mathscr{X} \times \mathscr{X} \longrightarrow \mathbb{R}$ which can be thought of as a "comparison function." Instead of using a mapping $\phi : \mathscr{X} \longrightarrow \mathscr{F}$ to represent each data point $x \in \mathscr{X}$ by $\phi(x) \in \mathscr{F}$, the data are represented by the $p \times p$ matrix of pairwise comparisons $k_{k,l} = k(x_k, x_l)$.

SVMs build linear separating hyperplanes in the feature space associated with the kernel. Hence, if the kernel is nonlinear SVMs can produce nonlinear boundaries by constructing a linear boundary in a transformed version of the original space (Figure 3.5).

The most commonly used kernels are given in Table 3.1. Kernels can also be defined for specific problems, such as string kernels for protein sequence data classification [216] or tree kernels for network inference [102] for example.

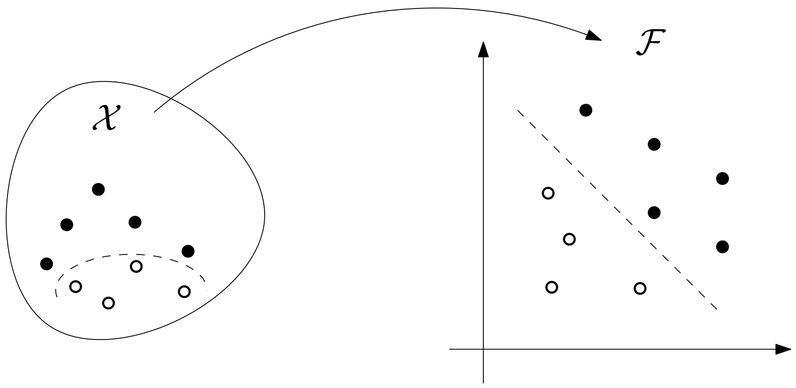


Figure 3.5: A nonlinear boundary is obtained (left) by constructing a linear boundary in a transformed version (right) of the original space (left; figure redrawn from Schölkopf et al. [216]).

Table 3.1: Commonly used kernels.

Kernel	Parameters	Description
Linear	None	$k\left(x_k, x_l\right) = x_k x_l^T$
Polynomial	d	$k\left(x_k, x_l\right) = \left(x_k x_l^T + 1\right)^d$
Gaussian radial basis function (RBF)	σ	$k\left(x_k, x_l\right) = \exp\left(\frac{-\lvert x_k - x_l \rvert^2}{2\sigma^2}\right)$
Gaussian	σ	$k\left(x_k, x_l\right) = \frac{1}{2\pi\sigma^2}\exp\left(\frac{-x_k^2 - x_l^2}{2\sigma^2}\right)$

3.7.3 Imbalanced classes

A (training) data set is *imbalanced* if the classes are not (approximately) equally represented [31]. In two-class classification, this means that one of the classes is (sometimes heavily) underrepresented compared to the other one. In *S. cerevisiae*'s NCR, for example, the "positive" class, i.e., the genes annotated as targets of NCR, is composed of only 41 genes out of almost $6,000$ genes (Chapters 7 and 8).

This is problematic mainly in terms of accuracy. Usually, algorithms minimize the risk to which the minority class contributes only marginally. In the example given previously, a classifier that always predicts the majority class, i.e., that predicts all genes as being insensitive to NCR, has a low risk but is useless for predicting putative NCR genes.

3.7.3.1 Sampling methods

Classifiers can be adapted to imbalanced data sets by using undersampling or oversampling (both techniques can also be combined). Undersampling consists in discarding instances of the overrepresented class. In case of oversampling, instances from the minority class are duplicated. The examples to be removed or duplicated are typically chosen randomly but they can also be chosen according to some prior knowledge.

Several studies suggest that undersampling leads to better results, while oversampling produces only small changes, if any, in performance (Hoste [123] and references therein). However, none of the approaches consistently outperforms the other and some studies have even presented conflicting viewpoints on the usefulness of oversampling versus undersampling [31].

Despite their benefits, these approaches have obvious drawbacks. By discarding (potentially) useful data, undersampling implies a loss of information, whereas oversampling increases the training size without any gain in information which can lead to overfitting. Furthermore, it is difficult to determine a specific undersampling or oversampling rate which consistently leads to the best results.

Note, however, that this last issue is not restricted to classification tasks with imbalanced classes. Indeed, a balanced class distribution is not necessarily the best one for learning [263]. Finding the best class distribution for training a classifier is still an open question.

3.7.3.2 Other methods

Alternatively, one can modify the decision rule (3.13) by moving the decision threshold. In two-class classification, this consists in taking a threshold on the (possibly corrected) posterior probability above (below) 0.5 when classifying an example in the majority (minority) class. Another possibility consists in varying the cost matrix so as to penalize errors on the minority class more severly than those on the majority class. However, recent results suggest that oversampling and undersampling produce nearly the same classifiers as does moving the decision threshold and varying the cost matrix [166].

3.7.3.3 **Posterior probability correction**

Given a random vector $\mathbf{x}$, the decision of a classifier is typically based on the (estimated) a posteriori probabilities $\hat{\mathbb{P}}\left(y \mid \mathbf{x}\right)$ of class membership, which rely on the a priori probabilities $\hat{\mathbb{P}}\left(y\right)$ of the classes estimated from the training set.

Unfortunately, the training data set does not always reflect the true a priori probabilities $\mathbb{P}\left(y\right)$ of the target classes, which can hinder the classifier's accuracy [204]. For example, biologists do not expect more than 200 NCR target genes (i.e., positive examples) out of almost $6,000$ genes in *S. cerevisiae.* As we will see in Chapter 7, the corresponding "true" a priori probabilities are not reflected in the training data set available.

We can however compute *corrected* a posteriori probabilities $\hat{\mathbb{P}}_c\left(y \mid \mathbf{x}\right)$ in terms of the outputs $\hat{\mathbb{P}}\left(y \mid \mathbf{x}\right)$ provided by the trained model using Bayes's theorem:

$$\hat{\mathbb{P}}_c\left(y \mid \mathbf{x}\right) = \frac{\frac{\hat{\mathbb{P}}_c(y)}{\hat{\mathbb{P}}(y)}\hat{\mathbb{P}}\left(y \mid \mathbf{x}\right)}{\sum_{k=1}^{K}\frac{\hat{\mathbb{P}}_c(y=c_k)}{\hat{\mathbb{P}}(y=c_k)}\hat{\mathbb{P}}\left(y=c_k \mid \mathbf{x}\right)} \,,$$

where $\hat{\mathbb{P}}_c\left(y\right)$ are the new a priori probabilities [204]. Note that the corrected posteriori probabilities are simply the a posteriori probabilities obtained from the training set weighted by the ratio of the new priors to the old priors [204]. The denominator ensures that the corrected a posteriori probabilities sum to one.

4

Gaussian Graphical Model

Graphical models are representations of multivariate probabilistic models in which conditional (in)dependence (Section 4.1) constraints are specified by graphs (Sections 4.4 and 4.5) [150]. More specifically, nodes represent random variables, and the absence of links between them represent conditional independence assumptions. These models are therefore regarded as a "marriage between probability theory and graph theory" [133].

Graphical models hence provide a framework to efficiently deal with the intrinsic uncertainty and complexity of many scientific disciplines, including bioinformatics. Indeed, probability theory provides a system of reasoning under uncertainty [187], while the graphical "language" provides an intuitively appealing representation of the model [133].

In particular, the notion of modularity—a complex system is built by combining simpler parts—underlying the graphical representation enables scientists to describe and handle complex problems by combining simpler elements [133, 150].

Further, the visual representation of the structure of a probabilistic model facilitates communication between scientists [150] and can be used to design new models [17]. In addition, inspection of the graph readily provides insights into the properties of the model [17].

Finally, graphs represent natural data structures for computers which enable complex computations to be efficiently expressed in terms of graphical manipulations [17, 150].

In this chapter, we consider *undirected graphical models*, also known as *Markov random fields*, in which the links have no directional significance. The other major class of graphical models, the *directed graphical models*, also known as *Bayesian networks*, in which the links have a particular directionality (indicated by arrows), will not be treated in this thesis.

More specifically, we focus on the Gaussian graphical model (GGM). We thus only introduce the key concepts of undirected graphical models that are needed to introduce GGMs (Section 4.6). For a more general treatment of graphical models, we refer to standard textbooks such as Cowell et al. [41], Cox and Wermuth [43], Edwards [70], Lauritzen [150], Whittaker [266].

We first present the concepts of (conditional) independence (Section 4.1), covariance (Section 4.2) and (partial) correlation (Section 4.3) which are central to graphical models. We then introduce the notation and terminology of graphs (Section 4.4). Next, we introduce the Markov properties and the concept of faithfulness (Section 4.5) which provide the connection between the probabilistic notion of (conditional) independence and the graphical “language.”

We then define the GGM (Section 4.6) and present the different approaches to GGM selection (Section 4.7). We put particular emphasis on GGM selection in the “small n, large p” setting (Section 4.7.1) which is crucial in bioinformatics applications.

4.1 Independence and conditional independence

We start with the basic notion of *independence* before introducing the essential concept of *conditional independence.* Henceforth, we consider continuous (real-valued) random variables (since we are interested in gene expression data) and assume their probability density functions to exist. We denote by $f_{\mathbf{x}}$ the density function of the random variable $\mathbf{x}$.

4.1.1 Independence

Definition 4.1.1 (independence)**.** *The random vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *are* independent *if and only if the joint probability density function* $f_{\mathbf{xy}}$ *satisfies*

$$f_{\mathbf{xy}}(x,y) = f_{\mathbf{x}}(x)\, f_{\mathbf{y}}(y) \;\; , \quad \forall x, y \; . \tag{4.1}$$

This relationship is denoted by $\mathbf{x} \perp\!\!\!\perp \mathbf{y}$.

The symbol ‘$\perp\!\!\!\perp$’ is the usual notation for independence due to Dawid [48]. Hence, $\mathbf{x}$ and $\mathbf{y}$ are independent if and only if the joint probability density function factorizes into the product of the marginal density functions. From (4.1), it is clear that the independence relation is symmetric in $\mathbf{x}$ and $\mathbf{y}$.

Interestingly, to establish independence, it is sufficient to show that the joint density function factorizes into the product of two factors, one not involving $\mathbf{x}$ and the other not involving $\mathbf{y}$, rather than it factorizes into the product of the marginals [48, 49]. This is known as the *factorization criterion* for independence [266], which is formalized in the proposition given below.

Proposition 4.1.1 (Whittaker [266])**.** *The random vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *are independent if and only if there exist two functions* g *and* h *such that*

$$f_{\mathbf{xy}}(x,y) = g(x)\, h(y) \;\; , \quad \forall x, y \; . \tag{4.2}$$

For example [258], let $\mathbf{x}$ and $\mathbf{y}$ have joint density

$$f_{\mathbf{xy}}(x,y) = \begin{cases} 2e^{-(x+2y)} & \text{if } x > 0 \text{ and } y > 0 \; , \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

This density function can be decomposed as in (4.2) with

$$g(x) = \begin{cases} 2e^{-x} & \text{if } x > 0 \; , \\ 0 & \text{otherwise,} \end{cases}$$

and

$$h(y) = \begin{cases} e^{-2y} & \text{if } y > 0 \ , \\ 0 & \text{otherwise.} \end{cases}$$

Thus, $\mathbf{x}$ and $\mathbf{y}$ are independent.

4.1.2 Conditional independence

Definition 4.1.2 (conditional independence)**.** *The random vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *are* conditionally independent *given (conditioned on) the random vector* $\mathbf{z}$ *if and only if*

$$f_{\mathbf{xy}|\mathbf{z}}(x, y \mid z) = f_{\mathbf{x}|\mathbf{z}}(x \mid z)\, f_{\mathbf{y}|\mathbf{z}}(y \mid z) \ , \quad \forall x, y \ , \tag{4.4}$$

for all z for which $f_{\mathbf{z}}(z) > 0$. This is written as $\mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}$.

Equation (4.4) is equivalent [266] to

$$f_{\mathbf{x}|\mathbf{yz}}(x \mid y, z) = f_{\mathbf{x}|\mathbf{y}}(x \mid y) \ . \tag{4.5}$$

This latter expression shows that the conditional independence of $\mathbf{x}$ from $\mathbf{y}$ means that $\mathbf{z}$ can be removed from the conditioning set.

Equation (4.4) is also equivalent [266] to

$$f_{\mathbf{xyz}}(x, y, z) = \frac{f_{\mathbf{xz}}(x, z)\, f_{\mathbf{yz}}(y, z)}{f_{\mathbf{z}}(z)} \ . \tag{4.6}$$

This second reformulation illustrates the fact that conditional independence can be rephrased entirely in terms of joint and marginal densities. In fact, as with independence, the *factorization criterion* for conditional independence [266] states that conditional independence should not necessarily factorize in joint and marginal distributions, as formalized in the following proposition.

Proposition 4.1.2 (Whittaker [266])**.** *The random vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *are conditionally independent given* $\mathbf{z}$*, if and only if there exists two functions g and h such that*

$$f_{\mathbf{xyz}}(x, y, z) = g(x, z)\, h(y, z) \ , \quad \forall x, y \ , \tag{4.7}$$

for all z for which $f_{\mathbf{z}}(z) > 0$.

Note that Propositions 4.1.1 and 4.1.2 only require the existence of a factorization into coordinate functions; there is no requirement for the factors to be unique [266].

The ternary relation $\mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}$ has the following properties [48, 49, 150], where h denotes an arbitrary measurable function on the sample space of $\mathbf{x}$:

$$\text{if } \mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z} \quad \text{then} \quad \mathbf{y} \perp\!\!\!\perp \mathbf{x} \mid \mathbf{z} \ ; \tag{4.8}$$

$$\text{if } \mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z} \text{ and } \mathbf{u} = h(\mathbf{x}) \quad \text{then} \quad \mathbf{u} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z} \ ; \tag{4.9}$$

$$\text{if } \mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z} \text{ and } \mathbf{u} = h(\mathbf{x}) \quad \text{then} \quad \mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid (\mathbf{z}, \mathbf{u}) \ ; \tag{4.10}$$

$$\text{if } \mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z} \text{ and } \mathbf{x} \perp\!\!\!\perp \mathbf{w} \mid (\mathbf{y}, \mathbf{z}) \quad \text{then} \quad \mathbf{x} \perp\!\!\!\perp (\mathbf{w}, \mathbf{y}) \mid \mathbf{z} \ . \tag{4.11}$$

Conditional independence can be regarded as expressing the notion of "irrelevance" in a given context, in the sense that the relation $\mathbf{x} \perp\!\!\!\perp \mathbf{y} \mid \mathbf{z}$ can be interpreted as "if we know $\mathbf{z}$, information about $\mathbf{y}$ is irrelevant for knowledge of $\mathbf{x}$" [70]. Although not being rigourous, this reformulation is helpful to grasp the intuition behind conditional independence (see also Lauritzen [150]).

Interestingly, the definition of independence (Definition 4.1.1) can also be rephrased in terms of the conditional and marginal density functions of $\mathbf{x}$ (or, equivalently, $\mathbf{y}$). "Intuitively, $\mathbf{x} \perp\!\!\!\perp \mathbf{y}$ if any information received about $\mathbf{y}$ does not alter uncertainty about $\mathbf{x}$" [48], formally

$$\mathbf{x} \perp\!\!\!\perp \mathbf{y} \iff f_{\mathbf{x}|\mathbf{y}}(x \mid y) = f_{\mathbf{x}}(x) \ , \quad \forall x, y \ ; \tag{4.12}$$

that is, $\mathbf{x} \perp\!\!\!\perp \mathbf{y}$ if and only if the conditional density function of $\mathbf{x}$ given $\mathbf{y}$ is equal to the marginal density function of $\mathbf{x}$. The advantage of this characterization is that it does not involve the density of $\mathbf{y}$. Note that the conditional distribution should not necessarily be the marginal distribution of $\mathbf{x}$, but simply a function g not involving x [48, 49]:

$$\mathbf{x} \perp\!\!\!\perp \mathbf{y} \iff f_{\mathbf{x}|\mathbf{y}}(x \mid y) = g(x) \ , \quad \forall x, y \ . \tag{4.13}$$

4.2 Covariance (matrix)

Definition 4.2.1 (covariance). *The* covariance *of two real valued random variables* $\mathbf{x}$ *and* $\mathbf{y}$*, each with finite variance, is defined as*

$$\mathrm{Cov}(\mathbf{x}, \mathbf{y}) = \mathbb{E}\left((\mathbf{x} - \mathbb{E}(\mathbf{x}))(\mathbf{y} - \mathbb{E}(\mathbf{y}))\right) \ . \tag{4.14}$$

A simple rearrangement of (4.14) leads to

$$\mathrm{Cov}(\mathbf{x}, \mathbf{y}) = \mathbb{E}(\mathbf{xy}) - \mathbb{E}(\mathbf{x})\,\mathbb{E}(\mathbf{y}) \ . \tag{4.15}$$

Note that $\mathbb{E}(\mathbf{xy})$ exists because $\mathbf{x}$ and $\mathbf{y}$ have finite variances [129]. We remark that

$$\mathrm{Cov}(\mathbf{x}, \mathbf{x}) = \mathrm{Var}(\mathbf{x}) \, .$$

Definition 4.2.2 (covariance matrix). *The* covariance matrix *of a real valued p-dimensional random vector* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$ *is defined as*

$$\Sigma = \mathrm{Var}(\mathbf{x}) = \mathbb{E}\left((\mathbf{x} - \mathbb{E}(\mathbf{x}))(\mathbf{x} - \mathbb{E}(\mathbf{x}))^T\right) \ . \tag{4.16}$$

The (i, j)-th entry of the covariance matrix Σ is thus given by $\mathrm{Cov}(\mathbf{x}_i, \mathbf{x}_j)$.

Definition 4.2.3 (concentration matrix). *The* concentration matrix *of a real valued p-dimensional random vector* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$ *is the inverse of its covariance matrix Σ:*

$$\Omega = \Sigma^{-1} \ . \tag{4.17}$$

4.3 Correlation and partial correlation

Let $\mathbf{x}$ and $\mathbf{y}$ be two real valued random variables, each with finite variance.

4.3.1 Correlation

Definition 4.3.1 (correlation)**.** *If* $\mathbf{x}$ *and* $\mathbf{y}$ *are nondegenerate (i.e.,* $\mathrm{Var}\left(\mathbf{x}\right) \neq 0$ *and* $\mathrm{Var}\left(\mathbf{y}\right) \neq 0$*) then the* correlation *of* $\mathbf{x}$ *and* $\mathbf{y}$ *is defined as*

$$\mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right) = \frac{\mathrm{Cov}\left(\mathbf{x}, \mathbf{y}\right)}{\sqrt{\mathrm{Var}\left(\mathbf{x}\right) \mathrm{Var}\left(\mathbf{y}\right)}} \; . \tag{4.18}$$

Note that correlation is undefined for degenerate random variables. From (4.18), we see that correlation is invariant to changes in location and scale [5], and is symmetric in the random variables.

Using the Cauchy-Schwartz inequality, it can be shown [231] that

$$-1 \leq \mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right) \leq 1 \; .$$

To simplify the notation, we will sometimes write $\rho_{(\mathbf{x},\mathbf{y})}$ instead of $\mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right)$. Furthermore, when the random variables are indexed, as $\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_p$, for example, we will often simply write $\rho_{(i,j)}$ instead of $\rho_{(\mathbf{x}_i,\mathbf{x}_j)}$.

4.3.1.1 Correlation and independence

Before establishing the relationship between correlation and independence, we give the definition of *uncorrelatedness.*

Definition 4.3.2 (uncorrelatedness)**.** *The variables* $\mathbf{x}$ *and* $\mathbf{y}$ *are said to be* uncorrelated *if*

$$\mathrm{Cov}\left(\mathbf{x}, \mathbf{y}\right) = 0 \; . \tag{4.19}$$

Because of (4.15), (4.19) is equivalent to

$$\mathbb{E}\left(\mathbf{x}\mathbf{y}\right) = \mathbb{E}\left(\mathbf{x}\right) \mathbb{E}\left(\mathbf{y}\right) \; . \tag{4.20}$$

Note that nondegenerate random variables with finite variance are uncorrelated if and only if their correlation is zero.

Importantly, correlation is a measure of *linear dependence* (equivalently, uncorrelatedness is a measure of *linear independence*) and it does not capture more complex forms of dependence [78]. Three cases can arise [78]:

1. If $\mathbf{x}$ and $\mathbf{y}$ are "perfectly" linearly dependent, that is

$$\exists\, a \in \mathbb{R} \setminus \{0\}\,, b \in \mathbb{R} : \; \mathbb{P}\left(\mathbf{y} = a\mathbf{x} + b\right) = 1 \; , \tag{4.21}$$

then $\mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right) = \mathrm{sgn}\left(a\right)$, where $\mathrm{sgn}\left(\cdot\right)$ is the *sign function* defined as:

$$\mathrm{sgn}\left(c\right) = \begin{cases} -1 & \text{if } c < 0 \; , \\ 0 & \text{if } c = 0 \; , \\ 1 & \text{if } c > 0 \; . \end{cases} \tag{4.22}$$

Hence, $\left|\mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right)\right| = 1$.

2. If $\mathbf{x}$ and $\mathbf{y}$ are not "perfectly" linearly dependent, that is

$$\forall\, a \in \mathbb{R} \setminus \{0\}\,, b \in \mathbb{R} :\ \mathbb{P}\left(\mathbf{y} = a\mathbf{x} + b\right) < 1\ , \tag{4.23}$$

   then $|\mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right)| < 1$.

3. If $\mathbf{x}$ and $\mathbf{y}$ are uncorrelated, then $\mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right) = 0$.

The following lemmata and theorem clarify the connection between independence and correlation, which are too often incorrectly assumed to be equivalent [78].

Lemma 4.3.1. *If* $\mathbf{x}$ *and* $\mathbf{y}$ *are independent random variables then* $\mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right) = 0$*.*

Proof: If $\mathbf{x} \perp\!\!\!\perp \mathbf{y}$, it follows from the definitions of independence (Definition 4.1.1) and expectation that (4.20) holds true. Hence, (4.19) holds true. By definition of correlation (4.18), the proof is complete. □

The converse to Lemma 4.3.1 is true *only* for elliptical distributions.

Lemma 4.3.2 (McNeil et al. [173]). *If* $\mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right) = 0$ *then* $\mathbf{x}$ *and* $\mathbf{y}$ *are independent if and only if they have an elliptical joint probability distribution.*

The best-known member of the family of elliptical distributions is the normal distribution. Hence, it follows from the two preceding lemmata that independence is equivalent to uncorrelatedness in the Gaussian case.

Theorem 4.3.1. *If the joint probability distribution of* $\mathbf{x}$ *and* $\mathbf{y}$ *is normal, then* $\mathbf{x}$ *and* $\mathbf{y}$ *are independent if and only if* $\mathrm{Cor}\left(\mathbf{x}, \mathbf{y}\right) = 0$*.*

4.3.2 Partial correlation

Definition 4.3.3 (partial correlation)**.** *In a variable set* $\mathcal{V}$*, with cardinality* $\mathrm{Card}\left(\mathcal{V}\right) \geq 2$*, the* partial correlation *between two random variables* $\mathbf{x}, \mathbf{y} \in \mathcal{V}$*,* $\mathbf{x} \neq \mathbf{y}$*, given a (possibly empty) set of random variables* $\mathcal{Z} \subseteq \mathcal{V} \setminus \{\mathbf{x}, \mathbf{y}\}$*, denoted as* $\rho_{(\mathbf{x},\mathbf{y}|\,\mathcal{Z})}$*, is the correlation of* $\mathbf{x}$ *and* $\mathbf{y}$ *if* $\mathcal{Z}$ *is empty, and the correlation of the residuals* $\mathbf{r}_{(\mathbf{x}|\,\mathcal{Z})}$ *and* $\mathbf{r}_{(\mathbf{y}|\,\mathcal{Z})}$ *resulting from the linear regression of* $\mathbf{x}$ *on* $\mathcal{Z}$ *and of* $\mathbf{y}$ *on* $\mathcal{Z}$*, respectively, otherwise.*

Written more compactly, we have that:

$$\rho_{(\mathbf{x},\mathbf{y}|\,\mathcal{Z})} \equiv \begin{cases} \rho_{(\mathbf{x},\mathbf{y})} & \text{if } \mathcal{Z} = \emptyset\ , \\ \rho_{\left(\mathbf{r}_{(\mathbf{x}|\,\mathcal{Z})},\mathbf{r}_{(\mathbf{y}|\,\mathcal{Z})}\right)} & \text{otherwise.} \end{cases}$$

The set $\mathcal{Z}$ is referred to as the *conditioning set.* Its cardinality, denoted as q, is the *order* of the partial correlation. The corresponding correlation is referred to as a q-order partial correlation [30]. When $\mathcal{Z} = \mathcal{V} \setminus \{\mathbf{x}, \mathbf{y}\}$, the $(p-2)$-order partial correlation $\rho_{(\mathbf{x},\mathbf{y}|\,\mathcal{Z})}$ (where $\mathrm{Card}\left(\mathcal{Z}\right) = p$ is the total number of variables) is often referred to as the *full-order partial correlation* between $\mathbf{x}$ and $\mathbf{y}$.

4.3.2.1 **Algebraic formulas**

By definition, partial correlations of order $q = 0$ (i.e., conditioning on the empty set) are given by (4.18). Partial correlations of order $q > 0$ can be computed by means of a recursive formula or by matrix inversion.

For $\mathcal{Z} \subseteq \mathcal{V} \setminus \{\mathbf{x}, \mathbf{y}\}, \mathcal{Z} \neq \emptyset$, the *recursive formula* for the partial correlation coefficient $\rho_{(\mathbf{x},\mathbf{y}|\mathcal{Z})}$ is given [231] by:

$$\rho_{(\mathbf{x},\mathbf{y}|\mathcal{Z})} = \frac{\rho_{(\mathbf{x},\mathbf{y}|\mathcal{Z}\setminus\{\mathbf{z_0}\})} - \rho_{(\mathbf{x},\mathbf{z_0}|\mathcal{Z}\setminus\{\mathbf{z_0}\})}\,\rho_{(\mathbf{y},\mathbf{z_0}|\mathcal{Z}\setminus\{\mathbf{z_0}\})}}{\sqrt{\left(1-\rho^2_{(\mathbf{x},\mathbf{z_0}|\mathcal{Z}\setminus\{\mathbf{z_0}\})}\right)\left(1-\rho^2_{(\mathbf{y},\mathbf{z_0}|\mathcal{Z}\setminus\{\mathbf{z_0}\})}\right)}}\ , \quad \text{for any } \mathbf{z_0} \in \mathcal{Z}\ . \tag{4.24}$$

Hence, q-partial correlations can be computed from $(q-1)$-partial correlations, for $q = 1, \ldots, \mathrm{Card}(\mathcal{V}) - 2$.

Implementing (4.24) as a recursive algorithm yields an exponential time complexity. By using dynamic programming, however, the time complexity of the algorithm (Appendix F) can be brought down to $O\left(q^3\right)$.

Let Σ and Ω denote the covariance (4.16) and the concentration matrices (4.17) of the variables $\mathcal{Z} \cup \{\mathbf{x}, \mathbf{y}\}$, respectively. The partial correlation coefficient $\rho_{(\mathbf{x},\mathbf{y}|\mathcal{Z})}$ can also be obtained by *matrix inversion* [150]:

$$\rho_{(\mathbf{x},\mathbf{y}|\mathcal{Z})} = \frac{-\omega_{\mathbf{xy}}}{\sqrt{\omega_{\mathbf{xx}}\omega_{\mathbf{yy}}}}\ , \tag{4.25}$$

where $\omega_{\mathbf{xy}}$ is the element of the concentration matrix corresponding to the variables $\mathbf{x}$ and $\mathbf{y}$. From (4.25) we note that the partial correlation $\rho_{(\mathbf{x},\mathbf{y}|\mathcal{Z})}$ is zero if and only if the corresponding element of the concentration matrix is zero:

$$\rho_{(\mathbf{x},\mathbf{y}|\mathcal{Z})} = 0 \iff \omega_{\mathbf{xy}} = 0\ . \tag{4.26}$$

Computing partial correlations of order q with (4.25) requires the inversion of a $(q+2) \times (q+2)$ covariance matrix which has a time complexity of $O\left(q^3\right)$, similarly to the recursive formula.

However, there is an important difference between the recursive formula and the matrix inversion approach. Indeed, (4.24) only gives the value of $\rho_{(\mathbf{x},\mathbf{y}|\mathcal{Z})}$. On the other hand, the concentration matrix enables to compute all the $\binom{q+2}{2}$ possible partial correlations in $\mathcal{Z} \cup \{\mathbf{x}, \mathbf{y}\}$. Indeed, suppose that $\mathcal{Z} = \{\mathbf{z}\}$ and thus that $q = 1$. The entry corresponding to the variables $\mathbf{x}$ and $\mathbf{y}$ gives (after normalizing as in (4.25)) the value of $\rho_{(\mathbf{x},\mathbf{y}|\mathbf{z})}$ while the entries corresponding to the variables $\mathbf{x}$ and $\mathbf{z}$, and $\mathbf{y}$ and $\mathbf{z}$, give (after normalization) the values of $\rho_{(\mathbf{x},\mathbf{z}|\mathbf{y})}$ and $\rho_{(\mathbf{y},\mathbf{z}|\mathbf{x})}$, respectively.

Hence, the matrix inversion approach returns $\binom{q+2}{2}$ partial correlations of order q instead of one for the recursive formula with the same time complexity. This can be useful when many partial correlations (and not just one) have to be computed. However, note that when the value of q is small compared to p, the gain is often negligible (Section 6.6.2).

4.3.2.2 Partial correlation and conditional independence

Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)^T \in \mathbb{R}^p$ denote a p-dimensional random variable, indexed by $\mathcal{V} = \{1, \ldots, p\}$, of mean vector μ and positive definite covariance matrix Σ.

Proposition 4.3.1 (Lauritzen [150])**.** *Assume that $\mathbf{x} \sim \mathcal{N}_p(\mu, \Sigma)$, where Σ is regular.*[1] *Then it holds for $i, j \in \mathcal{V}$, $i \neq j$, that*

$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathcal{V}\setminus\{i,j\}} \iff \omega_{ij} = 0 \ ,$$

where $\Omega = \{\omega_{ij}\}_{i,j\in\mathcal{V}} = \Sigma^{-1}$ is the concentration matrix of the distribution.

In other words, two variables are conditionally independent (given the remaining variables) in the multivariate Gaussian case if and only if the corresponding entry in the concentration matrix is zero. This is true for any elliptical distribution [8]. From (4.26), we also have (in the elliptical case) that:

$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathcal{V}\setminus\{i,j\}} \iff \rho_{(i,j|\mathcal{V}\setminus\{i,j\})} = 0 \ . \tag{4.27}$$

4.3.3 Partial correlation and linear regression

We have already mentioned that correlation is a measure of linear dependence (Section 4.3.1.1). We now establish the connection between (partial) correlation and linear regression (Section 3.6).

Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)^T \in \mathbb{R}^p$ denote a p-dimensional random vector with zero mean. We regress each $\mathbf{x}_i$ in turn on all remaining variables, i.e., on the set $\{\mathbf{x}_j\}_{j\neq i}$. Hence, we have to determine the vectors $\beta^{(i)} = \left(\beta_1^{(i)}, \ldots, \beta_{i-1}^{(i)}, \beta_{i+1}^{(i)}, \ldots, \beta_p^{(i)}\right)^T$, $i = 1, \ldots, p$, where $\beta_j^{(i)}$ is the coefficient of variable $\mathbf{x}_j$ in the linear regression of $\mathbf{x}_i$. Note that there is no need for an intercept term (under squared loss) since $\mathbf{x}$ has zero mean (Section 3.6.1).

The linear predictor of $\mathbf{x}_i$ in terms of $\{\mathbf{x}_j\}_{j\neq i}$ that minimizes the squared error,

$$\beta^{(i)} = \underset{\beta\in\mathbb{R}^p}{\arg\min} \left(\mathbf{x}_i - \sum_{\substack{j=1 \\ j\neq i}}^{p} \mathbf{x}_j \beta_j^{(i)} \right)^2 \quad , \quad \forall\, i \in \{1, \ldots, p\} \ , \tag{4.28}$$

i.e., the "best" linear predictor under square loss, is given [42, 43] by:

$$\beta_j^{(i)} = \rho_{(i,j|\mathcal{K})} \sqrt{\frac{\operatorname{Var}(\mathbf{x}_i \mid \mathcal{K} \cup \{\mathbf{x}_j\})}{\operatorname{Var}(\mathbf{x}_j \mid \mathcal{K} \cup \{\mathbf{x}_i\})}} \ , \tag{4.29}$$

where $\mathcal{K} = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\} \setminus \{\mathbf{x}_i, \mathbf{x}_j\}$, and $\operatorname{Var}(\mathbf{x}_i \mid \mathcal{K} \cup \{\mathbf{x}_j\})$ and $\operatorname{Var}(\mathbf{x}_j \mid \mathcal{K} \cup \{\mathbf{x}_i\})$ are conditional variances. The *conditional variance* of a random variable $\mathbf{x}_i$ given a set $\mathcal{K}$ of random variables is defined as:

$$\operatorname{Var}(\mathbf{x}_i \mid \mathcal{K}) = \mathbb{E}\left((\mathbf{x}_i - \mathbb{E}(\mathbf{x}_i \mid \mathcal{K}))^2 \mid \mathcal{K} \right) \ . \tag{4.30}$$

[1] See Appendix G for definitions on matrices.

Since

$$\rho_{(i,j|\,\mathcal{K})} = \frac{-\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}} \quad \text{and} \quad \mathrm{Var}\left(\mathbf{x}_i \mid \mathcal{K} \cup \{\mathbf{x}_j\}\right) = \frac{1}{\omega_{ii}} \ ,$$

we have that

$$\beta_j^{(i)} = \frac{-\omega_{ij}}{\omega_{ii}} \quad \text{and} \quad \sqrt{\beta_j^{(i)}\beta_i^{(j)}} = \left|\rho_{(i,j|\,\mathcal{K})}\right| \ .$$

Hence, the parameters $\beta^{(i)}$, $i = 1, \ldots, p$, in (4.28), are determined by the covariance matrix (or, equivalently, the concentration matrix) of $\mathbf{x}$.

Linear regression and covariance matrices are thus intimately related. Therefore, Gaussian graphical model (GGM) selection (Section 4.6) in the "small n, large p" data setting (Section 4.7.1) can be performed by using robust estimators of the covariance matrix or by resorting to regularization techniques for linear regression.

4.3.4 Multiple correlation

Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)^T \in \mathbb{R}^p$ denote a p-dimensional random vector. The multiple correlation $R^2_{(1|\,2,\ldots,p)}$ between $\mathbf{x}_1$ and $\mathbf{x}_2, \ldots, \mathbf{x}_p$, which generalizes correlation to more than one regressor, is defined [231] as

$$R^2_{(1|\,2,\ldots,p)} = 1 - \frac{\mathrm{Var}\left(\mathbf{x}_1 \mid \mathbf{x}_2, \ldots, \mathbf{x}_p\right)}{\mathrm{Var}\left(\mathbf{x}_1\right)} \ ,$$

where the conditional variance $\mathrm{Var}\left(\mathbf{x}_1 \mid \mathbf{x}_2, \ldots, \mathbf{x}_p\right)$ is defined in (4.30). The multiple correlation represents the proportion of the variance of $\mathbf{x}_1$ explained by $\mathbf{x}_2, \ldots, \mathbf{x}_p$. Since

$$0 \le \mathrm{Var}\left(\mathbf{x}_1 \mid \mathbf{x}_2, \ldots, \mathbf{x}_p\right) \le \mathrm{Var}\left(\mathbf{x}_1\right) \ ,$$

we have that

$$0 \le R^2_{(1|\,2,\ldots,p)} \le 1 \ .$$

When $\mathbf{x}_1$ is a linear combination of $\mathbf{x}_2, \ldots, \mathbf{x}_p$, then the conditional variance is related to the (partial) correlations of the corresponding variables [231] as follows:

$$\mathrm{Var}\left(\mathbf{x}_1 \mid \mathbf{x}_2, \ldots, \mathbf{x}_p\right) = \mathrm{Var}\left(\mathbf{x}_1\right)\left(1 - \rho^2_{(1,2)}\right)\left(1 - \rho^2_{(1,3|\,2)}\right)\cdots\left(1 - \rho^2_{(1,p|\,2,3,\ldots,p-1)}\right) \ . \tag{4.31}$$

From (4.31), we see that the conditional variance decreases each time an additional variable is included, unless $\mathbf{x}_1$ and the added variable (say $\mathbf{x}_k$) are conditionally uncorrelated (i.e., $\rho_{(1,k|\,2,3,\ldots,k-1)} = 0$) in which case it is unchanged. Therefore:

$$R^2_{(1|\,2)} \le R^2_{(1|\,2,3)} \le \cdots \le R^2_{(1|\,2,3\ldots,p-1)} \le R^2_{(1|\,2,3\ldots,p)} \ . \tag{4.32}$$

Hence, adding a variable cannot decrease the multiple correlation.

If the regressor variables are uncorrelated [113] then

$$R^2_{(1|\,2,3\ldots,p)} = \sum_{i=2}^{p} \rho^2_{(1,i)} \ .$$

However, correlated variables are not always redundant [113]. The perhaps natural belief that

$$R^2_{(1|\,2,3\ldots,p)} \leq \sum_{i=2}^{p} \rho^2_{(1,i)}$$

is always verified is therefore *erroneous* [113]. Indeed, it sometimes happens [113] that

$$R^2_{(1|\,2,3\ldots,p)} > \sum_{i=2}^{p} \rho^2_{(1,i)} \; .$$

Consequently, adding a variable to the regression may increase the relevance of another variable. The added variable is referred to as a "suppressor" or a "masking variable" [231].

This means that although the overall contribution (in terms of variance reduction) of a set of variables cannot decrease when a new variable is added (compare (4.32)), the contribution of a variable (considered individually) can increase when a new variable is added.

4.3.5 Geometric interpretation

Both an algebraic and a geometric viewpoint can be taken with respect to linear models in general [119], and (partial) correlation in particular [235]. The advantages of presenting the linear model within the setting of Euclidian geometry have been emphasized by several authors [28, 61, 113, 119, 231, 235]. Geometric concepts have proven particulary helpful to gain an intuitive grasp of the relation between correlation and partial correlation coefficients [113, 235]. For example, the recursive algebraic formula for the partial correlation coefficient given by (4.24) can be easily derived from the following geometric interpretation [231].

Without loss of generality, we assume all variables to be centered. Given n observations $x_{11}, \ldots, x_{1n}$ of a variable $\mathbf{x}_1$, we can represent this variable as a n-dimensional vector $x_1 = (x_{11}, \ldots, x_{n1})$.

This representation leads to the following interpretations. The (sample) correlation between two variables $\mathbf{x}_1$ and $\mathbf{x}_2$ is the cosine of the angle α between their representative vectors x_1 and x_2 (Figure 4.1).

The (sample) multiple correlation between the variable $\mathbf{y}$ and two (explanatory) variables $\mathbf{x}_1$ and $\mathbf{x}_2$ is the cosine of the angle β between the representative vector y of $\mathbf{y}$ and its orthogonal projection y' on the plane spanned by the representative vectors of $\mathbf{x}_1$ and $\mathbf{x}_2$ (Figure 4.1).

The (sample) partial correlation between $\mathbf{x}_1$ and $\mathbf{x}_2$ given $\mathbf{y}$ is the cosine of the angle γ between the components of x_1 and x_2 orthogonal to y (Figure 4.2).

4.4 Notation and terminology on graphs

We follow closely the presentation in Lauritzen [150]. Some definitions are due to Diestel [57] or Castelo and Roverato [30].

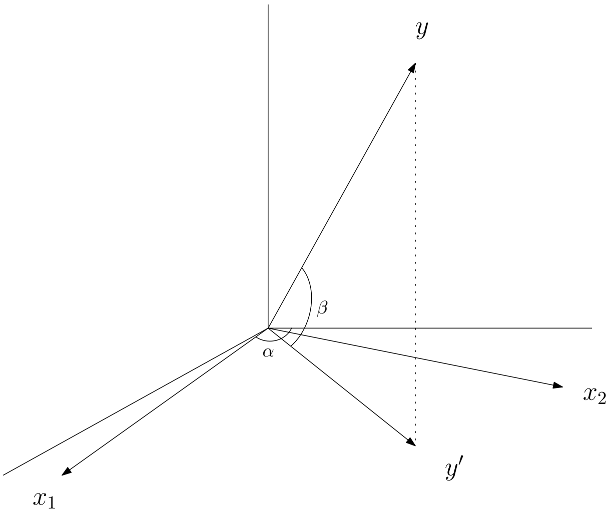


Figure 4.1: Correlation and multiple correlation illustrated with $n = 3$. See text for details.

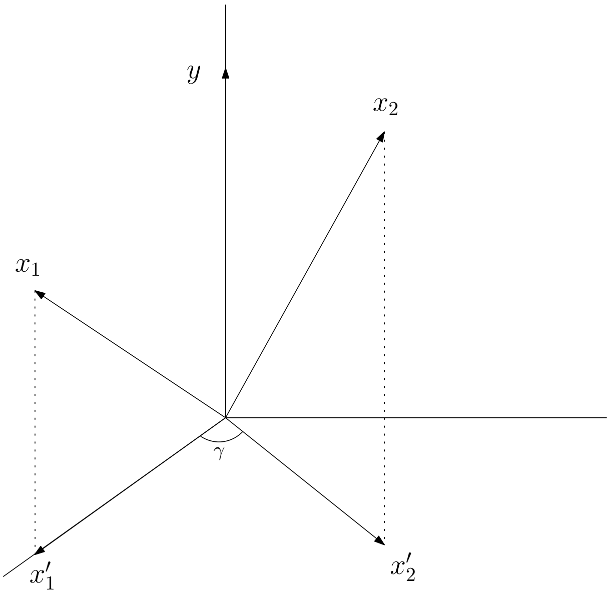


Figure 4.2: Partial correlation illustrated with $n = 3$. See text for details.

A *graph* is a pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of *vertices* (or *nodes*) and the set of *edges* $\mathcal{E}$ is a subset of the set $\mathcal{V} \times \mathcal{V}$ of ordered pairs of distinct vertices. We thus only consider *simple* graphs: there are no multiple edges and no loops.

Two graphs $G = (\mathcal{V}, \mathcal{E})$ and $G' = (\mathcal{V}', \mathcal{E}')$ are identical if $\mathcal{V} = \mathcal{V}'$ and $\mathcal{E} = \mathcal{E}'$. In this case, we simply write $G = G'$.

The set $\overline{\mathcal{E}}$ of *missing edges* of G is composed of the pairs $\{\alpha, \beta\}$ such that $\alpha \neq \beta$ and $\{\alpha, \beta\} \notin \mathcal{E}$.

An edge $\{\alpha, \beta\} \in \mathcal{E}$ is called *undirected* if $\{\beta, \alpha\} \in \mathcal{E}$ and is denoted by $\alpha \sim_G \beta$ (or, equivalently, $\beta \sim_G \alpha$). A graph having only undirected edges is an *undirected graph* (Figure 4.3). In such a graph, the edges are more conveniently represented as unordered pairs $\{\alpha, \beta\}$. Henceforth, we will consider only undirected graphs and refer to them as graphs.

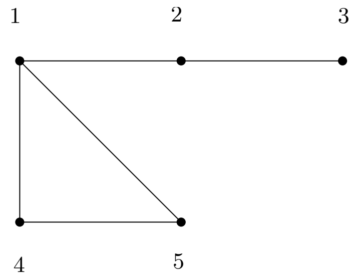


Figure 4.3: A simple (undirected) graph with node set $\mathcal{V} = \{1, \ldots, 5\}$ and edge set $\mathcal{E} = \{\{1,2\}, \{1,4\}, \{1,5\}, \{2,3\}, \{4,5\}\}$.

Two vertices α and β are said to be *adjacent* or *neighbours* if $\alpha \sim_G \beta$, and *non-adjacent* if $\alpha \nsim_G \beta$. The *boundary* of a vertex α, denoted as $\mathrm{bd}_G(\alpha)$, is the set of vertices that are neighbours to α. The *closure* of α is $\mathrm{cl}_G(\alpha) = \{\alpha\} \cup \mathrm{bd}_G(\alpha)$. In Figure 4.3, for example, $\mathrm{bd}_G(2) = \{1, 3\}$ and $\mathrm{cl}_G(2) = \{1, 2, 3\}$.

The definitions of boundary and closure are readily extended to a subset $\mathcal{A} \subseteq \mathcal{V}$. The expression $\mathrm{bd}_G(\mathcal{A})$ denotes the collection of nodes in the boundaries of vertices in $\mathcal{A}$ that are not themselves elements of $\mathcal{A}$:

$$\mathrm{bd}_G(\mathcal{A}) = \cup_{\alpha \in \mathcal{A}} \mathrm{bd}_G(\alpha) \setminus \mathcal{A} \ ,$$

and $\mathrm{cl}_G(\mathcal{A})$ denotes the collection of nodes in the closures of vertices in $\mathcal{A}$:

$$\mathrm{cl}_G(\mathcal{A}) = \cup_{\alpha \in \mathcal{A}} \mathrm{cl}_G(\alpha) \ .$$

The *edge proportion* γ_G is defined as the ratio between the number of edges and the number of possible edges of the graph:

$$\gamma_G = \frac{2 \times \mathrm{Card}\,(\mathcal{E})}{\mathrm{Card}\,(\mathcal{V})\,(\mathrm{Card}\,(\mathcal{V}) - 1)} \; .$$

A *sparse* graph is (informally) defined as a graph $G = (\mathcal{V}, \mathcal{E})$ in which the number of edges is much smaller than the possible number of edges:

$$\mathrm{Card}\,(\mathcal{E}) \ll \frac{\mathrm{Card}\,(\mathcal{V})\,(\mathrm{Card}\,(\mathcal{V}) - 1)}{2} \; .$$

If we let d_G denote the *average degree* of the nodes of the graph (i.e., the average number of neighbors of the graph's nodes),

$$d_G = \frac{1}{\mathrm{Card}\,(\mathcal{V})} \sum_{i \in \mathcal{V}} \mathrm{Card}\,(\mathrm{bd}_G\,(i)) \;\; ,$$

we have that the number of edges is given by

$$\mathrm{Card}\,(\mathcal{E}) = \frac{d_G \times \mathrm{Card}\,(\mathcal{V})}{2} \; ,$$

and thus the edge proportion can be expressed in terms of average degree and number of nodes as follows:

$$\gamma_G = \frac{d_G}{\mathrm{Card}\,(\mathcal{V}) - 1} \; . \tag{4.33}$$

A *path* of length n from α to β, denoted as $\alpha \mapsto_G \beta$, is a sequence $\alpha = \alpha_0, \ldots, \alpha_n = \beta$ of distinct vertices such that $\{\alpha_{i-1}, \alpha_i\} \in \mathcal{E}$ for all $i = 1, \ldots, n$. Since $\alpha \mapsto_G \beta$ implies $\beta \mapsto_G \alpha$, we write $\alpha \rightleftharpoons_G \beta$ (or, equivalently, $\beta \rightleftharpoons_G \alpha$). The relation $\rightleftharpoons$ is an equivalence relation and the corresponding equivalence classes $[\alpha]_G$, where

$$\beta \in [\alpha]_G \iff \alpha \rightleftharpoons_G \beta \; ,$$

are the *connectivity components* of G. If there is only one equivalence class, we say that G is *connected*. The graph in Figure 4.3 is connected.

For a pair of vertices (α, β) with $\alpha \neq \beta$, a set $\mathcal{C} \subseteq \mathcal{V}$ is said to be an (α, β)-*separator* if all paths from α to β intersect $\mathcal{C}$, i.e., have at least one vertex in $\mathcal{C}$. If either $\alpha \in \mathcal{C}$ or $\beta \in \mathcal{C}$ then we say that $\mathcal{C}$ is *trivial*. If no proper subset of $\mathcal{C}$ is a (α, β)-separator we say that $\mathcal{C}$ is *minimal*. We denote by $\mathcal{S}_G\,(\alpha, \beta)$ the set of all nontrivial minimal (α, β)-separators in G. Note that $\mathcal{S}_G\,(\alpha, \beta) = \{\emptyset\}$ if and only if α and β are in different connected components. In Figure 4.3, $\mathcal{S}_G\,(3, 4) = \{\{1\}, \{2\}\}$.

The subset $\mathcal{C}$ is said to *separate $\mathcal{A}$ from $\mathcal{B}$* if it is an (α, β)-separator for every $\alpha \in \mathcal{A}, \beta \in \mathcal{B}$. In Figure 4.3, the set $\{1\}$ separates the set $\{2, 3\}$ from the set $\{4, 5\}$.

The *connectivity* of α and β is the smallest cardinality of the sets in $\mathcal{S}_G\,(\alpha, \beta)$ and is denoted as $d_G\,(\alpha, \beta)$. In Figure 4.3, $d_G\,(3, 4) = 1$. It represents both the maximum number of independent paths between α and β in G and the minimum number of vertices that need to be removed from G to make α and β disconnected [57].

If $\mathcal{A} \subseteq \mathcal{V}$ is a subset of the vertex set, it induces a *subgraph* $G_{\mathcal{A}} = (\mathcal{A}, \mathcal{E}_{\mathcal{A}})$, where the edge set $\mathcal{E}_{\mathcal{A}} = \mathcal{E} \cap (\mathcal{A} \times \mathcal{A})$ is obtained from G by keeping edges with both endpoints in $\mathcal{A}$.

A graph is *complete* if all possible pairs of vertices form an (undirected or directed) edge. A subset is complete if it induces a complete subgraph. A *clique* is a maximal complete subset (with respect to $\subseteq$). The subset $\{1, 4, 5\}$ in Figure 4.3 is a clique.

To simplify the notation, we often drop the subscript whenever it is clear from the context which graph is referred to. So, for example, we write $\mathrm{bd}\,(\mathcal{A})$ instead of $\mathrm{bd}_G\,(\mathcal{A})$ whenever the reference to G is obvious.

4.5 Markov properties and faithfulness on undirected graphs

Let $G = (\mathcal{V}, \mathcal{E})$ denote an undirected graph on the nonempty set of random variables

$$\mathcal{V} = \{\mathbf{x}_i\}_{i=1,\dots,p} \ , \quad \text{where} \quad \mathbf{x}_i \in \mathbb{R} \ , \quad i = 1, \dots, p \ .$$

Let $F_{\mathcal{V}}$ denote the probability distribution of the random vector $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_p)$. Henceforth we do not distinguish between the random variable $\mathbf{x}_i$ and the corresponding node of the graph.

We now define the pairwise, local and global (undirected) Markov properties [150].

Definition 4.5.1 (pairwise, local and global Markov properties)**.** *The probability distribution $F_{\mathcal{V}}$ is said to obey*

- *the* pairwise Markov property *(P), relative to G, if for any pair $(\mathbf{x}_i, \mathbf{x}_j)$ of distinct non-adjacent vertices*

$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathcal{V} \setminus \{\mathbf{x}_i, \mathbf{x}_j\} \ ;$$

- *the* local Markov property *(L), relative to G, if for any vertex $\mathbf{x}_i \in \mathcal{V}$*

$$\mathbf{x}_i \perp\!\!\!\perp \mathcal{V} \setminus \mathrm{cl}_G\,(\mathbf{x}_i) \mid \mathrm{bd}_G\,(\mathbf{x}_i) \ ;$$

- *the* global Markov property *(G), relative to G, if for any triple $(\mathcal{A}, \mathcal{B}, \mathcal{S})$ of disjoint subsets of $\mathcal{V}$ such that $\mathcal{S}$ separates $\mathcal{A}$ from $\mathcal{B}$ in G*

$$\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{S} \ .$$

It can easily be shown [150] that the global Markov property (G) implies the local Markov property (L), which in turns implies the pairwise Markov property (P):

$$\text{(G)} \Longrightarrow \text{(L)} \Longrightarrow \text{(P)} \ . \tag{4.34}$$

For this reason, the global Markov property (G) is often simply referred to as *the* Markov property.

Further, it can be shown that the Markov properties are all equivalent under an additional constraint.

Theorem 4.5.1 (Pearl and Paz [189])**.** *If the probability distribution $F_{\mathcal{V}}$ is such that for all disjoint subsets $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ of $\mathcal{V}$ it holds that*

$$\textit{if } \mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid (\mathcal{C} \cup \mathcal{D}) \ \textit{ and } \mathcal{A} \perp\!\!\!\perp \mathcal{C} \mid (\mathcal{B} \cup \mathcal{D}) \ \textit{ then } \mathcal{A} \perp\!\!\!\perp (\mathcal{B} \cup \mathcal{C}) \mid \mathcal{D} \ , \tag{4.35}$$

then

$$\text{(G)} \Longleftrightarrow \text{(L)} \Longleftrightarrow \text{(P)} \ . \tag{4.36}$$

Proposition 4.5.1 (Lauritzen [150])**.** *Condition* (4.35) *holds if $F_{\mathcal{V}}$ has a positive and continuous density.*

Given that the density function of a multivariate normal distribution is positive and continuous [180], the Markov properties are all equivalent in the Gaussian case.

Corollary 4.5.1 (Lauritzen [150])**.** *If $F_{\mathcal{V}}$ is a multivariate normal distribution, then* (4.36) *holds.*

The probability distribution $F_{\mathcal{V}}$ is faithful to G if all the conditional independence relationships in $F_{\mathcal{V}}$ can be read off the graph G through the pairwise Markov property as formalized in the following definition [188].

Definition 4.5.2 (faithfulness)**.** *The probability distribution $F_{\mathcal{V}}$ is* faithful *to G if for all vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ and sets of vertices $\mathcal{K} \subseteq \mathcal{V} \setminus \{\mathbf{x}_i, \mathbf{x}_j\}$ with $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathcal{K}$ it holds that $\mathcal{K}$ separates $\mathbf{x}_i$ and $\mathbf{x}_j$ in G.*

When $F_{\mathcal{V}}$ is both Markov and faithful to G, then there is a one-to-one mapping between the graph and the conditional independences in the data, referred to as a *perfect map* [17]. Indeed, all the conditional independence relationships read off the graph G through the Markov property are present in $F_{\mathcal{V}}$ (Markov) and all the conditional independence relationships in $F_{\mathcal{V}}$ can be read off the graph G through the Markov property (faithfulness).

In the literature, the Markov and faithfulness assumptions are preconditions to prove correctness of algorithms [191]. A discussion on faithful distributions (and other types of distributions) with respect to properties of conditional independence can be found in Nilsson et al. [181].

4.6 Definition of Gaussian graphical model

The graphical interaction model for the multivariate normal distribution is called a *Gaussian graphical model* (GGM) and was first introduced by Dempster [52]. Standard textbooks on GGMs include Lauritzen [150], Chap. 5, Edwards [70], Chap. 3, and Whittaker [266], Chap. 6. The GGM has also been called a covariance selection model [52], a concentration graph model [43], and a Gaussian graphical Markov model [62]. We will sometimes refer to the GGM as the *concentration graph.*

Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph on the p-dimensional random variable $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)^T \in \mathbb{R}^p$. The GGM is defined as follows [150].

Definition 4.6.1 (Gaussian graphical model or concentration graph)**.** *The* Gaussian graphical model (GGM) *or* concentration graph *for $\mathbf{x}$ with undirected graph $G = (\mathcal{V}, \mathcal{E})$, denoted as $N_p(G)$, is given by assuming that $\mathbf{x}$ is distributed according to the p-variate normal distribution $\mathcal{N}_p(\mu, \Sigma)$, with mean $\mu = \mathbb{E}(\mathbf{x})$ and covariance matrix $\Sigma = \mathrm{Var}(\mathbf{x})$, which obeys the undirected pairwise Markov properties (Section* 4.5*) imposed by G:*

$$\alpha \nsim_G \beta \Longrightarrow \mathbf{x}_\alpha \perp\!\!\!\perp \mathbf{x}_\beta \mid \mathbf{x}_{\mathcal{V}\setminus\{\alpha,\beta\}} \ , \quad \forall\, \alpha, \beta \in \mathcal{V} \ , \tag{4.37}$$

where $\alpha \nsim_G \beta$ means that $\{\alpha, \beta\}$ does not form an edge in G (Section 4.4*).*

The density function of a multivariate normal distribution is positive and continuous [180]. Therefore, the restrictions imposed by the pairwise Markov properties (4.37) on the distribution of $\mathbf{x}$ imply, by Corollary 4.5.1, that $\mathbf{x}$ also obeys the undirected global and local Markov properties (Section 4.5) imposed by G.

Because of Proposition 4.3.1, the restrictions (4.37) can equivalently be expressed in terms of constraints on the concentration matrix (Definition 4.2.3):

$$\alpha \nsim_G \beta \Longrightarrow \omega_{\alpha\beta} = 0 \ , \quad \forall\, \alpha, \beta \in \mathcal{V} \ . \tag{4.38}$$

The constraints (4.38) justify the names of covariance selection model and concentration graph model given to $N_p\left(G\right)$ by Dempster [52] and Cox and Wermuth [43], respectively. The Gaussian graphical model on $\mathbb{R}^p$ induced by G can thus alternatively be defined as the family of multivariate normal distributions given by

$$N_p\left(G\right) = \left\{\mathcal{N}_p\left(\mu, \Sigma\right) \mid \mu \in \mathbb{R}^p,\, \Sigma^{-1} \in \mathcal{S}_p^+\left(G\right)\right\} \ , \tag{4.39}$$

where $\mathcal{S}_p^+\left(G\right)$ denotes the set of $p \times p$ symmetric positive definite matrices satisfying (4.38).

From (4.39), we see that the graph G acts as a *distribution filter* [17]: it restricts the family of p-variate normal distributions $\mathcal{N}_p\left(\mu, \Sigma\right)$, with Σ nonsingular, to the distributions for which $\Sigma^{-1} \in \mathcal{S}_p^+\left(G\right)$. In other words, it "filters out" the normal distributions that do not fulfill the conditions imposed on the concentration matrix.

4.7 Gaussian graphical model selection

Suppose we have a training set consisting of n i.i.d. observations

$$x_{1\cdot}, \ldots, x_{n\cdot} \sim \mathcal{N}_p\left(\mu, \Sigma\right) \ .$$

GGM selection (Section 3.1.3) consists in determining the graph G from the data. This involves finding the pattern of zeros in the inverse covariance matrix, since these zeros correspond to conditional independencies among the variables (4.38). From the previous section, we know that this can be based on an estimate of the concentration matrix or, equivalently, on the set of full-order partial correlations estimates $\left\{\hat{\boldsymbol{\rho}}_{(i,j|\mathcal{V}\setminus\{i,j\})}\right\}_{i,j\in\mathcal{V}}$.

4.7.1 The "small n, large p" data setting

Unfortunately, this is problematic in the "small n, large p" setting: the usual sample concentration matrix—the maximum likelihood estimate of the (population) concentration matrix—requires the sample covariance matrix to be positive definite and this holds, with probability one, if and only if $n > p$ [69].

To cope with this dimensionality issue, three methods have been proposed in the literature. The first one restricts the analysis to very small numbers of genes or gene clusters as to satisfy $n > p$ [138, 245, 246, 256, 273]. This approach avoids the issue at hand by solving a different problem. It is clearly unsatisfactory and will hence not be treated further.

The second approach, which we present in Chapter 5, uses *regularization* to infer robust estimators of the covariance matrix [45, 59, 145, 159, 174, 210, 212].

The third alternative, which is presented in Chapter 6, is to rely on *limited-order partial correlation graphs* [30, 51, 143, 144, 164, 267, 268], or *q-partial correlation graphs*, to approximate GGMs.

4.7.2 Model selection strategies

Once a robust estimate of the concentration matrix has been obtained, one has to find the pattern of zeros in it. Two model selection strategies, the constraint-based approach and the score-based search procedure, have been proposed in the literature [62]. A third one, the Bayesian approach [272] has only been used marginally so far [62] and will hence not be treated.

4.7.2.1 Constraint-based approach

The *constraint-based* approach consists in testing each of the $p\,(p-1)\,/2$ edges separately for inclusion by testing whether the corresponding full-order partial correlation is significantly different from zero [30, 62–64, 143, 174, 210, 212, 267]. Some authors [267] refer to it as the *hypothesis testing-based* approach. However, this is ambiguous since the search-based approach (Section 4.7.2.2) also uses hypothesis tests.

In a frequentist setting, this approach requires the distribution function of the sample full-order partial correlation $\hat{\boldsymbol{\rho}}_{(i,j|\mathcal{V}\setminus\{i,j\})}$ under the null hypothesis $\rho_{(i,j|\,\mathcal{V}\setminus\{i,j\})} = 0$ to address the statistical testing problem of non-zero full-order partial correlation:

$$H_0 : \rho_{(i,j|\,\mathcal{V}\setminus\{i,j\})} = 0 \qquad \text{versus} \qquad H_1 : \rho_{(i,j|\,\mathcal{V}\setminus\{i,j\})} \neq 0 \; . \tag{4.40}$$

Similarly to testing for zero correlation (Appendix H.1), a possible solution is to resort to Fisher's Z-transform of the full-order partial correlation:

$$Z_{(i,j|\,\mathcal{V}\setminus\{i,j\})} = \tanh^{-1} \hat{\boldsymbol{\rho}}_{(i,j|\mathcal{V}\setminus\{i,j\})} = \frac{1}{2}\log\left(\frac{1+\hat{\boldsymbol{\rho}}_{(i,j|\mathcal{V}\setminus\{i,j\})}}{1-\hat{\boldsymbol{\rho}}_{(i,j|\mathcal{V}\setminus\{i,j\})}}\right) \;,$$

which has an asymptotic normal distribution under the null hypothesis H_0 when the data follow a multivariate Gaussian distribution [6, 86, 87]. Using a significance level α, we reject the null-hypothesis H_0 against the two-sided alternative H_1 if

$$\sqrt{n-p-1}\, Z_{(i,j|\,\mathcal{V}\setminus\{i,j\})} > \Phi^{-1}\left(1-\alpha/2\right) \;, \tag{4.41}$$

where $\Phi\left(\cdot\right)$ denotes the cumulative distribution function of the standard normal distribution $\mathcal{N}\left(0,1\right)$.

Note that Drton and Perlman [62] use a different test than (4.40). In their approach, conservative simultaneous confidence intervals are computed for the entire set of full-order partial correlations. An edge is then included in the model if the corresponding confidence interval does not comprise 0.

Other tests include the generalized likelihood ratio test [150, 266–268] and the t-test for zero regression coefficients [30] (recall from Section 4.3.3 the connection between partial correlations and regression coefficients).

Alternatively, Schäfer and Strimmer [210, 212] fit a mixture of distributions model (where the null distribution is given by Hotelling [124] and the distribution of the true edges is a uniform distribution) to the observed partial correlation coefficients to take advantage of the networks' sparsity. Their method is inspired by similar approaches to detect differentially expressed genes (where it is assumed that the majority of investigated genes is not differentially expressed) [72, 75].

Subsequently, a *multiple testing correction* (Appendix H.2) procedure needs to be applied because of the parallel testing situation [67, 154]. Traditional techniques, which rely mostly on control of the family-wise error rate, are very conservative if the number of tests is large [67]. To alleviate this problem, Benjamini and Hochberg [14] proposed to control the *false discovery rate* (FDR; Appendix H.2) which measures the expected proportion of false positives out of the total number of rejections (instead of controlling the chance of any false positives). This approach and similar ones (such as *local fdr* [73]) have been widely used for GRN inference [67, 160, 192, 210, 212, 267, 268].

4.7.2.2 **Score-based search**

Score-based search procedures consider model selection as a combinatorial optimization problem. Models are selected by searching through the space of underlying graphs and maximizing a goodness-of-fit score, such as the Bayesian information criterion (BIC) [258], which evaluates the degree of fitness between a graph (in the search space) and the available data.

Testing all $2^{p(p-1)/2}$ possible graphs is of course hardly feasible, except for toy examples. Non-exhaustive search strategies have therefore been proposed. The search is usually performed greedily by defining a neighborhood structure for graphs and is terminated with a graph for which no neighboring graph achieves a higher score. Standard approaches include greedy stepwise forward-selection or backward-deletion [70]. In each step the edge selection or deletion (i.e., deciding whether a partial correlation is significantly different from zero or not) is typically done through hypothesis testing.

Contributions to Gene Regulatory Network Reverse Engineering from Gene Expression Data

5

Regularized Estimator for Gaussian Graphical Model Selection[1]

We propose an improved shrinkage estimator of the covariance matrix that corrects the bias of the optimal shrinkage intensity estimator of Ledoit and Wolf [153]'s shrinkage estimator through a parametric bootstrap approach. The applicability and usefulness of our estimator are demonstrated on both simulated and real expression data.

From the previous chapter, we know that Gaussian graphical model (GGM) selection reduces to estimating a covariance matrix. However, obtaining robust estimators for large empirical covariance matrices is a challenging and important issue, which has become particularly acute with the data flood phenomenon bioinformatics is experiencing for more than a decade now. Indeed, estimating large-scale covariance matrices is a common (though often implicit) task in functional genomics and transcriptome analysis [212]. Furthermore, in most cases, the available data describe a large number p of variables (on the order of hundreds or thousands) but only contain comparatively a small number n of samples (on the order of tens or hundreds), which renders this estimation an ill-posed problem.

The widely adopted solution is to rely either on the maximum likelihood estimate or on the related unbiased empirical covariance matrix (Section 5.2). Unfortunately, these commonly used estimators present serious defects in the "small n, large p" setting [153, 212]. To circumvent this problem, several regularization methods have been proposed since James and Stein [130] (Section 5.1).

In particular, Ledoit and Wolf [153] showed that they could improve the estimation of the covariance matrix by finding an optimal linear combination of the sample covariance matrix and a constrained covariance matrix, for which they provide an analytical solution (Section 5.2). Intuitively, their approach reduces to balancing bias and variance to reduce the mean squared error (MSE; recall Section 3.1.3). Unfortunately, the parameter defining shrinkage depends on unknown quantities and needs to be estimated consistently (Section 5.2.2).

[1] Parts of this chapter appeared in Kontos and Bontempi [145].

This chapter introduces the first main contribution of the thesis, which consists in an improved shrinkage estimator of the covariance matrix. We show that the optimal shrinkage intensity estimator of Ledoit and Wolf [153]'s shrinkage estimator (see also Schäfer and Strimmer [212]) is biased (Section 5.3). Consequently, we propose a parametric bootstrap approach (Section 3.2.1) to estimate this bias (Section 5.4) and derive a "bias-corrected" shrinkage estimator (Section 5.5). The applicability and usefulness of our estimator are demonstrated on both simulated and real expression data (Sections 5.6 and 5.7, respectively). Finally, Section 5.9 concludes the chapter.

5.1 Overview

We overview the main regularization approaches to obtain robust estimators of the covariance matrix, which includes the shrinkage estimator that we consider in the rest of the chapter (from Section 5.2 onwards).

Haff [111] showed that one can improve the estimation of the covariance matrix by using linear combinations of the sample covariance matrix and any positive semidefinite matrix (see also Anderson [5], Bickel and Levina [15], Daniels and Kass [44], Friedman [91]). This shrinkage approach was not new in statistical mathematics: it had already served previously, e.g., as original motivation for ridge regression (Section 3.6.3).

Recently, Ledoit and Wolf [151, 152, 153] proposed a shrinkage estimator, for the problem of portfolio selection in financial engineering, that is both statistically efficient and computationally fast. This approach was subsequently introduced in the bioinformatics literature to tackle the problem of large-scale genetic regulatory network inference from microarray data [212] and will be presented in Section 5.2.

Alternatively, Meinshausen and Bühlmann [174] use the lasso (Section 3.6.3). Indeed, partial correlation coefficients can be obtained from linearly regressing each variable in turn against the remaining ones (Section 4.3.3). Other regression techniques suited to the "small n, large p" setting have also been used [148, 157]. However, compared to Ledoit and Wolf [153]'s estimator, these approaches seemingly fail to uncover the topology of biological networks [212].

Another approach consists in using the resampling variance reduction technique [210, 271] of bootstrap aggregation or bagging (Section 3.2.1). The drawback of this method is that the sparsity is not accounted for when estimating the covariance matrix [158]. Consequently, although it provides better results than the sample covariance matrix, it has been shown to perform poorly compared to Ledoit and Wolf [153]'s shrinkage estimator [212]. Moreover, it is computationally much more expensive than the latter [212]. Note that the shrinkage estimator can be combined with bagging but empirical results suggest that this combination offers no further improvement beyond shrinkage [271].

Finally, robust estimators of the covariance matrix can be obtained through *penalized likelihood* maximization [9, 10, 45, 90, 156, 159, 275]. In particular, Banerjee et al. [9, 10] (see also d'Aspremont et al. [45], Friedman et al. [90]) proposed solving a maximum likelihood problem with an L_1-norm penalty term added to encourage sparsity in the inverse matrix:

$$\max_{X \in \mathcal{S}_p} \log \det(X) - \operatorname{tr}(SX) - \theta \|X\|_1 \ , \tag{5.1}$$

where $\mathcal{S}_p$ is the set of $p \times p$ symmetric positive definite covariance matrices, $\det(X)$ and $\operatorname{tr}(X)$ denote, respectively, the determinant and the trace of X, S is the sample covariance matrix, the term

$$\|X\|_1 = \sum_{i,j=1}^{p} |x_{ij}| \ ,$$

penalizes nonzero elements of X, and the scalar parameter $\theta > 0$ controls the trade-off between log-likelihood and the L_1-norm penalty (hence the sparsity of the solution).

The penalty term involving the sum of absolute values of the entries of X is used as a proxy for the number of its non-zero elements [10]. This penalization approach is similar to the regularization approaches for linear regression (Section 3.6.3).

Note that the classical maximum likelihood estimate of the covariance matrix Σ, i.e., the sample covariance matrix S, is recovered for $\theta = 0$ [10].

Yuan and Lin [275] showed that there is a close connection between the penalized-likelihood approach and Meinshausen and Bühlmann [174]'s method (see above)–the latter being, however, computationally faster than the former [275]. Hence, similarly to the techniques using regularization approaches to linear regression, the penalized-likelihood approach seemingly fails to uncover the topology of biological networks [212] and will not be treated further.

5.2 Shrinkage estimator

Let X denote a $n \times p$ matrix of n i.i.d. observations of p random variables with mean zero and covariance matrix Σ.

Let $\hat{\mathbf{S}}_{\mathrm{ML}}$ denote the maximum likelihood estimator of the covariance matrix Σ defined as

$$\hat{\mathbf{S}}_{\mathrm{ML}} = \frac{1}{n} X^T X \ ,$$

where X^T is the transpose of X. Let $\hat{\mathbf{S}}$ denote the related unbiased sample covariance matrix defined as

$$\hat{\mathbf{S}} = \frac{n}{n-1} \hat{\mathbf{S}}_{\mathrm{ML}} = \frac{1}{n-1} X^T X \ . \tag{5.2}$$

Despite being widely used, both estimators exhibit high variance [153]. To decrease their variance, and thus also to reduce their mean squared error (MSE), Ledoit and Wolf [153] proposed a (linear) *shrinkage estimator*, also known as a *biased estimator*. This estimator "shrinks" the sample covariance matrix $\hat{\mathbf{S}}$ towards a low-dimensional (biased) estimator $\hat{\mathbf{T}}$ of the covariance matrix Σ whose (i, j)-th element is defined by

$$\hat{\mathbf{t}}_{ij} = \begin{cases} \hat{\mathbf{s}}_{ii} & \text{if } i = j \ , \\ 0 & \text{if } i \neq j \ , \end{cases} \tag{5.3}$$

where $\hat{\mathbf{s}}_{ij}$ is the (i, j)-th element of $\hat{\mathbf{S}}$. The estimator $\hat{\mathbf{T}}$ is thus a diagonal matrix. We refer to Schäfer and Strimmer [212] for a list of commonly used low-dimensional estimators.

The linear shrinkage estimator $\hat{\mathbf{\Sigma}}_\lambda$ combines both estimators in a convex combination,[2] instead of choosing between one of these two extremes. The shrinkage estimator is defined as the linear combination of the estimators $\hat{\mathbf{S}}$ and $\hat{\mathbf{T}}$:

$$\hat{\mathbf{\Sigma}}_\lambda = \lambda \hat{\mathbf{T}} + (1 - \lambda) \hat{\mathbf{S}}, \tag{5.4}$$

where $\lambda \in [0, 1]$ represents the shrinkage intensity.

The number of parameters to be fitted in the constrained estimate $\hat{\mathbf{T}}$ is small compared to that of the unconstrained estimate $\hat{\mathbf{S}}$ (p parameters instead of $p\,(p+1)\,/2$). Hence, the constrained estimate $\hat{\mathbf{T}}$ will exhibit a lower variance than its unconstrained counterpart $\hat{\mathbf{S}}$. On the other hand, the former will exhibit considerable bias as an estimator of Σ (recall that the latter is unbiased).

The rationale behind the shrinkage estimator[3] is to minimize the MSE by finding the best trade-off between error due to bias and error due to variance (from (E.1) that the MSE can be decomposed in bias and variance terms). This idea of a trade-off between bias and variance can be traced back to the shrinkage technique of James and Stein [130].

5.2.1 Optimal shrinkage intensity

The *optimal* shrinkage intensity λ^* minimizes the expected quadratic loss:

$$\lambda^* = \underset{\lambda \in [0,1]}{\arg\min}\ \mathbb{E}\left(\left\| \hat{\mathbf{\Sigma}}_\lambda - \Sigma \right\|_F^2 \right) , \tag{5.5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, i.e.,

$$\|M\|_F = \sqrt{\operatorname{tr}\left(MM^T\right)} = \sqrt{\sum_{i=1}^{p} \sum_{j=1}^{p} m_{ij}^2} ,$$

and $\operatorname{tr}(\cdot)$ denotes matrix trace.

The value of λ^* is given by

$$\lambda^* = \frac{\sum_{i=1}^{p} \sum_{j=1}^{p} \left(\operatorname{Var}\left(\hat{\mathbf{s}}_{ij}\right) - \operatorname{Cov}\left(\hat{\mathbf{t}}_{ij}, \hat{\mathbf{s}}_{ij}\right)\right)}{\sum_{i=1}^{p} \sum_{j=1}^{p} \mathbb{E}\left(\left(\hat{\mathbf{t}}_{ij} - \hat{\mathbf{s}}_{ij}\right)^2\right)} , \tag{5.6}$$

where $\hat{\mathbf{s}}_{ij}$ is the (i, j)-th element of $\hat{\mathbf{S}}$. A derivation of (5.6) [153, 212] is given in Appendix I. It can be shown that λ^* always exists and is unique [153]. Note that this analytical expression is valid for any low-dimensional estimator of the covariance matrix. In the case where the constrained estimator is defined as in (5.3), then (5.6) reduces to

$$\lambda^* = \frac{\sum_i \sum_{j \neq i} \operatorname{Var}\left(\hat{\mathbf{s}}_{ij}\right)}{\sum_i \sum_{j \neq i} \mathbb{E}\left(\hat{\mathbf{s}}_{ij}^2\right)} . \tag{5.7}$$

[2] In practice, shrinkage is applied to the correlations rather than the covariances [212].

[3] Additionally, Ledoit and Wolf [153] proposed three other interpretations of the shrinkage approach: one involving eigenvalues of the covariance matrix, a Bayesian one, and one involving a projection in a Hilbert space.

The corresponding *optimal* shrinkage estimator of the covariance matrix

$$\hat{\mathbf{\Sigma}}^* \equiv \hat{\mathbf{\Sigma}}_{\lambda^*} = \lambda^* \hat{\mathbf{T}} + (1 - \lambda^*)\, \hat{\mathbf{S}} \;, \tag{5.8}$$

is referred to as *the* shrinkage estimator hereafter.

The *optimal* shrinkage estimator of the concentration matrix (Definition 4.2.3), simply referred to as *the shrinkage estimator of the concentration matrix*, is obtained by inverting $\hat{\mathbf{\Sigma}}^*$ as in (4.17):

$$\hat{\mathbf{\Omega}}^* = \left(\hat{\mathbf{\Sigma}}^*\right)^{-1} \;. \tag{5.9}$$

The *optimal* shrinkage estimator of the partial correlation $\rho_{(i,j|\mathcal{V}\backslash\{i,j\})}$ of variables $\mathbf{x}_i$ and $\mathbf{x}_j$ given the remaining variables $\mathcal{V}\backslash\{i,j\}$, simply referred to as *the shrinkage estimator of the partial correlation* $\rho_{(i,j|\mathcal{V}\backslash\{i,j\})}$, is obtained from (5.9) as in (4.25):

$$\hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{V}\backslash\{i,j\})} = \frac{-\hat{\boldsymbol{\omega}}^*_{ij}}{\sqrt{\hat{\boldsymbol{\omega}}^*_{ii}\hat{\boldsymbol{\omega}}^*_{jj}}} \;, \tag{5.10}$$

where $\hat{\boldsymbol{\omega}}^*_{ij}$ is the (i,j)-th element of $\hat{\mathbf{\Omega}}^*$.

Several insights into (5.6) can be given [212]. First, the shrinkage intensity diminishes when the variance of $\hat{\mathbf{S}}$ (the first term of the numerator of (5.6)) decreases. Hence, the influence of the target diminishes with increasing sample sizes.

Second, the shrinkage intensity decreases with increasing covariance between the two estimators (the second term of the numerator of (5.6)). This term adjusts for the fact that both estimators are inferred from the same data and that the prior information associated with $\hat{\mathbf{T}}$ is not independent of the data.

Finally, the shrinkage intensity diminishes with increasing mean squared difference between $\hat{\mathbf{S}}$ and $\hat{\mathbf{T}}$ (in the denominator of (5.6)). This protects the shrinkage estimate against an inappropriate target.

5.2.2 Estimating the optimal shrinkage intensity

In practice, one needs to obtain an estimate $\hat{\boldsymbol{\lambda}}^*$ of the optimal shrinkage intensity given by (5.7). This is achieved by replacing all expectations and variances in (5.7) by their unbiased sample counterparts [153, 212]:

$$\hat{\boldsymbol{\lambda}}^* = \frac{\sum_i \sum_{j \neq i} \widehat{\mathbf{Var}}\left(\hat{\mathbf{s}}_{ij}\right)}{\sum_i \sum_{j \neq i} \widehat{\mathbf{E}}\left(\hat{\mathbf{s}}^2_{ij}\right)} \;, \tag{5.11}$$

where

$$\widehat{\mathbf{E}}\left(\hat{\mathbf{s}}^2_{ij}\right) = \hat{\mathbf{s}}^2_{ij} \;,$$

and

$$\widehat{\mathbf{Var}}\left(\hat{\mathbf{s}}_{ij}\right) = \frac{n}{(n-1)^3} \sum_{k=1}^{n} \left(x_{ki}x_{kj} - \frac{1}{n} \sum_{k=1}^{n} x_{ki}x_{kj} \right)^2 \;, \tag{5.12}$$

with x_{ki} denoting the (k,i)-th element of X (i.e., the value of variable $\mathbf{x}_i$'s k-th sample). We refer to Appendix I for the derivation of (5.12).

Note that, in some finite samples, $\hat{\boldsymbol{\lambda}}^*$ may become negative or exceed one [212]. To avoid negative shrinkage or overshrinkage, $\hat{\boldsymbol{\lambda}}^*$ is truncated accordingly:

$$\hat{\boldsymbol{\lambda}}^*_{[0,1]} = \max\left(0, \min\left(1, \hat{\boldsymbol{\lambda}}^*\right)\right) \ .$$

5.2.3 Benefits

Ledoit and Wolf [153] showed that the shrinkage estimator $\hat{\boldsymbol{\Sigma}}^*$ (5.8) is both well-conditioned and more accurate than the sample covariance matrix $\hat{\mathbf{S}}$ asymptotically, and that the asymptotic results tend to hold well in finite samples. Further, they showed that $\hat{\boldsymbol{\Sigma}}^*$ outperforms several alternative state-of-the-art estimators to the sample covariance matrix $\hat{\mathbf{S}}$, including Haff's empirical Bayesian estimator, the so-called Stein-Haff estimator and the minimax (Ledoit and Wolf [153] and references therein).

Most importantly, by deriving the analytical solution (5.6), Ledoit and Wolf [153]'s method avoids computationally intensive procedures such as cross-validation [91, 243], bootstrap or Markov chain Monte Carlo (MCMC) methods.

As pointed out by Schäfer and Strimmer [212], the estimator proposed by Ledoit and Wolf [153] avoids all (main) drawbacks of most available estimators (see, e.g., Daniels and Kass [44] for an extensive review): it is not restricted to data with $p < n$, it does not assume specific underlying distributions and it is not computationally expensive.

5.3 Bias of the optimal shrinkage intensity estimator

Because of the linearity property of expectation, linear transformations of unbiased estimators are unbiased. Hence, the numerator and denominator of (5.11) are unbiased estimates of the numerator and denominator of (5.7), respectively. Unfortunately, the nonlinear transformation of unbiased estimators is not guaranteed to be unbiased. Hence, $\hat{\boldsymbol{\lambda}}^*$ is not necessarily an unbiased estimator of λ^* because of the quotient in (5.11). In fact, we will experimentally demonstrate in Sections 5.6 and 5.8 that it is biased.

5.4 Monte Carlo bias estimation

We present a Monte Carlo procedure to estimate the bias of the optimal shrinkage intensity estimator $\hat{\boldsymbol{\lambda}}^*$ (5.11) which is defined as

$$\text{Bias}\left(\hat{\boldsymbol{\lambda}}^*\right) = \mathbb{E}\left(\hat{\boldsymbol{\lambda}}^*\right) - \lambda^* \ . \tag{5.13}$$

Note that this estimation procedure will also be exploited by the estimator we propose (Section 5.5) and which attempts to correct this bias.

Suppose we have a $p \times p$ covariance matrix Σ. We generate B data matrices $X^{(b)}$, $b = 1, \ldots, B$, of dimension $p \times n$, where the n samples are drawn from the multivariate normal distribution with mean zero and covariance matrix Σ. Next, for all $X^{(b)}$, $b = 1, \ldots, B$, we compute the sample covariance matrices $\hat{\mathbf{S}}^{(b)}$, $b = 1, \ldots, B$, respectively, as in (5.2).

We then estimate the expected value of the optimal shrinkage intensity estimator (i.e., the first term of the right-hand side of (5.13)) by

$$\widehat{\mathbf{E}}\left(\hat{\boldsymbol{\lambda}}^*\right) = \frac{1}{B}\sum_{b=1}^{B}\hat{\boldsymbol{\lambda}}^*(b) \ , \tag{5.14}$$

where $\hat{\boldsymbol{\lambda}}^*(b)$ is the optimal shrinkage intensity estimator computed from the sample covariance matrix $\hat{\mathbf{S}}^{(b)}$ by (5.11).

Next, we determine the optimal shrinkage intensity λ^* (i.e., the second term of the right-hand side of (5.13)) by (5.7), where $\mathrm{Var}\left(\hat{\mathbf{s}}_{ij}\right)$ and $\mathbb{E}\left(\hat{\mathbf{s}}_{ij}\right)$ are given, respectively, by

$$\mathrm{Var}\left(\hat{\mathbf{s}}_{ij}\right) = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\mathbf{s}}_{ij}^{(b)} - \frac{1}{B}\sum_{b=1}^{B}\hat{\mathbf{s}}_{ij}^{(b)}\right)^2 \ , \quad i,j = 1,\ldots,p, \tag{5.15}$$

and

$$\mathbb{E}\left(\hat{\mathbf{s}}_{ij}^2\right) = \frac{1}{B}\sum_{b=1}^{B}\left(\hat{\mathbf{s}}_{ij}^{(b)}\right)^2 \ , \quad i,j = 1,\ldots,p. \tag{5.16}$$

Of course, (5.15) and (5.16) are estimators. However, since the number B of bootstrap replicates can be arbitrarily large, we can reasonably consider them as the true $\mathrm{Var}\left(\hat{\mathbf{s}}_{ij}\right)$ and $\mathbb{E}\left(\hat{\mathbf{s}}_{ij}\right)$, respectively.

Finally, plugging (5.15) and (5.16) into (5.7) gives:

$$\lambda^* = \frac{B\sum_i\sum_{j\neq i}\sum_{b=1}^{B}\left(\hat{\mathbf{s}}_{ij}^{(b)} - \frac{1}{B}\sum_{b=1}^{B}\hat{\mathbf{s}}_{ij}^{(b)}\right)^2}{(B-1)\sum_i\sum_{j\neq i}\sum_{b=1}^{B}\hat{\mathbf{s}}_{ij}^{(b)}} \ , \tag{5.17}$$

which, despite being an estimator (above), can be regarded as the “true” optimal shrinkage intensity [139].

Finally, we estimate the bias of the optimal shrinkage intensity estimator $\hat{\boldsymbol{\lambda}}^*$ by plugging (5.14) and (5.17) into (5.13):

$$\widehat{\mathbf{Bias}}\left(\hat{\boldsymbol{\lambda}}^*\right) = \widehat{\mathbf{E}}\left(\hat{\boldsymbol{\lambda}}^*\right) - \lambda^* \ . \tag{5.18}$$

5.5 Parametric bootstrap approach for bias correction

The bias estimation procedure we have presented in the previous section will be used to show the bias of the optimal shrinkage intensity estimator (5.11) in Section 5.6. It will also be used by the shrinkage estimator we now introduce and which attempts to correct for this bias.

The rationale behind this estimator is to estimate the bias of the standard shrinkage estimator. Indeed, if we knew its bias, $\mathrm{Bias}\left(\hat{\boldsymbol{\lambda}}^*\right)$, then

$$\hat{\boldsymbol{\lambda}}^*_{\mathrm{un}} = \hat{\boldsymbol{\lambda}}^* - \mathrm{Bias}\left(\hat{\boldsymbol{\lambda}}^*\right) \tag{5.19}$$

Algorithm 5.1: “Bias-corrected” shrinkage estimator.

```
Input: $p \times n$ data matrix $X$, number of bootstrap replications $B$.
Output: “Bias-corrected” shrinkage estimator $\hat{\boldsymbol{\Sigma}}^*_{\mathrm{bc}}$.
Estimate $\hat{\mathbf{S}}$ and $\hat{\mathbf{T}}$ as in (5.2) and (5.3), respectively ;
Estimate $\hat{\boldsymbol{\lambda}}^*$ as in (5.11) ;
Compute $\hat{\boldsymbol{\Sigma}}^*$ as in (5.4) with $\lambda = \hat{\boldsymbol{\lambda}}^*$ ;
Use $\hat{\boldsymbol{\Sigma}}^*$ to determine $\widehat{\mathbf{Bias}}_{\hat{\boldsymbol{\Sigma}}^*}\left(\hat{\boldsymbol{\lambda}}^*\right)$ (5.18) ;
$\hat{\boldsymbol{\lambda}}^*_{\mathrm{bc}} \longleftarrow \hat{\boldsymbol{\lambda}}^* - \widehat{\mathbf{Bias}}_{\hat{\boldsymbol{\Sigma}}^*}\left(\hat{\boldsymbol{\lambda}}^*\right)$ ;
$\hat{\boldsymbol{\Sigma}}^*_{\mathrm{bc}} \longleftarrow \hat{\boldsymbol{\lambda}}^*_{\mathrm{bc}}\hat{\mathbf{T}} + \left(1 - \hat{\boldsymbol{\lambda}}^*_{\mathrm{bc}}\right)\hat{\mathbf{S}}$ ;
return $\hat{\boldsymbol{\Sigma}}^*_{bc}$ ;
```

would be an unbiased estimator of λ^*, since (5.13)

$$\mathbb{E}\left(\hat{\boldsymbol{\lambda}}^*_{\mathrm{un}}\right) = \mathbb{E}\left(\hat{\boldsymbol{\lambda}}^*\right) - \left(\mathbb{E}\left(\hat{\boldsymbol{\lambda}}^*\right) - \lambda^*\right) = \lambda^* \; .$$

Of course, this bias is unknown and has to be estimated. Unfortunately, the bias estimation procedure described previously requires the “true” covariance matrix Σ (which we are trying to estimate) to be known. Thus, we face a circularity problem, namely that for an accurate estimate of the covariance matrix, a reliable estimate of the shrinkage intensity is needed, and vice versa.

However, although the shrinkage intensity estimator (and consequently the corresponding covariance matrix estimator) is biased, we can apply the bias estimation procedure described in Section 5.4 but with the covariance shrinkage estimator $\hat{\boldsymbol{\Sigma}}^*$ replacing the true covariance matrix Σ. Indeed, although $\hat{\boldsymbol{\Sigma}}^*$ is a biased estimator of Σ (because $\hat{\boldsymbol{\lambda}}^*$ is a biased estimator of λ^*, as will be shown in Section 5.8), it can be used as an approximation of Σ. Hence, we can reasonably assume that the bias estimated from $\hat{\boldsymbol{\Sigma}}^*$ instead of Σ as in Section 5.4,

$$\widehat{\mathbf{Bias}}_{\hat{\boldsymbol{\Sigma}}^*}\left(\hat{\boldsymbol{\lambda}}^*\right) = \widehat{\mathbf{E}}_{\hat{\boldsymbol{\Sigma}}^*}\left(\hat{\boldsymbol{\lambda}}^*\right) - \lambda^*_{\hat{\boldsymbol{\Sigma}}^*} \; ,$$

is close to the one we would obtain if we knew the true covariance matrix Σ.

The proposed “bias-corrected” optimal shrinkage intensity estimator is therefore obtained as in (5.19):

$$\hat{\boldsymbol{\lambda}}^*_{\mathrm{bc}} = \hat{\boldsymbol{\lambda}}^* - \widehat{\mathbf{Bias}}_{\hat{\boldsymbol{\Sigma}}^*}\left(\hat{\boldsymbol{\lambda}}^*\right) \; .$$

The detailed procedure to obtain the “bias-corrected” shrinkage estimator of the covariance matrix is given by Algorithm 5.1.

5.6 Experiments on synthetic gene regulatory networks

5.6.1 Data generation

We used the package GeneNet [184] for the statistical software R [196] to generate data sets with $p = 100$ variables as follows. First, we generated $R = 100$ covariance matrices $\Sigma^{(r)}_{p,\gamma_G}$,

$r = 1, \ldots, R$, of dimension $p \times p$ with edge proportion $\gamma_G \in \{0.01, 0.02, \ldots, 0.15\}$ (Table 5.1 gives the average node degree d_G (4.33) for each edge proportion γ_G) by an algorithm which guarantees that the resulting matrices are always positive definite [184, 210].

Table 5.1: Average node degree for each edge proportion $\gamma_G \in \{0.01, 0.02, \ldots, 0.15\}$.

Edge proportion (γ_G)	Average degree (d_G)
0.01	0.99
0.02	1.98
0.03	2.97
0.04	3.96
0.05	4.95
0.06	5.94
0.07	6.93
0.08	7.92
0.09	8.91
0.10	9.90
0.11	10.89
0.12	11.88
0.13	12.87
0.14	13.86
0.15	14.85

The algorithm generates a partial correlation matrix where the number of nonzero entries (outside the diagonal) is determined according to γ_G. Gaussian graphical model (GGM) theory (Section 4.6) shows that a nonzero entry in the partial correlation matrix implies that the two corresponding variables are dependent given the remaining variables (assuming a joint normal distribution) and form an edge in the corresponding GGM. Given that the partial correlation matrix is a “normalized” concentration matrix (4.25), the covariance matrix is derived from the inverse of the partial correlation matrix (4.17).

Although the model used for data generation is a simplification of real molecular processes, it is important to faithfully evaluate the prediction results. This is possible only if the true structure of the regulatory network is known.

Next, for each covariance matrix $\Sigma^{(r)}_{p,\gamma_G}$ and for each $n \in \{20, 40, 60, 80, 100, 1000\}$, $B = 1\,000$ data sets $X^{(r,b)}_{p,\gamma_G,n}$, $b = 1, \ldots, B$, of the desired sample size n were drawn from the multivariate normal distribution with mean zero and covariance matrix $\Sigma_{p,\gamma_G}^{(r)}$. All in all, we generated $15 \times 100 \times 6 \times 1\,000 = 9\,000\,000$ data sets.

5.6.2 Bias computation

For each covariance matrix $\Sigma_{p,\gamma_G}^{(r)}$ and for each n, we estimate the bias of $\hat{\boldsymbol{\lambda}}_{p,\gamma_G,n}^{*(r)}$ as described in Section 5.4:

$$\widehat{\mathbf{Bias}}\left(\hat{\boldsymbol{\lambda}}_{p,\gamma_G,n}^{*(r)}\right) = \widehat{\mathbf{E}}\left(\hat{\boldsymbol{\lambda}}_{p,\gamma_G,n}^{*(r)}\right) - \lambda_{p,\gamma_G,n}^{*(r)} \ , \qquad \forall r, p, \gamma_G, n \ , \tag{5.20}$$

where

$$\widehat{\mathbf{E}}\left(\hat{\boldsymbol{\lambda}}_{p,\gamma_G,n}^{*(r)}\right) = \frac{1}{B}\sum_{b=1}^{B} \hat{\boldsymbol{\lambda}}_{p,\gamma_G,n}^{*(r,b)} \ , \qquad \forall r, p, \gamma_G, n \ , \tag{5.21}$$

with $\hat{\boldsymbol{\lambda}}_{p,\gamma_G,n}^{*(r,b)}$ estimated from $X_{p,\gamma_G,n}^{(r,b)}$ as in (5.14), and where $\lambda_{p,\gamma_G,n}^{*(r)}$ is computed as in (5.17).

5.6.3 Bias correction

For each covariance matrix $\Sigma_{p,\gamma_G}^{(r)}$ and for each n, we apply Algorithm 1 to obtain an estimate $\hat{\boldsymbol{\Sigma}}_{\text{bc}}^{*}$ of the covariance matrix.

We compare the expected square loss of our "bias-corrected" shrinkage estimator $\hat{\boldsymbol{\Sigma}}_{\text{bc}}^{*}$ (Section 5.5) and of the "standard" shrinkage estimator $\hat{\boldsymbol{\Sigma}}^{*}$ (Section 5.2) across a wide range of parameters (Section 5.6.1). For each $\Sigma_{p,\gamma_G}^{(r)}$, we also infer the "optimal" covariance matrix $\hat{\boldsymbol{\Sigma}}_{\text{opt}}^{*}$, which is the covariance matrix obtained by using the "true" optimal lambda $\lambda_{p,\gamma_G,n}^{*(r)}$ (5.7) instead of $\hat{\boldsymbol{\lambda}}_{p,\gamma_G,n}^{*(r,b)}$.

As in Ledoit and Wolf [153], the benchmark is the expected square loss of the sample covariance matrix $\hat{\mathbf{S}}$. The expectations are computed (approximated) by averaging the losses over $R = 100$ Monte Carlo replications, and standard errors are also computed.

We also compute the percentage relative improvement (on the sample covariance matrix $\hat{\mathbf{S}}$) in average loss (PRIAL) of the three estimators $\hat{\boldsymbol{\Sigma}}^{*}$, $\hat{\boldsymbol{\Sigma}}_{\text{bc}}^{*}$ and $\hat{\boldsymbol{\Sigma}}_{\text{opt}}^{*}$. The PRIAL of an estimator $\hat{\boldsymbol{\Theta}}$, which was introduced by [153], is defined as:

$$\text{PRIAL}\left[\hat{\boldsymbol{\Theta}}\right] = 100 \times \frac{\left(\mathbb{E}\left(\left\|\hat{\mathbf{S}} - \Sigma\right\|_F^2\right) - \mathbb{E}\left(\left\|\hat{\boldsymbol{\Theta}} - \Sigma\right\|_F^2\right)\right)}{\mathbb{E}\left(\left\|\hat{\mathbf{S}} - \Sigma\right\|_F^2\right)} \ . \tag{5.22}$$

If the PRIAL is positive (negative), then $\hat{\boldsymbol{\Theta}}$ performs better (worse) than $\hat{\mathbf{S}}$. The PRIAL of the sample covariance matrix $\hat{\mathbf{S}}$ is zero by definition. The PRIAL cannot exceed 100% [153].

Finally, we also report the percentage of "wins," that is the proportion of Monte Carlo replications for which our "bias-corrected" shrinkage estimator produces a smaller square loss than the "standard" shrinkage estimator. Hence, when this value is greater (resp. smaller) than 50%, our estimator performs better (resp. worse) than the "standard" shrinkage estimator in the majority of cases.

5.7 Inferring the gene regulatory network of *Escherichia coli* from expression data

To illustrate the applicability of the proposed estimator, we apply it to the problem chosen by Schäfer and Strimmer [212] to illustrate the effectiveness of the shrinkage estimator in bioinformatics, namely the inference of (a part of) the genetic regulatory network (GRN) of *Escherichia coli* (*E. coli*) from a microarray data set.

The experiment (Schäfer and Strimmer [212] and references therein) measures the stress response of *E. coli* during expression of the recombinant protein SOD (human superoxide dismutase). The resulting data monitors all 4 289 protein coding genes of *E. coli* 8, 15, 22, 45, 68, 90, 150, and 180 minutes after induction of SOD. Among these genes, $p = 102$ were identified as differentially expressed in one or more of the $n = 8$ samples [212].

The objective is to infer the GRN among these 102 preselected genes [212] from the partial correlation matrix (Section 5.6.1). To identify the edges in the regulatory network from the partial correlation matrix (obtained by inverting the "bias-corrected" covariance matrix), we adopt a search heuristic which is based on large-scale multiple testing of edges using local fdr (Section 4.7.2.1), which returns a threshold to be applied on the partial correlation coefficients (i.e., pairs of genes whose partial correlation coefficients in absolute value are higher than the threshold are inferred as edges). Expected square losses can of course not be computed since we do not know the true covariance matrix.

5.8 Results and discussion

5.8.1 Synthetic gene regulatory networks

Table 5.2 shows the mean and standard deviation of the bias values $\left\{\widehat{\mathbf{Bias}}\left(\hat{\lambda}^{*(r)}_{p,\gamma_G,n}\right), r = 1, \ldots, R\right\}$, for each possible combination of γ_G and n (recall that $p = 100$). Figure 5.1 plots the mean of the bias values versus γ_G for each n (the standard deviations have been omitted for clarity).

The results suggest that the optimal shrinkage intensity estimator is biased in the "small n, large p" setting, particularly for small sample values. Further, we note that while the bias is positive for the smallest edge proportions, it becomes negative (i.e., the shrinkage intensity estimator underestimates the optimal intensity) for higher values.

Tables 5.3 and 5.4 present the results obtained with the four estimators $\hat{\mathbf{S}}$, $\hat{\mathbf{\Sigma}}^*$, $\hat{\mathbf{\Sigma}}^*_{\text{bc}}$ and $\hat{\mathbf{\Sigma}}^*_{\text{opt}}$ in terms of estimated square loss, PRIAL and "wins" (Section 5.6.3). For reasons of space and clarity, we only report the results for $n \in \{20, 40, 60\}$ (that we are interested by the behavior of the estimators in the "small n, large p" setting) and $\gamma_G \in \{0.01, 0.02, \ldots, 0.10\}$.

The results suggest that for small edge proportions (0.01 and 0.02), the "bias-corrected" shrinkage estimator $\hat{\mathbf{\Sigma}}^*_{\text{bc}}$ performs better than the "standard" shrinkage estimator $\hat{\mathbf{\Sigma}}^*$ except for the smallest sample size ($n = 20$). For edge proportions ranging from 0.04 to 0.06, $\hat{\mathbf{\Sigma}}^*_{\text{bc}}$ performs better for the smallest size ($n = 20$), while it outperforms $\hat{\mathbf{\Sigma}}^*$ for most sample sizes as the edge proportion increases (0.08, 0.09 and 0.10). Of course, the

Table 5.2: Mean (standard deviation) of bias values.

γ_G	Number of samples n		
	20	40	60
0.01	0.016 (0.0153)	0.022 (0.0071)	0.019 (0.0042)
0.02	0.015 (0.016)	0.026 (0.0085)	0.023 (0.0051)
0.03	-0.017 (0.004)	0.007 (0.0028)	0.011 (0.0024)
0.04	-0.029 (0.0021)	-0.003 (0.0016)	0.004 (0.0015)
0.05	-0.036 (0.0013)	-0.009 (0.0011)	-0.001 (0.001)
0.06	-0.040 (0.0011)	-0.012 (0.001)	-0.004 (0.001)
0.07	-0.042 (8e-04)	-0.015 (8e-04)	-0.006 (8e-04)
0.08	-0.044 (8e-04)	-0.016 (9e-04)	-0.008 (7e-04)
0.09	-0.045 (8e-04)	-0.018 (7e-04)	-0.009 (7e-04)
0.10	-0.046 (6e-04)	-0.019 (7e-04)	-0.010 (7e-04)
0.11	-0.047 (7e-04)	-0.019 (7e-04)	-0.011 (6e-04)
0.12	-0.047 (7e-04)	-0.020 (6e-04)	-0.011 (7e-04)
0.13	-0.048 (6e-04)	-0.020 (7e-04)	-0.012 (7e-04)
0.14	-0.048 (6e-04)	-0.021 (6e-04)	-0.012 (6e-04)
0.15	-0.048 (6e-04)	-0.021 (7e-04)	-0.012 (7e-04)
	80	100	1000
0.01	0.016 (0.0028)	0.014 (0.002)	0.002 (1e-04)
0.02	0.021 (0.0035)	0.018 (0.0025)	0.003 (1e-04)
0.03	0.012 (0.0018)	0.012 (0.0014)	0.003 (1e-04)
0.04	0.006 (0.0013)	0.007 (0.0011)	0.003 (1e-04)
0.05	0.002 (9e-04)	0.004 (9e-04)	0.002 (1e-04)
0.06	-4e-04 (8e-04)	0.002 (8e-04)	0.002 (2e-04)
0.07	-0.002 (8e-04)	-1e-04 (6e-04)	0.002 (2e-04)
0.08	-0.004 (7e-04)	-0.002 (6e-04)	0.002 (2e-04)
0.09	-0.005 (6e-04)	-0.003 (7e-04)	0.002 (2e-04)
0.10	-0.006 (7e-04)	-0.003 (6e-04)	0.002 (3e-04)
0.11	-0.006 (7e-04)	-0.004 (6e-04)	0.002 (3e-04)
0.12	-0.007 (7e-04)	-0.005 (6e-04)	0.002 (3e-04)
0.13	-0.007 (6e-04)	-0.005 (6e-04)	0.002 (3e-04)
0.14	-0.008 (7e-04)	-0.005 (6e-04)	0.001 (3e-04)
0.15	-0.008 (5e-04)	-0.006 (5e-04)	0.001 (3e-04)

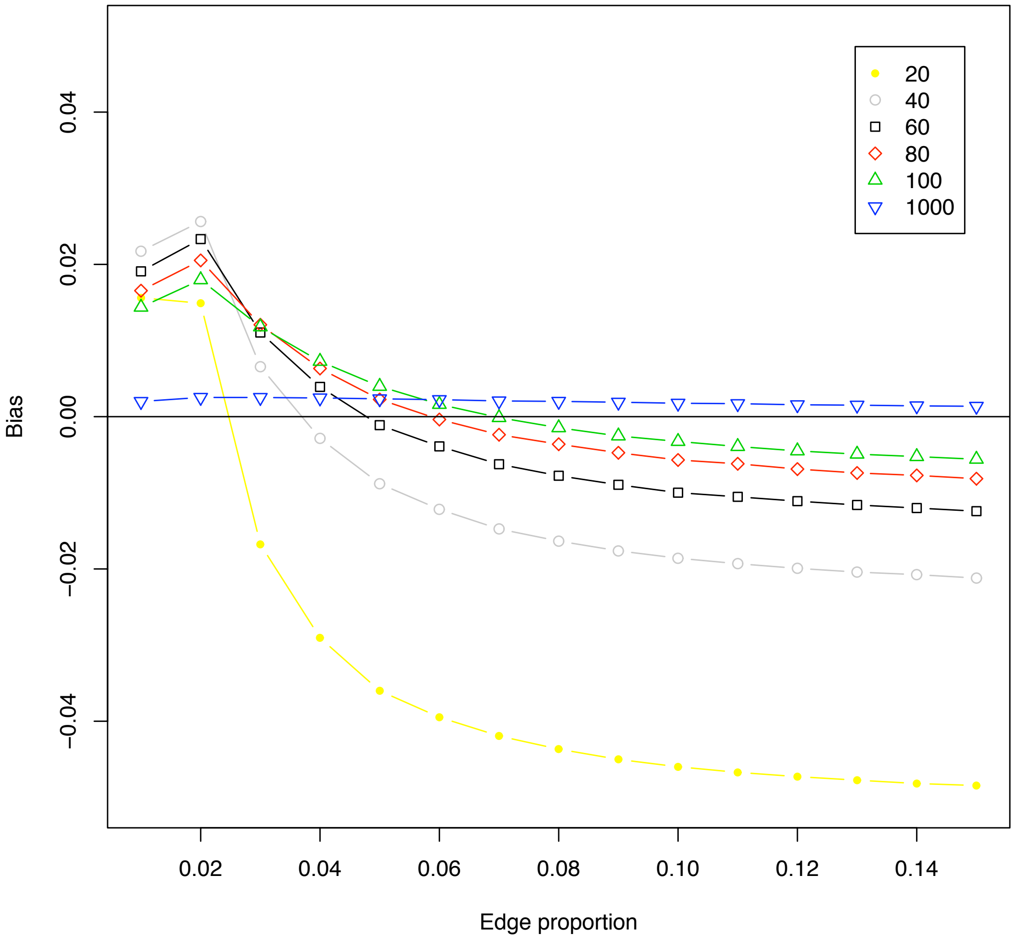


Figure 5.1: Plot of mean bias values (standard deviations are given in Table 5.2) versus edge proportion for different sample sizes.

Table 5.3: Mean (and standard error) of expected square loss and PRIAL (%) of the sample covariance matrix $\hat{\mathbf{S}}$, the "standard" shrinkage estimator $\hat{\mathbf{\Sigma}}^{*}$, the "bias-corrected" shrinkage estimator $\hat{\mathbf{\Sigma}}^{*}_{\text{bc}}$ and the "optimal" shrinkage estimator $\hat{\mathbf{\Sigma}}^{*}_{\text{opt}}$ for edge proportion $\gamma_G \in \{0.01, 0.02, \ldots, 0.05\}$. The percentage of "wins" (Section 5.6.3) is reported in the last column.

Edge prop.	Sample size	$\hat{S}$	$\hat{\Sigma}^{*}$	$\hat{\Sigma}^{*}_{\text{bc}}$	$\hat{\Sigma}^{*}_{\text{opt}}$	WINS
γ	n					
0.01	20	494.96 (0.043)	234.74 (0.049)	235.15 (0.049)	232.77 (0.049)	34.2 %
		0.0 %	52.574 %	52.491 %	52.972 %	
	40	240.34 (0.023)	154.21 (0.023)	153.79 (0.023)	153.27 (0.023)	64 %
		0.0 %	35.836 %	36.009 %	36.229 %	
	60	158.74 (0.014)	115.46 (0.014)	115.17 (0.014)	114.97 (0.014)	68.7 %
		0.0 %	27.266 %	27.449 %	27.57 %	
0.02	20	490.17 (0.042)	219.75 (0.049)	221.21 (0.05)	216.04 (0.046)	26.2 %
		0.0 %	55.169 %	54.871 %	55.925 %	
	40	237.78 (0.022)	148.26 (0.026)	148.07 (0.025)	145.89 (0.024)	66.3 %
		0.0 %	37.648 %	37.729 %	38.643 %	
	60	157.14 (0.014)	112.66 (0.018)	112.24 (0.017)	111 (0.016)	68.4 %
		0.0 %	28.306 %	28.569 %	29.36 %	
0.03	20	510.57 (0.02)	100.14 (0.011)	100.57 (0.011)	99.428 (0.011)	40.7 %
		0.0 %	80.386 %	80.302 %	80.526 %	
	40	248.27 (0.01)	82.914 (0.0085)	83.119 (0.0085)	82.565 (0.0084)	28.9 %
		0.0 %	66.603 %	66.52 %	66.744 %	
	60	164.33 (0.0068)	70.355 (0.0063)	70.404 (0.0063)	70.083 (0.0062)	28.1 %
		0.0 %	57.188 %	57.158 %	57.353 %	
0.04	20	514.14 (0.016)	68.474 (0.006)	68.329 (0.006)	67.674 (0.0059)	52 %
		0.0 %	86.682 %	86.71 %	86.837 %	
	40	250.41 (0.0083)	59.627 (0.0047)	59.755 (0.0047)	59.416 (0.0046)	35.8 %
		0.0 %	76.188 %	76.137 %	76.272 %	
	60	165.33 (0.0057)	53.092 (0.004)	53.151 (0.0041)	52.935 (0.004)	35.2 %
		0.0 %	67.888 %	67.852 %	67.983 %	
0.05	20	515.77 (0.014)	49.693 (0.0033)	49.24 (0.0033)	48.74 (0.0032)	62.4 %
		0.0 %	90.365 %	90.453 %	90.55 %	
	40	251.51 (0.0072)	44.556 (0.0027)	44.615 (0.0028)	44.369 (0.0027)	43.3 %
		0.0 %	82.284 %	82.261 %	82.359 %	
	60	166.41 (0.0051)	40.728 (0.0023)	40.763 (0.0024)	40.6 (0.0024)	41.8 %
		0.0 %	75.526 %	75.505 %	75.603 %	

Table 5.4: Mean (and standard error) of expected square loss and PRIAL (%) of the sample covariance matrix $\hat{\mathbf{S}}$, the "standard" shrinkage estimator $\hat{\mathbf{\Sigma}}^*$, the "bias-corrected" shrinkage estimator $\hat{\mathbf{\Sigma}}^*_{\text{bc}}$ and the "optimal" shrinkage estimator $\hat{\mathbf{\Sigma}}^*_{\text{opt}}$ for edge proportion $\gamma_G \in \{0.06, 0.07, \ldots, 0.10\}$. The percentage of "wins" (Section 5.6.3) is reported in the last column.

Edge prop. γ	Sample size n	$\hat{S}$	$\hat{\Sigma}^*$	$\hat{\Sigma}^*_{\text{bc}}$	$\hat{\Sigma}^*_{\text{opt}}$	WINS
0.06	20	516.95 (0.013)	39.275 (0.0024)	38.586 (0.0022)	38.21 (0.0022)	69.2 %
		0.0 %	92.403 %	92.536 %	92.608 %	
	40	251.83 (0.0068)	35.609 (0.0019)	35.628 (0.002)	35.424 (0.0019)	48.2 %
		0.0 %	85.86 %	85.852 %	85.933 %	
	60	166.21 (0.0044)	33.116 (0.0018)	33.151 (0.0018)	33.016 (0.0017)	39.8 %
		0.0 %	80.076 %	80.054 %	80.135 %	
0.07	20	517.03 (0.012)	31.441 (0.0014)	30.697 (0.0011)	30.359 (0.001)	71.4 %
		0.0 %	93.919 %	94.063 %	94.128 %	
	40	252.3 (0.0064)	28.776 (0.00097)	28.759 (0.00098)	28.585 (0.00094)	51.8 %
		0.0 %	88.595 %	88.601 %	88.67 %	
	60	166.71 (0.0045)	27.124 (0.0009)	27.144 (0.00091)	27.019 (0.00088)	45.6 %
		0.0 %	83.73 %	83.718 %	83.793 %	
0.08	20	517.95 (0.012)	26.246 (0.0014)	25.347 (0.00099)	25.071 (0.00089)	77.6 %
		0.0 %	94.933 %	95.106 %	95.16 %	
	40	252.47 (0.0061)	24.045 (0.00089)	24.01 (0.00088)	23.851 (0.00086)	52.9 %
		0.0 %	90.476 %	90.49 %	90.553 %	
	60	167.18 (0.0041)	22.854 (0.00079)	22.855 (0.00079)	22.754 (0.00078)	51.2 %
		0.0 %	86.33 %	86.329 %	86.39 %	
0.09	20	518.89 (0.012)	22.547 (0.0014)	21.517 (0.00096)	21.236 (0.00087)	81.3 %
		0.0 %	95.655 %	95.853 %	95.907 %	
	40	252.47 (0.006)	20.548 (0.00085)	20.503 (0.00085)	20.348 (0.00082)	57.3 %
		0.0 %	91.861 %	91.879 %	91.94 %	
	60	166.93 (0.0039)	19.63 (0.00076)	19.635 (0.00077)	19.539 (0.00076)	49.2 %
		0.0 %	88.241 %	88.238 %	88.295 %	
0.10	20	518.82 (0.011)	20.358 (0.0012)	19.321 (0.00062)	19.067 (0.00051)	82.5 %
		0.0 %	96.076 %	96.276 %	96.325 %	
	40	252.76 (0.0058)	18.553 (0.00057)	18.49 (0.00054)	18.348 (0.0005)	58.1 %
		0.0 %	92.66 %	92.685 %	92.741 %	
	60	167.15 (0.0038)	17.804 (0.0005)	17.807 (0.00051)	17.714 (0.00049)	49.6 %
		0.0 %	89.348 %	89.347 %	89.402 %	

"optimal" shrinkage estimator $\hat{\mathbf{\Sigma}}^*_{\text{opt}}$ always performs best. Since we are mainly interested in small sample sizes and edge proportions around 0.05, we can conclude that, in terms of mean squared error, the proposed estimator performs better than the standard one.

5.8.2 Gene regulatory network of *Escherichia coli*

The "standard" shrinkage approach yields an optimal shrinkage intensity of $\hat{\boldsymbol{\lambda}}^* = 0.180$, while our "bias-corrected" shrinkage estimator (with $B = 100$ bootstrap replications) gives $\hat{\boldsymbol{\lambda}}^*_{\text{bc}} = 0.015$. Hence $\widehat{\mathbf{Bias}}_{\hat{\mathbf{\Sigma}}^*}\left(\hat{\boldsymbol{\lambda}}^*\right) = 0.165$. With a cut-off of 0.2 on the local fdr [212], the resulting networks, denoted by $\mathcal{N}_{\text{S}}$ and $\mathcal{N}_{\text{bc}}$, contain 136 edges (2.64% of possible edges) and 160 edges (3.11% of possible edges), respectively. They are shown in Figures 5.2 and 5.3, respectively.

The covariance matrices $\hat{\mathbf{\Sigma}}^*$ and $\hat{\mathbf{\Sigma}}^*_{\text{bc}}$ (used to infer $\mathcal{N}_{\text{S}}$ and $\mathcal{N}_{\text{bc}}$, respectively) both have full rank (102) and are well-conditioned (condition numbers of 2.68 and 2.80, respectively). In contrast, the standard covariance matrix $\hat{\mathbf{S}}$ has only rank 8 and is ill-conditioned (infinite condition number). We already see the benefits of the shrinkage estimator for inferring the covariance matrix irrespective of the shrinkage approach (i.e., "standard" or "bias-corrected").

The gene sucA, which is involved in the citric acid cycle, has 25 neighbors in $\mathcal{N}_{\text{bc}}$ while it only has 18 neighbors in $\mathcal{N}_{\text{S}}$. The "hub" connectivity structure (pointed out by Schäfer and Strimmer [212]) of this gene is thus more pronounced with the network $\mathcal{N}_{\text{bc}}$ inferred with our method.

The edges connecting the genes lacA, lacZ and lacY in $\mathcal{N}_{\text{S}}$ and $\mathcal{N}_{\text{bc}}$ are the strongest (i.e., with the largest absolute values of partial correlation, and correspondingly also with the smallest local fdr values) in each network, respectively. This is interesting [212] since the experiment was based on these genes: lacA, lacY and lacZ are induced by IPTG (isopropyl-beta-D-thiogalactopyranoside) dosage and initiate recombinant protein synthesis [215]. Further, we note that lacZ and lacY have, respectively, 10 and 7 neighbors in $\mathcal{N}_{\text{bc}}$, while they only have, respectively, 8 and 6 neighbors in $\mathcal{N}_{\text{S}}$. In particular, lacZ, which is related to the genes cchB, nuoA and ibpA [215], is only 4 edges distant from the gene cchB in $\mathcal{N}_{\text{bc}}$ while it is 5 edges distant in $\mathcal{N}_{\text{S}}$ (there is no difference for nuoA and ibpA).

These results suggest that from a biological point of view, the benefits of the shrinkage estimator are more pronounced with the "bias-corrected" approach.

Note that the different shrinkage intensity values obtained with the "standard" and "bias-corrected" approaches, respectively, imply different rankings of the edges (based on the absolute values of partial correlation) because the covariance matrix is *inverted* to obtain the partial correlation matrix. So, for example, decreasing the cut-off on the local fdr to retain the top 125 edges with the "bias-corrected" approach will not yield the same network as $\mathcal{N}_{\text{S}}$.

Finally, in terms of performance, executing the "bias-corrected" method on the *E. coli* data set using the statistical software R required less than 10 seconds of CPU time on a 2.2 GHz Intel Core 2 Duo laptop with 2 GB RAM running Mac OS X. Although the "standard"

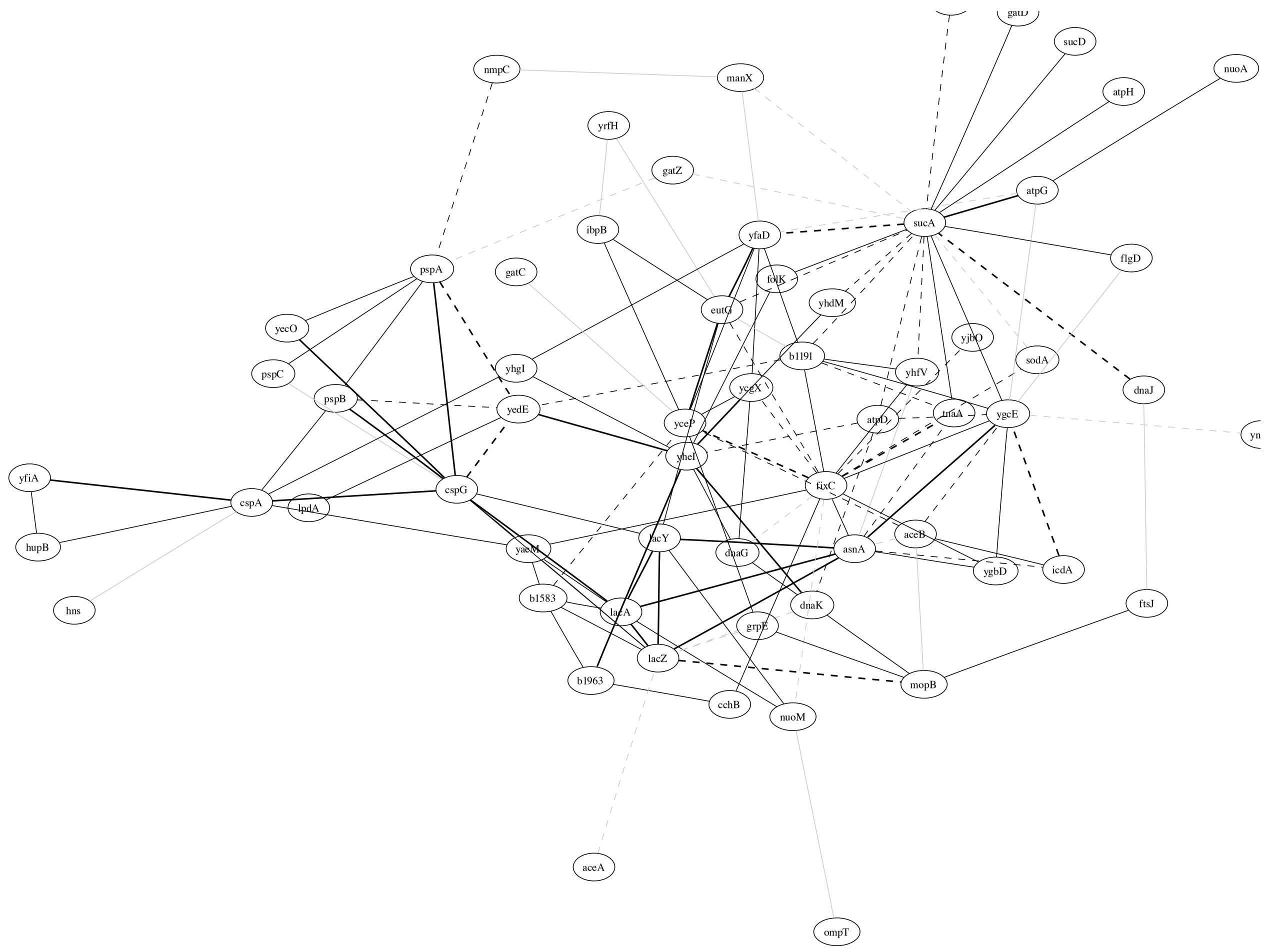


Figure 5.2: Gene networks of *E. coli* inferred by the “standard” shrinkage estimator. Full and dotted edges indicate positive and negative partial correlation, respectively. The edge thickness represents its weight in terms of partial correlation.

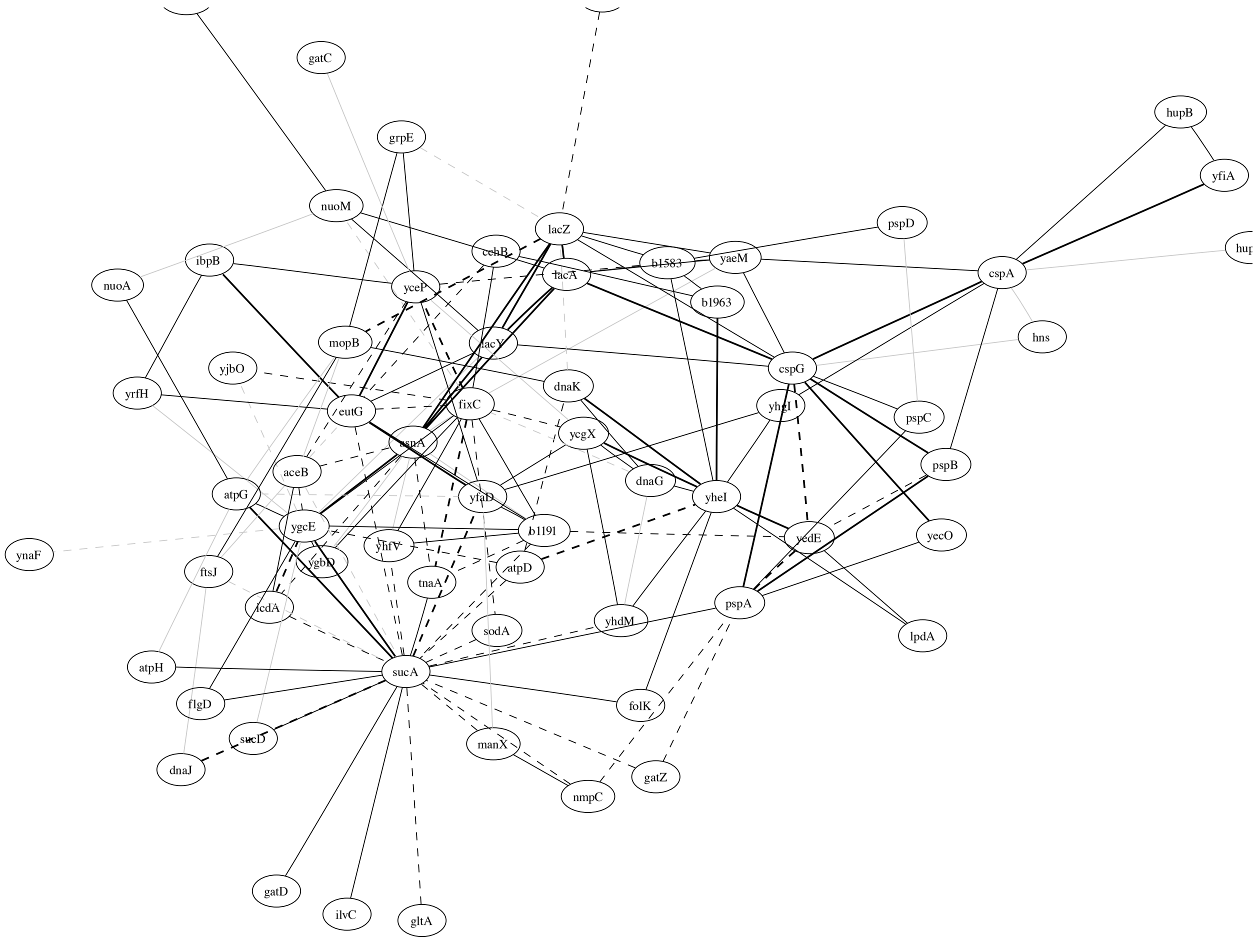


Figure 5.3: Gene networks of *E. coli* inferred by the "bias-corrected" shrinkage estimator. Full and dotted edges indicate positive and negative partial correlation, respectively. The edge thickness represents its weight in terms of partial correlation.

approach required less than 1 second, this example shows that the computational overhead of our approach is negligible when applied to real data.

5.9 Conclusion

Gaussian graphical models are widely used to infer large-scale GRNs from expression data. Unfortunately, in the “small n, large p” setting characteristic of microarray data, the usual estimator—the sample covariance matrix—is ill-suited.

First, we showed that the “standard” shrinkage estimator, despite successfully coping with the important challenge of inferring a well-conditioned covariance matrix in this setting, is biased.

Next, we proposed a “bias-corrected” shrinkage estimator based on a parametric bootstrap bias estimation procedure that improves upon the “standard” shrinkage estimator with negligible computational overhead.

We first illustrated the effectiveness of our covariance matrix estimator on synthetic data by showing that it improved upon the standard shrinkage estimator in terms of mean squared error.

We then assessed the ability of our estimator on a GRN inference task. For comparison purposes, we used the same inference task as in Schäfer and Strimmer [212], namely the reverse engineering of a subnetwork of *E. coli*’s GRN. The network inferred with our estimator has a more pronounced “hub topology,” as expected by biologists.

Since estimating large-scale covariance matrices is a common (though often implicit) task in functional genomics and transcriptome analysis, the proposed approach should be of interest to users and practitioners even outside the field of GRN reverse engineering.

Algorithm to Efficiently Infer q-Partial Correlation Graphs for Gaussian Graphical Model Selection[1]

We propose the q-nested procedure, an algorithm to efficiently infer q-partial correlation graphs for GGM selection. By adopting a screening procedure, we iteratively build nested graphs by discarding the less relevant edges. Moreover, by conditioning only on relevant variables, we diminish the problems related to multiple testing. We show that our algorithm outperforms state-of-the-art methods on simulated data.

Inferring Gaussian graphical models (GGMs), which are *full-order partial correlation graphs*, in the "small n, large p" setting prevalent in bioinformatics is an ill-posed problem (Section 4.7.1). In the previous chapter, we studied a first alternative to cope with this dimensionality issue, which consisted in the use of regularization to estimate the covariance matrix and we proposed a new shrinkage estimator.

A second alternative consists in approximating GGMs by *limited-order partial correlation graphs*, or *q-partial correlation graphs*. This approach was shown to be satisfactory for inferring biological networks [51, 255, 267].

This chapter introduces the second main contribution of the thesis, which consists in an algorithm to efficiently infer such graphs for GGM selection. We start with the presentation of independence graphs and 0-partial correlation graphs (Section 6.1). After highlighting their limitations to approximate GGMs, we introduce independence graphs of higher order and q-partial correlation graphs (Section 6.2). With the help of Castelo and Roverato [30]'s q-partial correlation graph theory (Section 6.3), we illustrate the effectiveness of q-partial correlation graphs to approximate GGMs. After emphasizing some serious problems encountered when inferring high-order partial correlation graphs (Section 6.4), we present our algorithm, the q-nested procedure, for coping with these issues (Sections 6.5). Instances of its application to simulated data are given in Section 6.7. The applicability and usefulness of our method are demonstrated on simulated data. In particular, we show that our algorithm outperforms state-of-the-art methods on simulated data.

[1] Parts of this chapter appeared in Kontos and Bontempi [143, 144].

6.1 Independence and correlation graphs

The first and simplest model to approximate GGMs to infer gene regulatory networks (GRNs) from "small n, large p" microarray data is that of *independence graph* where two nodes (genes) $\mathbf{x}_i$ and $\mathbf{x}_j$ are not connected if and only if they are (marginally) independent:

$$\mathbf{x}_i \nsim \mathbf{x}_j \iff \mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \ , \quad \forall i, j \in \{1, \ldots, p\} \ ,$$

where p is the number of genes.

In the bioinformatics literature, these graphs are known as *gene relevance networks* and were first introduced by Butte et al. [29] who used mutual information as a measure of (in)dependence.

In the multivariate normal case, the dependence relations are completely determined by the correlations between the variables (Section 4.3):

$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \iff \rho_{(i,j)} = 0 \ , \quad \forall i, j \in \{1, \ldots, p\} \ .$$

In this case, the independence graph is a *correlation graph* (also known as a *covariance graph*) where two nodes (genes) $\mathbf{x}_i$ and $\mathbf{x}_j$ are not connected if and only if their correlation is zero:

$$\mathbf{x}_i \nsim \mathbf{x}_j \iff \rho_{(i,j)} = 0 \ , \quad \forall i, j \in \{1, \ldots, p\} \ .$$

Inferring such a graph thus consists in determining the correlation matrix from data. Next, pairs of genes are connected if their respective correlation is significantly different from zero (in the sample case one has to resort to statistical testing).

Despite their relative ease of construction, these graphs suffer from a major drawback: they represent the marginal independence structure of the genes which is a strong indicator for independence, but a weak criterion for measuring dependence, since more or less all genes will be marginally (i.e., directly or indirectly) dependent [211] (Figure 6.1).

Furthermore, current biological knowledge suggests that genes do not interact in pairs independently of all the remaining genes [163]. Indeed, interactions between pairs of genes are influenced by other genes–hence the need for inferring large-scale GRNs (Chapter 1).

6.2 q-Partial (correlation) graphs

Due to the obvious limitations of independence graphs, some authors have inferred independence graphs of higher order, referred to as *q-partial graphs*. For an order $q \in \{0, \ldots, p-2\}$, two nodes (genes) $\mathbf{x}_i$ and $\mathbf{x}_j$ are not connected in such a graph if and only if there exists a conditioning subset $\mathcal{S}$ of the remaining genes of size at most q,

$$\mathcal{S} \subseteq \{1, \ldots, p\} \setminus \{i, j\} \quad \text{and} \quad \mathrm{Card}\,(\mathcal{S}) \leq q \ ,$$

such that $\mathbf{x}_i$ and $\mathbf{x}_j$ are conditionally independent with respect to $\mathcal{S}$:

$$\mathbf{x}_i \nsim \mathbf{x}_j \iff \mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathcal{S}} \ , \quad \forall i, j \in \{1, \ldots, p\} \ . \tag{6.1}$$

More formally, let $\mathbf{x}_{\mathcal{V}}$ be a random vector indexed by $\mathcal{V} = \{1, \ldots, p\}$ with probability distribution $F_{\mathcal{V}}$ and let $G = (\mathcal{V}, \mathcal{E})$ be the associated undirected graph.

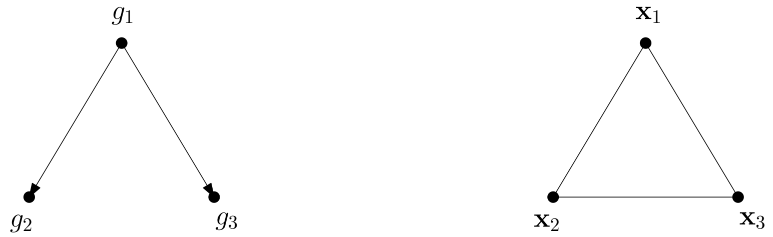


Figure 6.1: A simple gene regulatory network (GRN; Section 2.3) consisting of 3 genes (left). Note that genes are denote by g and not by $\mathbf{x}$ to emphasize that the variables are not the genes but their expression levels (which are denoted by $\mathbf{x}$ in the graph on the right). The arrow pointing from gene g_1 to gene g_2 (resp. g_3) means that g_1 regulates g_2 (resp. g_3). The expression levels $\mathbf{x}_2$ and $\mathbf{x}_3$ of, respectively, genes g_2 and g_3 are highly correlated with each other because g_2 and g_3 are both regulated by gene g_1. The spurious relation between the expression levels $\mathbf{x}_2$ and $\mathbf{x}_3$ will therefore be inferred in the independence graph (right). Note that the directions of the identified connections are not inferred in the independence graph (right).

Hypothesis 6.1. *We assume that $F_{\mathcal{V}}$ is both Markov and faithful (Section* 4.5*) with respect to G.*

For a subset $\mathcal{S} \subseteq \mathcal{V}$, we denote by $\mathbf{x}_{\mathcal{S}}$ the subvector of $\mathbf{x}$ indexed by $\mathcal{S}$, and by $F_{\mathcal{S}}$ the associated marginal distribution.

q-Partial graphs[2] are defined as follows [30].

Definition 6.2.1 (q-partial graph)**.** *For a random vector $\mathbf{x}_{\mathcal{V}}$ and an integer $0 \leq q \leq p-2$, the q-*partial graph *of $\mathbf{x}_{\mathcal{V}}$, denoted by $G_{(q)} = \big(\mathcal{V}, \mathcal{E}_{(q)}\big)$, is the undirected graph where the edge $\{i,j\} \notin \mathcal{E}_{(q)}$ if and only if there exists a (possibly empty) set $\mathcal{S} \subseteq \mathcal{V} \setminus \{i,j\}$ with cardinality* $\mathrm{Card}\,(\mathcal{S}) \leq q$ *such that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathcal{S}}$ holds in $F_{\mathcal{V}}$.*

We note that the q-partial graph generalizes the independence graph which is recovered by taking $q = 0$ (i.e., conditioning on the empty set).

In the sequel of the chapter, we make the following assumption.

Hypothesis 6.2. *We assume the vector $\mathbf{x}_{\mathcal{V}}$ to have a multivariate normal distribution with mean vector μ and positive definite covariance matrix Σ.*

In this case, the measure of partial (in)dependence is the partial correlation (4.27):

$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathcal{S}} \iff \rho_{(i,j|\mathcal{S})} = 0 \ . \tag{6.2}$$

The conditions (6.1) are thus equivalent to:

$$\mathbf{x}_i \nsim \mathbf{x}_j \iff \rho_{(i,j|\mathcal{S})} = 0 \ , \quad \forall i,j \in \{1,\ldots,p\} \ .$$

Under Hypothesis 6.2, the q-partial graph becomes a *q-partial correlation graph.*

[2] Recall that we only consider undirected graphs and do not distinguish between the edges $\{i,j\}$ and $\{j,i\}$.

Definition 6.2.2 (q-partial correlation graph)**.** *For a random vector* $\mathbf{x}_{\mathcal{V}} \sim \mathcal{N}_p(\mu, \Sigma)$ *and an integer* $0 \leq q \leq p-2$*, the* q-partial correlation graph *of* $\mathbf{x}_{\mathcal{V}}$*, denoted by* $G_{(q)} = \left(\mathcal{V}, \mathcal{E}_{(q)}\right)$*, is the undirected graph where the edge* $\{i,j\} \notin \mathcal{E}_{(q)}$ *if and only if there exists a (possibly empty) set* $\mathcal{S} \subseteq \mathcal{V} \setminus \{i,j\}$ *with cardinality* $\operatorname{Card}(\mathcal{S}) \leq q$ *such that* $\rho_{(i,j|\mathcal{S})} = 0$*.*

We note that the q-partial correlation graph generalizes the GGM (Definition 4.6.1) which is recovered by taking $q = p-2$ (i.e., conditioning on all remaining variables).

The use of q-partial (correlation) graphs with $q > 0$ has been limited to $q = 1$ [164, 267, 268] and $q = 2$ [51]. Due to computational issues (Section 6.4), larger values of q have only been used by Castelo and Roverato [30] who apply a randomization procedure (Section 6.3), and Kontos and Bontempi [143, 144] who apply a nested procedure that we present in this chapter (Section 6.5).

Authors relying on information-theoretic measures of (in)dependence have resorted to conditional mutual information [161, 276]. Note that it is also possible to combine partial correlation and information-theoretic (in)dependence measures [199].

6.3 The q-partial correlation graph theory

The use of q-partial correlation graphs has lead to the q-partial graph theory that we now present. This theory, introduced by Castelo and Roverato [30], provides a common framework for graphs inferred from q-partial correlation graphs and GGMs. In particular, it "clarifies the connection between the sparseness of the concentration graph [i.e., the GGM] and the usefulness of marginal distributions [...] under the assumption of faithfulness" [30] (Section 6.3.1).

6.3.1 Connection between GGMs and q-partial correlation graphs

We now clarify the connection between GGMs and q-partial correlation graphs. Under the assumption of faithfulness (Hypothesis 6.1), it can easily be shown [30, 267] that each edge of G is also an edge of $G_{(q)}$.

Proposition 6.3.1 (Castelo and Roverato [30])**.** *Let* $G = (\mathcal{V}, \mathcal{E})$ *and* $G_{(q)} = \left(\mathcal{V}, \mathcal{E}_{(q)}\right)$ *be the GGM and the q-partial correlation graph of* $\mathbf{x}_{\mathcal{V}}$*, respectively. Then*

$$\mathcal{E} \subseteq \mathcal{E}_{(q)} \ , \quad q = 0, \ldots, p-2 \ .$$

However, for $G_{(q)}$ to be useful as a proxy of G, we need to quantify the closeness of the two graphs by characterizing the missing edges of G that are also missing in $G_{(q)}$. To do so we first require two definitions due to Castelo and Roverato [30] which are related to the notion of connectivity (Section 4.4).

Definition 6.3.1 (outer connectivity)**.** *Let* $i \neq j$ *be two vertices of an undirected graph* $G = (\mathcal{V}, \mathcal{E})$*. The* outer connectivity *of* i *and* j *is defined as*

$$d_G^{out}(i,j) = d_{G_{ij}}(i,j) = \min_{S \in \mathcal{S}_{G_{ij}}(i,j)} \operatorname{Card}(S) \ ,$$

where $\mathcal{S}_{G_{ij}}(i,j)$ *is the set of all nontrivial minimal* (i,j)*-separators (Section* 4.4*) in* $G_{ij} = (\mathcal{V}, \mathcal{E} \setminus \{i,j\})$ *and* $\operatorname{Card}(S)$ *is the cardinality of* S*.*

Hence, $d_G^{out}(i,j)$ is the connectivity of i and j in the graph obtained by removing edge $\{i,j\}$, if present, from G (Figure 6.2 for an illustration).

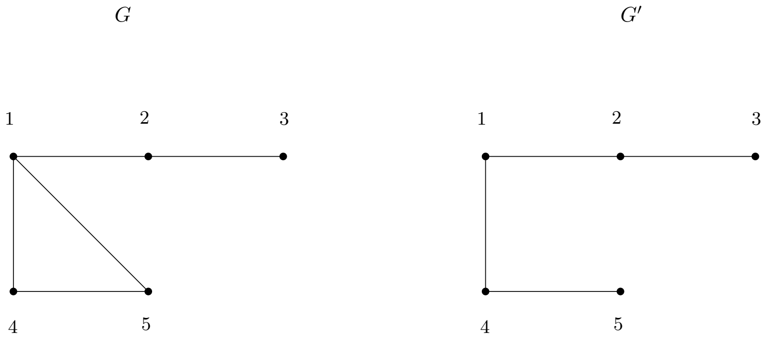


Figure 6.2: In graph G on the left, the outer connectivity of nodes 2 and 5 is equal to their connectivity, $d_G^{out}(2,5) = d_G(2,5) = 1$, because they do not form an edge. However, the outer connectivity of nodes 1 and 5 in G is equal to their connectivity in the graph G' on the right where the edge $\{1,5\}$ has been removed, $d_G^{out}(1,5) = d_{G'}(1,5) = 1$, which is different from their connectivity in G, $d_G(1,5) = 2$.

This definition of outer connectivity is extended to the set $\overline{\mathcal{E}}$ of missing edges of G as follows.

Definition 6.3.2 (outer connectivity of the missing edges). *The* outer connectivity of the missing edges *of* $G = (\mathcal{V}, \mathcal{E})$ *is defined as*

$$d_G^{out}\left(\overline{\mathcal{E}}\right) = \max_{\{i,j\}\in\overline{\mathcal{E}}} d_G^{out}(i,j) \ .$$

With these definitions in hand, we can introduce the following proposition which states that a missing edge in G is missing also in $G_{(q)}$ if and only if the outer connectivity of the corresponding vertices is smaller than or equal to q.

Proposition 6.3.2 (Castelo and Roverato [30]). *Let* $G = (\mathcal{V}, \mathcal{E})$ *and* $G_{(q)} = \left(\mathcal{V}, \mathcal{E}_{(q)}\right)$ *be the concentration and the q-partial correlation graph of* $\mathbf{x}_{\mathcal{V}}$*, respectively. If* $\{i,j\} \in \overline{\mathcal{E}}$ *then* $\{i,j\} \in \overline{\mathcal{E}}_{(q)}$ *if and only if*

$$d_G^{out}(i,j) \leq q \ . \tag{6.3}$$

In other words, a missing edge in G is missing also in $G_{(q)}$ if and only if there exists a marginal distribution of $\mathbf{x}_{\mathcal{V}}$ of dimension $(q+2)$ in which the variables are conditionally independent.

Intuitively, larger values of q should be preferred. This is confirmed by the following (edge) inclusion relation which derives from Proposition 6.3.2 and generalizes Proposition 6.3.1.

Corollary 6.3.1 (Castelo and Roverato [30]). *Let $G_{(q)} = \left(\mathcal{V}, \mathcal{E}_{(q)}\right)$ and $G_{(r)} = \left(\mathcal{V}, \mathcal{E}_{(r)}\right)$ be the q-partial and the r-partial graph of $\mathbf{x}_{\mathcal{V}}$, respectively. If $r \leq q$ then*

$$\mathcal{E} \subseteq \mathcal{E}_{(q)} \subseteq \mathcal{E}_{(r)} \ . \tag{6.4}$$

The inclusion relation (6.4) can be extended to the nodes' neighborhoods.

Corollary 6.3.2. *Let $G_{(q)} = \left(\mathcal{V}, \mathcal{E}_{(q)}\right)$ and $G_{(r)} = \left(\mathcal{V}, \mathcal{E}_{(r)}\right)$ be the q-partial and the r-partial graph of $\mathbf{x}_{\mathcal{V}}$, respectively. If $r \leq q$ then*

$$\mathrm{bd}_G(i) \subseteq \mathrm{bd}_{G_{(q)}}(i) \subseteq \mathrm{bd}_{G_{(r)}}(i) \ . \tag{6.5}$$

If (6.3) is satisfied for all the missing edges of G then the q-partial correlation graph is identical to G (Figure 6.3).

Proposition 6.3.3 (Castelo and Roverato [30]). *Let $G = (\mathcal{V}, \mathcal{E})$ and $G_{(q)} = \left(\mathcal{V}, \mathcal{E}_{(q)}\right)$ be the concentration and the q-partial correlation graph of $\mathbf{x}_{\mathcal{V}}$, respectively. Then $G = G_{(q)}$ if and only if*

$$d_G^{out}\left(\overline{\mathcal{E}}\right) \leq q \ .$$

Unfortunately, there is no direct connection between the degree of sparseness of G and the outer degree of its missing edges. Sparseness is only useful as long as it implies small separators (Section 4.4) for non-adjacent vertices [30]. However, one can easily find sparse graphs in which two non-adjacent vertices have a high value of outer connectivity. It is even possible to find examples for which the missing edges' outer connectivity for a graph G is less than that of a sparser graph, as illustrated in Figure 6.4.

6.3.2 Determining the usefulness of q-partial correlation graphs on $G_{(q)}$

The results presented so far provide necessary and sufficient conditions to determine the usefulness of q-partial correlation graphs. However, these conditions require G to be known. We now determine how information on the structure of G can be extracted from $G_{(q)}$ with a theorem and a corollary.

Theorem 6.3.1 (Castelo and Roverato [30]). *Let $G = (\mathcal{V}, \mathcal{E})$ and $G_{(q)} = \left(\mathcal{V}, \mathcal{E}_{(q)}\right)$ be the concentration and the q-partial correlation graph of $\mathbf{x}_{\mathcal{V}}$, respectively. If $\{i, j\} \in \mathcal{E}_{(q)}$ then a sufficient condition for the relation $\{i, j\} \in \mathcal{E}$ to hold true is*

$$d_{G_{(q)}}^{out}(i, j) \leq q \ . \tag{6.6}$$

Corollary 6.3.3 (Castelo and Roverato [30]). *Let $G = (\mathcal{V}, \mathcal{E})$ and $G_{(q)} = \left(\mathcal{V}, \mathcal{E}_{(q)}\right)$ be the concentration and the q-partial correlation graph of $\mathbf{x}_{\mathcal{V}}$, respectively. A sufficient condition for the relation $G = G_{(q)}$ to hold true is*

$$d_{G_{(q)}}^{out}\left(\overline{\mathcal{E}}_{(q)}\right) \leq q \ . \tag{6.7}$$

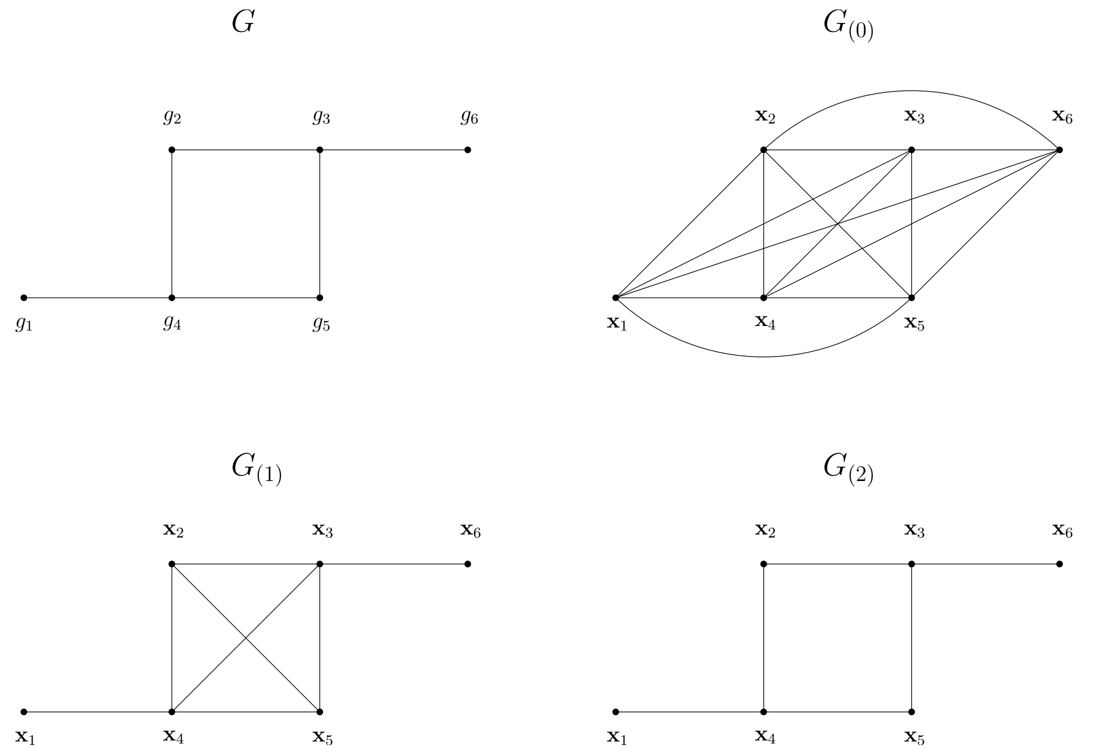


Figure 6.3: Outer connectivity illustrated. A simple GRN G consisting of 6 genes is depicted (upper left). Note that we have ignored the directionality of the edges. No pair of non-connected genes in G can be separated by the empty set (because G is connected), hence $G_{(0)}$ is the complete graph and is different from G. Some pairs of non-connected genes in G cannot be separated by any subset of at most 1 gene (for example $\mathbf{x}_3$ and $\mathbf{x}_4$) hence $G_{(1)} \neq G$. However all pairs of non-connected genes in G can be separated by at least one subset of at most 2 genes. Therefore we have $G_{(2)} = G_{(3)} = G_{(4)} = G$.

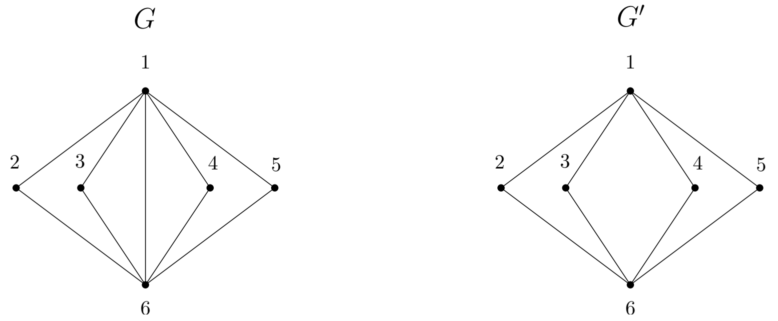


Figure 6.4: On the left, a graph $G = (\mathcal{V}, \mathcal{E})$ with $d_G^{out}\left(\overline{\mathcal{E}}\right) = 2$. On the right, a sparser graph $G' = (\mathcal{V}, \mathcal{E}')$, i.e., $\mathcal{E}' \subset \mathcal{E}$, with $d_{G'}^{out}\left(\overline{\mathcal{E}}\right) = 4$. The missing edge sets of G and G' are given by $\mathcal{E} = \{\{2,3\},\{2,4\},\{2,5\},\{3,4\},\{3,5\},\{4,5\}\}$ and $\mathcal{E}' = \mathcal{E} \cup \{\{1,6\}\}$, respectively.

Theorem 6.3.1 and Corollary 6.3.3 give weaker results than Propositions 6.3.2 and 6.3.3, respectively, since they only give sufficient conditions. However, they are of more practical use because conditions (6.6) and (6.7) can be checked on $G_{(q)}$.

Note however that the computation of the outer connectivity of two vertices is a NP-hard problem [30]. In practice, one has to derive upper and lower bounds to this number [202].

6.4 Issues arising when inferring q-partial correlation graphs

In principle, q-partial correlation graphs can be inferred for any $q \in \{0, \ldots, p-2\}$. However, two serious issues drastically hinder their applicability. First, the computation of $\binom{p-2}{q}$ q-partial correlations for each of the $p\,(p-1)\,/2$ possible pairs of genes is computationally intensive even for small networks, and often intractable for large networks, except if q takes on (very) small or (very) large values.

Second, an edge is added to the q-partial correlation graph if all of $\binom{p-2}{q}$ null hypotheses are rejected. But if the value of $\binom{p-2}{q}$ is large then most, or even all, of the edges are removed as the number of false negatives increases. Indeed, despite correcting for multiple testing (Appendix H.2), the probability that at least one hypothesis of zero q-order partial correlation is wrongly non-rejected increases dramatically with the number of performed tests [30].

Therefore, unless full-order partial correlations are considered (i.e., through regularization approaches; Chapter 5), the existing algorithms to reverse engineer limited-order partial correlation graphs [51, 164, 267, 268] are restricted to $q \leq 2$.

The single approach that allows for higher values of q (up to $q = 20$ for $p = 150$ genes) was proposed by Castelo and Roverato [30]. In a nutshell, their approach[3] consists, for each pair of genes, to compute only a small number (typically a few hundreds) of randomly chosen q-partial correlations instead of considering all $\binom{p-2}{q}$ q-partial correlations.

Unfortunately, their method suffers a considerable drawback. If two genes interact with each other “through” other genes, the probability of randomly selecting the subset that, conditioned upon, renders the two genes independent is extremely low. For example, suppose that two genes (g_1 and g_2) interact with each other “through” two other genes (g_3 and g_4) in a GRN of $p = 5000$ genes (Figure 6.5). Among 100 randomly selected subsets of size two, the probability that one of these subsets is composed of the two genes g_3 and g_4 is given by the probability mass function of the hypergeometric distribution (Appendix J):

$$\frac{\binom{2}{2}\binom{(5000-2)-2}{100-2}}{\binom{5000-2}{100}} = 3.96 \times 10^{-4} \,.$$

If the genes g_1 and g_2 interact “through” three (resp. four) other genes and if we randomly select subsets of size three (resp. four), the corresponding probability is given by 7.78×10^{-6} (resp. 1.51×10^{-7}).

[3] Castelo and Roverato [30] also define a new quantity, the non-rejection rate, that they use to address the statistical problem of zero q-partial correlation. However, when referring to Castelo and Roverato [30]'s approach, we only refer to the random selection of q-partial correlations.

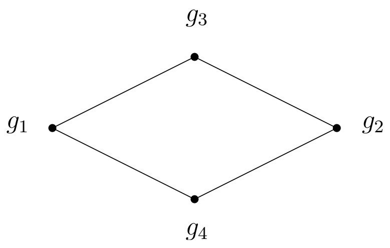


Figure 6.5: A simple GRN consisting of 4 genes. Note that we have ignored the directionality of the edges. Genes g_1 and g_2 interact with each other "through" genes g_3 and g_4. The remaining 4996 genes are not shown, neither the edges connecting genes g_1, g_2, g_3, g_4 with the remaining genes.

Hence, these problems need to be addressed for q-partial correlation graphs to be applicable in practice. It is the aim of our q-nested procedure (Section 6.5) to tackle both problems.

Note that two additional (albeit less serious) issues can arise. First, the assumption of faithfulness (Hypothesis 6.1) is sometimes violated. This implies that a missing edge in $G_{(q)}$ may be present in $G_{(r)}$, with $r > q$ (Figure 6.6). This undesirable effect is well-known in the literature on causal inference where it is referred to as the "explaining away effect" [187, 188, 227]. However, this problem has a weak impact on the estimates of partial correlation [30, 191]. In other words, if an edge is present in $G_{(r)}$ it is very unlikely, in practice, to be absent in $G_{(q)}$, with $r > q$. The assumption of faithfulness (Hypothesis 6.1) is thus reasonable [191]. Moreover, our q-nested procedure will tackle this issue as well.

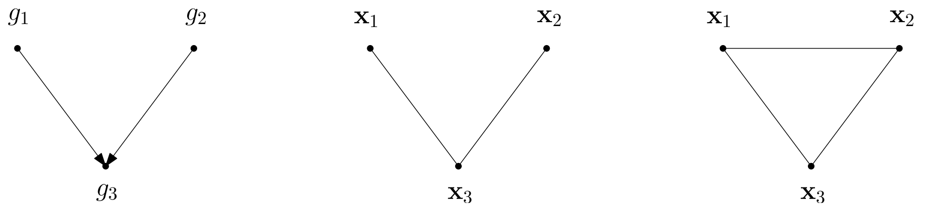


Figure 6.6: A simple GRN consisting of 3 genes (left). Gene g_3 is regulated by both genes g_1 and g_2, which are not correlated. Variables $\mathbf{x}_1$ and $\mathbf{x}_2$ are therefore not connected in the independence graph (center), i.e., the 0-partial graph. Because of the lack of faithfulness, a spurious connection between $\mathbf{x}_1$ and $\mathbf{x}_2$ will be inferred when conditioning on gene $\mathbf{x}_3$ in the 1-partial graph (right).

Finally, computing partial correlations of order q requires the inversion of an estimate of the covariance matrix estimated from an $n \times (q+2)$ data matrix, where n is the number of samples. Unfortunately, the sample covariance matrix is positive definite with probability one [69] if and only if

$$q < n - 2 \ . \tag{6.8}$$

This is, however, not problematic. Indeed, we will use the shrinkage estimator (Section 5.2), which can cope with dimensionality issues, to compute partial correlations. Moreover, condition (6.8) is not stringent since microarray data typically consist of several tens or hundreds of samples.

6.5 The q-nested procedure

Given the serious issues hindering the applicability of q-partial correlation graphs and the lack of methods to tackle these problems (Section 6.4), we present a new approach, the *q-nested procedure*, to infer q-partial (correlation) graphs.

We first assume that partial correlations are known (Section 6.5.1) before moving to the case where they have to be estimated from data (Section 6.5.2).

6.5.1 Population version

The two main characteristics of our procedure are as follows. First, we take advantage of the information provided by the $(q-1)$-partial correlation graph when inferring a q-partial correlation graph through the inclusion relation (6.4) to reduce the number of pairs of genes for which q-partial correlations have to be computed (Section 6.5.1.1). This reduces the computational issue (Section 6.4). Moreover, this procedure avoids the problems that may arise if the assumption of faithfulness (Hypothesis 6.1) is violated (Section 6.4).

Second, we prove that for any given pair of genes only a small number (out of $\binom{p-2}{q}$) of q-partial correlations have to be computed (Sections 6.5.1.2 and 6.5.1.3). More specifically, we show that for any given pair of genes (i, j), the conditioning sets of size q used to computed the q-partial correlations can be chosen in the smallest of i and j's neighborhoods (Section 6.5.1.2) or in the intersection of both neighborhoods (Section 6.5.1.3). This further reduces the computational issue and it also diminishes the problem related to multiple testing (hence decreasing the number of false negatives).

6.5.1.1 Edge screening

Recall from Corollary 6.3.1 that, under the assumption of faithfulness (Hypothesis 6.1), if $0 \le r \le q \le p-2$ then

$$\mathcal{E} \subseteq \mathcal{E}_{(q)} \subseteq \mathcal{E}_{(r)} \ .$$

This inclusion relation implies that every missing edge in $G_{(r)}$ is also missing in $G_{(q)}$. Hence, if we have inferred $G_{(r)}$, we only need to compute q-partial correlations for genes that form an edge in $G_{(r)}$ to infer $G_{(q)}$, and not for all $p\,(p-1)\,/2$ possible edges.

More specifically, let

$$\mathcal{E}^{(-1)} = \mathcal{V} \times \mathcal{V} = \{\{i,j\} \mid i,j \in \mathcal{V}, i \neq j\}$$

be the set of all possible edges (i.e., of all unordered pairs of genes). We can do a screening according to (marginal) correlations by building the set:

$$\mathcal{E}^{(0)} = \left\{\{i,j\} \in \mathcal{E}^{(-1)} \mid \rho_{(i,j)} = \rho_{(i,j|\emptyset)} \neq 0\right\} . \tag{6.9}$$

We then continue the screening using higher-order partial correlations and building the sets

$$\begin{aligned}\mathcal{E}^{(q+1)} &= \left\{\{i,j\} \in \mathcal{E} \mid \rho_{(i,j|\mathcal{S})} \neq 0, \text{ for all } \mathcal{S} \subseteq \mathcal{V} \setminus \{i,j\} \text{ with } \mathrm{Card}\,(\mathcal{S}) \leq q+1\right\} , \\ &= \left\{\{i,j\} \in \mathcal{E}^{(q)} \mid \rho_{(i,j|\mathcal{S})} \neq 0, \text{ for all } \mathcal{S} \subseteq \mathcal{V} \setminus \{i,j\} \text{ with } \mathrm{Card}\,(\mathcal{S}) = q+1\right\} ,\end{aligned} \tag{6.10}$$

for $q \in \{0, \ldots, p-3\}$, ending up with a nested sequence of sets:

$$\mathcal{E}^{(0)} \supseteq \mathcal{E}^{(1)} \supseteq \cdots \supseteq \mathcal{E}^{(k)} \supseteq \cdots \supseteq \mathcal{E}^{(p-2)} .$$

Assuming the underlying graph is sparse, this screening may substantially reduce the dimensionality of the problem.

6.5.1.2 Smallest neighborhood search

Recall from Section 4.4 that the boundary of vertex i in G is the set of vertices adjacent to i and is denoted by $\mathrm{bd}_G\,(i)$.

Proposition 6.5.1. *Let $\{i,j\} \in \mathcal{E}^{(q)}$ for a given $q \in \{0, \ldots, p-3\}$. If there exists a set $\mathcal{S}$, with $\mathrm{Card}\,(\mathcal{S}) = q+1$, such that $\rho_{(i,j|\mathcal{S})} = 0$ (and thus $\{i,j\} \notin \mathcal{E}^{(q+1)}$), then $\rho_{\left(i,j|\,\mathrm{bd}_{G_{(q)}}(i)\setminus\{j\}\right)} = 0$ and $\rho_{\left(i,j|\,\mathrm{bd}_{G_{(q)}}(j)\setminus\{i\}\right)} = 0$.*

Proof: Since $\{i,j\} \notin \mathcal{E}^{(q+1)}$ there exists a subset $\mathcal{S} \subseteq \mathcal{V} \setminus \{i,j\}$, with $\mathrm{Card}\,(\mathcal{S}) = q+1$ (recall that $\{i,j\} \in \mathcal{E}^{(q)}$), such that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathcal{S}}$. By Theorem 4.5.1 and Corollary 4.5.1, the Markov properties are all equivalent under the assumption of normality (Hypothesis 6.2). Hence, we have that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathrm{bd}_G(i)}$ and $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathrm{bd}_G(j)}$. By Corollary 6.3.2, we have that $\mathrm{bd}_G\,(i) \subseteq \mathrm{bd}_{G_{(q)}}\,(i)$ and $\mathrm{bd}_G\,(j) \subseteq \mathrm{bd}_{G_{(q)}}\,(j)$. Since $\{i,j\} \notin \mathcal{E}$ (recall that $\{i,j\} \notin \mathcal{E}^{(q+1)}$), it follows that $j \notin \mathrm{bd}_G\,(i)$ and $i \notin \mathrm{bd}_G\,(j)$, and thus that $\mathrm{bd}_G\,(i) \subseteq \mathrm{bd}_{G_{(q)}}\,(i) \setminus \{j\}$ and $\mathrm{bd}_G\,(j) \subseteq \mathrm{bd}_{G_{(q)}}\,(j) \setminus \{i\}$. We hence have that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathrm{bd}_{G_{(q)}}(i)\setminus\{j\}}$ and $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathrm{bd}_{G_{(q)}}(j)\setminus\{i\}}$, which under Hypothesis 6.2 is equivalent to $\rho_{\left(i,j|\,\mathrm{bd}_{G_{(q)}}(i)\setminus\{j\}\right)} = 0$ and $\rho_{\left(i,j|\,\mathrm{bd}_{G_{(q)}}(j)\setminus\{i\}\right)} = 0$. □

Proposition 6.5.2. *Let $\{i,j\} \in \mathcal{E}^{(q)}$ for a given $q \in \{0, \ldots, p-3\}$. If*

$$\min\left(\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}\,(i)\right), \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}\,(j)\right)\right) \leq q+1 ,$$

then $\{i,j\} \in \mathcal{E}^{(q+1)}$.

Proof: Without loss of generality, assume that $\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right) \leq q+1$. By contraposition. Assume $\{i,j\} \notin \mathcal{E}^{(q+1)}$. Hence, there exists a subset $\mathcal{S} \subseteq \mathcal{V} \setminus \{i,j\}$, with $\mathrm{Card}(\mathcal{S}) = q+1$ (recall that $\{i,j\} \in \mathcal{E}^{(q)}$), such that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathcal{S}}$, which implies that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathrm{bd}_G(i)}$ by Corollary 4.5.1. By Corollary 6.3.2, we have that $\mathrm{bd}_G(i) \subseteq \mathrm{bd}_{G_{(q)}}(i)$. Since $\{i,j\} \notin \mathcal{E}$ (recall that $\{i,j\} \notin \mathcal{E}^{(q+1)}$), it follows that $j \notin \mathrm{bd}_G(i)$, and thus that $\mathrm{bd}_G(i) \subseteq \mathrm{bd}_{G_{(q)}}(i) \setminus \{j\}$. We hence have that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathrm{bd}_{G_{(q)}}(i)\setminus\{j\}}$, which contradicts the assumption that $\{i,j\} \in \mathcal{E}^{(q)}$ since $\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i) \setminus \{j\}\right) \leq q$. □

The following corollary follows by recursive application of Proposition 6.5.2 and by noting that

$$\mathrm{Card}\left(\mathrm{bd}_{G_{(q+1)}}(i)\right) \leq \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right) ,$$

and

$$\mathrm{Card}\left(\mathrm{bd}_{G_{(q+1)}}(j)\right) \leq \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(j)\right) ,$$

since $\mathcal{E}^{(q+1)} \subseteq \mathcal{E}^{(q)}$.

Corollary 6.5.1. *Let $\{i,j\} \in \mathcal{E}^{(q)}$ for a given $q \in \{0,\ldots,p-3\}$. If*

$$\min\left(\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right), \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(j)\right)\right) \leq q+1 ,$$

then $\{i,j\} \in \mathcal{E}$.

To simplify the notation, we let $\mathrm{bd}_{G_{(q)}}(i,j)$ denote the smallest of i and j's neighborhoods:

$$\mathrm{bd}_{G_{(q)}}(i,j) = \begin{cases} \mathrm{bd}_{G_{(q)}}(i) & \text{if } \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right) \leq \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(j)\right) , \\ \mathrm{bd}_{G_{(q)}}(j) & \text{otherwise} , \end{cases}$$

and we let $\mathrm{bd}^*_{G_{(q)}}(i,j)$ denote the smallest of i and j's neighborhoods excluding i or j:

$$\mathrm{bd}^*_{G_{(q)}}(i,j) = \begin{cases} \mathrm{bd}_{G_{(q)}}(i) \setminus \{j\} & \text{if } \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right) \leq \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(j)\right) , \\ \mathrm{bd}_{G_{(q)}}(j) \setminus \{i\} & \text{otherwise} , \end{cases}$$

Proposition 6.5.3. *Let $\{i,j\} \in \mathcal{E}^{(q)}$ for a given $q \in \{0,\ldots,p-3\}$. If*

$$\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i,j)\right) = q+2 ,$$

then $\{i,j\} \in \mathcal{E}^{(q+1)}$ if and only of $\rho_{\left(i,j \mid \mathrm{bd}^*_{G_{(q)}}(i,j)\right)} \neq 0$.

Proof: Without loss of generality, assume that $\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right) = q+2$.

Sufficiency ($\Longrightarrow$). By contraposition. If $\rho_{\left(i,j \mid \mathrm{bd}_{G_{(q)}}(i)\setminus\{j\}\right)} = 0$ then

$$\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathrm{bd}_{G_{(q)}}(i)\setminus\{j\}}$$

under Hypothesis 6.2. This contradicts the assumption that $\{i,j\} \in \mathcal{E}^{(q+1)}$ since

$$\text{Card}\left(\text{bd}_{G_{(q)}}(i) \setminus \{j\}\right) = q+1 \, .$$

Necessity ($\Longleftarrow$). By contraposition. If $\{i,j\} \notin \mathcal{E}^{(q+1)}$, then there exists a subset $\mathcal{S}$, with $\text{Card}(\mathcal{S}) = q+1$ (that $\{i,j\} \in \mathcal{E}^{(q)}$), such that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathcal{S}}$, which implies that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\text{bd}_G(i)}$ by Corollary 4.5.1. An immediate consequence of Corollary 6.3.1 is that $\text{bd}_G(i) \subseteq \text{bd}_{G_{(q)}}(i)$. Since $\{i,j\} \notin \mathcal{E}$, it follows that $j \notin \text{bd}_G(i)$, and thus that $\text{bd}_G(i) \subseteq \text{bd}_{G_{(q)}}(i) \setminus \{j\}$. We hence have that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\text{bd}_{G_{(q)}}(i)\setminus\{j\}}$, which contradicts the assumption that $\rho_{\left(i,j \mid \text{bd}_{G_{(q)}}(i)\setminus\{j\}\right)} \neq 0$. □

Corollary 6.5.2. *Let $\{i,j\} \in \mathcal{E}^{(q)}$ for a given $q \in \{0,\dots,p-3\}$. If*

$$\text{Card}\left(\text{bd}_{G_{(q)}}(i,j)\right) = q+2 \, ,$$

then $\{i,j\} \in \mathcal{E}$ if and only of $\rho_{\left(i,j \mid \text{bd}^*_{G_{(q)}}(i,j)\right)} \neq 0$.

Proof: Without loss of generality, assume that $\text{Card}\left(\text{bd}_{G_{(q)}}(i)\right) = q+2$.

Sufficiency ($\Longrightarrow$). By Corollary 6.3.1, $\{i,j\} \in \mathcal{E}$ implies that $\{i,j\} \in \mathcal{E}^{(q+1)}$. By Proposition 6.5.3, we then have that $\rho_{\left(i,j \mid \text{bd}_{G_{(q)}}(i)\setminus\{j\}\right)} \neq 0$.

Necessity ($\Longleftarrow$). By Proposition 6.5.3, we have that $\{i,j\} \in \mathcal{E}^{(q+1)}$. We then have that $\{i,j\} \in \mathcal{E}$ by Corollaries 6.3.2 and 6.5.1. □

To summarize, three cases can arise at step $q+1$ for each $\{i,j\} \in \mathcal{E}^{(q)}$.

Case 1.

If

$$\min\left(\text{Card}\left(\text{bd}_{G_{(q)}}(i)\right), \text{Card}\left(\text{bd}_{G_{(q)}}(j)\right)\right) \leq q+1 \, ,$$

we apply Corollary 6.5.1. Note that no partial correlation needs to be computed. Furthermore, the edge does not even need to be considered in the subsequent steps of the algorithm as we know that $\{i,j\} \in \mathcal{E}$ (Figure 6.7).

Case 2.

If

$$\min\left(\text{Card}\left(\text{bd}_{G_{(q)}}(i)\right), \text{Card}\left(\text{bd}_{G_{(q)}}(j)\right)\right) = q+2 \, ,$$

we apply Corollary 6.5.2. Note that we simply need to compute a single partial correlation to know whether $\{i,j\} \in \mathcal{E}$ or not (Figure 6.8).

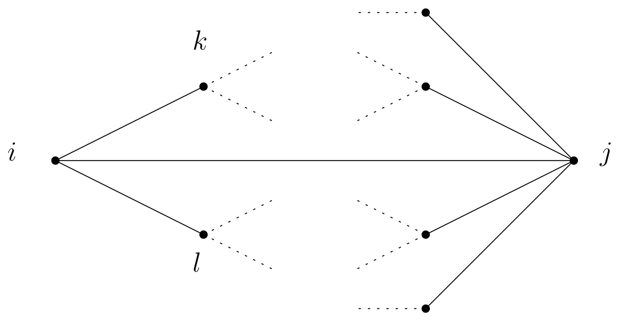


Figure 6.7: A 2-partial correlation graph. The node i has only two neighbors besides j (k and l). Hence, if $\{i,j\} \notin \mathcal{E}$, then we should have $\{i,j\} \notin \mathcal{E}^{(2)}$. Since this is not case, we can conclude that $\{i,j\} \in \mathcal{E}$ by Corollary 6.5.1.

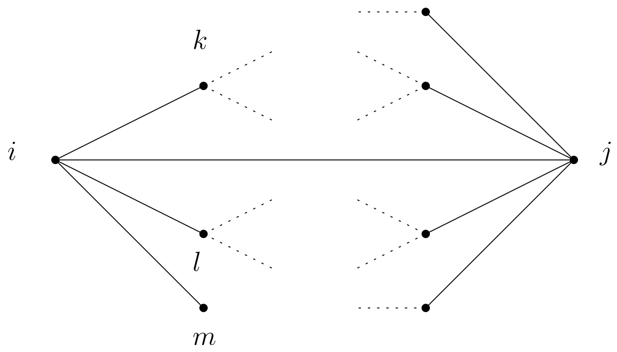


Figure 6.8: A 2-partial correlation graph. The node i has three neighbors besides j (k, l and m). By Corollary 6.5.2, $\{i,j\} \in \mathcal{E}$ if and only if $\rho_{(i,j|\,k,l,m)} \neq 0$.

Algorithm 6.1: q-Nested procedure (version 1).

```
Input: $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)^T$
Output: $q$-partial correlation graph
1 $q \leftarrow 0$
2 $\mathcal{E}^{(0)} \leftarrow \{\{i,j\} \in \mathcal{E}^{(-1)} \mid \rho_{(i,j)} = \rho_{(i,j|\emptyset)} \neq 0\}$
3 while $q < p - 2$ do
4     $\mathcal{E}^{(q+1)} \leftarrow \left\{\{i,j\} \in \mathcal{E}^{(q)} \mid \min\left(\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right), \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(j)\right)\right) \leq q+1\right\}$
5     $\mathcal{E}^{(q+1)} \leftarrow \mathcal{E}^{(q+1)} \cup$
      $\left\{\{i,j\} \in \mathcal{E}^{(q)} \mid \rho_{(i,j|\mathcal{S})} \neq 0,\ \text{for all } \mathcal{S} \subseteq \mathrm{bd}^*_{G_{(q)}}(i,j) \text{ with } \mathrm{Card}(\mathcal{S}) = q+1\right\}$
6     $q \leftarrow q + 1$
  end
```

Case 3.

Otherwise, i.e., if

$$\min\left(\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right), \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(j)\right)\right) > q + 2\ ,$$

we take advantage of Proposition 6.5.1. At step $q+1$, we consider only subsets of $\mathrm{bd}_{G_{(q)}}(i)$ or $\mathrm{bd}_{G_{(q)}}(j)$ (in practice, we choose the smallest neighborhood) of size $q+1$, instead of examining all possible subsets (of size $q+1$). However, Proposition 6.5.1 suggests that we condition on the whole neighborhood of i or j, not simply on subsets thereof. If we consider only subsets of size $q+1$, we might incorrectly keep edges that should be removed. Indeed, assume that after step $q+1$ there exists (at least) one subset $\mathcal{S} \subseteq \mathcal{V} \setminus \{i,j\}$ of size $q+1$ such that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_\mathcal{S}$, but there is no subset $\mathcal{S}' \subseteq \mathrm{bd}_{G_{(q)}}(i) \setminus \{j\}$ of size $q+1$ such that $\mathbf{x}_i \perp\!\!\!\perp \mathbf{x}_j \mid \mathbf{x}_{\mathcal{S}'}$. In this case, the edge $\{i,j\}$ is not removed at step $q+1$ as it should be.

However, this is not problematic. Indeed, as the order increases and as the neighborhoods of i and j can only decrease in size (Corollary 6.3.2), the edge will be ultimately removed, albeit at a later step. Thus, the advantage of looking at a smaller number of subsets is offset by the fact that some edges that could have been removed at step $q+1$ will only be removed at a later step $q' > q+1$. Therefore, before the q-nested procedure terminates, the intermediary graphs are not necessarily q-partial correlation graphs (Definition 6.2.2).

The detailed description of the q-nested procedure is given in Algorithm 6.1. In Lines 1 and 2, we respectively initialize q and $\mathcal{E}^{(0)}$ (6.9). Then, as long as q has not reached its maximum value of $p-2$ (Line 3), we loop through Lines 4 to 6 where we update $\mathcal{E}^{(q+1)}$ (Lines 4 and 5) and q (Lines 6). Note the difference between Lines 4 and 5, and (6.10). Indeed, thanks to Corollary 6.5.1 (i.e., Case 1), we know that for $\{i,j\} \in \mathcal{E}^{(q)}$, if

$$\min\left(\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right), \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(j)\right)\right) \leq q + 1\ ,$$

then $\{i,j\} \in \mathcal{E}$. Thus, $\{i,j\}$ can be automatically added to $\mathcal{E}^{(q+1)}$ without having to compute any partial correlation (Line 4). Then (Line 5), we add to $\mathcal{E}^{(q+1)}$ the edges

$\{i, j\} \in \mathcal{E}^{(q)}$ for which there exists no subset of cardinality $q + 2$ in the smallest of i and j's neighborhoods such that the corresponding partial correlation vanishes. Indeed, either

$$\min\left(\text{Card}\left(\text{bd}_{G_{(q)}}(i)\right), \text{Card}\left(\text{bd}_{G_{(q)}}(j)\right)\right) = q + 2 ,$$

and by Corollary 6.5.2 (i.e., Case 2) we have that $\{i, j\} \in \mathcal{E}$. Thus, $\{i, j\}$ must be added to $\mathcal{E}^{(q+1)}$. Note that in the next iteration, this edge will "fall" into Case 1 (Line 4). Either

$$\min\left(\text{Card}\left(\text{bd}_{G_{(q)}}(i)\right), \text{Card}\left(\text{bd}_{G_{(q)}}(j)\right)\right) > q + 2 ,$$

and Case 3 applies.

6.5.1.3 Intersection of neighborhoods search

We improve the results obtained in the previous section by proving that, when testing for the presence of an edge $\{i, j\}$ at step $q+1$, any separating subset is composed exclusively of nodes that belong to *both* the neighborhoods of i and j in $G_{(q)}$.

Proposition 6.5.4. *Let $\{i, j\} \in \mathcal{E}^{(q)}$ for a given $q \in \{0, \ldots, p-3\}$. If $\{i, j\} \notin \mathcal{E}^{(q+1)}$, then all sets $\mathcal{S} \subseteq \{1, \ldots, p\} \setminus \{i, j\}$ such that $\text{Card}(\mathcal{S}) = q + 1$ and $\rho_{(i,j|\mathcal{S})} = 0$ satisfy $\mathcal{S} \subseteq \text{bd}_{G_{(q)}}(i) \cap \text{bd}_{G_{(q)}}(j)$.*

Proof: First note that there exists at least one set $\mathcal{S} \subseteq \{1, \ldots, p\} \setminus \{i, j\}$ such that $\text{Card}(\mathcal{S}) = q + 1$ and $\rho_{(i,j|\mathcal{S})} = 0$ since $i \nsim_{G_{(q+1)}} j$ and $i \sim_{G_{(q)}} j$.

Now, recall from (4.24) that a partial correlation of order $q + 1$, $q \in \{0, \ldots, p-3\}$, between i and j can be computed from partial correlations of order q as follows:

$$\rho_{(i,j|\mathcal{K})} = \frac{\rho_{(i,j|\mathcal{K}\setminus\{k\})} - \rho_{(i,k|\mathcal{K}\setminus\{k\})}\,\rho_{(j,k|\mathcal{K}\setminus\{k\})}}{\sqrt{\left(1 - \rho^2_{(i,k|\mathcal{K}\setminus\{k\})}\right)\left(1 - \rho^2_{(j,k|\mathcal{K}\setminus\{k\})}\right)}} , \quad \text{for any } k \in \mathcal{K} , \tag{6.11}$$

where $\mathcal{K} \subseteq \{1, \ldots, p\} \setminus \{i, j\}$ with $\text{Card}(\mathcal{K}) = q + 1$. For any set $\mathcal{S} \subseteq \{1, \ldots, p\} \setminus \{i, j\}$ with $\text{Card}(\mathcal{S}) = q + 1$ and $\rho_{(i,j|\mathcal{S})} = 0$, we hence have by (6.11) that

$$\rho_{(i,j|\mathcal{S}\setminus\{s\})} = \rho_{(i,s|\mathcal{S}\setminus\{s\})}\,\rho_{(j,s|\mathcal{S}\setminus\{s\})} , \quad \text{for any } s \in \mathcal{S} . \tag{6.12}$$

Because $i \sim_{G_{(q)}} j$, we have that

$$\rho_{(i,j|\mathcal{Z})} \neq 0 , \quad \text{for all subsets } \mathcal{Z} \subset \{1, \ldots, p\} \setminus \{i, j\} \text{ with } \text{Card}(\mathcal{Z}) \leq q . \tag{6.13}$$

Consequently, any set $\mathcal{S}$ must be such that the two terms in the right-hand side of (6.12) are different from zero, i.e.,

$$\rho_{(i,s|\mathcal{S}\setminus\{s\})} \neq 0 \quad \text{and} \quad \rho_{(j,s|\mathcal{S}\setminus\{s\})} \neq 0 , \quad \text{for any } s \in \mathcal{S} , \tag{6.14}$$

otherwise the left-hand side of (6.12) would be equal to zero and (6.13) would not hold. Since $\text{Card}(\mathcal{S} \setminus \{s\}) = q$, it follows from (6.14) that $i \sim_{G_{(q)}} s$ and $j \sim_{G_{(q)}} s$, for any $s \in \mathcal{S}$. Hence, all nodes in $\mathcal{S}$ are neighbors of i and j in $G_{(q)}$: $\mathcal{S} \subseteq \text{bd}_{G_{(q)}}(i) \cap \text{bd}_{G_{(q)}}(j)$. □

Proposition 6.5.4 constitutes an improvement with respect to Algorithm 6.1 in that it (potentially) further reduces the size of the successive conditioning sets since

$$\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)\right) \leq \min\left(\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i)\right), \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(j)\right)\right) .$$

The two following corollaries analyze the case where

$$\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)\right) \leq q .$$

Corollary 6.5.3. *Let $\{i,j\} \in \mathcal{E}^{(q)}$ for a given $q \in \{0,\ldots,p-3\}$. If*

$$\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)\right) \leq q ,$$

then $\{i,j\} \in \mathcal{E}^{(q+1)}$.

Proof: By contraposition. Assume $\{i,j\} \notin \mathcal{E}^{(q+1)}$. Hence, there exists a subset $\mathcal{S} \subseteq \mathcal{V} \setminus \{i,j\}$, with $\mathrm{Card}(\mathcal{S}) = q+1$ (recall that $\{i,j\} \in \mathcal{E}^{(q)}$), such that $\rho_{(i,j|\mathcal{S})} = 0$. By Proposition 6.5.4, we have that $\mathcal{S} \subseteq \mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)$, which contradicts the fact that $\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)\right) \leq q$. □

The following corollary follows by a recursive application of Corollary 6.5.3, by noting that

$$\mathrm{Card}\left(\mathrm{bd}_{G_{(q+1)}}(i) \cap \mathrm{bd}_{G_{(q+1)}}(j)\right) \leq \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)\right) ,$$

since $\mathcal{E}^{(q+1)} \subseteq \mathcal{E}^{(q)}$, $\forall q \in \{0,\ldots,p-3\}$.

Corollary 6.5.4. *Let $\{i,j\} \in \mathcal{E}^{(q)}$ for a given $q \in \{0,\ldots,p-3\}$. If*

$$\mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)\right) \leq q ,$$

then $\{i,j\} \in \mathcal{E}$.

Algorithm 6.2 gives the second version of our q-nested procedure. It differs from the first version (Algorithm 6.2) in Lines 4 and 5 where we use the results from Proposition 6.5.4 and Corollary 6.5.4, respectively.

6.5.2 Sample version

Partial correlation coefficients are obtained from the shrinkage estimator of the covariance/concentration matrix as in (5.10). Although the sample covariance matrix could be used as well, the former is a more accurate estimator than the latter (Section 5.2).

In a frequentist approach to inference, we require the distribution function of the sample partial correlation coefficient $\hat{\boldsymbol{\rho}}_{(i,j|\mathcal{S})}$ under the null hypothesis $\rho_{(i,j|\mathcal{S})} = 0$ for all $\mathcal{S} \subseteq \mathcal{V} \setminus \{i,j\}$ to address the statistical testing problem of non-zero partial correlation

$$H_0 : \rho_{(i,j|\mathcal{S})} = 0 \qquad \text{versus} \qquad H_1 : \rho_{(i,j|\mathcal{S})} \neq 0 . \tag{6.15}$$

Algorithm 6.2: q-Nested procedure (version 2).

```
Input: $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)^T$
Output: $q$-partial correlation graph
$q \leftarrow 0$
$\mathcal{E}^{(0)} \leftarrow \{\{i,j\} \in \mathcal{E}^{(-1)} \mid \rho_{(i,j)} = \rho_{(i,j|\emptyset)} \neq 0\}$
while $q < p - 2$ do
    $\mathcal{E}^{(q+1)} \leftarrow \left\{\{i,j\} \in \mathcal{E}^{(q)} \mid \mathrm{Card}\left(\mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)\right) \leq q\right\}$
    $\mathcal{E}^{(q+1)} \leftarrow \mathcal{E}^{(q+1)} \cup$
    $\left\{\{i,j\} \in \mathcal{E}^{(q)} \mid \rho_{(i,j|\mathcal{S})} \neq 0, \text{ for all } \mathcal{S} \subseteq \mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j) \text{ with } \mathrm{Card}(\mathcal{S}) = q+1\right\}$
    $q \leftarrow q + 1$
end
```

Consider an edge $\{i,j\} \in \mathcal{E}^{(q)}$ for which we need to determine whether $\{i,j\} \in \mathcal{E}^{(q+1)}$ or not. We hence require the distribution function of the sample partial correlation $\hat{\boldsymbol{\rho}}_{(i,j|\mathcal{S})}$ under the null hypothesis $\rho_{(i,j|\mathcal{S})} = 0$ to address the statistical testing problem of non-zero partial correlation (6.15) for all $\mathcal{S} \subseteq \mathrm{bd}^*_{G_{(q)}}(i,j)$ (Algorithm 6.1) or $\mathcal{S} \subseteq \mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)$ (Algorithm 6.2) with $\mathrm{Card}(\mathcal{S}) = q+1$.

Without loss of generality, let us consider Algorithm 6.2. An edge $\{i,j\}$ is removed ($\{i,j\} \notin \mathcal{E}^{(q+1)}$) if there exists a subset $\mathcal{S} \subseteq \mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j)$ with $\mathrm{Card}(\mathcal{S}) = q+1$ such that H_0 is not reject. Hence, if the smallest partial correlation (in absolute value) is different from zero, then all partial correlations are different from zero and the edge is not removed. Otherwise, the edge is discarded. The edge $\{i,j\}$ is thus removed if H_0 is not reject in the following test:

$$H_0 : \min_{\substack{\mathcal{S} \subseteq \mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j),\\ \mathrm{Card}(\mathcal{S}) = q+1}} \left|\rho_{(i,j|\mathcal{S})}\right| = 0 \quad \text{vs.} \quad H_1 : \min_{\substack{\mathcal{S} \subseteq \mathrm{bd}_{G_{(q)}}(i) \cap \mathrm{bd}_{G_{(q)}}(j),\\ \mathrm{Card}(\mathcal{S}) = q+1}} \left|\rho_{(i,j|\mathcal{S})}\right| \neq 0 \,.$$

Similarly to testing for zero correlation (Appendix H.1), a possible solution is to resort to Fisher's Z-transform of the q-partial correlation:

$$Z_{(i,j|\mathcal{S})} = \tanh^{-1} \hat{\boldsymbol{\rho}}_{(i,j|\mathcal{S})} = \frac{1}{2} \log\left(\frac{1 + \hat{\boldsymbol{\rho}}_{(i,j|\mathcal{S})}}{1 - \hat{\boldsymbol{\rho}}_{(i,j|\mathcal{S})}}\right) ,$$

which has an asymptotic normal distribution under the null hypothesis H_0 when the data follow a multivariate Gaussian distribution [6, 86, 87]. Using a significance level α, we reject the null-hypothesis H_0 against the two-sided alternative H_1 if

$$\sqrt{n - (q+1) - 3}\, Z_{(i,j|\mathcal{S})} > \Phi^{-1}(1 - \alpha/2) \ , \tag{6.16}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution $\mathcal{N}(0,1)$. Note that (6.16) implies

$$q + 1 < n - 3 \,. \tag{6.17}$$

The sample version of the q-nested algorithm is obtained by modifying Algorithms 6.1 and 6.2 as follows. First, replace in Line 3 the condition $q < p - 2$ by

$$q < \min(n-3, p-2) \tag{6.18}$$

because of (6.17) (note that we have $q < n-3$ and not $q+1 < n-3$ as in (6.17) because of Line 6). Of course, in the "small n, large p" data setting,

$$\min(n-3, p-2) = n-3 \; .$$

Second, replace in Lines 2 and 5 the statements about, respectively, $\rho_{(i,j)} = \rho_{(i,j|\emptyset)} \neq 0$ and $\rho_{(i,j|\mathcal{S})} \neq 0$ by the statistical hypothesis test described above. Note that we correct the p-values over the multiple tests for all edges using the Benjamini-Hochberg correction for controlling the false discovery rate (Section 4.7.2.1).

Of course, other stopping criteria than (6.18) can be used. The user might for example choose a maximum value for q. Another example is given in Kontos and Bontempi [143] where we stopped the inference procedure at step q' as soon as:

$$\mathcal{E}^{(q')} = \mathcal{E}^{(q'-1)} \; . \tag{6.19}$$

One can easily verify that (6.19) is only a "heuristic" stopping criterion as it does not ensure that the algorithm will return the graph $G_{(p-2)}$ (see Figure 6.9 for an example).

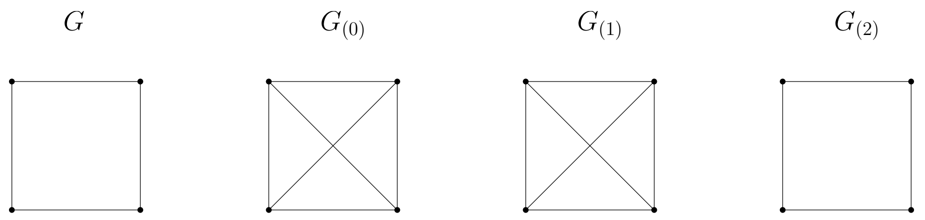


Figure 6.9: An illustration of the possible failure of the "heuristic" stopping criterion adopted in Kontos and Bontempi [143]. The algorithm will return $G_{(1)}$ which is different from $G_{(2)}$.

6.6 Additional remarks

We briefly discuss the Gaussian assumption (Hypothesis 6.2) and highlight an important difference between the two versions of our q-nested procedure (Algorithms 6.1 and 6.2).

6.6.1 The Gaussian assumption

In GGMs (Definition 4.6.1) it is assumed that the data follow a multivariate normal distribution (Hypothesis 6.2). Although the normality of microarray data is a disputed question [103, 269], we remark that the results obtained in Section 6.5 are valid for any distribution (as long as Hypothesis 6.1 is satisfied). Indeed, although we formulated statements of independence in terms of zero partial correlations in Section 6.5.1.2, this was not necessary to prove the results obtained in this section. Any test for conditional independence can thus be used. Therefore the first version of the q-nested procedure (Algorithm 6.1) constitutes a general procedure that can be used also outside the multivariate normal case. The second version of the q-nested procedure (Algorithm 6.2), however, requires Hypothesis 6.2.

6.6.2 Computational aspects

We compare the number of partial correlations to be computed in the worst case for inferring a q-partial (correlation) graph with the "standard" approach, Castelo and Roverato [30]'s method and our q-nested procedure (Algorithm 6.2). Table 6.2 summarizes the results.

As mentioned previously (Section 6.4), unless full-order partial correlations are considered (i.e., through regularization approaches; Chapter 5), the standard algorithms to reverse engineer limited-order partial correlation graphs are restricted to $q \leq 2$ given the very large number of partial correlation that need to be computed. Indeed, for each of the $p\,(p-1)\,/2$ possible edges, $\binom{p-2}{q}$ partial correlations have to be considered in the worst case, yielding a total of

$$\frac{p\,(p-1)}{2}\binom{p-2}{q}$$

partial correlations.

Castelo and Roverato [30] cope with this problem by considering only a fixed number r of randomly chosen subsets for the computation of partial correlations. Hence, the total number of partial correlations that are computed in the worst case is given by:

$$\frac{p\,(p-1)}{2}\min\left(r,\binom{p-2}{q}\right).$$

Despite reducing the computational burden of inferring q-partial (correlation) graphs, the number of edges to be considered ($p\,(p-1)\,/2$) remains high and the random selection of subsets raises some concerns (Section 6.4).

Our approach (Algorithms 6.1 and 6.2) both reduces the number of edges to be considered and the number of partial correlations per edge to be computed. For a q-partial correlation graph, Algorithms 6.1 and 6.2 require the computation of[4]

$$\sum_{k=0}^{q}\sum_{i,j\in\mathcal{E}^{(k-1)}}\binom{\mathrm{Card}\left(\mathrm{bd}^{*}_{G_{(q)}}(i,j)\right)}{k}$$

[4] Recall that the binomial coefficient $\binom{m}{k}=0$ if $k>m$.

and

$$\sum_{k=0}^{q}\sum_{i,j\in\mathcal{E}^{(k-1)}}\binom{\mathrm{Card}\left(\mathrm{bd}_{G_{(k-1)}}(i)\cap\mathrm{bd}_{G_{(k-1)}}(j)\right)}{k}$$

partial correlations in the worst case, respectively. We note that, although our approach does not require the underlying graph to be sparse, its efficiency relies on this assumption. Indeed, only in this case are $\mathrm{Card}\left(\mathrm{bd}^{*}_{G_{(q)}}(i,j)\right)$ and $\mathrm{Card}\left(\mathrm{bd}_{G_{(k-1)}}(i)\cap\mathrm{bd}_{G_{(k-1)}}(j)\right)$ relatively small.

Table 6.1: Number of partial correlations to be computed in the worst case for inferring a q-partial correlation graph with the "standard" approach, Castelo and Roverato [30]'s method and our q-nested procedure.

Method	Number of partial correlations
Standard	$\frac{p(p-1)}{2}\binom{p-2}{q}$
Castelo and Roverato [30]	$\frac{p(p-1)}{2}\min\left(r,\binom{p-2}{q}\right)$
Algorithm 6.1	$\sum_{k=0}^{q}\sum_{i,j\in\mathcal{E}^{(k-1)}}\binom{\mathrm{Card}\left(\mathrm{bd}^{*}_{G_{(q)}}(i,j)\right)}{k}$
Algorithm 6.2	$\sum_{k=0}^{q}\sum_{i,j\in\mathcal{E}^{(k-1)}}\binom{\mathrm{Card}\left(\mathrm{bd}_{G_{(k-1)}}(i)\cap\mathrm{bd}_{G_{(k-1)}}(j)\right)}{k}$

The time complexity for the computation of a partial correlation is $O\left(q^3\right)$ (Section 4.3.2.1). However, by using the matrix inversion approach, we obtain the values of $\binom{q+2}{2}$ for the price of one (Section 4.3.2.1).

In the standard case, the time complexity is thus given by

$$O\left(q^3\frac{\binom{p}{2}\binom{p-2}{q}}{\binom{q+2}{2}}\right)=O\left(q^3\binom{p}{q+2}\right) \tag{6.20}$$

To determine the time complexity in the remaining cases (Castelo and Roverato [30]'s method and Algorithms 6.1 and 6.2), we cannot simply divide by $\binom{q+2}{2}$ as in (6.20) since all the $\binom{q+2}{2}$ partial correlations computed by the matrix inversion formula will not necessarily be used. Indeed, depending on the random selection of subsets in Castelo and Roverato [30]'s method and on the evolution of the screening process in our q-nested procedure, some partial correlations returned by the matrix inversion formula might not be required.

Table 6.2: Worst case time complexity for inferring a q-partial correlation graph with the "standard" approach, Castelo and Roverato [30]'s method and our q-nested procedure.

Method	Number of partial correlations
Standard	$O\left(q^3\binom{p}{q+2}\right)$
Castelo and Roverato [30]	$O\left(q^3\frac{p(p-1)}{2}\min\left(r,\binom{p-2}{q}\right)\right)$
Algorithm 6.1	$O\left(q^3\sum_{k=0}^{q}\sum_{i,j\in\mathcal{E}^{(k-1)}}\binom{\operatorname{Card}\left(\operatorname{bd}^*_{G_{(q)}}(i,j)\right)}{k}\right)$
Algorithm 6.2	$O\left(q^3\sum_{k=0}^{q}\sum_{i,j\in\mathcal{E}^{(k-1)}}\binom{\operatorname{Card}\left(\operatorname{bd}_{G_{(k-1)}}(i)\cap\operatorname{bd}_{G_{(k-1)}}(j)\right)}{k}\right)$

We adopt a conservative stance and assume that we use the matrix inversion formula each time a partial correlation is computed, hence discarding the remaining partial correlations obtained as a "by-product" of the formula's application (recall that the O-notation gives an asymptotic upper bound; Appendix F). We hence obtain the following time complexities for Castelo and Roverato [30]'s approach:

$$O\left(q^3\frac{p\,(p-1)}{2}\min\left(r,\binom{p-2}{q}\right)\right),$$

for Algorithm 6.1:

$$O\left(q^3\sum_{k=0}^{q}\sum_{i,j\in\mathcal{E}^{(k-1)}}\binom{\operatorname{Card}\left(\operatorname{bd}^*_{G_{(q)}}(i,j)\right)}{k}\right),$$

and for Algorithm 6.2:

$$O\left(q^3\sum_{k=0}^{q}\sum_{i,j\in\mathcal{E}^{(k-1)}}\binom{\operatorname{Card}\left(\operatorname{bd}_{G_{(k-1)}}(i)\cap\operatorname{bd}_{G_{(k-1)}}(j)\right)}{k}\right).$$

In any case, we note that for small values of q, the value of $\binom{q+2}{2}=(q+2)(q+1)/2$ is small. Hence, the relative advantage of the standard approach with respect to the matrix inversion formula compared to the remaining approaches is negligible.

Note that using the matrix inversion formula does not increases memory requirements since for each edge we are only interested in the smallest q-partial correlation. Hence, at each moment, we only need to store one partial correlation per edge.

6.7 Experimental setup

6.7.1 Datasets

We use the package GeneNet [184] for the statistical software R [196] to generate 10 networks with $p=60$ genes with average node degree $d_G\in\{2.5,5\}$ and sample size $n=30$.

More specifically, 10 random "true" full-order partial correlation $p \times p$ matrices are generated by an algorithm which guarantees that the resulting matrices are always positive definite [184, 210]. The non-zero entries of these matrices correspond to the edges of the "true" networks. The edge proportion is determined according to (4.33) for $d_G \in \{2.5, 5\}$. Next, for each network, simulated data of the desired sample size $n = 30$ are drawn from the multivariate normal distribution with mean zero and the "true" correlation structure.

Although the model used for data generation is a simplification of real molecular processes, it is important to faithfully evaluate the prediction results. This is possible only if the true structure of the regulatory network is known.

6.7.2 Methods

We compare our q-nested procedure (Algorithm 6.2) to the q-partial correlation graph (Section 6.2) with $q \in \{0, 1, 2\}$ (recall that $q = 0$ corresponds to the correlation graph; Section 6.1), Castelo and Roverato [30]'s method with $q \in \{1, 2\}$ and 30 randomly selected q-partial correlations (we do not consider larger values because of the relative small number of genes considered), and Ledoit and Wolf [153]'s shrinkage estimator (Section 5.2). We implemented all these methods in R [196]. We did not consider q-partial correlation graphs with larger values of q because they are computationally too demanding to infer. We also perform a negative control by drawing partial correlations from the (continuous) uniform distribution on the interval $[0, 1]$.

6.7.3 Validation

A network inference problem can be seen as a binary decision problem where the inference algorithm plays the role of a classifier: for each pair of genes, the algorithm either adds an edge or not (Appendix K). Each pair of genes is thus assigned a positive label (an edge) or a negative label (no edge). A positive label (an edge) predicted by the algorithm is considered as a true positive (TP) or as a false positive (FP) depending on the presence or not of the corresponding edge in the underlying true network, respectively. The true and false negatives (TN and FN, respectively) are defined analogously (Appendix K).

We compute the area under the precision-recall curve (AUC-PR; Appendix K.2). The varying thresholds are applied to the partial correlations returned by the different methods. Note that for each edge, the smallest partial correlation (i.e., the one providing the more "evidence" against the presence of the edge) is considered.

Concerning the q-nested procedure, the values considered for the removed edges are the last computed partial correlations. Suppose for example that the procedure stops after q' steps. If an edge has been removed at step $q'' < q'$, it is the value of the q''-partial correlation (which is thus not significantly different from zero since the edge has been removed) that is taken into account for that edge.

6.8 Results and discussion

Table 6.3 compares the different methods' AUC-PR values. Box plots of the AUC-PR values are shown in Figures 6.10.

First, we remark that the q-partial correlation graphs (1-pcor and 2-pcor) do not achieve better results than the 0-correlation graph (cor). This can be explained by the large number of 1-partial and 2-partial correlations ($p-2=58$ and $\binom{p-2}{2}=1653$, respectively) that have to be computed for each edge. The chance that any of these is arbitrarily small is thus high (Section 6.4). Furthermore, their computational cost is much higher than that of the 0-correlation graph (cor).

Second, we note that the results obtained with Castelo and Roverato [30]'s approach with $q=1$ (castelo-1) are identical to those obtained with pcor-1. Indeed, despite being relatively small, the number of random q-partial correlations considered (30) is close to the total number of 1-partial correlations ($p-2=58$). The same conclusion can be drawn when comparing Castelo and Roverato [30]'s approach with $q=2$ (castelo-2) with pcor-2. Although this might be surprising at first sight, it seems that by picking a relatively large number of random q-partial correlations (30) compared to the total number of nodes (60), Castelo and Roverato [30]'s approach is robust to the problem of selecting q-partial correlations randomly (Section 6.4). However, for larger networks, this will most probably not be the case.

Third, we observe that the shrinkage approach (shrink) does not achieve better results than the 0-correlation graph (cor) either.

Finally, we note that our approach (qnested) is the only one to outperform all the other approaches.

We also remark that all results are significant compared to the negative control (random).

If we take into account the computational time required to infer the graphs (box plots of the CPU times on a 2.4 GHz AMD Opteron quad-core computer with 32 GB RAM running a Linux distribution are given in Figures 6.11 and 6.12), we notice that our approach remains the more attractive. Indeed, its computational time is only slightly higher than those of the cor and shrink methods.

For 8 graphs (out of 10) the q-nested procedure stops after 2 steps (and thus returns 1-partial correlation graphs). For the 2 remaining graphs, it returns 2-partial correlation graphs.

The average number of edges in the final graphs returned by the q-nested procedure is 63.9 while the number of edges in the graphs to be inferred is 150. On the other hand, the 0-partial correlation graphs (which are also the graphs inferred in the first step of the q-nested procedure) contain on average 422.2 edges. Here we see that it might be interesting to look at the whole sequence of graphs returned by the q-nested procedure (which is another of its advantage compared to the other methods), and not just at the final graph. Indeed, the first graphs in the sequence (such as the 0-partial correlation graph) contain too many edges (mainly false positives). As the procedure progresses, additional edges are removed and only a few remain. In some cases, too few remain and there are thus several false negatives. It might thus be interesting to combine information of the 0-partial correlation graphs and the 1-partial correlation graphs for example.

Nonetheless, our q-nested procedure clearly outperforms the state-of-the-art methods to reverse engineer q-partial correlation graphs. It enables to accurately infer these graphs while maintaining a reasonable computation time compared to the “simplest” methods (such as cor and shrink). A potential drawback of the q-nested procedure is that a false negative (i.e., incorrectly removing an edge) “propagates” to the next steps of the procedure.

Table 6.3: Comparison of AUC-PR values. The values in row i and column j report the number of method i's AUC-PR values which are larger than, equal to and smaller than method j's, respectively.

	cor	pcor-1	pcor-2	castelo-1	castelo-2	qnested	shrink	random
cor		9 / 0 / 1	10 / 0 / 0	9 / 0 / 1	10 / 0 / 0	0 / 0 / 10	4 / 0 / 6	10 / 0 / 0
pcor-1	1 / 0 / 9		9 / 0 / 1	0 / 10 / 0	9 / 0 / 1	0 / 0 / 10	2 / 0 / 8	10 / 0 / 0
pcor-2	0 / 0 / 10	1 / 0 / 9		1 / 0 / 9	0 / 10 / 0	0 / 0 / 10	0 / 0 / 10	10 / 0 / 0
castelo-1	1 / 0 / 9	0 / 10 / 0	9 / 0 / 1		9 / 0 / 1	0 / 0 / 10	2 / 0 / 8	10 / 0 / 0
castelo-2	0 / 0 / 10	1 / 0 / 9	0 / 10 / 0	1 / 0 / 9		0 / 0 / 10	0 / 0 / 10	10 / 0 / 0
qnested	10 / 0 / 0	10 / 0 / 0	10 / 0 / 0	10 / 0 / 0	10 / 0 / 0		10 / 0 / 0	10 / 0 / 0
shrink	6 / 0 / 4	8 / 0 / 2	10 / 0 / 0	8 / 0 / 2	10 / 0 / 0	0 / 0 / 10		10 / 0 / 0
random	0 / 0 / 10	0 / 0 / 10	0 / 0 / 10	0 / 0 / 10	0 / 0 / 10	0 / 0 / 10	0 / 0 / 10	

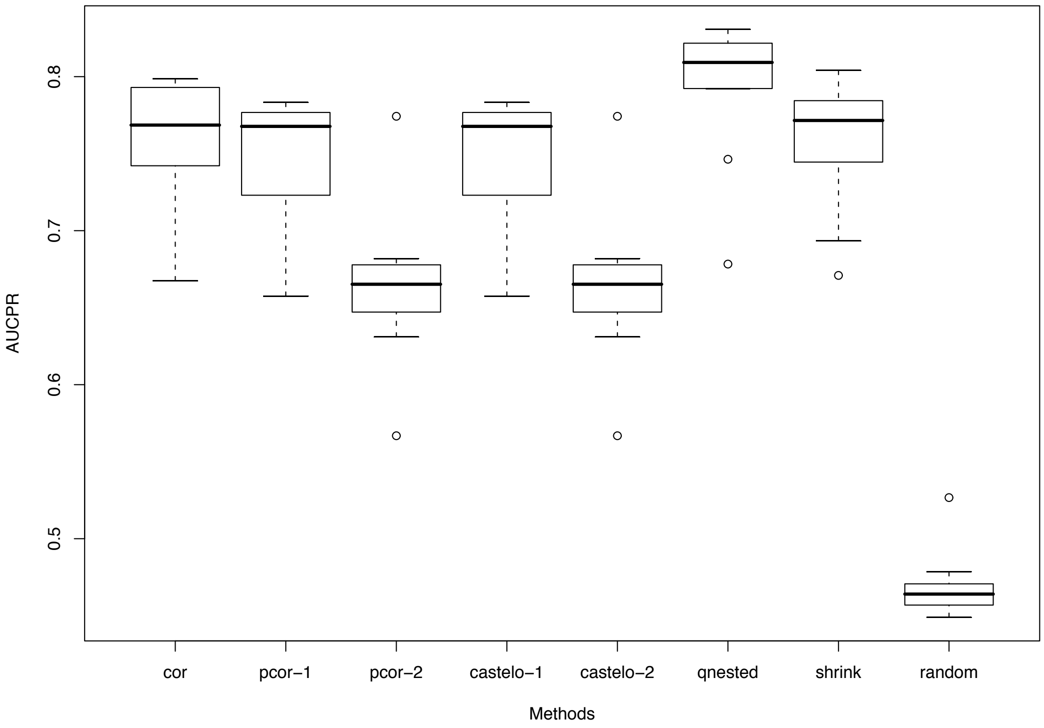


Figure 6.10: Box plots of AUCPR values.

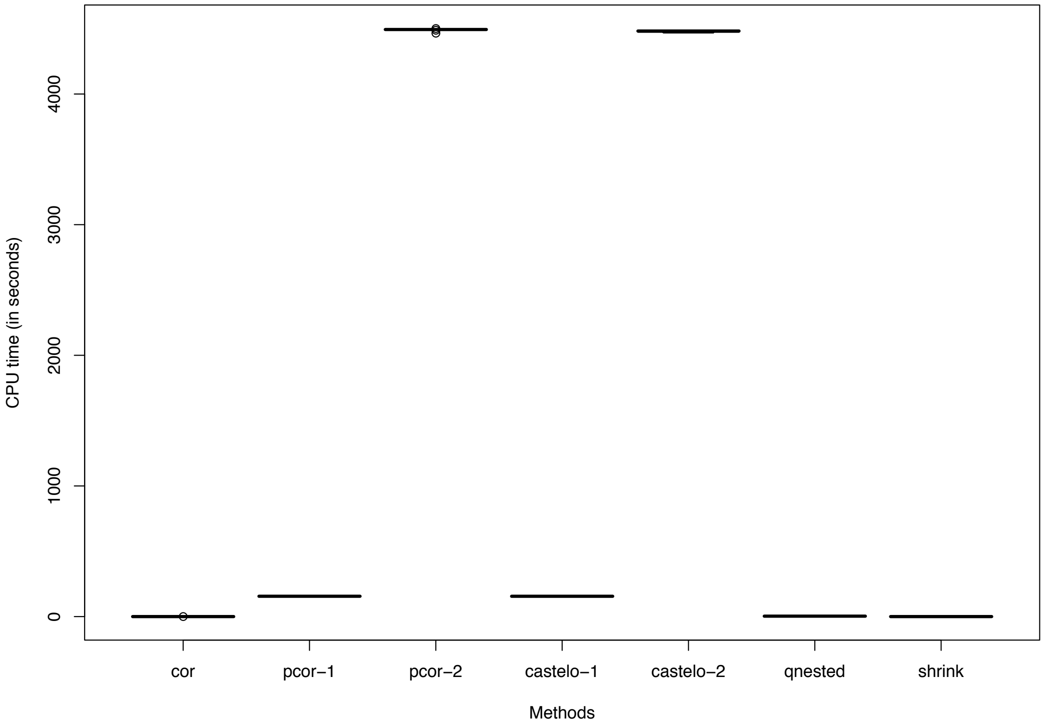
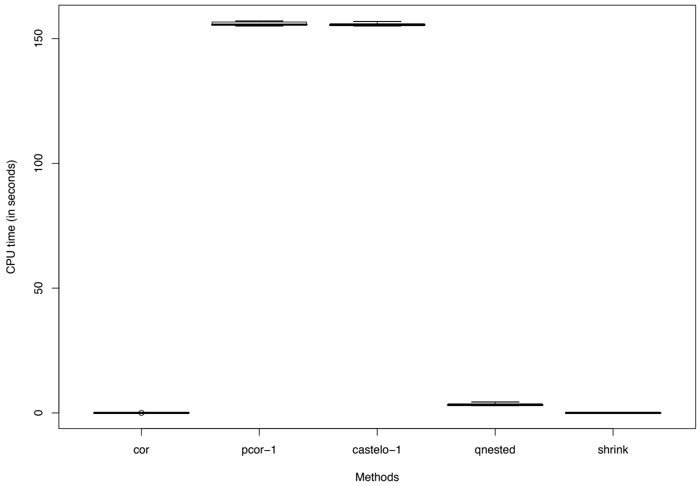
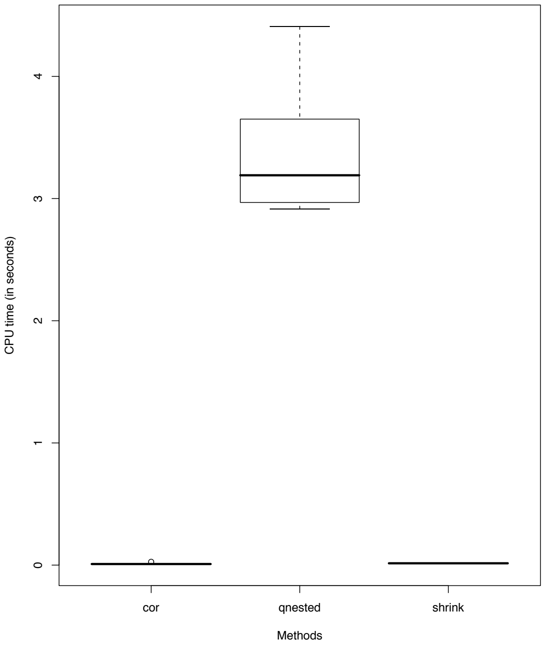


Figure 6.11: CPU time (in seconds) for all methods except random.



(a) All methods except random, pcor-2 and castelo-2.



(b) Methods cor, qnested and shrink.

Figure 6.12: CPU time (in seconds) for selected methods.

Contributions to Nitrogen Catabolite Repression Target Gene Prediction

7

Two-Class Classification for NCR Target Gene Prediction[1]

We formulate the identification of putative NCR genes in the yeast S. cerevisiae as a supervised two-class classification problem and extend the method by Godard et al. [107]. We show that our approach makes significant and biologically valid predictions, and we identify previously uncharacterized variables.

In the first part of the thesis, we used Gaussian graphical models (GGMs; Chapter 4) for inferring gene regulatory networks (GRNs; Section 2.3) from gene expression data (Chapters 5 and 6). We now tackle the important and challenging problem of gene function prediction. More specifically, we are interested in identifying genes involved in the yeast *S. cerevisiae*'s nitrogen catabolite repression (NCR; Section 2.2). In this chapter, we adopt a "standard" classification (Section 3.7) approach to tackle this problem. We will then make a connection with the first part of the thesis by proposing a new approach for predicting NCR genes based on a network inference paradigm (Chapter 8).

Recall from Section 2.2 that NCR is a selection mechanism that consists in the specific inhibition of transcriptional activation of genes encoding the permeases and catabolic enzymes needed to degrade poor nitrogen sources. All known nitrogen catabolite pathways are regulated by four regulators (Gln3, Gat1, Dal80, and Deh1). Moreover, approximatively 40 genes have been annotated as NCR-sensitive.

The ultimate goal is to identify all genes involved in NCR. This challenge has mainly been tackled by three genome-wide experimental studies [11, 107, 214], one of which [107] stems from the ARC project that supported the work presented in this thesis (see the Preface). In Godard et al. [107], we also proposed a bioinformatics approach, which we refer to as Godard et al. [107]'s approach, to complement the experimental study. Indeed bioinformatics methods offer the possibility to identify putative NCR genes and to discard uninteresting genes, hence strengthening the results of the experimental study. We adopted a "standard" classification approach to this function prediction task [121, 219]. More specifically, we formulated the identification of putative NCR genes in the yeast *S. cerevisiae* as a supervised two-class classification problem (Section 7.1). The

[1] Parts of this chapter appeared in Kontos et al. [146, 147] and in (the supplemental material of) Godard et al. [107].

(trained) classifiers predict whether genes are NCR-sensitive or not based on the number of occurrences of NCR-related motifs in their upstream noncoding sequences.

The third main contribution of the thesis consists in extending this two-class classification approach (Section 7.2). Instead of focusing on NCR-related motifs in the upstream noncoding sequences of the genes, we concentrate solely on the `GATA` motif. Indeed, the promoter regions of NCR target genes typically contain several `5'-GATA-3'` core sequences, which we will refer to as GATA boxes, recognized by the GATA family transcription factors (Godard et al. [107] and references therein). We specify a large number of variables related to this motif (Section 7.2.2). These variables define characteristics that biologists (who took part in the aforementioned ARC project) hypothesize to be relevant to NCR. Our goal mainly consists in determining new properties that could be determinant in NCR.

We also define a negative training set of manually-selected genes known to be insensitive to NCR (Section 7.2.1), thus avoiding the computational expensive undersampling approach (Section 3.7.3) adopted previously [107]. Besides, different classifiers (Section 7.2.3) and variable selection methods (Section 7.2.4) are compared.

We then show the effectiveness of our approach (Section 7.3). In particular, we show that all classifiers make significant and biologically valid predictions by comparing these predictions to annotated and putative NCR genes (Section 7.3.1), and by performing several negative controls. Moreover, the inferred NCR genes significantly overlap with putative NCR genes identified in three aforementioned genome-wide experimental studies (Section 7.3.2). These results suggest that our approach can successfully identify potential NCR genes. Hence, the dimensionality of the problem of identifying all genes involved in NCR is drastically reduced. Finally, we identify previously uncharacterized variables. Further experimental analysis is however required to determine whether these variables indeed play a role in NCR.

7.1 Two-class classification approach

In Godard et al. [107] we formulate the identification of putative NCR genes as a supervised two-class classification problem (Section 3.7) based on the number of occurrences of NCR-related motifs in their upstream noncoding sequences. Our approach is directly inspired by the one introduced in Simonis et al. [219] to discriminate co-regulated from non-co-regulated genes.

Based on prior biological knowledge, we defined a set of 9 motifs as potentially relevant for the NCR regulation:

- the canonical GATA box (`GATAAG`);
- the non-complete GATA box (`GATAAH`, where `H` means "not `G`");
- the degenerate GATA box (`GATTA`);
- GATA pairs formed of the canonical motif (`GATAAG`) in the three possible relative orientations, i.e., tandem (`GATAAGn{0,60}GATAAG`), convergent (`GATAAGn{0,60}CTTATC`) and divergent (`CTTATCn{0,60}GATAAG`);
- GATA pairs formed of the shortened motif (`GATAA`) in the three possible relative orientations, i.e., tandem (`GATAAn{0,60}GATAA`), convergent (`GATAAn{0,60}TTATC`) and

divergent (TTATCn{0,60}GATAA).

Next, we used the *oligo-analysis* tool from the Regulatory Sequence Analysis Tools (RSAT; available from http://rsat.ulb.ac.be/rsat/) [250] to detect over-represented oligonucleotides (for all sizes between 5 and 8) in the promoter sequences of the 41 ANCR genes, leading to a total of 56 significantly over-represented motifs (Appendix L.2). Quite consistently, most of these motifs were variants of the GATA box, and the most significant among them was the canonical GATA box GATAAG. Since some annotated motifs were also detected by *oligo-analysis*, we generated a non-redundant list of 62 motifs of interest (Appendix L.2).

Subsequently, we retrieved the upstream sequences of all 5869 yeast genes over 800 base pairs (bp) upstream from the start codon using the *retrieve sequence* tool from RSAT. When the upstream open reading frame (ORF) is closer than 800 bp, we retrieved a shorter sequence to discard coding sequences.

Finally, we used the program *dna-pattern* from RSAT to count the occurrences of the 62 motifs in each of the 5869 yeast gene promoters.

We applied linear discriminant analysis (Section 3.7.2.3) to classify genes into two classes (NCR versus not NCR) based on the number of occurrences of NCR-related motifs in their upstream noncoding sequences. Note that the genes play the role of samples and the motifs are the variables. As a positive training set, i.e., genes regulated by NCR, we used a set of 41 genes annotated as NCR sensitive (ANCR; see Table L.2).

Since we did not dispose of any reliable negative set for the training, i.e., genes not regulated by NCR, we applied the same undersampling strategy (Section 3.7.3) as the one described in Simonis et al. [219] by randomly selecting a (first) set of 123 (= 3 × 41) genes in the yeast genome (the process is then repeated 10 times as explained hereafter). The multiplicative factor 3 determining the size of the negative group was chosen through leave-one-out cross-validation.

Since the number of variables ($p = 62$) is larger than the number of genes in the positive training set ($n = 41$), we applied forward stepwise selection (Section 3.4) to select the subset of variables giving the most accurate classification. The efficiency of a classification was estimated using leave-one-out cross-validation. After this phase of training and variable selection, the discriminant function was then applied to each yeast gene to estimate its posterior probability to be NCR sensitive and to assign it to a class (NCR or not NCR).

The whole process was repeated 10 times with different negative groups in order to reduce the number of fluctuations due to random selection. A list of 100 genes predicted to be subject to NCR was finally obtained (Table S3 in the supplemental material of Godard et al. [107]) by selecting the genes for which the median posterior probability was greater than 0.5 and for which the posterior probability was greater than 0.5 in at least 6 iterations among 10.

7.2 Method: Extending Godard et al.'s approach

Our method consists in extending the two-class classification approach of Godard et al. [107] as follows:

- we use a negative training set composed of "non-NCR" genes instead of relying on the computationally expensive strategy of undersampling (Section 7.2.1);
- we define variables that reflect properties of the occurrences of the `GATA` motif in the upstream noncoding sequences of the yeast genes (Section 7.2.2); and
- we compare various classifiers (Section 7.2.3) and variable selection techniques (Section 7.2.4).

The classifier takes as input a data matrix X containing n rows (one per gene) and p columns (one per variable). The n genes constitute the samples. The p variables reflect properties of the occurrences of the `GATA` motif in the upstream noncoding sequences of the yeast genes (Section 7.2.2). Hence, each variable is a n-dimensional vector. The classifier is trained on a number $n_t \ll n$ of positive and negative training samples, i.e., genes that are known to be NCR-sensitive and insensitive, respectively. The trained classifier is then used to make predictions for genes not used in the training phase.

7.2.1 Training sets

As a positive training set, denoted by ANCR, we use 37 of the 41 genes previously annotated as NCR-responding [107] (Appendix L). Four genes are discarded because none of them were identified as NCR-responding in any of the three genome-wide experimental and bioinformatics studies described in Bar-Joseph et al. [11], Godard et al. [107], Scherens et al. [214]. The negative training set, denoted by NNCR, is composed of 90 manually-selected genes, known to be insensitive to NCR, most of which being involved in housekeeping cellular functions unrelated to nitrogen metabolism (Appendix L).

7.2.2 Variables

The promoter regions of NCR target genes typically contain several `5'-GATA-3'` core sequences, which we will refer to as GATA boxes, recognized by the GATA family transcription factors (Godard et al. [107] and references therein). Hence, we define 585 variables related to the GATA boxes in the upstream noncoding sequences of the yeast genes. These variables define characteristics that biologists (who took part in the ARC project mentioned previously) hypothesize to be relevant to NCR.

Since the variables rely on the availability of the upstream noncoding sequences, we retrieved them for all yeast genes over 800 base pairs (bp) upstream from the start codon using the collection of software tools provided by RSAT. When the upstream open reading frame (ORF) is closer than 800 bp, a shorter sequence is retrieved to discard coding sequences.

We now describe the 585 variables (see Table 7.1 for a summary)

- **Number of GATA boxes**: The annotated NCR genes (ANCR) are characterized by a relatively large number of GATA boxes (Figure 7.1) compared to the genes known

Table 7.1: Abbreviations and short descriptions of variables.

Abbreviation	Description
NUM	Number of GATA boxes
1-GAP, 2-GAP, 3-GAP, B-GAP	First, second and third smallest, and biggest GATA gaps
M-GAP, MI-GAP, SD-GAP	Mean, median and standard deviation (sd) of all GATA gaps
i-MINDIST ($i = 2, \ldots, 5$)	Minimum number of bp spanning over i GATA boxes
UP-i-MER ($i = 1, 2, 3$)	`N{1,i}GATA`
DOWN-i-MER ($i = 1, 2, 3$)	`GATAN{1,i}`
GAP-i-MER ($i = 1, 2$)	`N{1,i}GATAN{1,i}`
F-POS, L-POS	Positions of the first and of the last GATA boxes, resp.
M-POS, MI-POS, SD-POS	Mean, median and sd of the positions of all GATA boxes

to be insensitive to NCR (NNCR; see Figures 7.2 and 7.3). We therefore define a variable NUM which counts the number of GATA boxes in the upstream noncoding sequences.

- **GATA gap**: Further, we note that GATA boxes often come in pairs separated by only few bp. We therefore define 11 variables related to the number of bp separating two consecutive GATA boxes in the upstream noncoding sequences. The "gap" between two consecutive GATA boxes is referred to as a GATA gap (Figure 7.4). The variables 1-GAP, 2-GAP, 3-GAP and B-GAP measure (in bp) the first, second and third smallest, and biggest GATA gaps, respectively. The variables M-GAP, MI-GAP and SD-GAP measure (in bp) the mean, median and standard deviation of all GATA gaps, respectively. Finally, the variables i-MINDIST, $i = 2, \ldots, 5$, measure the minimum number of bp spanning over i GATA boxes (Figure 7.4).
- **k-Mers**: When searching for over-represented motifs in the upstream noncoding sequences of ANCR genes, it appears that variants of GATA boxes are relatively frequent, as for example the following motifs: `GATAAG` and `GATAAH`. Hence, we define the variables UP-i-MER ($i = 1, 2, 3$), DOWN-i-MER ($i = 1, 2, 3$) and GAP-i-MER ($i = 1, 2$) that count the following k-mers, respectively: `N{1,i}GATA`, `GATAN{1,i}` and `N{1,i}GATAN{1,i}`, where `N{1,i}` is a motif of length comprised between 1 and i, and where `N` represents any nucleotide (`A`, `C`, `G` or `T`). There are respectively 84 $\left(= 4 + 4^2 + 4^3\right)$, 84 and 400 $\left(= 4^2 + 2 \times 4^3 + 4^4\right)$ variables `N{1,i}GATA`, `GATAN{1,i}` and `N{1,i}GATAN{1,i}`.
- **Positions of GATA boxes**: Finally, we define 5 variables relative to the positions of the GATA boxes in the upstream noncoding sequences. The position of a GATA box is defined as the number of bp separating its first bp from the start codon of the gene. The variables F-POS and L-POS measure the positions of the first (i.e., the closest to the start codon) and of the last (i.e., the farthest from the start codon) GATA boxes, respectively. The variables M-POS, MI-POS and SD-POS measure the mean, median and standard deviation of the positions of all GATA boxes, respectively.

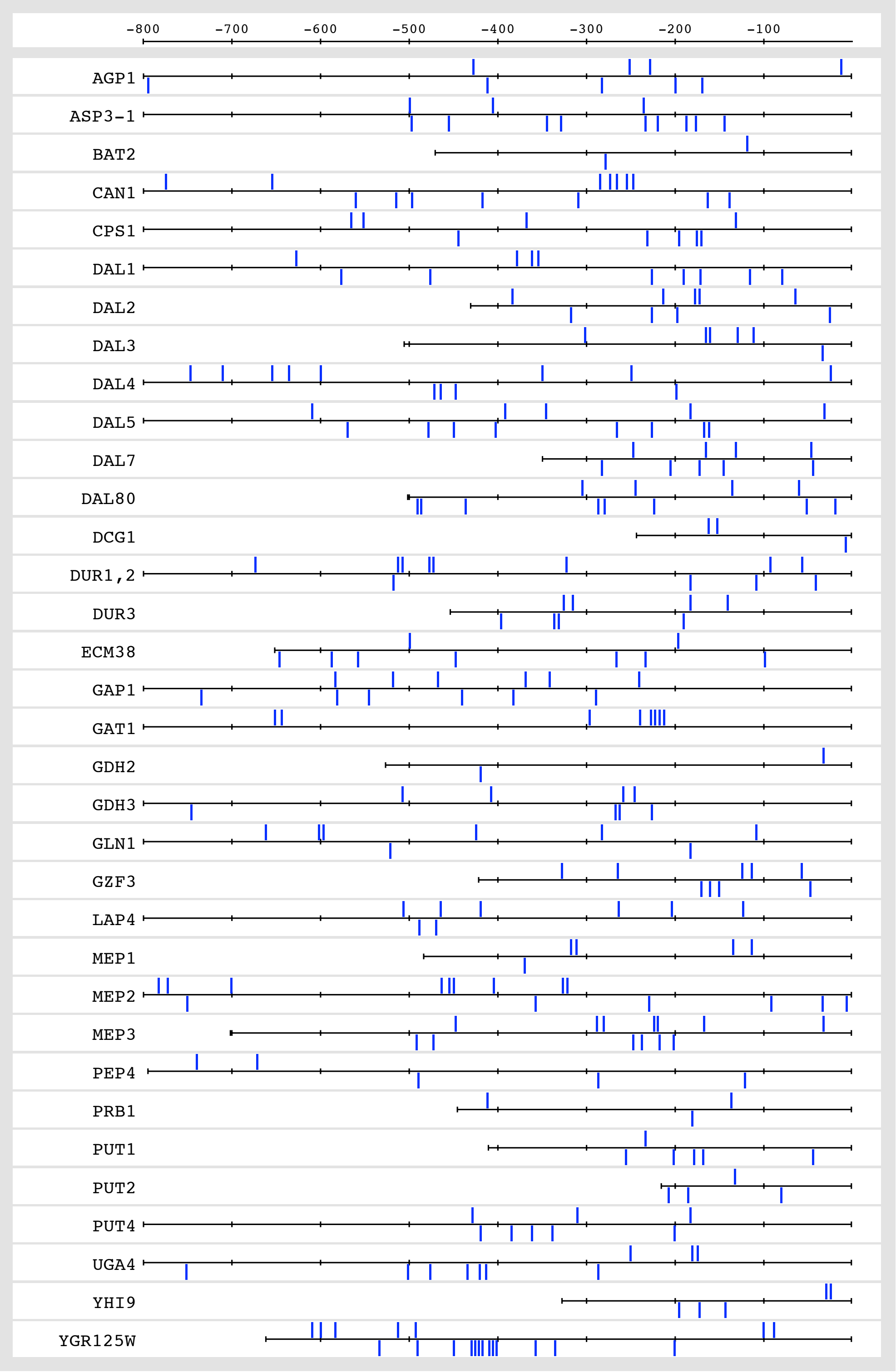


Figure 7.1: Graphical map of the GATA boxes in the upstream noncoding sequences of ANCR genes generated with RSAT [250]. Each horizontal line represents the noncoding sequence of a gene over 800 bp upstream from the start codon. When the upstream ORF is closer than 800 bp, the sequence is shortened to discard coding sequences. In each noncoding sequence, the blue vertical bars localize the GATA boxes.

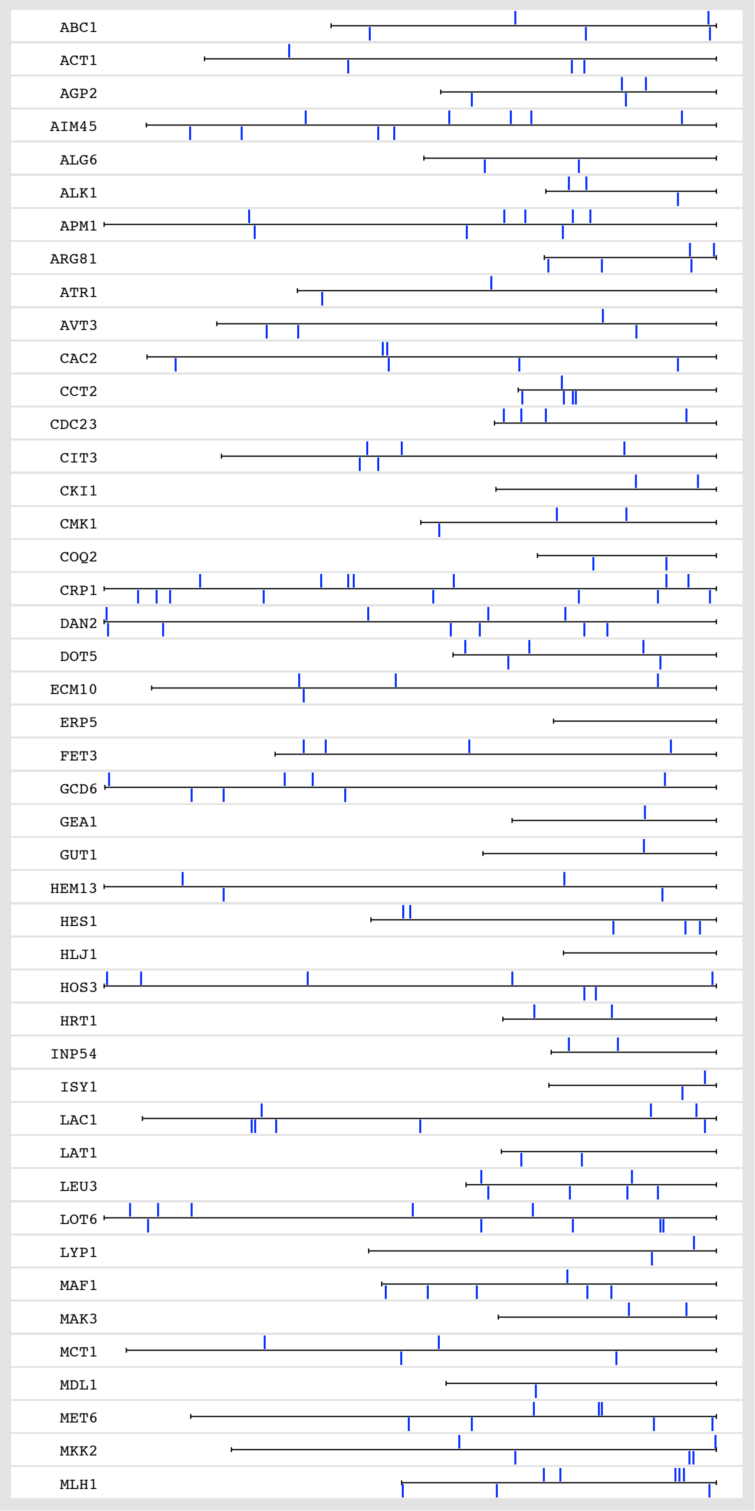


Figure 7.2: Graphical map of the GATA boxes in the upstream noncoding sequences of NNCR genes generated with RSAT [250] (part 1 of 2). See Figure 7.1's caption for more details.

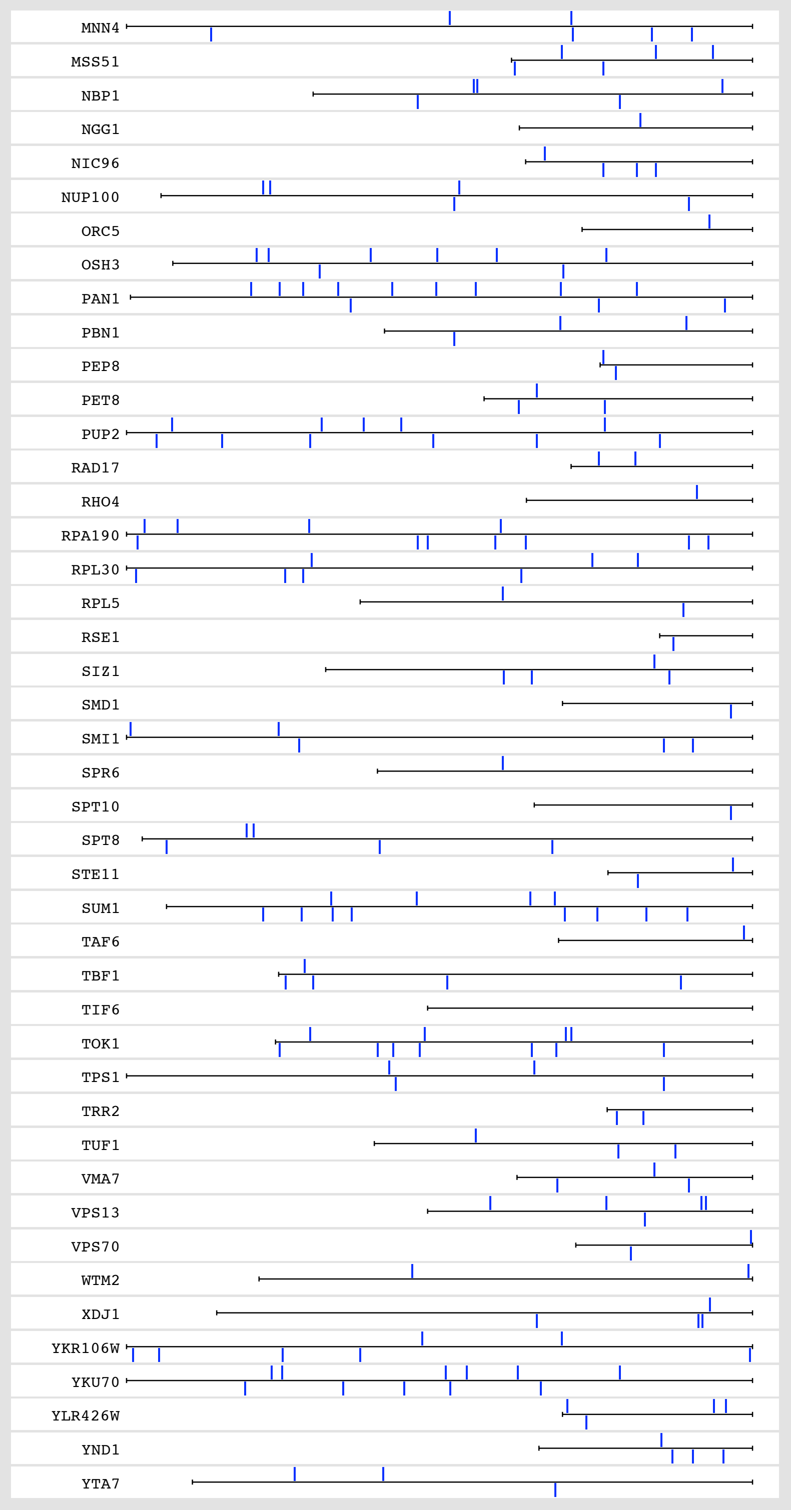


Figure 7.3: Graphical map of the GATA boxes in the upstream noncoding sequences of NNCR genes generated with RSAT [250] (part 2 of 2). See Figure 7.1's caption for more details.

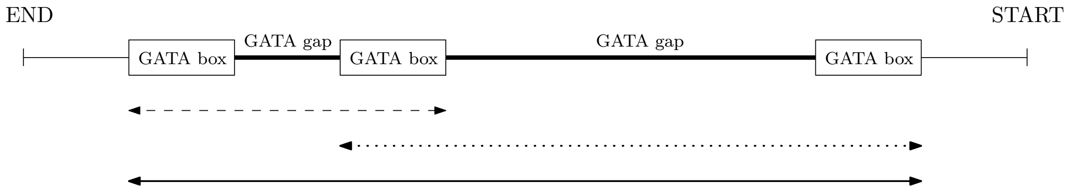


Figure 7.4: The figure shows the upstream noncoding sequence of a gene containing three GATA boxes. Note that the sequence "starts" on the right (at the start codon) "ends" on the left. The "gaps" between these boxes are referred to as GATA gaps (thick lines) and are measured in bp. The dashed line represents the minimum number of bp spanning over two GATA boxes. The dotted line also spans over two GATA boxes but the number of bp is larger than for the dashed line. Finally, the solid line represents the minimum number of bp spanning over three GATA boxes.

7.2.3 Classifiers

We compare three classifiers (Section 3.7.2): naive Bayes (NB), k-nearest-neighbors (KNN), where leave-one-out error is used to choose the number of neighbors, and, as a linear classifier, linear kernel support vector machine (SVM).

The classifiers provide estimates of the posterior probabilities that rely on the prior probabilities estimated from the training set. Unfortunately, these prior probabilities do not reflect the expected prior probabilities of the target classes. Therefore, we adjust the posterior probabilities returned by the classifiers with respect to new prior probabilities using Bayes's theorem (Section 3.7.3.3). These new priori probabilities are chosen according to prior biological knowledge: more or less 200 genes are expected to be targets of NCR [107]. Hence, we set the prior probability of a gene to be target of NCR to $200/n$, where $n = 5869$ is the total number of yeast genes considered.

7.2.4 Variable selection

Because of the high-dimensionality of the classification task (the number of variables is greater than the number of samples), we compare two variable selection methods to improve prediction performance and enhance interpretability (Section 3.4).

First, we use a filter method (Section 3.4.2) based on the Gram-Schmidt orthogonalization procedure where the number of selected variables is determined according to leave-one-out cross-validation (Appendix M). The ranking of variables through orthogonalization has many interesting features: it is computationally fast, it takes into account the collinearity between variables (i.e., if two variables are almost collinear in observation space, the fact that one of them is selected will tend to drive the other to a much lower rank in the list) and it allows an incremental construction of the model, so that training can be terminated without using all variables [229]. Although this method assumes linearity and is based on the minimization of a squared error loss (which is not always the most appropriate for classification), it gives relatively good results for classification tasks [229].

Second, we use a wrapper method (Section 3.4.2) consisting of a forward stepwise procedure where the prediction performance is assessed by means of stratified 10-fold cross-validation (Section 3.1.2). The performance measure used is the balanced error rate (BER) defined as the average of the errors on each class:

$$\text{BER} \ = 0.5 \left(\frac{\text{FP}}{\text{FP} + \text{TN}} + \frac{\text{FN}}{\text{FN} + \text{TP}} \right) \ , \tag{7.1}$$

where TP and FP are the true and false positives, respectively, and TN and FN are the true and false negatives, respectively. The threshold on the corrected posterior probability (Section 3.7.3.3) is 0.5. By using the prediction performance of a given learning machine to assess the relative usefulness of subsets of variables, wrappers offer a simple and powerful way to address the problem of variable selection [109, 140]. A greedy search strategy, such as forward selection, is both computationally advantageous and robust against overfitting [109].

7.3 Results and discussion

7.3.1 Validation

We assess the quality of the variable selection methods and classifiers through cross-validation: leave-one-out (l-o-o) cross validation for the filter variable selection method and 10-fold cross validation in the wrapper case. We use two performance measures. The first one is the BER (7.1). The threshold on the corrected posterior probability is 0.5. Results are shown in the "BER" column of Table 7.2. The best combinations of variable selection method and classifier, i.e., those having a BER not significantly higher than the lowest BER according to McNemar's test [58] with p-value $<$ .05, are marked with an asterisk (*).

The second performance measure is the area under the receiver operator characteristic (ROC) curve (AUC) (Appendix K.1). Results are given in the "AUC" column of Table 7.2. The ROC curves are shown in Figure 7.5.

Given the scarcity of the data and the risk of the variable selection procedure to overfit the selected variables to the training set, we perform a negative control to determine whether the results are significant or not. We empirically estimate the random rate of correct classification by running the same procedure but with randomized data sets obtained by randomly sampling the labels of the training set. Results are shown in the "negative control" columns of Table 7.2. The values reported are the mean and standard deviation over 10 repetitions. McNemar's test suggests that the linear classifier (the linear kernel SVM) performs best independently of the variable selection method.

Table 7.2: Performance assessment. VS and CLASS stand for variable selection method and classifier, respectively.

				Negative control	
VS	CLASS	BER	AUC	BER	AUC
Filter	NB	0.31	0.93	0.49 ± 0.022	0.50 ± 0.072
	KNN	0.18	0.90	0.51 ± 0.021	0.51 ± 0.077
	SVM	0.13*	0.93	0.48 ± 0.060	0.50 ± 0.097
Wrapper	NB	0.24	0.95	0.49 ± 0.054	0.50 ± 0.130
	KNN	0.20	0.97	0.48 ± 0.045	0.52 ± 0.100
	SVM	0.13*	0.95	0.47 ± 0.066	0.58 ± 0.130

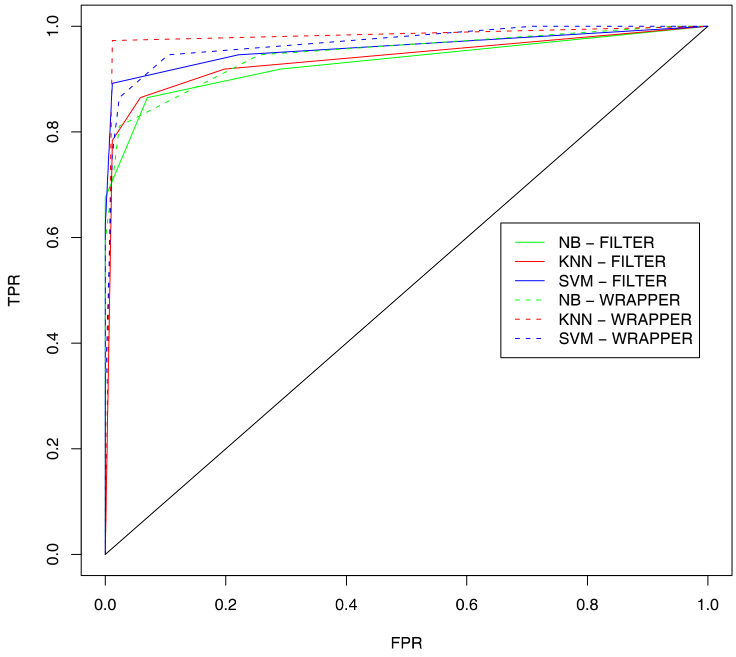


Figure 7.5: ROC curves.

7.3.2 Gene set comparisons

For each combination of variable selection method and classifier, we compare the set of predicted NCR genes, obtained with a threshold of 0.5 on the corrected posterior probability, with each of the three sets identified in the three aforementioned studies [11, 107, 214], respectively. More specifically, we compute for each combination of variable selection method and classifier, and for each set, the F-measure (K.1). Results are given in Table 7.3.

We also compute overlapping p-values on the basis of the cumulative distribution function of the hypergeometric distribution (Appendix J), to assess the significance of the overlap between two sets and to account for the artificial increase in the overlap that occurs when the number of predicted NCR genes increases (i.e., with decreasing threshold on the corrected posterior probability). Results are shown in Table 7.3.

Table 7.3: Gene set comparisons. VS and CLASS stand for variable selection method and classifier, respectively.

		F-measure (p-value)		
VS	CLASS	Bar-Joseph et al. [11]	Godard et al. [107]	Scherens et al. [214]
Filter	NB	0.05 (2.9×10^{-16})	0.09 (3.5×10^{-7})	0.06 (2.4×10^{-13})
	KNN	0.06 (9.4×10^{-9})	0.09 (4.8×10^{-5})	0.07 (1.1×10^{-7})
	SVM	0.11 (1.5×10^{-13})	0.15 (9.0×10^{-10})	0.14 (8.2×10^{-14})
Wrapper	NB	0.07 (9.1×10^{-11})	0.11 (7.7×10^{-18})	0.08 (4.3×10^{-16})
	KNN	0.12 (7.7×10^{-14})	0.20 (7.0×10^{-28})	0.16 (5.2×10^{-26})
	SVM	0.13 (8.9×10^{-11})	0.16 (7.2×10^{-14})	0.13 (2.6×10^{-11})

7.3.3 Variable selection

The improvement of prediction performance with variable selection is confirmed by the number of variables returned by the wrapper approach. Indeed, for all classifiers, the number of selected variables is small (in the order of tens) compared to the total number of variables (585).

The 6 variables selected by the filter variable selection method are shown in Table 7.4. The first variable is a DOWN-2-MER while the remaining ones are GAP-i-MERs. Two of these variables (`GATAAG` and `CAGATAAG`) appear in the set of motifs used in Godard et al. [107] (Table L.5). The variables selected by the filter method in the l-o-o validation procedure along with their respective frequencies are given in Table 7.6. We remark that the selected variables (Table 7.4) are almost always selected during the l-o-o procedure.

Table 7.4: Variables selected by the filter variable selection method. The variable marked with one asterisk (*) appears in the set of motifs used in Godard et al. [107].

Rank	Variable
1.	GATAAG*
2.	TAGATAA
3.	CGATAGG
4.	AAGATATT
5.	CAGATAAG*
6.	GGATAAG

Interestingly, the first variable, GATAAG, which is always selected in the l-o-o validation procedure, is known to be potentially relevant for the NCR regulation [38, 224]. There is currently no known relationship between the five other motifs and NCR.

The 17 variables selected by the linear SVM wrapper variable selection method are given in Table 7.5. The top selected variables are k-mers (UP-i-MER, DOWN-i-MER and GAP-i-MER. Two of these variables (GATAAG and CGATAA) appear in the set of motifs used in Godard et al. [107] (Table L.5). Interestingly, the first selected variable (GATAAG) is also the first one selected by the filter method. Another motif (TAGATAA) also appears in both rankings. Unfortunately, except for GATAAG, no other variable is frequently selected in the 10-fold cross-validation procedure. Some variables from Table 7.5 (rank 5 and ranks 7 to 17) are even never selected during the cross-validation procedure. This might be due to the larger "perturbations" that 10-fold cross-validation imposes on the data set compared to l-o-o. However, the filter selection method appeared as robust with 10-fold cross-validation as with l-o-o.

It is noteworthy that no variable related to GATA gaps has been identified in our approach as potential relevant for NCR, as was expected by the biologists involved in the ARC project mentioned previously. Nonetheless, the five variables identified by the filter method for which there exists no known relationship with NCR are interesting candidates for further analysis.

7.3.4 Final predictions and comparison with Godard et al.'s approach

Genes for which the corrected posterior probability is larger than 0.5 for both linear kernel SVMs with filter and wrapper variable selection methods, respectively, are predicted as NCR (Tables 7.8 and 7.9). This set is composed of 264 putative NCR genes. Indeed, McNemar's test suggests that the linear kernel SVM performs best independently of the variable selection method (Section 7.3.1). The final corrected posterior probability of these genes is the average of the corrected posterior probabilities as returned by the SVMs. The description of the top 25 genes is given in Table 7.10.

We compute the intersections of the sets of putative NCR genes identified in this chapter and the one inferred in the bioinformatics study of Godard et al. [107] (Section 7.1) with the sets of known and annotated NCR genes (RNCR and ANCR) and all possible

Table 7.5: Variables selected by the linear kernel SVM wrapper variable selection method. The variable marked with an asterisk (*) appears in the set of motifs used in Godard et al. [107].

Rank	Variable
1.	GATAAG*
2.	GGGATATA
3.	CTGGATA
4.	GATAGT
5.	GGGATAA
6.	TAGATAA
7.	ATGGATA
8.	GATAAGT
9.	TGATATT
10.	AGGATACT
11.	TTGATAAT
12.	CGATAA*
13.	TCGATAAA
14.	GATAT
15.	GTAGATA
16.	GATATAA
17.	ATGATAGT

combinations of intersections and unions of the sets $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$ arising from the aforementioned experimental studies [11, 107, 214]. We also assess the significancy of these intersections by computing p-values from the hypergeometric distribution (Appendix J). Results are presented in Table 7.11. We note that the intersections are (highly) significant for both sets.

7.4 Conclusion

We showed that all classifiers make significant (Section 7.3.1) and biologically valid (Section 7.3.2) predictions by comparing the predictions to annotated and putative NCR genes, and by performing several negative controls. However, McNemar's test suggests that the linear classifier (the linear kernel SVM) performs best independently of the variable selection method.

In particular, the inferred NCR genes significantly overlap with putative NCR genes identified in the three aforementioned genome-wide experimental studies [11, 107, 214], comparably to Godard et al. [107]'s bioinformatics approach. However, the latter approach yields more significant results. A possible explanation of this difference is the use of a single (small) set of 90 non-NCR genes. Although this is less computationally expensive than the undersampling strategy adopted in Godard et al. [107], it is probable that the 90 negative examples do not faithfully represent the whole set of non-NCR genes (which comprises almost all $\sim 6\,000$ yeast genes). Therefore, training classifiers on a single small set might

Table 7.6: Frequency of variables selected by the filter variable selection method in the l-o-o validation procedure.

Variable	Frequency
GATAAG*	1.000
CGATAGG*	0.953
TAGATAA*	0.942
AAGATATT*	0.779
CAGATAAG*	0.709
GGATAAG*	0.686
TAGATACC	0.174
CCGATATT	0.174
CGGATATT	0.105
CAGATAAT	0.070
AGATAAC	0.047
TTAGATA	0.047
AGGATATT	0.035
GCGATAAC	0.035
CTGATATT	0.035
GATAGGC	0.035
GTGATAAG	0.023
CAGATAAA	0.023
GTGATAAA	0.023
CCCGATA	0.012
TAGATAAG	0.012
CTGATACC	0.012
CCGATAAC	0.012
TTGATAGG	0.012
GAGATATG	0.012
TAGATATC	0.012
ACGATACT	0.012
AAGATACC	0.012
TCGATAA	0.012
2.gap	0.012
ACGATAAG	0.012
CTGATAAT	0.012
CTGATAT	0.012
TTGATAAG	0.012
ATAGATA	0.012
TAGATACA	0.012
CGCGATA	0.012
GAGATACC	0.012
GTAGATA	0.012
TCGATAAT	0.012
TTGATAG	0.012
TGGATACA	0.012
GATAAC*	0.012
GATATCT	0.012

Table 7.7: Frequency of variables selected by the linear kernel SVM wrapper variable selection method in the 10-fold cross-validation procedure.

Variable	Frequency
GATAAG*	1.00
ATGATAGC	0.20
ACCGATA	0.20
GGATAAG	0.20
GATAGTC	0.10
CCGGATA	0.10
GGGATATA*	0.10
AAGATATG	0.10
TGATAAA	0.10
GAGATACG	0.10
GCCGATA	0.10
CCCGATA	0.10
TCGATAGT	0.10
GATATGC	0.10
TCGATA	0.10
GATAAGG	0.10
GATACTC	0.10
GGGATATG	0.10
CAGATACT	0.10
TGATAGT	0.10
TAGATA	0.10
GATAGA	0.10
TGATACT	0.10
sd.pos	0.10
GTGATAAG	0.10
GATATT	0.10
AGGATACG	0.10
TCGATAAG	0.10
GAGATAC	0.10
ATGATAAT	0.10
GGGATAC	0.10
CTGGATA*	0.10
GGATACA	0.10
CTGATATT	0.10
TAGATAA*	0.10
TTGATAG	0.10
TAGGATA	0.10
CAGATATC	0.10
AGGATACA	0.10
GCGATATT	0.10
ACGATAAC	0.10
GGATATC	0.10
AGGATAGG	0.10
GATACCA	0.10
GAGGATA	0.10
ACAGATA	0.10
ATGATATG	0.10
GATAGT*	0.10
TGTGATA	0.10
TGATATG	0.10
AGGATAG	0.10
CGATA	0.10
CGGATAGG	0.10
CAGATACG	0.10
TTAGATA	0.10
GATACAA	0.10
GATAGCC	0.10
GATAACG	0.10
GATAGTA	0.10
CTGATAGA	0.10
ACGATAAA	0.10
AAGGATA	0.10

Table 7.8: Final prediction of NCR genes. Part 1 of 2.

Rank	Gene	Cor. post. prob.
1.	FCY21	1
2.	MGA2	1
3.	FRS2	1
4.	NPR2	1
5.	YER060W	1
6.	UBI4	1
7.	YIR033W	1
8.	MOH1	1
9.	APL1	1
10.	RHO3	1
11.	ECM37	1
12.	SAG1	1
13.	OPT2	1
14.	SPO14	1
15.	RSM10	1
16.	DAL1	1
17.	YFL022C	1
18.	PRP46	1
19.	RPL25	1
20.	YGK3	1
21.	SET5	1
22.	AMD2	1
23.	MEP3	1
24.	BAT1	1
25.	DAL4	1
26.	HTD2	0.966
27.	RSF2	0.964
28.	YEL062W	0.959
29.	GUD1	0.959
30.	MEP2	0.956
31.	DUR1,2	0.953
32.	HNM1	0.942
33.	YGR038C-A	0.937
34.	GDH1	0.93
35.	VPS21	0.923
36.	AVT1	0.907
37.	FCY2	0.884
38.	YGL196W	0.882
39.	MIG1	0.879
40.	IST1	0.876
41.	CIN5	0.876
42.	SAM35	0.875
43.	ARF2	0.874
44.	HXT17	0.869
45.	PRP2	0.863
46.	URK1	0.863
47.	KSP1	0.862
48.	AVT7	0.856
49.	IST3	0.856
50.	MCH4	0.856
51.	SSA4	0.855
52.	SLX9	0.854
53.	TOM20	0.854
54.	HRB1	0.844
55.	SDS23	0.842
56.	GLE2	0.838
57.	ARG1	0.834
58.	VAC17	0.828
59.	CHA1	0.828
60.	VTC2	0.825
61.	LEA1	0.818
62.	THI6	0.818
63.	HIM1	0.818
64.	GAP1	0.817
65.	ADE16	0.816
66.	YLL039C	0.814
67.	LYS20	0.813
68.	PHO13	0.808
69.	YGR125W	0.806
70.	ARA2	0.806
71.	ARG80	0.806
72.	SMP3	0.804
73.	MRPL23	0.804
74.	GTO1	0.802
75.	CYS4	0.802
76.	PCM1	0.802
77.	SOM1	0.802
78.	CUP9	0.798
79.	YEF3	0.796
80.	ALO1	0.793
81.	YHC1	0.791
82.	LEE1	0.789
83.	LGE1	0.789
84.	MRP7	0.781
85.	YBL049W	0.771
86.	COX19	0.771
87.	ORC1	0.77
88.	SMA2	0.77
89.	OLE1	0.762
90.	LSM4	0.761
91.	YJR005W	0.76
92.	YIL118W	0.76
93.	YIL146C	0.76
94.	SAP190	0.758
95.	YJR004C	0.756
96.	GAT1	0.751
97.	HMS1	0.75
98.	GYP1	0.749
99.	PRE7	0.748
100.	YPR194C	0.747

Table 7.9: Final prediction of NCR genes. Part 2 of 2.

Rank	Gene	Cor. post. prob.
101.	ERR3	0.745
102.	ERR1	0.745
103.	ERR2	0.745
104.	DAL3	0.744
105.	RAT1	0.742
106.	ATG1	0.741
107.	BSP1	0.731
108.	SNO4	0.731
109.	PEX2	0.725
110.	OPT1	0.725
111.	YKR031C	0.725
112.	YDR041W	0.724
113.	YIR027C	0.724
114.	YPL150W	0.723
115.	YPL151C	0.723
116.	YOL127W	0.723
117.	FUI1	0.723
118.	LAP3	0.723
119.	NAR1	0.723
120.	YOL128C	0.723
121.	AQY1	0.723
122.	VPH1	0.722
123.	HSP33	0.717
124.	HSP32	0.717
125.	PAU12	0.711
126.	PTP3	0.71
127.	DMA2	0.71
128.	YHR207C	0.704
129.	YDR242W	0.704
130.	CIS1	0.702
131.	SES1	0.702
132.	YPR138C	0.701
133.	YSP3	0.699
134.	RPS0B	0.699
135.	EHD3	0.691
136.	KRS1	0.691
137.	CPT1	0.69
138.	AVT4	0.688
139.	TIM44	0.688
140.	YKE4	0.688
141.	YHR208W	0.681
142.	ALD2	0.678
143.	NRG1	0.677
144.	FBP26	0.677
145.	SKO1	0.677
146.	MRC1	0.676
147.	ASI1	0.67
148.	YIR028W	0.669
149.	ARG4	0.666
150.	CSN9	0.665
151.	YHR067W	0.665
152.	RGT2	0.665
153.	ILV2	0.665
154.	FLC3	0.664
155.	YJR127C	0.664
156.	YDL237W	0.663
157.	YDL238C	0.659
158.	ORT1	0.654
159.	SRY1	0.653
160.	YNL142W	0.652
161.	YBR208C	0.652
162.	YGL077C	0.65
163.	UGA3	0.65
164.	GLT1	0.65
165.	RTS3	0.649
166.	YGR038C-A	0.649
167.	RVB2	0.648
168.	YOR375C	0.648
169.	PHO5	0.645
170.	OAZ1	0.64
171.	KTR6	0.64
172.	OSH7	0.64
173.	ISF1	0.638
174.	RTA1	0.636
175.	YOR089C	0.635
176.	YJR001W	0.606
177.	REG1	0.603
178.	SEC1	0.602
179.	TRM82	0.602
180.	ICL1	0.598
181.	YPK1	0.598
182.	YER056C	0.595
183.	FUN12	0.593
184.	RBG1	0.593
185.	TRP4	0.591
186.	GAL83	0.59
187.	CLU1	0.59
188.	YIL089W	0.588
189.	ENT2	0.579
190.	FZO1	0.577
191.	DTR1	0.577
192.	IZH3	0.574
193.	YGL035C	0.565
194.	HEM4	0.555
195.	SPL2	0.553
196.	ARO9	0.553
197.	GNA1	0.547
198.	QDR2	0.547
199.	SEO1	0.545
200.	ZPS1	0.535

Table 7.10: Ranking of genes (top 20) by decreasing corrected posterior probability (Table 7.8) with their description (retrieved with RSAT [250]).

Rank	Sys. name	Stand. name	Description
1.	YER060W	FCY21	Putative purine-cytosine permease, very similar to Fcy2p but cannot substitute for its function [YER060W;FCY21;FCY21;YER060W;856788;6320902; NP_010981]
2.	YIR033W	MGA2	ER membrane protein involved in regulation of OLE1 transcription, acts with homolog Spt23p; inactive ER form dimerizes and one subunit is then activated by ubiquitin/proteasome-dependent processing followed by nuclear targeting [YIR033W;MGA2;MGA2;YIR033W;854851;6322224;NP_012299]
3.	YFL022C	FRS2	Alpha subunit of cytoplasmic phenylalanyl-tRNA synthetase, forms a tetramer with Frs1p to form active enzyme; evolutionarily distant from mitochondrial phenylalanyl-tRNA synthetase based on protein sequence, but substrate binding is similar [YFL022C;FRS2;FRS2;YFL022C;850522; 14318497;NP_116631]
4.	YEL062W	NPR2	Protein with a possible role in regulating expression of nitrogen permeases; transcription is induced in response to proline and urea; contains two PEST sequences; null mutant is resistant to cisplatin and doxorubicin [YEL062W;NPR2;NPR2;YEL062W;856647;37362640; NP_010852]
5.	YER060W	FCY21	Putative purine-cytosine permease, very similar to Fcy2p but cannot substitute for its function [YER060W;FCY21;FCY21;YER060W;856788;6320902; NP_010981]
6.	YLL039C	UBI4	Ubiquitin, becomes conjugated to proteins, marking them for selective degradation via the ubiquitin-26S proteasome system; essential for the cellular stress response; encoded as a polyubiquitin precursor comprised of 5 head-to-tail repeats [YLL039C;UBI4;UBI4;YLL039C;850620;6322989;UB14;SCD2; NP_013061]
7.	YIR033W	MGA2	ER membrane protein involved in regulation of OLE1 transcription, acts with homolog Spt23p; inactive ER form dimerizes and one subunit is then activated by ubiquitin/proteasome-dependent processing followed by nuclear targeting [YIR033W;MGA2;MGA2;YIR033W;854851;6322224;NP_012299]
8.	YBL049W	MOH1	Protein of unknown function, has homology to kinase Snf7p; not required for growth on nonfermentable carbon sources; essential for viability in stationary phase [YBL049W;MOH1;MOH1;YBL049W;852231;6319422;NP_009504]
9.	YJR005W	APL1	Beta-adaptin, large subunit of the clathrin associated protein complex (AP-2); involved in vesicle mediated transport; similar to mammalian beta-chain of the clathrin associated protein complex [YJR005W;APL1;APL1;YJR005W;853461;6322464;YAP80;NP_012538]
10.	YIL118W	RHO3	Non-essential small GTPase of the Rho/Rac subfamily of Ras-like proteins involved in the establishment of cell polarity; GTPase activity positively regulated by the GTPase activating protein (GAP) Rgd1p [YIL118W;RHO3;RHO3;YIL118W;854688;6322073;NP_012148]
11.	YIL146C	ECM37	Non-essential protein of unknown function [YIL146C;ECM37;ECM37;YIL146C;854660;6322045;NP_012120]
12.	YJR004C	SAG1	Alpha-agglutinin of alpha-cells, binds to Aga1p during agglutination, N-terminal half is homologous to the immunoglobulin superfamily and contains binding site for a-agglutinin, C-terminal half is highly glycosylated and contains GPI anchor [YJR004C;SAG1;SAG1;YJR004C;853460;6322463; AG(ALPHA)1;NP_012537]
13.	YPR194C	OPT2	Oligopeptide transporter; member of the OPT family, with potential orthologs in S. pombe and C. albicans [YPR194C;OPT2;OPT2;YPR194C;856324; 6325452;NP_015520]
14.	YKR031C	SPO14	Phospholipase D, catalyzes the hydrolysis of phosphatidylcholine, producing choline and phosphatidic acid; involved in Sec14p-independent secretion; required for meiosis and spore formation; differently regulated in secretion and meiosis [YKR031C;SPO14;SPO14;YKR031C;853902;6322883;PLD1; NP_012956]
15.	YDR041W	RSM10	Mitochondrial ribosomal protein of the small subunit, has similarity to E. coli S10 ribosomal protein; essential for viability, unlike most other mitoribosomal proteins [YDR041W;RSM10;RSM10;YDR041W;851611;6320246;NP_010326]
16.	YIR027C	DAL1	Allantoinase, converts allantoin to allantoate in the first step of allantoin degradation; expression sensitive to nitrogen catabolite repression [YIR027C; DAL1; DAL1;YIR027C;854845;6322218;NP_012293]
17.	YFL022C	FRS2	Alpha subunit of cytoplasmic phenylalanyl-tRNA synthetase, forms a tetramer with Frs1p to form active enzyme; evolutionarily distant from mitochondrial phenylalanyl-tRNA synthetase based on protein sequence, but substrate binding is similar [YFL022C;FRS2;FRS2;YFL022C;850522; 14318497;NP_116631]
18.	YPL151C	PRP46	Splicing factor that is found in the Cef1p subcomplex of the spliceosome [YPL151C;PRP46;PRP46;YPL151C;855952;6325106;NTC50;NP_015174]
19.	YOL127W	RPL25	Primary rRNA-binding ribosomal protein component of the large (60S) ribosomal subunit, has similarity to E. coli L23 and rat L23a ribosomal proteins; binds to 26S rRNA via a conserved C-terminal motif [YOL127W;RPL25;RPL25;YOL127W;853993;6324445;NP_014514]
20.	YOL128C	YGK3	Protein kinase related to mammalian glycogen synthase kinases of the GSK-3 family; GSK-3 homologs (Mck1p, Rim11p, Mrk1p, Ygk3p) are involved in control of Msn2p-dependent transcription of stress responsive genes and in protein degradation [YOL128C;YGK3;YGK3;YOL128C;853992;6324444; NP_014513]

Table 7.11: Hypergeometric p-values: comparisons of Godard et al. [107]'s approach to the one proposed in this chapter. Note that Godard et al. [107] refers to the 100 genes identified by the bioinformatics procedure described in Section 7.1, while $\mathcal{P}_g$ refers to the 140 genes identified experimentally (i.e., with DNA microarrays) in Godard et al. [107].

Set (number of genes)	Godard et al. [107]	Chapter 7
RNCR (4)	3 (1.78×10^{-5})	1 (1.27×10^{-1})
RCNR+ANCR (38)	30 (1.69×10^{-48})	9 (2.99×10^{-6})
$\mathcal{P}_g$ (140)	32 (5.92×10^{-29})	25 (2.82×10^{-12})
$\mathcal{P}_s$ (87)	26 (8.62×10^{-27})	17 (2.49×10^{-9})
$\mathcal{P}_b$ (83)	23 (7.37×10^{-23})	18 (1.34×10^{-10})
$\mathcal{P}_g \cup \mathcal{P}_s$ (188)	40 (2.82×10^{-35})	30 (3.35×10^{-13})
$\mathcal{P}_g \cup \mathcal{P}_b$ (197)	39 (4.97×10^{-33})	32 (2.95×10^{-14})
$\mathcal{P}_s \cup \mathcal{P}_b$ (149)	36 (1.10×10^{-33})	27 (2.41×10^{-13})
$\mathcal{P}_g \cup \mathcal{P}_s \cup \mathcal{P}_b$ (240)	46 (7.39×10^{-39})	36 (7.77×10^{-15})
$\mathcal{P}_g \cap \mathcal{P}_s$ (39)	18 (9.39×10^{-23})	12 (2.40×10^{-9})
$\mathcal{P}_g \cap \mathcal{P}_b$ (26)	16 (4.74×10^{-23})	11 (2.14×10^{-10})
$\mathcal{P}_s \cap \mathcal{P}_b$ (21)	13 (6.27×10^{-19})	8 (1.86×10^{-7})
$\mathcal{P}_g \cap \mathcal{P}_s \cap \mathcal{P}_b$ (16)	12 (4.03×10^{-19})	7 (3.65×10^{-7})

not be optimal to discriminate all NCR from non-NCR genes.

No variable related to GATA gaps has been identified in our approach as potential relevant for NCR, as was expected by biologists. Nonetheless, several of the identified variables are known or hypothesized to be relevant to NCR. Therefore, the previously uncharacterized variables that were selected (Section 7.3.3) are promising candidates for further experimental analysis.

Given the problems related to the selection of a negative training set, we will not consider further the two-class classification approaches presented in this chapter, but rather adopt (in the next chapter) a new inference paradigm based on network inference which does not require a negative training set.

8

Network Inference Approach to NCR Target Gene Prediction[1]

We propose a network inference approach to predict NCR target genes. We reverse engineer a Gaussian graphical model (GGM) and exploit the topology of the inferred network for functional information. The network structure can give further insight into the considered problem.

In the previous chapter, we presented and extended a "standard" two-class classification approach for inferring nitrogen catabolite repression (NCR) target genes. Despite delivering promising results, these two-class classification approaches suffer from a major drawback: they require a negative training set. Indeed, we have available four regulators and a few tens of annotated NCR genes, but *a priori* there are no known "non-NCR" genes.

We first adopted an undersampling strategy (Section 3.7.3) to determine the choice of a negative training set (Section 7.1). This approach is based on the assumption that, since biologists expect at most a few hundreds genes to be involved in NCR, almost all[2] *S. cerevisiae*'s genes are not related with this process. Despite being a reasonable assumption, the ensuing strategy is computationally expensive: the whole inference procedure as to be performed several times (we performed 10 iterations in Godard et al. [107]) with different negative groups to reduce the variability of the predictions.

As an alternative, we then used a set of 90 manually-selected genes known to be insensitive to NCR (Section 7.2). Unfortunately, given the large number of "non-NCR" genes, selecting only 90 genes to represent the whole class of negative examples might no be ideal. Indeed, we cannot ascertain that these examples capture all of the features that characterize non-NCR genes. Moreover, these 90 genes might even be the "easiest" to discriminate from NCR genes. Hence, building a classifier that discriminates these 90 genes from NCR genes does not necessarily lead to a classifier that accurately classifies NCR from all non-NCR genes.

[1] Parts of this chapter appeared in Kontos et al. [142].

[2] Biologists are of course expecting some additional genes to be involved in NCR, otherwise we would not be trying to tackle the problem of inferring NCR target genes in the first place.

Obviously, the problem at hand corresponds more to *one-class classification* [233] than to two-class classification. One-class classification tries to discriminate one class of objects from all other possible objects by learning from a training set containing only the objects of that class. This observation leads us to the fourth main contribution of the thesis, which consists in a network inference approach to one-class classification (Section 8.1). In a nutshell, our approach consists in inferring a Gaussian graphical model (GGM) based on the number of occurrences of NCR-related motifs in the upstream noncoding sequences of the genes. To circumvent the dimensionality issue, we use Ledoit and Wolf [153]'s shrinkage estimator (Section 5.2). Given a set of NCR related genes, we then exploit the topology of the inferred network for functional information. More specifically, the neighbors of the genes of interest (the NCR regulators or/and the annotated NCR genes) are identified as putative NCR genes.

This approach does not require a negative training set[3] and thus avoids the problems encountered with the methods introduced in Chapter 7.

Furthermore, the network structure can give further insight into the considered problem. Indeed, "in real world applications, graphical [...] models are not only a tool for operations such as classification or prediction, but usually the network structures of the models themselves are also of great interest" [160]. This approach provides a more subtle and rich picture of the considered problem. Although we ultimately look at the neighbors of the genes of interest, the network topology offers the possibility for biologists to conduct a more detailed and refined analysis. While a standard classification approach only predicts NCR genes, a network approach also gives information on the interactions between the inferred NCR genes as well as on their interactions with the remaining genes. We deem that a network inference approach is more adequate to deal with such a problem.

Finally, this procedure is by far less computationally expensive that the two two-class classification approaches introduced previously. The feature selection and training phases are replaced by the inference of a regularized covariance matrix.

Interestingly, a similar network inference approach to function prediction has recently been proposed by Fitch and Jones [88].

8.1 Method: Inferring putative NCR genes with GGMs

Suppose we have a set $\mathcal{C}$ of genes known (or hypothesised) to be involved in NCR. We will refer to it as the "*core set*". It is either composed of the known NCR regulators, $\mathcal{C} = \text{RNCR}$, or the set of annotated NCR genes, $\mathcal{C} = \text{ANCR}$ (Section 8.2).

Further suppose we have a $p \times n$ data matrix X, where p is the number of genes (variables) and n is the number of samples. Note an important difference with the methods presented in Chapter 7 where genes were samples (to be classified). In this network approach, however, genes play the role of variables among which multivariate dependencies are to be inferred. Hence, each sample is a p-dimensional vector. The samples correspond to motifs relevant to the NCR regulation. We use the same $n = 62$ motifs as Godard

[3] Nevertheless, we will use negative validation sets for comparison purposes.

et al. [107] (Section 7.1). Hence, the (i, j)-th entry of matrix X, denoted x_{ij}, represents the number of occurrences of motif j in the upstream noncoding sequence of gene i.

The proposed approach consists in inferring (linear) dependencies between yeast genes by computing full-order partial correlations from the data matrix X using the shrinkage estimator presented in Section 5.2. These multivariate dependencies are then exploited by selecting the genes that are correlated (in terms of partial correlation) with at least one gene of the core set $\mathcal{C}$.

More specifically, for a given threshold t, the set $\mathcal{I}_t$ of inferred NCR genes is given by:

$$\mathcal{I}_t = \left\{ j \in \mathcal{A} \setminus \mathcal{C} : \max_{i \in \mathcal{C}} \left| \hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{A}\setminus\{i,j\})} \right| \geq t \right\} \ , \tag{8.1}$$

where $\mathcal{A}$ denotes the set of all yeast genes, and $\left| \hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{A}\setminus\{i,j\})} \right|$ is the absolute value of the shrinkage estimator of the full-order partial correlation between genes i and j (5.10).

Hence, $\mathcal{I}_t$ is composed of the genes (not in $\mathcal{C}$) for which the full-order partial correlation with at least one gene of $\mathcal{C}$ is greater (in absolute value) than the threshold t. In other words, genes which are dependent (i.e., not independent) of at least one gene in $\mathcal{C}$ (given the remaining genes in the genome) are inferred as NCR-sensitive. Inversely, genes which are independent of all genes in $\mathcal{C}$ are not included in $\mathcal{I}_t$.

The detailed description of our algorithm is given by Algorithm 8.1. In Line 1, we infer the covariance matrix (which entirely determines the GGM; recall Section 4.6) using Ledoit and Wolf [153]'s shrinkage estimator (Section 5.2). Next, we estimate the concentration matrix $\hat{\boldsymbol{\Omega}}^*$ (Line 2) as in (5.9). Finally, we build the set $\mathcal{I}_t$ of inferred genes (Line 3) as in (8.1).

Algorithm 8.1: Inference of putative NCR genes.

```
Input: Data set $X$, core set $\mathcal{C}$ and threshold $t$.
Output: Set $\mathcal{I}_t$ of inferred genes.
1 Compute the shrinkage covariance matrix $\hat{\boldsymbol{\Sigma}}^*$ as in (5.8)
2 $\hat{\boldsymbol{\Omega}}^* \leftarrow \left(\hat{\boldsymbol{\Sigma}}^*\right)^{-1}$
3 Compute the shrinkage full-order partial correlations $\hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{A}\setminus\{i,j\})}$ for all $i, j \in \mathcal{A}$
  from $\hat{\boldsymbol{\Omega}}^*$ as in (5.10)
4 $\mathcal{I}_t = \left\{ j \in \mathcal{A} \setminus \mathcal{C} : \max_{i \in \mathcal{C}} \left| \hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{A}\setminus\{i,j\})} \right| \geq t \right\}$
5 return $\mathcal{I}_t$
```

Graphically, our method consists in drawing an edge between pairs of genes whose full-order partial correlation exceeds the threshold t. Such a graph would be defined as

$G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ represents the set of genes and

$$\mathcal{E} = \left\{ (i,j) \in \mathcal{V} \times \mathcal{V} : \left| \hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{A}\setminus\{i,j\})} \right| \geq t \right\}$$

defines the set of edges. The inferred NCR genes $\mathcal{I}_t$ are then simply the neighbors (i.e., the boundary; recall Section 4.4) of the genes in the core set $\mathcal{C}$:

$$\mathcal{I}_t = \mathrm{bd}_G \left(\mathcal{C} \right) \ ,$$

as illustrated in Figure 8.1.

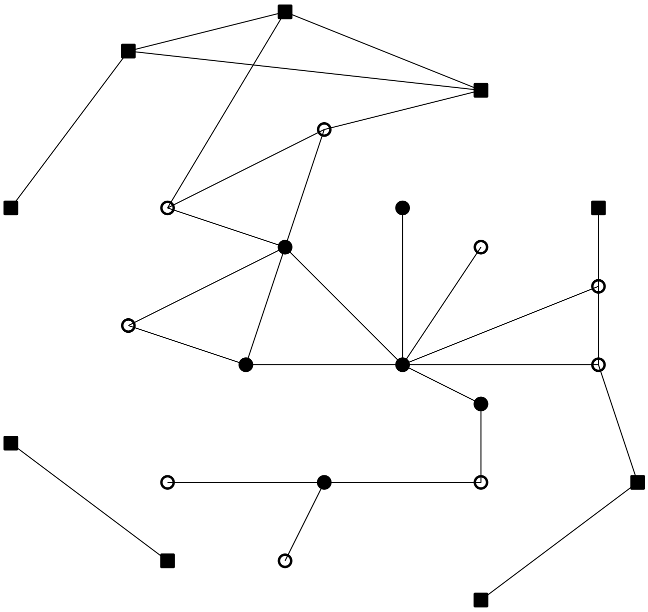


Figure 8.1: The graph where edges correspond to partial correlations (in absolute value) higher than the threshold t. The dots correspond to the genes in the core set $\mathcal{C}$, the circles represent the genes in $\mathcal{I}_t$, while the remaining genes are symbolized by squares.

Concerning the threshold t, we vary its value and use measures of performance related to receiver operator characteristic (ROC) curves (Section 8.2.1). Of course, the number of inferred genes diminishes with increasing threshold values.

The proposed method can be seen as a network version of a (supervised) two-class classifier. For a given threshold t, the genes in $\mathcal{I}_t$ are classified as "positive", i.e., inferred as NCR-sensitive, and the genes not in $\mathcal{I}_t$ (i.e., genes in $\mathcal{A} \setminus (\mathcal{C} \cup \mathcal{I}_t)$) are classified as "negative."

Note that the full-order partial correlations only depend on the input data matrix X. Hence, they only need to be computed once when considering different core sets and/or different threshold values as long as the same data set is used.

In terms of performance, executing the proposed method (i.e., inferring the partial correlation matrix and building the set $\mathcal{I}_t$) using the statistical software R requires less than 1 minute of CPU time on a 2.2 GHz Intel Core 2 Duo laptop with 2 GB RAM running Mac OS X.

8.2 Experimental setup

Our approach requires a set $\mathcal{C}$ of core genes to be defined (Section 8.1). Given the available data (Section 2.2), we use the set of known regulatory genes, $\mathcal{C}$ = RNCR, in Section 8.2.2, and the set of annotated NCR genes, $\mathcal{C}$ = ANCR, in Section 8.2.3. In both cases, positive and negative validation sets, denoted by $\mathcal{P}$ and $\mathcal{N}$, respectively, are defined to assess the predictive power of our approach using the performance measure presented in Section 8.2.1. Note that we perform negative controls to evaluate the significancy of our results in Section 8.2.4. Eventually, we present the procedure for the "final" predictions in Section 8.2.5.

8.2.1 Performance measure

As explained in Section 8.1, our method can be assessed as a two-class classifier. Hence, we use the area under the receiver operator characteristic (ROC) curve (AUC) as the performance measure.

However, our interest does not lie in the entire range of FPRs, but rather on very low false positive rates such as FPR < 0.05. We therefore also compute the partial area under the ROC curve (pAUC). Here, we focus on FPR $\in (0, u]$ with $u = 0.05$ and denote the corresponding area by pAUC$_u$. Because the magnitude of pAUC$_u$ depends on u, it is normalized by dividing it by u.

We also report jackknife estimates of standard deviations to be able to compare the (p)AUC values.

8.2.2 Known regulators and negative validation set

We assess the ability of our approach to recover the annotated NCR genes (ANCR) from the set RNCR of known regulators. We thus take $\mathcal{C}$ = RNCR and $\mathcal{P}$ = ANCR. Concerning the negative validation set $\mathcal{N}$, we consider two alternatives. As already mentioned, we can reasonably assume most of the $\sim 6\,000$ yeast genes not to be targets of NCR. Hence, we first take $\mathcal{N} = \mathcal{A} \setminus \{\mathcal{C} \cup \mathcal{P}\}$ (that $\mathcal{A}$ denotes the set of all yeast genes). Next, we consider the aforementioned set of 90 manually-selected genes known to be insensitive to NCR, i.e., $\mathcal{N}$ = NNCR.

8.2.3 Annotated NCR genes

We now run our method with $\mathcal{C}$ = ANCR. The inferred genes are first validated with $\mathcal{P}$ = ANCR through a leave-one-out procedure. Next, we use the gene sets $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$ identified in the aforementioned experimental studies [11, 107, 214].

8.2.3.1 Leave-one-out

The ANCR genes form a biologically meaningful set since they are all targets of NCR. Hence, we can expect that any given gene $i \in$ ANCR is strongly correlated (in terms of partial correlation) to at least one other gene in ANCR. If not, this would imply that gene i interacts indirectly (i.e., through other genes) with the other ANCR genes (by definition of partial correlation) and would be in contradiction with the hypothesis that the ANCR genes form a biologically coherent set.

If our approach to inferring NCR genes is sound, then for each gene $i \in$ ANCR the maximal full-order partial correlation (in absolute value) of gene i with a gene in ANCR $\setminus$ $\{i\}$,

$$\hat{\boldsymbol{\rho}}_i^{\max}(\textsc{ancr}) = \max_{j \in \mathrm{ANCR} \setminus \{i\}} \left| \hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{A} \setminus \{i,j\})} \right| ,$$

should be high, relative to the same quantity computed for all genes k not in ANCR ($k \in \mathcal{A} \setminus$ ANCR):

$$\hat{\boldsymbol{\rho}}_k^{\max}(\textsc{ancr}) = \max_{j \in \mathrm{ANCR}} \left| \hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{A} \setminus \{i,j\})} \right| .$$

To assess the usefulness of our approach, we thus estimate the p-value p_i, which represents the probability of randomly obtaining a score at least as high as ω_i^{max}, for all $i \in$ ANCR, by the empirical p-value:

$$\hat{\mathbf{p}}_i = \frac{\mathrm{Card}\left(\{k \in \mathcal{A} \setminus \mathrm{ANCR} : \omega_k^{max}(\textsc{ancr}) \geq \omega_i^{max}(\textsc{ancr})\}\right)}{\mathrm{Card}\left(\mathcal{A} \setminus \mathrm{ANCR}\right)} , \tag{8.2}$$

where Card$(\mathcal{Z})$ denotes the cardinality of set $\mathcal{Z}$.

8.2.3.2 Experimental studies

We also use the genes identified in the aforementioned experimental studies [11, 107, 214] to validate our approach. Of course, these genes are only putative NCR genes and the three sets identified only partially overlap. Still, in absence of any other validation data, we use these sets to complement the leave-one-out validation procedure described previously. More specifically, we consider all possible combinations of intersections and unions of these three sets (Section 8.3).

8.2.4 Negative control

We perform negative controls to determine whether the results are significant or not. More specifically, for the experiments described in Sections 8.2.2 and 8.2.3.2, respectively, we run our method with 1 000 randomly chosen core sets $\mathcal{C}_r^i \subset \mathcal{A}$ of cardinality $\left|\mathcal{C}_r^i\right| = |\mathcal{C}|$, $i = 1, \ldots, 1\,000$. We then perform the validation procedure as described in Sections 8.2.2

and 8.2.3, respectively, and report the mean and standard deviation of the AUC values obtained.

8.2.5 Final predictions

Finally, we consider as core set $\mathcal{C}$ the known NCR regulators (RNCR) and the annotated NCR genes (ANCR), i.e., $\mathcal{C} = \text{RNCR} \cup \text{ANCR}$, to predict the genes' "NCR-sensitivity". Specifically, we compute for each gene $j \in \mathcal{A}$ its maximal partial correlation (in absolute value), denoted by ω_j^{max}, with a gene in the core set (except with itself if $j \in \mathcal{C}$), $\mathcal{C} \setminus \{j\}$; formally:

$$\hat{\boldsymbol{\rho}}_j^{\max} = \hat{\boldsymbol{\rho}}_j^{\max}\left(\text{RNCR} \cup \text{ANCR}\right) = \max_{i \in \mathcal{C} \setminus \{j\}} \left| \hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{A} \setminus \{i,j\})} \right|, \quad \forall j \in \mathcal{A}. \tag{8.3}$$

This ω_j^{max} quantity is the "inferred NCR-sensitivity" of gene j.

We can thus rank all genes according to ω_j^{max}. In order to identify *the* set of putative NCR genes, we determine a threshold on ω_j^{max}. Since the RNCR and ANCR genes form a biologically meaningful set, we set the threshold equal to the median of the core genes' ω_j^{max} (i.e., the genes in $\mathcal{C} = \text{RNCR} \cup \text{ANCR}$):

$$\operatorname*{median}_{j \in \mathcal{C}} \hat{\boldsymbol{\rho}}_j^{\max} = \operatorname*{median}_{j \in \mathcal{C}} \left\{ \max_{i \in \mathcal{C} \setminus \{j\}} \left| \hat{\boldsymbol{\rho}}^*_{(i,j|\mathcal{A} \setminus \{i,j\})} \right| \right\} . \tag{8.4}$$

8.3 Results and discussion

8.3.1 Known regulators

Tables 8.1 and 8.2 present, respectively, the AUC and pAUC values for $\mathcal{C} = \text{RNCR}$, both for the GGM and the independence graph (Section 8.2.2). First, we note that all results are significant given the (p)AUC values obtained in the negative control cases. Next, we note the relatively high values for the GGM compared to the independence graph, which demonstrate the ability of the proposed method to successfully recover the annotated NCR genes from the four known NCR regulators. This suggests that our method (Section 8.1) is able to identify genes relevant to NCR more efficiently than with the independence graph.

8.3.2 Annotated NCR genes

AUC and pAUC values for $\mathcal{C} = \text{ANCR}$ and $\mathcal{N} = \mathcal{A} \setminus \{\mathcal{C} \cup \mathcal{P}\}$ are shown in Tables 8.3 and 8.4, respectively (Section 8.2.3.2). Results for $\mathcal{C} = \text{ANCR}$ and $\mathcal{N} = \text{NNCR}$ are given in Tables 8.5 and 8.6, respectively. Like previously, we note that all results are significant given the AUC values obtained in the negative control cases. However, this is not always the case with the pAUC values. We also note that the best AUC values are obtained for sets $\mathcal{P}$ that contain genes found in at least two of the aforementioned studies [11, 107, 214] (i.e., for intersections of at least two of the sets $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$). In other words, the stronger is the "consensus" on the NCR-sensitivity of a gene, the higher is the probability of this gene to be identified as such by our approach. In the case of pAUC values, however,

Table 8.1: AUC values (and jackknife estimates of standard deviations) for $\mathcal{C}$ = RNCR and $\mathcal{P}$ = ANCR are given in the "AUC" columns. The first row corresponds to $\mathcal{N} = \mathcal{A}\backslash\{\mathcal{C}\cup\mathcal{P}\}$ and the second one to $\mathcal{N}$ = NNCR. The "negative control" column shows the mean and standard deviation of the AUC over 1 000 repetitions.

	GGM		Independence graph	
Negative set $\mathcal{N}$	AUC	Neg. control	AUC	Neg. control
$\mathcal{A}\setminus\{\mathcal{C}\cup\mathcal{P}\}$	0.910 (0.034)	0.501 (0.049)	0.678 (0.11)	0.499 (0.049)
NNCR	0.909 (0.039)	0.501 (0.059)	0.668 (0.14)	0.501 (0.059)

Table 8.2: pAUC values (and jackknife estimates of standard deviations) for $\mathcal{C}$ = RNCR and $\mathcal{P}$ = ANCR are given in the "pAUC" columns. The first row corresponds to $\mathcal{N} = \mathcal{A}\setminus\{\mathcal{C}\cup\mathcal{P}\}$ and the second one to $\mathcal{N}$ = NNCR. The "negative control" column shows the mean and standard deviation of the pAUC over 1 000 repetitions.

	GGM		Independence graph	
Negative set $\mathcal{N}$	pAUC	Neg. control	pAUC	Neg. control
$\mathcal{A}\setminus\{\mathcal{C}\cup\mathcal{P}\}$	0.023 (0.01)	0.001 (0.001)	0.005 (0.005)	0.001 (0.001)
NNCR	0.03 (0.018)	0.002 (0.002)	0.006 (0.006)	0.002 (0.002)

there is no notable difference between the sets. We remark that the GGM only slightly outperforms the independence graph.

Concerning the leave-one-out procedure (Section 8.2.3.1), 22, 24 and 31 out of 37 ANCR genes are significantly identified as such (Table 8.7), with P-val. ≤ 0.05, P-val. ≤ 0.10 and P-val. ≤ 0.15, respectively. This is to be compared with the 26, 28, 17 and 16 ANCR genes identified as such in the aforementioned experimental studies Bar-Joseph et al. [11], Godard et al. [107], Scherens et al. [214] and by the independence graph (P-val. ≤ 0.10), respectively.

8.3.3 Final predictions

Table 8.8 ranks by decreasing "inferred NCR-sensitivity" the genes whose "inferred NCR-sensitivity" (8.3) is higher than the threshold (8.4). The threshold's value is 0.0179 yielding a set of 95 putative NCR genes. The description of the top 25 genes is given in Table 8.9.

Figure 8.2 depicts a graph composed of the core set's genes (i.e., RNCR and ANCR genes) and the putative NCR genes (i.e., the non-bold genes of Table 8.8), and the edges between them whose corresponding full-order partial correlations are above the threshold (8.4).

Interestingly, out of the 38 genes which are NCR regulators (RNCR) and/or targets of NCR (ANCR), 21 appear in this set of 95 putative NCR genes and 24 appear in the top 600 genes ($\sim$ 10% of all genes) ranked by decreasing "inferred NCR-sensitivity," respectively

Table 8.3: AUC values (and jackknife estimates of standard deviations) for $\mathcal{C}$ = ANCR and $\mathcal{N} = \mathcal{A} \setminus \{\mathcal{C} \cup \mathcal{P}\}$ are given in the "AUC" columns. For $\mathcal{P}$, we consider all possible combinations (in terms of unions and intersections) of the sets $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$ (from which we remove genes in $\mathcal{C}$). The "negative control" column shows the mean and standard deviation of the AUC over 1 000 repetitions.

	GGM		Independence graph	
$\mathcal{P}$	AUC	Neg. control	AUC	Neg. control
$\mathcal{P}_g \setminus \mathcal{C}$	0.696 (0.021)	0.499 (0.028)	0.638 (0.028)	0.499 (0.027)
$\mathcal{P}_s \setminus \mathcal{C}$	0.709 (0.030)	0.499 (0.039)	0.650 (0.052)	0.500 (0.040)
$\mathcal{P}_b \setminus \mathcal{C}$	0.656 (0.034)	0.498 (0.038)	0.637 (0.049)	0.501 (0.036)
$\{\mathcal{P}_g \cup \mathcal{P}_s\} \setminus \mathcal{C}$	0.690 (0.018)	0.501 (0.024)	0.635 (0.026)	0.501 (0.025)
$\{\mathcal{P}_g \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.671 (0.017)	0.500 (0.023)	0.628 (0.026)	0.502 (0.023)
$\{\mathcal{P}_s \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.671 (0.026)	0.501 (0.028)	0.634 (0.036)	0.500 (0.027)
$\{\mathcal{P}_g \cup \mathcal{P}_s \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.669 (0.017)	0.500 (0.021)	0.624 (0.024)	0.500 (0.021)
$\{\mathcal{P}_g \cap \mathcal{P}_s\} \setminus \mathcal{C}$	0.792 (0.053)	0.500 (0.064)	0.703 (0.071)	0.499 (0.069)
$\{\mathcal{P}_g \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0.836 (0.034)	0.501 (0.085)	0.779 (0.078)	0.498 (0.084)
$\{\mathcal{P}_s \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0.867 (0.032)	0.499 (0.107)	0.805 (0.073)	0.499 (0.109)
$\{\mathcal{P}_g \cap \mathcal{P}_s \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0.855 (0.047)	0.493 (0.144)	0.736 (0.103)	0.503 (0.145)

Table 8.4: pAUC values (and jackknife estimates of standard deviations) for $\mathcal{C}$ = ANCR and $\mathcal{N} = \mathcal{A} \setminus \{\mathcal{C} \cup \mathcal{P}\}$ are given in the "pAUC" columns. For $\mathcal{P}$, we consider all possible combinations (in terms of unions and intersections) of the sets $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$ (from which we remove genes in $\mathcal{C}$). The "negative control" column shows the mean and standard deviation of the pAUC over 1 000 repetitions.

	GGM		Independence graph	
$\mathcal{P}$	pAUC	Neg. control	pAUC	Neg. control
$\mathcal{P}_g \setminus \mathcal{C}$	0.005 (0.002)	0.001 (0.001)	0.002 (0.002)	0.001 (0.001)
$\mathcal{P}_s \setminus \mathcal{C}$	0.005 (0.002)	0.001 (0.001)	0.003 (0.002)	0.001 (0.001)
$\mathcal{P}_b \setminus \mathcal{C}$	0.004 (0.002)	0.001 (0.001)	0.002 (0.002)	0.001 (0.001)
$\{\mathcal{P}_g \cup \mathcal{P}_s\} \setminus \mathcal{C}$	0.005 (0.002)	0.001 (0.001)	0.003 (0.002)	0.001 (0.001)
$\{\mathcal{P}_g \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.005 (0.001)	0.001 (0.001)	0.002 (0.002)	0.001 (0.001)
$\{\mathcal{P}_s \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.005 (0.002)	0.001 (0.001)	0.003 (0.002)	0.001 (0.001)
$\{\mathcal{P}_g \cup \mathcal{P}_s \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.005 (0.002)	0.001 (0)	0.003 (0.002)	0.001 (0)
$\{\mathcal{P}_g \cap \mathcal{P}_s\} \setminus \mathcal{C}$	0.006 (0.003)	0.001 (0.001)	0 (0.003)	0.001 (0.002)
$\{\mathcal{P}_g \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0.005 (0.003)	0.001 (0.002)	0.001 (0.005)	0.001 (0.002)
$\{\mathcal{P}_s \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0 (0.001)	0.001 (0.002)	0.002 (0.004)	0.001 (0.002)
$\{\mathcal{P}_g \cap \mathcal{P}_s \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0 (0.001)	0.001 (0.003)	0 (0)	0.001 (0.003)

Table 8.5: AUC values (and jackknife estimates of standard deviations) for $\mathcal{C}$ = ANCR and $\mathcal{N}$ = NNCR are given in the "AUC" columns. For $\mathcal{P}$, we consider all possible combinations (in terms of unions and intersections) of the sets $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$ (from which we remove genes in $\mathcal{C}$). The "negative control" column shows the mean and standard deviation of the AUC over 1 000 repetitions.

	GGM		Independence graph	
$\mathcal{P}$	AUC	Neg. control	AUC	Neg. control
$\mathcal{P}_g \setminus \mathcal{C}$	0.656	0.500 (0.042)	0.689	0.500 (0.042)
$\mathcal{P}_s \setminus \mathcal{C}$	0.686	0.500 (0.049)	0.647	0.500 (0.049)
$\mathcal{P}_b \setminus \mathcal{C}$	0.629	0.500 (0.048)	0.732	0.500 (0.048)
$\{\mathcal{P}_g \cup \mathcal{P}_s\} \setminus \mathcal{C}$	0.653	0.500 (0.039)	0.670	0.501 (0.039)
$\{\mathcal{P}_g \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.637	0.500 (0.038)	0.694	0.500 (0.039)
$\{\mathcal{P}_s \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.646	0.500 (0.041)	0.682	0.500 (0.041)
$\{\mathcal{P}_g \cup \mathcal{P}_s \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.634	0.499 (0.038)	0.676	0.500 (0.037)
$\{\mathcal{P}_g \cap \mathcal{P}_s\} \setminus \mathcal{C}$	0.778	0.500 (0.075)	0.716	0.500 (0.076)
$\{\mathcal{P}_g \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0.777	0.501 (0.090)	0.853	0.500 (0.090)
$\{\mathcal{P}_s \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0.820	0.500 (0.116)	0.857	0.500 (0.114)
$\{\mathcal{P}_g \cap \mathcal{P}_s \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0.784	0.498 (0.150)	0.895	0.500 (0.147)

Table 8.6: pAUC values (and jackknife estimates of standard deviations) for $\mathcal{C}$ = ANCR and $\mathcal{N}$ = NNCR are given in the "pAUC" columns. For $\mathcal{P}$, we consider all possible combinations (in terms of unions and intersections) of the sets $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$ (from which we remove genes in $\mathcal{C}$). The "negative control" column shows the mean and standard deviation of the pAUC over 1 000 repetitions.

	GGM		Independence graph	
$\mathcal{P}$	AUC	Neg. control	AUC	Neg. control
$\mathcal{P}_g \setminus \mathcal{C}$	0.006 (0.002)	0.002 (0.001)	0.004 (0.003)	0.002 (0.001)
$\mathcal{P}_s \setminus \mathcal{C}$	0.007 (0.004)	0.002 (0.001)	0.004 (0.003)	0.001 (0.001)
$\mathcal{P}_b \setminus \mathcal{C}$	0.005 (0.003)	0.002 (0.001)	0.003 (0.004)	0.002 (0.001)
$\{\mathcal{P}_g \cup \mathcal{P}_s\} \setminus \mathcal{C}$	0.006 (0.002)	0.002 (0.001)	0.004 (0.003)	0.002 (0.001)
$\{\mathcal{P}_g \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.006 (0.002)	0.002 (0.001)	0.003 (0.003)	0.001 (0.001)
$\{\mathcal{P}_s \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.007 (0.003)	0.002 (0.001)	0.004 (0.003)	0.002 (0.001)
$\{\mathcal{P}_g \cup \mathcal{P}_s \cup \mathcal{P}_b\} \setminus \mathcal{C}$	0.006 (0.002)	0.001 (0.001)	0.004 (0.003)	0.001 (0.001)
$\{\mathcal{P}_g \cap \mathcal{P}_s\} \setminus \mathcal{C}$	0.009 (0.005)	0.002 (0.002)	0 (0.004)	0.002 (0.002)
$\{\mathcal{P}_g \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0.006 (0.003)	0.002 (0.002)	0.004 (0.01)	0.002 (0.002)
$\{\mathcal{P}_s \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0.001 (0.009)	0.001 (0.003)	0.004 (0.004)	0.002 (0.003)
$\{\mathcal{P}_g \cap \mathcal{P}_s \cap \mathcal{P}_b\} \setminus \mathcal{C}$	0 (0.008)	0.002 (0.004)	0 (0)	0.001 (0.004)

Table 8.7: p-values of the leave-one-out procedure (Section 8.2.3). Genes with p-value ≤ 0.10 are shown in boldface.

Gene	p-value	Gene	p-value	Gene	p-value	Gene	p-value
AGP1	1.06e-01	**DAL3**	5.32e-03	**GAT1**	1.03e-02	PRB1	3.34e-01
ASP3-1	1.71e-04	**DAL4**	1.01e-02	GDH2	5.35e-01	PUT1	1.20e-01
ASP3-2	1.71e-04	**DAL5**	1.10e-02	**GDH3**	1.20e-03	**PUT2**	8.32e-02
ASP3-3	1.71e-04	**DAL7**	8.57e-04	**GLN1**	8.81e-02	PUT4	1.85e-01
ASP3-4	1.71e-04	**DAL80**	3.43e-03	**GZF3**	6.86e-03	**UGA4**	7.20e-03
BAT2	1.69e-01	**DCG1**	1.18e-02	LAP4	3.47e-01	YGR125W	1.22e-01
CAN1	3.26e-03	**DUR1,2**	1.05e-02	**MEP1**	2.74e-03	YHI9	1.11e-01
CPS1	1.09e-01	DUR3	1.39e-01	**MEP2**	1.03e-02		
DAL1	1.01e-02	ECM38	1.18e-01	**MEP3**	3.09e-03		
DAL2	3.09e-03	**GAP1**	2.91e-03	PEP4	2.44e-01		

Table 8.8: Ranking by decreasing "inferred NCR-sensitivity" of the genes whose "inferred NCR-sensitivity" (8.3) is higher than the threshold (8.4). Known NCR regulators (RNCR) and annotated NCR genes (ANCR) are shown in boldface and marked with two (**) and one asterisks (*), respectively. Genes in one of the three sets of putative NCR genes $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$ (but not in RNCR nor in ANCR) are shown in boldface (without asterisk).

Rank	Gene	P-cor	Rank	Gene	P-cor	Rank	Gene	P-cor
1.	**YSP3**	0.076	34.	ARA2	0.0327	67.	RTG2	0.0213
2.	**ASP3-1***	0.073	35.	ARG80	0.0327	68.	**MOH1**	0.021
3.	**ASP3-2***	0.073	36.	FBP26	0.0314	69.	PXR1	0.0208
4.	**ASP3-3***	0.073	37.	RBG1	0.028	70.	YOR1	0.0205
5.	**ASP3-4***	0.073	38.	FUN12	0.028	71.	ALD2	0.0204
6.	CUP9	0.072	39.	YNR071C	0.028	72.	IST1	0.0201
7.	SSA4	0.0684	40.	DMA2	0.0278	73.	CWH41	0.02
8.	PPM1	0.0643	41.	LYS20	0.0277	74.	FMP41	0.0198
9.	PPZ2	0.0642	42.	MGA2	0.0267	75.	**DAL1***	0.0198
10.	SRY1	0.0592	43.	**SLX9**	0.0267	76.	**DAL4***	0.0198
11.	**AVT7**	0.059	44.	TOM20	0.0267	77.	LSC2	0.0196
12.	**GDH3***	0.0587	45.	**DAL3***	0.0263	78.	**GAT1****	0.0195
13.	**DAL7***	0.0587	46.	FRS2	0.0257	79.	**MEP2***	0.0195
14.	**ECM37**	0.0578	47.	RAT1	0.0249	80.	RPL25	0.019
15.	**YOL019W**	0.0573	48.	VPS21	0.0247	81.	**DUR1,2***	0.0189
16.	IRR1	0.0567	49.	IST3	0.0243	82.	YMR010W	0.0189
17.	RTA1	0.0556	50.	PRP46	0.0241	83.	PTI1	0.0188
18.	YHR202W	0.0549	51.	YPL150W	0.0241	84.	YAL061W	0.0188
19.	**YIL089W**	0.0546	52.	NAR1	0.0239	85.	**DAL5***	0.0186
20.	UBI4	0.048	53.	**LAP3**	0.0239	86.	PEX3	0.0184
21.	YAL037W	0.0472	54.	YHC1	0.0236	87.	UBX5	0.0184
22.	YAL037C-A	0.0449	55.	HTD2	0.0236	88.	DED81	0.0184
23.	**MEP1***	0.0436	56.	**UGA4***	0.0233	89.	YHR020W	0.0184
24.	YGR121W-A	0.0423	57.	**GZF3****	0.0233	90.	THI80	0.0182
25.	**GAP1***	0.0423	58.	MPD1	0.0231	91.	YHL015W-A	0.0181
26.	SPO14	0.0386	59.	YOR289W	0.0231	92.	YPR091C	0.0181
27.	**DAL2***	0.0383	60.	**GUD1**	0.0226	93.	YLR446W	0.0181
28.	**MEP3***	0.0376	61.	**YDL237W**	0.0226	94.	IME1	0.018
29.	**NPR2**	0.0375	62.	SNU13	0.0219	95.	PTR3	0.018
30.	**AVT4**	0.0363	63.	ILV2	0.0219			
31.	HXT5	0.0331	64.	**YGK3**	0.0218			
32.	**CAN1***	0.033	65.	YHR112C	0.0214			
33.	**DAL80****	0.033	66.	YHR113W	0.0214			

Table 8.9: Ranking of genes (top 25) by decreasing partial correlation (8.3) with their description (retrieved with RSAT [250]).

Rank	Sys. name	Stand. name	P-cor	Description
1.	YOR003W	YSP3	0.076	Putative precursor to the subtilisin-like protease III [YOR003W;YSP3;YSP3;YOR003W;854164;6324576;NP_014645]
2.	YLR155C	ASP3-1	0.073	Cell-wall L-asparaginase II, involved in asparagine catabolism; expression is induced during nitrogen starvation; four copies of ASP3 are present in the genome reference strain S288C [YLR155C;ASP3-1;ASP3-1;YLR155C;850850;6323184;ASP3;NP_013256]
3.	YLR157C	ASP3-2	0.073	Cell-wall L-asparaginase II, involved in asparagine catabolism; expression is induced during nitrogen starvation; four copies of ASP3 are present in the genome reference strain S288C [YLR157C;ASP3-2;ASP3-2;YLR157C;850852;6323186;ASP3;NP_013258]
4.	YLR158C	ASP3-3	0.073	Cell-wall L-asparaginase II, involved in asparagine catabolism; expression is induced during nitrogen starvation; four copies of ASP3 are present in the genome reference strain S288C [YLR158C;ASP3-3;ASP3-3;YLR158C;850855;6323187;ASP3;NP_013259]
5.	YLR160C	ASP3-4	0.073	Cell-wall L-asparaginase II, involved in asparagine catabolism; expression is induced during nitrogen starvation; four copies of ASP3 are present in the genome reference strain S288C [YLR160C;ASP3-4;ASP3-4;YLR160C;850857;6323189;ASP3;NP_013261]
6.	YPL177C	CUP9	0.072	Homeodomain-containing transcriptional repressor of PTR2, which encodes a major peptide transporter; imported peptides activate ubiquitin-dependent proteolysis, resulting in degradation of Cup9p and de-repression of PTR2 transcription [YPL177C; CUP9;CUP9;YPL177C;855926;6325080;NP_015148]
7.	YER103W	SSA4	0.0684	Heat shock protein that is highly induced upon stress; plays a role in SRP-dependent cotranslational protein-membrane targeting and translocation; member of the HSP70 family; cytoplasmic protein that concentrates in nuclei upon starvation [YER103W;SSA4;SSA4; YER103W;856840;6320950;YG107;NP_011029]
8.	YDR435C	PPM1	0.0643	Carboxyl methyl transferase, methylates the C terminus of the protein phosphatase 2A catalytic subunit (Pph21p or Pph22p), which is important for complex formation with regulatory subunits [YDR435C;PPM1;PPM1;YDR435C;852045;6320643;NP_010723]
9.	YDR436W	PPZ2	0.0642	Serine/threonine protein phosphatase Z, isoform of Ppz1p; involved in regulation of potassium transport, which affects osmotic stability, cell cycle progression, and halotolerance [YDR436W;PPZ2;PPZ2;YDR436W;852046;6320644;NP_010724]
10.	YKL218C	SRY1	0.0592	3-hydroxyaspartate dehydratase, deaminates L-threo-3-hydroxyaspartate to form oxaloacetate and ammonia; required for survival in the presence of hydroxyaspartate [YKL218C;SRY1;SRY1;YKL218C;853662;6322631;NP_012704]
11.	YIL088C	AVT7	0.059	Putative transporter, member of a family of seven S. cerevisiae genes (AVT1-7) related to vesicular GABA-glycine transporters [YIL088C; AVT7;AVT7;YIL088C;854721;6322103;NP_012178]
12.	YAL062W	GDH3	0.0587	NADP(+)-dependent glutamate dehydrogenase, synthesizes glutamate from ammonia and alpha-ketoglutarate; rate of alpha-ketoglutarate utilization differs from Gdh1p; expression regulated by nitrogen and carbon sources [YAL062W;GDH3;GDH3;YAL062W;851237;6319256; FUN51;NP_009339]
13.	YIR031C	DAL7	0.0587	Malate synthase, role in allantoin degradation unknown; expression sensitive to nitrogen catabolite repression and induced by allophanate, an intermediate in allantoin degradation [YIR031C;DAL7;DAL7;YIR031C;854849;6322222;MSL2;MLS2;NP_012297]
14.	YIL146C	ECM37	0.0578	Non-essential protein of unknown function [YIL146C;ECM37;ECM37;YIL146C;854660;6322045;NP_012120]
15.	YOL019W	YOL019W	0.0573	Protein of unknown function; green fluorescent protein (GFP)-fusion protein localizes to the cell periphery and vacuole [YOL019W; YOL019W;854141;6324554;NP_014623]
16.	YIL026C	IRR1	0.0567	Subunit of the cohesin complex, which is required for sister chromatid cohesion during mitosis and meiosis and interacts with centromeres and chromosome arms, essential for viability [YIL026C;IRR1;IRR1;YIL026C;854786;6322163;SCC3;NP_012238]
17.	YGR213C	RTA1	0.0556	Protein involved in 7-aminocholesterol resistance; has seven potential membrane-spanning regions [YGR213C;RTA1;RTA1;YGR213C; 853127;6321652;NP_011729]
18.	YHR202W	YHR202W	0.0549	Putative protein of unknown function; green fluorescent protein (GFP)-fusion protein localizes to the vacuole, while HA-tagged protein is found in the soluble fraction, suggesting cytoplasmic localization [YHR202W;YHR202W;856609;6321996;NP_012072]
19.	YIL089W	YIL089W	0.0546	Putative protein of unknown function [YIL089W;YIL089W;854719;6322102;NP_012177]
20.	YLL039C	UBI4	0.048	Ubiquitin, becomes conjugated to proteins, marking them for selective degradation via the ubiquitin-26S proteasome system; essential for the cellular stress response; encoded as a polyubiquitin precursor comprised of 5 head-to-tail repeats [YLL039C;UBI4;UBI4;YLL039C; 850620;6322989;UB14;SCD2;NP_013061]
21.	YAL037W	YAL037W	0.0472	Putative protein of unknown function [YAL037W;YAL037W;851194;6319280;NP_009363]
22.	YAL037C-A	YAL037C-A	0.0449	Putative protein of unknown function [YAL037C-A;YAL037C-A;1466428;33438756;NP_878040]
23.	YGR121C	MEP1	0.0436	Ammonium permease; belongs to a ubiquitous family of cytoplasmic membrane proteins that transport only ammonium (NH4+); expression is under the nitrogen catabolite repression regulation [YGR121C;MEP1;MEP1;YGR121C;853019;6321559;AMT1;NP_011636]
24.	YGR121W-A	YGR121W-A	0.0423	Putative protein of unknown function [YGR121W-A;YGR121W-A;1466458;33438796;NP_878078]
25.	YKR039W	GAP1	0.0423	General amino acid permease; localization to the plasma membrane is regulated by nitrogen source [YKR039W;GAP1;GAP1;YKR039W; 853912;6322892;NP_012965]

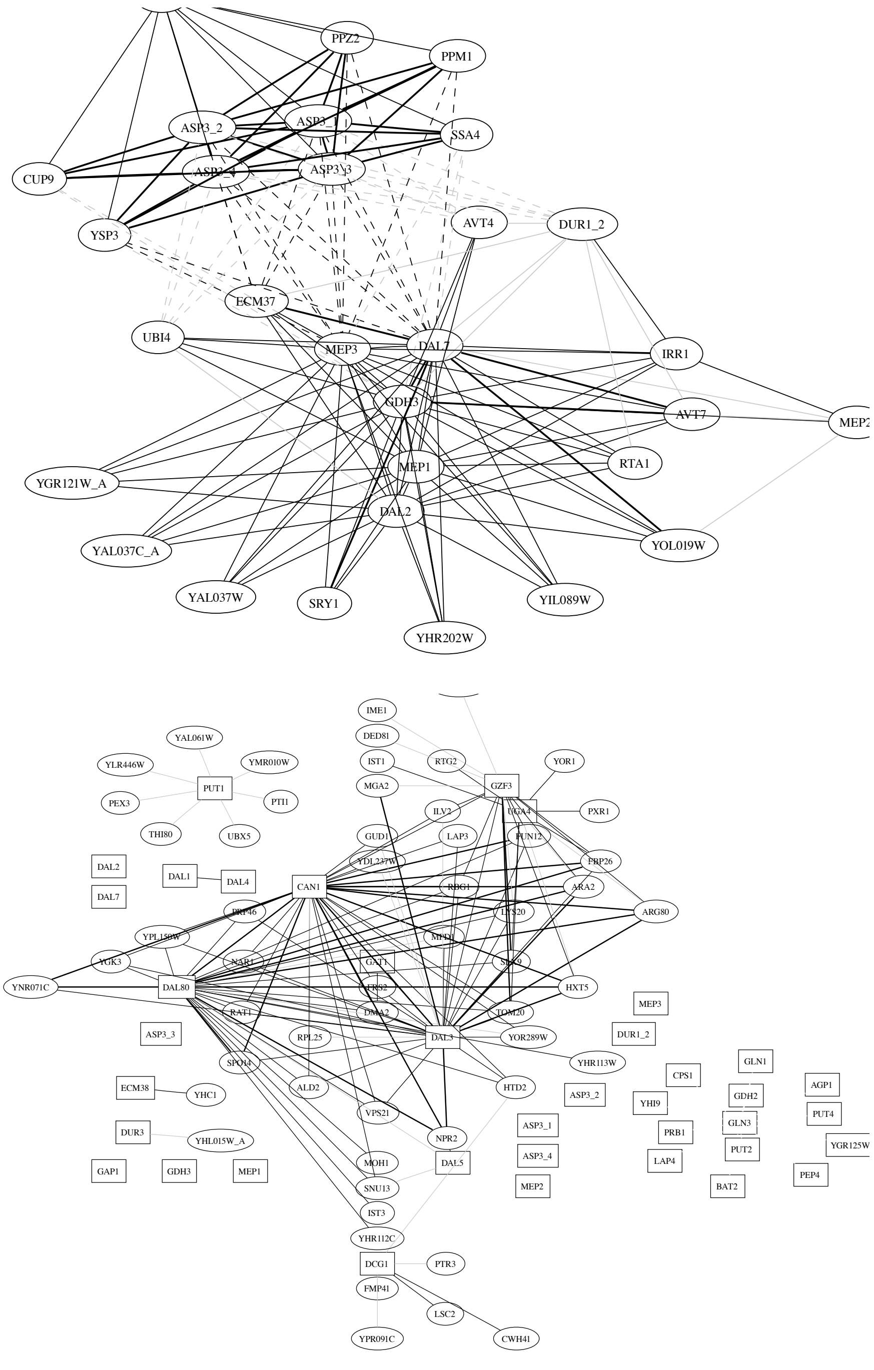


Figure 8.2: Graph composed of the core set's genes (i.e., RNCR and ANCR genes; depicted as rectangles) and the putative NCR genes (i.e., the non-bold genes of Table 8.8; depicted as ovals), and the edges between them whose corresponding full-order partial correlations are above the threshold (8.4).

(see also Figure 8.3). Among the 14 genes not appearing in the top 600 genes, one (PUT4) is known to be a "difficult case" because its two GATA-boxes are non-canonical [107].

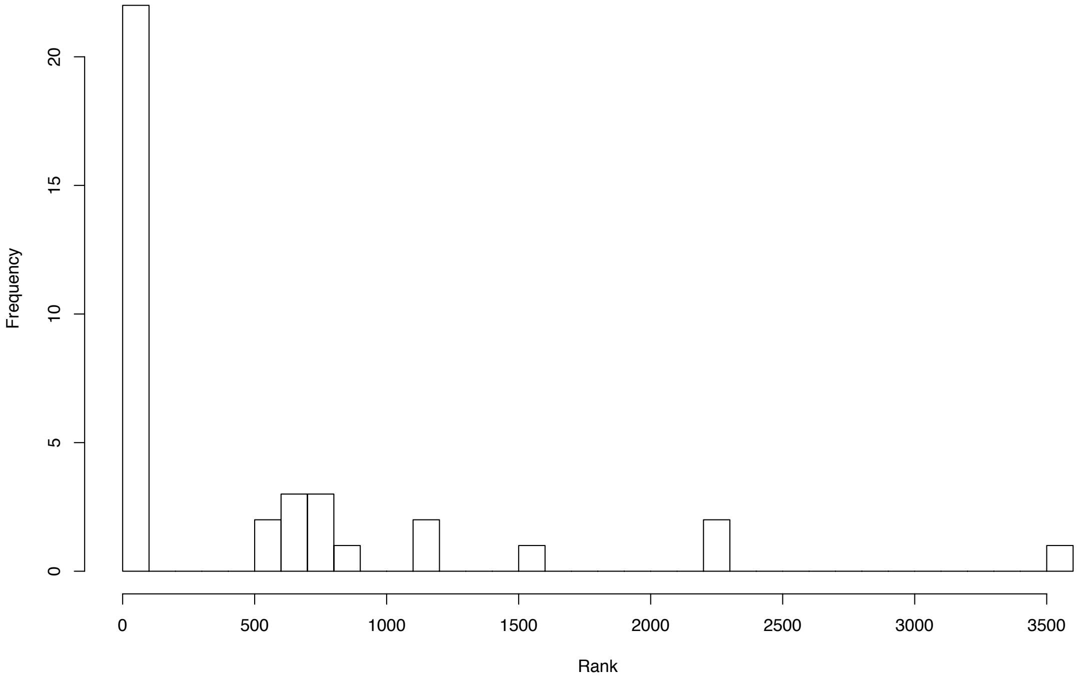


Figure 8.3: Histogram of NCR genes' rank in the final predictions' ranking.

Moreover, out of the 16 genes identified in each of the three aforementioned studies [11, 107, 214] (i.e., the intersection of the sets $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$), 9 appear in the set of 95 putative NCR genes and 12 appear in the top 600 genes ($\sim$ 10% of all genes) ranked by decreasing "inferred NCR-sensitivity," respectively. In other words, putative genes having the largest "consensus" are relatively well ranked by our approach.

8.3.4 Comparisons

We compute the intersections of the sets of putative NCR genes identified in Godard et al. [107], in Chapter 7 and in Chapter 8 with the sets of known and annotated NCR genes (RNCR and ANCR) and all possible combinations of intersections and unions of the sets $\mathcal{P}_g$, $\mathcal{P}_s$ and $\mathcal{P}_b$ arising from the aforementioned experimental studies [11, 107, 214]. We also assess the significancy of these intersections by computing p-values from the hypergeometric distribution (Appendix J). Results are presented in Table 8.10. Except for the RNCR genes with Chapter 7's approach, all the p-values are smaller than 0.0001 and are thus highly significant for all three sets. However, differences exist between the three methods. The most significant results are obtained by Godard et al. [107]'s approach, followed by the methods proposed in this chapter and Chapter 7, respectively.

Interestingly, there are large intersections between the sets of genes identified by the three methods (Figure 8.4), and at the same time, each method still identifies genes not inferred as NCR-sensitive by the other two. This suggests a certain complementarity between the different approaches.

Table 8.10: Number of genes in common and hypergeometric p-values. Note that Godard et al. [107] refers to the 100 genes identified by the bioinformatics procedure described in Section 7.1, while $\mathcal{P}_g$ refers to the 140 genes identified experimentally (i.e., with DNA microarrays) in Godard et al. [107].

Set (number of genes)	Godard et al. [107]	Chapter 7	Chapter 8
RNCR (4)	3 (1.78×10^{-5})	1 (1.27×10^{-1})	3 (4.48×10^{-5})
RCNR+ANCR (38)	30 (1.69×10^{-48})	9 (2.99×10^{-6})	19 (3.85×10^{-22})
$\mathcal{P}_g$ (140)	32 (5.92×10^{-29})	25 (2.82×10^{-12})	20 (2.65×10^{-11})
$\mathcal{P}_s$ (87)	26 (8.62×10^{-27})	17 (2.49×10^{-9})	19 (3.02×10^{-14})
$\mathcal{P}_b$ (83)	23 (7.37×10^{-23})	18 (1.34×10^{-10})	13 (3.20×10^{-8})
$\mathcal{P}_g \cup \mathcal{P}_s$ (188)	40 (2.82×10^{-35})	30 (3.35×10^{-13})	29 (5.40×10^{-17})
$\mathcal{P}_g \cup \mathcal{P}_b$ (197)	39 (4.97×10^{-33})	32 (2.95×10^{-14})	24 (7.47×10^{-12})
$\mathcal{P}_s \cup \mathcal{P}_b$ (149)	36 (1.10×10^{-33})	27 (2.41×10^{-13})	25 (1.31×10^{-15})
$\mathcal{P}_g \cup \mathcal{P}_s \cup \mathcal{P}_b$ (240)	46 (7.39×10^{-39})	36 (7.77×10^{-15})	33 (9.79×10^{-18})
$\mathcal{P}_g \cap \mathcal{P}_s$ (39)	18 (9.39×10^{-23})	12 (2.40×10^{-9})	10 (9.34×10^{-9})
$\mathcal{P}_g \cap \mathcal{P}_b$ (26)	16 (4.74×10^{-23})	11 (2.14×10^{-10})	9 (2.73×10^{-9})
$\mathcal{P}_s \cap \mathcal{P}_b$ (21)	13 (6.27×10^{-19})	8 (1.86×10^{-7})	7 (2.35×10^{-7})
$\mathcal{P}_g \cap \mathcal{P}_s \cap \mathcal{P}_b$ (16)	12 (4.03×10^{-19})	7 (3.65×10^{-7})	7 (2.54×10^{-8})

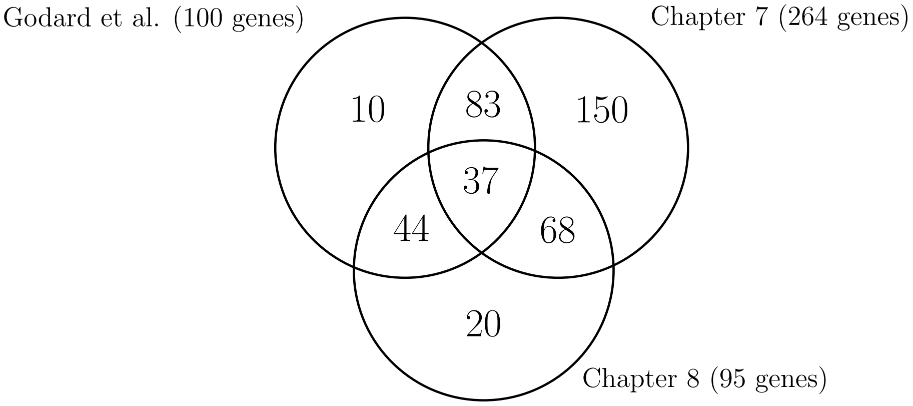


Figure 8.4: Intersections between the sets of genes identified in Godard et al. [107], in Chapter 7 and in Chapter 8.

8.4 Conclusion

We proposed an approach based on Gaussian graphical models (GGMs) to identify putative NCR genes from putative NCR regulatory motifs and over-represented motifs in the upstream noncoding sequences of annotated NCR genes. Because of the high-dimensionality of the data, we used a shrinkage estimator of the covariance matrix to infer the GGMs, which is statistically efficient and fast to compute.

We showed that our approach makes significant and biologically valid predictions by comparing these predictions to annotated and putative NCR genes, and by performing negative controls (Section 8.3). We also showed that the GGM is more effective (overall) than the independence graph. This result underlines the importance of being able to distinguish direct from indirect interactions. Moreover, this also shows that increasing the threshold to remove the spurious interactions inferred by the independence graph does not solve the problem. These results suggest that our approach can successfully identify potential NCR genes in *S. cerevisiae*. Nonetheless, we note that the independence graph also produces significant results (compared to the negative control) and that the GGM does not always outperform it.

However, the results obtained with Godard et al. [107]'s approach are more significant than those obtained with the proposed procedure. Nonetheless, the proposed method performs better than the extended classification approach of Chapter 7. Moreover, the visualization of the graph inferred with the proposed method offers the possibility to biologists to conducted a more refined analysis.

Note that the proposed approach can readily be adapted to any type of data (e.g., expression data), and to any biological process of interest in any sequenced organism. It only requires a (possibly small) set of genes known (or hypothesised) to be involved in the biological process of interest and a data matrix whose samples are related to this process, e.g. over-represented motifs in the upstream noncoding sequences. Of course, other type of data, or even combination of different data (e.g., over-represented motifs and expression data) can also be used. Finally, note that we do not endorse the GGM as the “true model” of multivariate dependencies between genes. Rather, we see it as a useful exploratory tool.

9

Conclusion

9.1 Summary of main results

With the advent of high-throughput technologies, biology is experiencing an unprecedented data surge. The main challenge–for which computers have become indispensable–is to extract useful information from this wealth of data. We have tackled two such tasks in this thesis: the reverse engineering of gene regulatory networks (GRNs) from DNA microarray data and the inference of nitrogen catabolite repression (NCR) target genes in the yeast *Saccharomyces cerevisae.*

9.1.1 Reverse engineering gene regulatory networks from DNA microarray data

The process of reverse engineering GRNs from DNA microarray data is far from being trivial because of the poor information content of expression data, which are corrupted by substantial amounts of measurement noise, and the combinatorial nature of the problem. Indeed, gene expression levels are regulated by the combined action of multiple gene products. Moreover, the "small n, large p" data setting renders learning tasks in molecular biology more challenging.

We tackled this problem of reverse engineering GRNs from DNA microarray data with Gaussian graphical models (GGMs). These models have become very popular in bioinformatics as they enable to distinguish between direct and indirect interactions. Unfortunately, GGM selection is an ill-posed problem in the "small n, large p" setting. Indeed, the usual sample concentration matrix—the maximum likelihood estimate of the (population) concentration matrix—requires the sample covariance matrix to be positive definite and this holds, with probability one, if and only if $n > p$.

To cope with this dimensionality issue, two approaches have been proposed in the literature. The first one uses regularization and the second one uses limited-order partial correlation graphs, or q-partial correlation graphs. The underlying idea of these approaches is to restrict the complexity of the models. Indeed, by introducing (a priori undesirable) bias in the model selection procedure, one reduces the variance of the selected model thereby increasing its accuracy. However, issues arise in both cases. Our two first contributions, which consist in a new shrinkage estimator and an algorithm–the *q-nested procedure*–to efficiently infer q-partial correlation graphs tackle these problems.

First, we showed that the optimal shrinkage intensity estimator of Ledoit and Wolf [153]'s shrinkage estimator is biased. Subsequently, we proposed an improved shrinkage estimator of the covariance matrix that corrects this bias through a parametric bootstrap approach. The applicability and usefulness of our estimator were demonstrated on both simulated and real expression data.

Our second contribution consists in the q-nested procedure, an algorithm to efficiently infer q-partial correlation graphs for GGM selection. Indeed, serious issues arise when inferring q-partial correlation graphs with the existing methods and hinder the applicability of these graphs for GGM selection. We showed that our algorithm efficiently copes with these problems and outperforms state-of-the-art methods on simulated data.

9.1.2 Predicting nitrogen catabolite repression target genes

The second important and challenging task that we addressed in the thesis is gene function prediction. Often, biologists know the function of some (but not all) genes with respect to a specific process and their goal is to infer other genes involved in this process.

In particular, we tackled the inference of NCR target genes in the yeast *Saccharomyces cerevisae*. The study of such a model organism is indispensable for the understanding of more complex ones. NCR is the process studied in the ARC project that supported the work presented in this thesis. It is an important biological process in *S. cerevisae* which involves an essential nutrient for all life forms: nitrogen. The ultimate goal is to identify all genes involved in NCR.

We first tackled this problem of inferring NCR target genes by adopting a "standard" two-class classification approach. We then used GGMs to propose a new approach for predicting NCR genes based on a network inference paradigm. The network structure gives further insight into the considered problem by providing a more subtle and rich picture. We deem that a network inference approach is more adequate to deal with such a problem.

9.2 Future work

Extensions of the proposed methods include the combination of multiple sources of information (for example expression data *and* occurrences of motifs of interest in upstream noncoding sequences) and the use of prior knowledge to further increase their accuracy.

Although GGM selection methods seem robust to the assumption of independent and identically distributed (i.i.d.) data, it would be interesting to adapt the proposed GGM selection methods (the improved shrinkage estimator and the q-nested procedure) to cases where this assumption does not hold, such as with time-series data [206].

Concerning the q-nested procedure, it would also be interesting to study the effect of the topology of the graph to be inferred (e.g., a graph whose degree distribution follows a power law) on the performance of the algorithm. Additionally, the relative influence of the screening procedure and the selection of partial correlations in the neighborhoods could be assessed independently of each other. Furthermore, nonlinear measures of independence, such as (conditional) mutual information, could be integrated to our algorithm. Another important issue is the control of Type II error (false negative) in multiple testing. Indeed, the Type II error is only controlled indirectly through the Type I error (false positive). However, in large sparse graphs such as GRNs, it might be useful to focus on controlling the Type II error in order for all edges not to be removed [30].

Concerning the inference of putative NCR genes, future work is mainly concerned by further experimental validation of the obtained results. Indeed, it would be interesting to assess the quality of the inferred set of genes by means of automated tools that query biological databases (e.g., FatiGO). Ideally, the putative NCR genes could be tested in vitro for NCR-sensitivity.

Given the availability of ever-growing volumes of data and the constant need to extract useful knowledge from them, we venture that the use of robust methods such as the ones developed in this thesis are–and will remain–of uttermost importance for scientists–who are “drowning in information”–not to “be starved for knowledge.” Hopefully, we will be able one day to build a holistic view of biological systems.

Appendices

A

Notation

Unless otherwise stated, we adopt the following notational conventions:

- random variables are denoted in boldface and their realizations are denoted in normal font;
- matrices are denoted in uppercase while vectors and scalars are denoted in lowercase;
- sets are denoted by the calligraphic symbols $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{G}$, $\mathcal{V}$, $\mathcal{E}\ldots$, except for domains of variables which are denoted by the calligraphic symbols $\mathscr{X}$, $\mathscr{Y}$, $\ldots$;
- estimators are denoted in boldface (since they are random variables) and surmounted by a hat (e.g., $\hat{\boldsymbol{\Theta}}$ is an estimator of the unknown quantity Θ);
- the input and output of a learning machine are denoted by the p-dimensional random vector $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$ and the random variable $\mathbf{y}$, respectively;
- the i-th sample of variables $\mathbf{x}$ and $\mathbf{y}$ are denoted by $x_{i\cdot} = (x_{i1}, \ldots, x_{ip})^T$ and y_i, $i = 1, \ldots, n$, respectively;
- we let $X = (x_{1\cdot}, \ldots, x_{n\cdot})^T$ denote the $n \times p$ data matrix with i-th row given by $x_{i\cdot}$ and we let $y = (y_1, \ldots, y_n)^T$ denote the n-dimensional *response vector*;
- we let $D_n = \{(x_{i\cdot}, y_i), i = 1, \ldots, n\}$ denote the data set available observations.

Equations are numbered only if they are referenced in the text.

A.1 **Probability**

$\mathbb{P}(A)$	Probability of event A.
$\mathcal{X}$	Sample space.
$F_{\mathbf{x}}(x)$	Cumulative distribution function of random variable $\mathbf{x}$.
$f_{\mathbf{x}}(x)$	Probability density function of continuous random variable $\mathbf{x}$.
$m_{\mathbf{y}}(y)$	Probability mass function of discrete random variable $\mathbf{y}$.
$\mathbb{E}(\mathbf{x})$	Expectation of random variable $\mathbf{x}$.
$\widehat{\mathbf{E}}(\mathbf{x})$	Estimate of $\mathbb{E}(\mathbf{x})$.
$\mathrm{Var}(\mathbf{x})$	Variance of random variable $\mathbf{x}$.
$\widehat{\mathbf{Var}}(\mathbf{x})$	Estimate of $\mathrm{Var}(\mathbf{x})$.
$\mathrm{Cov}(\mathbf{x}, \mathbf{y})$	Covariance between random variables $\mathbf{x}$ and $\mathbf{y}$.
$\mathrm{Cor}(\mathbf{x}, \mathbf{y})$ or $\rho_{(\mathbf{x}, \mathbf{y})}$	Correlation between random variables $\mathbf{x}$ and $\mathbf{y}$.
$\mathrm{Bias}(\mathbf{x})$	Bias of random variable $\mathbf{x}$.

A.2 Matrices

$\|\cdot\|_F$	Frobenius norm.
$\mathrm{tr}\left(\cdot\right)$	Trace.
Σ	Covariance matrix.
$\hat{\mathbf{S}}_{\mathrm{ML}}$	Maximum likelihood estimator of Σ.
$\hat{\mathbf{S}}$	Unbiased sample covariance matrix.
$\hat{\mathbf{T}}$	Low-dimensional (biased) estimator of Σ.
$\hat{\boldsymbol{\Sigma}}_{\lambda}$	Linear shrinkage estimator.
λ	Shrinkage intensity.
λ^*	Optimal shrinkage intensity.
$\hat{\boldsymbol{\lambda}}^*$	Estimate of the optimal shrinkage intensity.
$\hat{\boldsymbol{\Omega}}^*$	Optimal shrinkage estimator of the concentration matrix.
$\hat{\boldsymbol{\omega}}^*_{ij}$	The (i, j)-th element of $\hat{\boldsymbol{\Omega}}^*$.
$\hat{\boldsymbol{\lambda}}^*_{\mathrm{bc}}$	"Bias-corrected" optimal shrinkage intensity estimator.
$\hat{\boldsymbol{\Sigma}}^*_{\mathrm{bc}}$	"Bias-corrected" optimal shrinkage estimator of Σ.
$\hat{\boldsymbol{\rho}}^*_{(i,j\|\mathcal{S})}$	Optimal shrinkage estimator of the partial correlation $\rho_{(i,j\|\mathcal{S})}$.

A.3 Graph

$\mathcal{V}$	Vertex set.
$\mathcal{E}$	Edge set.
$G = (\mathcal{V}, \mathcal{E})$	Graph G with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$.
$\overline{\mathcal{E}}$	Missing edge set.
$\{\alpha, \beta\}$, $\alpha \sim_G \beta$	Edge from node α to node β.
$\mathrm{bd}_G\left(\alpha\right)$	Boundary of node α.
$\mathrm{cl}_G\left(\alpha\right)$	Closure of node α.
γ_G	Edge proportion in G.
d_G	Average degree of nodes in G.
$\alpha \mapsto_G \beta$	Path from α to β.
$\mathcal{S}_G\left(\alpha, \beta\right)$	Set of all nontrivial minimal (α, β)-separators in G.
$d_G\left(\alpha, \beta\right)$	Connectivity of α and β.
$G_{(q)} = \left(\mathcal{V}, \mathcal{E}_{(q)}\right)$	q-Partial (correlation) graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}_{(q)}$.
$d_G^{out}\left(i, j\right)$	Outer connectivity of nodes i and j.
$d_G^{out}\left(\overline{\mathcal{E}}\right)$	Outer connectivity of the missing edges.

B

Abbreviations and Acronyms

ANCR Annotated NCR genes.

AUC Area under the ROC curve.

AUC-PR Area under the PR curve.

BER Balanced error rate.

BLUE Best linear unbiased estimator.

bp Base pairs.

DNA Deoxyribonucleic acid.

ERM Empirical risk minimization.

FDR False discovery rate.

FP False positive.

FPR False positive rate.

GGM Gaussian graphical model.

GRN Gene regulatory network.

GSO Gram-Schmidt orthogonalization.

i.i.d. Independent and identically distributed.

KNN k-Nearest neighbors.

LARS Least angle regression.

LDA Linear discriminant analysis.

loo Leave-one-out.

MCMC Markov chain Monte Carlo.

MISE Mean integrated squared error.

MLE Maximum likelihood estimator.

mRNA Messenger RNA.

MSE Mean squared error.

NB Naive Bayes.

NCR Nitrogen catabolite repression.

NNCR Genes known to be insensitive to NCR.

ODE Ordinary differential equation.

OLS Ordinary least squares.

ORF Open reading frame.

pAUC Partial area under the ROC curve.

PCA Principal components analysis.

PR Precision-recall.

PRIAL Percentage relative improvement in average loss.

QDA Quadratic discriminant analysis.

RDA Regularized discriminant analysis.

RNCR NCR regulators.

ROC Receiver operator characteristic.

RSAT Regulatory Sequence Analysis Tools.

SVD Singular value decomposition.

SVM Support vector machine.

TF Transcription factor.

TP True positive.

TPR True positive rate.

C

Biology Glossary[1]

DNA microarray: Miniaturized array of a large number (into the thousands) of unique DNA sequences spotted robotically onto glass slides or other solid substrates. Microarrays are used to simultaneously study large numbers of genes and their regulation, by probing with labeled nucleic acids taken from biological samples.

Eukaryote: Any of the single-celled or multicellular organisms whose cell contains a distinct, membrane-bound nucleus (e.g., *Saccharomyces cerevisae*).

Gene expression: The conversion of the information from the gene into mRNA via transcription and then to protein via translation resulting in the phenotypic manifestation of the gene.

Gene (expression) regulation: The modulation of any of the stages of gene expression, hence, it encompasses the various systems that control and determine which genes are switched on and off, and when, how long, and to what extent the genes are expressed.

Hybridisation: The process of forming a double-stranded nucleic acid from two complementary strands of DNA (or RNA).

Messenger RNA (mRNA): Single stranded rNA molecule that specifies the amino acid sequence of one or more polypeptide chains. This information is translated during protein synthesis when ribosomes bind to the mRNA. In prokaryotes, mRNA is normally formed by splicing a large primary transcript from a dNA sequence and protein synthesis starts while the mRNA is still being synthesised. Prokaryote mRNAs are usually very short lived. In contrast, in eukaryotes the primary transcripts (hnRNA) are synthesised in the nucleus and they are extensively processed to give the mRNA that is exported to the cytoplasm where protein synthesis takes place.

Open reading frame: A reading frame in a sequence of nucleotides in dNA that contains no termination codons and so can potentially translate as a polypeptide chain.

Prokaryote: Any of the group of organisms primarily characterized by the lack of true nucleus and other membrane-bound cell compartments: such as mitochondria and chloroplasts, and by the possession of a single loop of stable chromosomal DNA in the nucleiod region and cytoplasmic structures, such as plasma membrane, vacuoles, primitive cytoskeleton, and ribosomes.

[1] Source: `http://www.biology-online.org/`

Regulatory gene: A gene that is involved in the production of a substance that controls or regulates the expression of one or more genes.

Reverse transcription: The process of making a double stranded DNA molecule from a single stranded RNA template through the enzyme, reverse transcriptase. It is called reverse transcription because it is a process in opposite or reverse of transcription.

Ribosome: A molecule consisting of two subunits that fit together and work as one to build proteins according to the genetic sequence held within the messenger RNA (mRNA). Using the mRNA as a template, the ribosome traverses each codon, pairing it with the appropriate amino acid. This is done through interacting with transfer RNA (tRNA) containing a complementary anticodon on one end and the appropriate amino acid on the other.

DNA sequencing: Any lab technique used to find out the sequence of nucleotide bases in a DNA molecule or fragment.

Transcriptome: The set of all mRNA molecules (or transcripts) in one or a population of biological cells for a given set of environmental circumstances. Therefore, unlike the genome, which is fixed for a given organism (apart from genetic polymorphism), the transcriptome varies depending upon the context of the experiment.

D

Probability Essentials

A *probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ consists of a set Ω (called the sample space), a σ-algebra $\mathcal{F}$ of subsets of Ω (called the events), and a probability measure $\mathbb{P}$ on the measurable space $(\Omega, \mathcal{F})$ [16, 82, 83, 129].

Definition D.0.1 (cumulative distribution function)**.** *The* cumulative distribution function *(CDF) of the random variable* $\mathbf{x}$ *is the function* $F_{\mathbf{x}} : \mathbb{R} \to [0, 1]$ *defined by*

$$F_{\mathbf{x}} = \mathbb{P}\left(\mathbf{x} \leq x\right) \ .$$

Definition D.0.2 (discrete random variable and probability mass function)**.** *A random variable* $\mathbf{y}$ *is* discrete *if it takes countably many values. The* probability mass function *of* $\mathbf{y}$ *is defined by*

$$m_{\mathbf{x}}\left(x\right) = \mathbb{P}\left(\mathbf{x} = x\right) \ .$$

Definition D.0.3 (continuous random variable and probability density function)**.** *A random variable* $\mathbf{x}$ *is* continuous *if there exists a function* $f_{\mathbf{x}}$ *such that* $f_{\mathbf{x}} \geq 0$ *for all* x*,*

$$\int_{-\infty}^{\infty} f_{\mathbf{x}}(x) dx = 1 \ ,$$

and for every $a \leq b$*,*

$$\mathbb{P}\left(a < \mathbf{x} < b\right) = \int_{a}^{b} f_{\mathbf{x}}(x) dx \ .$$

The function $f_{\mathbf{x}}$ *is called the* probability density function *(PDF). We have that*

$$F_{\mathbf{x}}(x) = \int_{-\infty}^{x} f_{\mathbf{x}}(t) dt \ ,$$

and $f_{\mathbf{x}}(x) = \frac{d}{dx} F_{\mathbf{x}}(x)$ *at all points* x *at which* $F_{\mathbf{x}}$ *is differentiable.*

Definition D.0.4 (marginal density function)**.** *If* $(\mathbf{x}, \mathbf{y})$ *have joint distribution with density function* $f_{\mathbf{x},\mathbf{y}}$*, then the* marginal density functions *are*

$$f_{\mathbf{x}}(x) = \int f_{\mathbf{x},\mathbf{y}}(x, y) dy \ , \quad f_{\mathbf{y}}(y) = \int f_{\mathbf{x},\mathbf{y}}(x, y) dx \ .$$

Definition D.0.5 (empirical distribution function)**.** *The* empirical distribution function $\hat{F}_n$ *is the CDF that puts mass* $1/n$ *at each of the* n *data points* $x_i, i = 1, \ldots, n$*:*

$$\hat{F}_n(x) = \frac{\sum_{i=1}^{n} I\left(x_i \leq x\right)}{n} .$$

where I *is the indicator function*

$$I\left(x_i \leq x\right) = \begin{cases} 1 & \text{if } x_i \leq x \ , \\ 0 & \text{if } x_i > x \ . \end{cases}$$

E

Bias-Variance Decomposition

Let $\hat{\boldsymbol{\alpha}}$ be the estimate returned by a given model for an (unknown) parameter α. As the complexity of the model increases, it is able to better "extract information" from the available sample. It will therefore return a more accurate estimate on average than would a simpler model. Its bias, defined as

$$\mathrm{Bias}\left(\hat{\boldsymbol{\alpha}}\right) = \alpha - \mathbb{E}\left(\hat{\boldsymbol{\alpha}}\right) \ ,$$

will hence be smaller than for a simpler model.

On the other hand, as the complexity increases, the model is also more sensitive to the sample than a simpler model would be. Its variance, defined as

$$\mathrm{Var}\left(\hat{\boldsymbol{\alpha}}\right) = \mathbb{E}\left(\left(\alpha - \mathbb{E}\left(\hat{\boldsymbol{\alpha}}\right)\right)^2\right) \ ,$$

will hence be higher.

Importantly, both bias and variance contribute to the mean squared error (MSE) as characterized by the *bias-variance decomposition* [116] of the MSE of the estimator $\hat{\boldsymbol{\Theta}}$ of Θ:

$$\mathrm{MSE}\left(\hat{\boldsymbol{\alpha}}\right) = \left(\mathrm{Bias}\left(\hat{\boldsymbol{\alpha}}\right)\right)^2 + \mathrm{Var}\left(\hat{\boldsymbol{\alpha}}\right) \ . \tag{E.1}$$

This decomposition is easily derived from the well-known relation for the variance of a random variable $\boldsymbol{\theta}$,

$$\mathrm{Var}\left(\boldsymbol{\theta}\right) = \mathbb{E}\left(\boldsymbol{\theta}^2\right) - \left(\mathbb{E}\left(\boldsymbol{\theta}\right)\right)^2 \ , \tag{E.2}$$

by taking $\boldsymbol{\theta} = \alpha - \hat{\boldsymbol{\alpha}}$, and by recalling that

$$\mathrm{MSE}\left(\hat{\boldsymbol{\alpha}}\right) = \mathbb{E}\left(\left(\alpha - \hat{\boldsymbol{\alpha}}\right)^2\right) \ .$$

F

O-notation

The O-notation is used to measure an algorithm's asymptotic efficiency and allows us to compare the relative performance of alternative algorithms [39]. More specifically, it is used in computational complexity theory to analyze the time (typically worst case or average case running time) or space (i.e., memory usage) complexity of an algorithm as a function of the size of the input [39].

For a given function $g(n)$, we denote by $O\left(g(n)\right)$ the set of functions [39]

$$O\left(g(n)\right) = \left\{f(n) \, : \, \exists\, c, n_0 > 0 \text{ such that } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0\right\} \; .$$

The O-notation thus gives an asymptotic upper bound on a function.

G

Matrices

We give a few basic definitions and results on matrices [108] used throughout the thesis.

We consider *square real* matrices,[1] i.e., $p \times p$ matrices whose elements consist entirely of real numbers. The set of $p \times p$ real matrices is denoted by $\mathbb{R}^{p \times p}$.

A matrix $A \in \mathbb{R}^{p \times p}$ whose elements outside the diagonal are zero, i.e., $a_{ij} = 0$, $i, j = 1, \dots, p$, $i \neq j$, is called a *diagonal* matrix.

A matrix $A \in \mathbb{R}^{p \times p}$ is *symmetric* if $a_{ij} = a_{ji}$, $i, j = 1, \dots, p$.

The *inverse* of a matrix $A \in \mathbb{R}^{p \times p}$ is a matrix denoted by A^{-1} such that

$$AA^{-1} = A^{-1}A = I_p \ ,$$

where I_p denotes the $p \times p$ identity matrix. Note that $A^{-1} \in \mathbb{R}^{p \times p}$. A square real matrix has an inverse if and only if its determinant is nonzero. A matrix possessing an inverse is called *nonsingular*, *regular*, or *invertible*. A square real matrix that is not invertible is called *singular*.

A matrix $A \in \mathbb{R}^{p \times p}$ is *positive definite* if

$$x^T A x > 0 \ , \quad \text{for all } x \in \mathbb{R}^p \setminus \{0\} \ ,$$

where 0 denotes the zero vector of dimension p. It is *positive semidefinite* if

$$x^T A x \geq 0 \ , \quad \text{for all } x \in \mathbb{R}^p \setminus \{0\} \ .$$

The determinant of a positive definite matrix is always positive, so a positive definite matrix is always nonsingular.

The *condition number* for matrix inversion with respect to a matrix norm $\|\cdot\|$ of a matrix $A \in \mathbb{R}^{p \times p}$ is defined by:

$$\kappa\left(A\right) = \begin{cases} \|A\| \left\|A^{-1}\right\| & \text{if } A \text{ is nonsingular;} \\ +\infty & \text{otherwise.} \end{cases}$$

The condition number is a measure of stability or sensitivity of a matrix (or the linear system it represents) to numerical operations. Matrices with condition numbers near 1 are said to be *well-conditioned.* Matrices with condition numbers much greater than one are said to be *ill-conditioned.*

[1] Most definitions and results given here apply to or can be generalized to nonsquare and/or nonreal matrices.

H

Hypothesis Testing

H.1 Correlation

In a frequentist setting, testing whether the correlation is significantly different from zero requires the distribution function of the sample correlation $\hat{\boldsymbol{\rho}}_{(i,j)}$ under the null hypothesis $\rho_{(i,j)} = 0$ to address the statistical testing problem of non-zero correlation:

$$H_0 : \rho_{(i,j)} = 0 \qquad \text{versus} \qquad H_1 : \rho_{(i,j)} \neq 0 \, . \tag{H.1}$$

A possible solution is to resort to Fisher's Z-transform of the correlation:

$$Z_{(i,j)} = \tanh^{-1} \hat{\boldsymbol{\rho}}_{(i,j)} = \frac{1}{2} \log \left(\frac{1 + \hat{\boldsymbol{\rho}}_{(i,j)}}{1 - \hat{\boldsymbol{\rho}}_{(i,j)}} \right) \, ,$$

which has an asymptotic normal distribution under the null hypothesis H_0 [6, 86, 87]. Using a significance level α, we reject the null-hypothesis H_0 against the two-sided alternative H_1 if

$$\sqrt{n-3}\, Z_{(i,j)} > \Phi^{-1} \left(1 - \alpha/2\right) \, , \tag{H.2}$$

where $\Phi\left(\cdot\right)$ denotes the cumulative distribution function of the standard normal distribution $\mathcal{N}\left(0,1\right)$.

H.2 Multiple testing correction

A *Type I error*, or *false positive*, is committed by rejecting a true null hypothesis when it is actually true. A *Type II error*, or *false negative*, is committed by failing to reject a false null hypothesis [67].

Ideally, one would like to simultaneously minimize both the number of Type I errors and Type II errors. Unfortunately, this is impossible and one seeks a trade-off between the two types of errors. A classical approach is to specify an acceptable level α of the Type I error rate and derive testing procedures that aim to minimize the Type II error rate among the tests with Type I error level at most α [67].

Standard approaches to multiple testing control the family-wise error rate (FWER), that is, the chance of committing at least one Type I error.

These techniques are often criticized because they are very conservative, especially for large-scale testing problems such as those encountered when reverse engineering gene regulatory networks, leading to large numbers of Type II errors (i.e., these methods lack power) [67]. When many tests are performed simultaneously and a large proportion of null hypotheses are expected to be false, one may be prepared to tolerate some Type I errors, provided their number is small in comparison to the number of rejected hypotheses [67]. Therefore, multiple testing procedures for controlling the proportion of Type I errors among the rejected hypotheses, such as the *false discovery rate* (FDR) [14], have become popular for large-scale testing problems.

I

Shrinkage Estimator

I.1 Variances of individual entries

The empirical unbiased variances of the individual entries of $\hat{\mathbf{S}}$ (5.2) are computed as follows (recall that the observations are i.i.d.):

$$\begin{aligned}
\widehat{\mathbf{Var}}\left(\hat{s}_{ij}\right) &= \left(\frac{n}{n-1}\right)^2 \widehat{\mathbf{Var}}\left(\frac{1}{n}\sum_{k=1}^{n} x_{ki}x_{kj}\right) \ , \\
&= \frac{1}{\left(n-1\right)^2}\sum_{k=1}^{n}\widehat{\mathbf{Var}}\left(x_{ki}x_{kj}\right) \ , \\
&= \frac{n}{\left(n-1\right)^2}\widehat{\mathbf{Var}}\left(x_{ki}x_{kj}\right) \ , \\
&= \frac{n}{\left(n-1\right)^3}\sum_{k=1}^{n}\left(x_{ki}x_{kj} - \frac{1}{n}\sum_{k=1}^{n} x_{ki}x_{kj}\right)^2 \ .
\end{aligned}$$

I.2 Optimal shrinkage intensity

The expected quadratic loss is given by:

$$\begin{aligned}
\mathbb{E}\left(\left\|\hat{\mathbf{\Sigma}}_\lambda - \Sigma\right\|_F^2\right) &= \mathbb{E}\left(\sum_{i=1}^{p}\sum_{j=1}^{p}\left(\hat{\mathbf{\Sigma}}_{\lambda,ij} - \Sigma_{ij}\right)^2\right) \ , \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p}\mathbb{E}\left(\left(\hat{\mathbf{\Sigma}}_{\lambda,ij} - \Sigma_{ij}\right)^2\right) \ , \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p}\left\{\mathrm{Var}\left(\hat{\mathbf{\Sigma}}_{\lambda,ij}\right) + \left(\mathbb{E}\left(\hat{\mathbf{\Sigma}}_{\lambda,ij}\right) - \Sigma_{ij}\right)^2\right\} \ , \qquad \text{(I.1)}
\end{aligned}$$

where we used (E.2) to derive (I.1). From the definition of $\hat{\mathbf{\Sigma}}_{\lambda,ij}$ (5.4), we have that

$$\mathrm{Var}\left(\hat{\mathbf{\Sigma}}_{\lambda,ij}\right) = \lambda^2\,\mathrm{Var}\left(\hat{\mathbf{T}}_{ij}\right) + \left(1-\lambda\right)^2\,\mathrm{Var}\left(\hat{\mathbf{S}}_{ij}\right) + 2\lambda\left(1-\lambda\right)\mathrm{Cov}\left(\hat{\mathbf{T}}_{ij}, \hat{\mathbf{S}}_{ij}\right) \ , \qquad \text{(I.2)}$$

and

$$\begin{aligned}\mathbb{E}\left(\hat{\mathbf{\Sigma}}_{\lambda,ij}\right) &= \lambda\,\mathbb{E}\left(\hat{\mathbf{T}}_{ij}\right) + (1-\lambda)\;\mathbb{E}\left(\hat{\mathbf{S}}_{ij}\right)\ , \\ &= \lambda\,\mathbb{E}\left(\hat{\mathbf{T}}_{ij} - \hat{\mathbf{S}}_{ij}\right) + \mathbb{E}\left(\hat{\mathbf{S}}_{ij}\right)\ . \end{aligned} \tag{I.3}$$

Plugging (I.2) and (I.3) in (I.1), and taking the derivative of (I.1) with respect to λ, we obtain:

$$\begin{aligned}&\frac{\delta}{\delta\lambda}\,\mathbb{E}\left(\left\|\hat{\mathbf{\Sigma}}_{\lambda} - \Sigma\right\|_F^2\right) = \\ &\quad \sum_{i=1}^{p}\sum_{j=1}^{p}\Big\{2\lambda\,\mathrm{Var}\left(\hat{\mathbf{T}}_{ij}\right) - 2\,(1-\lambda)\;\mathrm{Var}\left(\hat{\mathbf{S}}_{ij}\right) + 2\,(1-2\lambda)\,\mathrm{Cov}\left(\hat{\mathbf{T}}_{ij},\hat{\mathbf{S}}_{ij}\right) \\ &\qquad +2\,\mathbb{E}\left(\hat{\mathbf{T}}_{ij} - \hat{\mathbf{S}}_{ij}\right)\left(\lambda\,\mathbb{E}\left(\hat{\mathbf{T}}_{ij} - \hat{\mathbf{S}}_{ij}\right) + \mathbb{E}\left(\hat{\mathbf{S}}_{ij}\right) - \Sigma_{ij}\right)\Big\}\ .\end{aligned}$$

We rearrange the terms to get:

$$\begin{aligned}&\frac{\delta}{\delta\lambda}\,\mathbb{E}\left(\left\|\hat{\mathbf{\Sigma}}_{\lambda} - \Sigma\right\|_F^2\right) = \\ &\lambda\left\{\sum_{i=1}^{p}\sum_{j=1}^{p}\Big\{2\,\mathrm{Var}\left(\hat{\mathbf{T}}_{ij}\right) + 2\,\mathrm{Var}\left(\hat{\mathbf{S}}_{ij}\right) - 4\,\mathrm{Cov}\left(\hat{\mathbf{T}}_{ij},\hat{\mathbf{S}}_{ij}\right) + 2\,\mathbb{E}\left(\hat{\mathbf{T}}_{ij} - \hat{\mathbf{S}}_{ij}\right)^2\Big\}\right\} \\ &-\left\{\sum_{i=1}^{p}\sum_{j=1}^{p}\Big\{2\,\mathrm{Var}\left(\hat{\mathbf{S}}_{ij}\right) - 2\,\mathrm{Cov}\left(\hat{\mathbf{T}}_{ij},\hat{\mathbf{S}}_{ij}\right) - 2\,\mathbb{E}\left(\hat{\mathbf{T}}_{ij} - \hat{\mathbf{S}}_{ij}\right)\left(\mathbb{E}\left(\hat{\mathbf{S}}_{ij}\right) - \Sigma_{ij}\right)\Big\}\right\}\ .\end{aligned} \tag{I.4}$$

By noting that

$$\mathrm{Var}\left(\hat{\mathbf{T}}_{ij}\right) + \mathrm{Var}\left(\hat{\mathbf{S}}_{ij}\right) - 2\,\mathrm{Cov}\left(\hat{\mathbf{T}}_{ij},\hat{\mathbf{S}}_{ij}\right) = \mathrm{Var}\left(\hat{\mathbf{T}}_{ij} - \hat{\mathbf{S}}_{ij}\right)\ ,$$

and

$$\mathbb{E}\left(\hat{\mathbf{S}}_{ij}\right) - \Sigma_{ij} = \mathrm{Bias}\left(\hat{\mathbf{S}}_{ij}\right)\ ,$$

and by setting (I.4) to zero, we obtain:

$$\lambda^* = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p}\left(\mathrm{Var}\left(\hat{\mathbf{S}}_{ij}\right) - \mathrm{Cov}\left(\hat{\mathbf{T}}_{ij},\hat{\mathbf{S}}_{ij}\right) - \mathrm{Bias}\left(\hat{\mathbf{S}}_{ij}\right)\right)}{\sum_{i=1}^{p}\sum_{j=1}^{p}\mathbb{E}\left(\left(\hat{\mathbf{T}}_{ij} - \hat{\mathbf{S}}_{ij}\right)^2\right)}\ . \tag{I.5}$$

Since $\hat{\mathbf{S}}$ is unbiased, i.e., $\mathrm{Bias}\left(\hat{\mathbf{S}}_{ij}\right) = 0$, (I.5) reduces to

$$\lambda^* = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p}\left(\mathrm{Var}\left(\hat{\mathbf{S}}_{ij}\right) - \mathrm{Cov}\left(\hat{\mathbf{T}}_{ij},\hat{\mathbf{S}}_{ij}\right)\right)}{\sum_{i=1}^{p}\sum_{j=1}^{p}\mathbb{E}\left(\left(\hat{\mathbf{T}}_{ij} - \hat{\mathbf{S}}_{ij}\right)^2\right)}\ .$$

J

Hypergeometric Distribution

The *hypergeometric distribution* is a discrete probability distribution that describes the number of successes in a sequence of draws from a finite population without replacement [82].

Let there be g ways for a "good" selection and b ways for a "bad" selection out of a total of $g + b$ possibilities. The probability that the total number of successful selections $\mathbf{x}$ is equal to k in a sequence of n draws is given by:

$$\mathbb{P}\left(\mathbf{x} = k\right) = \frac{\binom{g}{k}\binom{b}{n-k}}{\binom{g+b}{n}} ,$$

where $\binom{a}{b}$ is the usual binomial coefficient, i.e., the number of b-element subsets of an a-element set. The random variable $\mathbf{x}$ is said to follow a hypergeometric distribution with parameters $g, b, n \in \mathbb{N}$.

K

Evaluating the Performance of Classifiers

We present measures based on receiver operator characteristic curves (Appendix K.1) and on precision-recall curves (Appendix K.2) to assess the performance of two-class classifiers (Section 3.7).

A label predicted as positive by a classifier is a *true positive* (TP) if the label is positive and a *false positive* (FP) otherwise. *True* and *false negatives* (TN and FN, respectively) are defined analogously.

K.1 Receiver operator characteristic curve

A *receiver operator characteristic* (ROC) *curve* is a graphical plot of the *true positive rate*

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \,,$$

versus the false positive rate

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \,,$$

for different values of the threshold.

The use of ROC curves is recommended when evaluating binary decision problems in order to avoid effects related to the chosen threshold [81, 193].

The *area under the ROC curve* (AUC) reduces ROC performance to a single scalar value representing the expected performance [81]. Being a portion of the area of the unit square, its value is comprised between 0 and 1. It corresponds to the "probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance" [81]. Note that a random classifier produces the diagonal line between $(0, 0)$ and $(1, 1)$, hence achieving an AUC of 0.5.

Sometimes we are not interested in the entire range of FPRs, but rather on very low false positive rates such as $\mathrm{FPR} < 0.05$. We then compute the *partial area under the ROC curve* (pAUC) [131, 172], which is a summary measure of the ROC curve used to make statistical inference when only a region of the ROC space is of interest. It is defined as the area below the ROC curve on $(0, u]$, with $u \in (0, 1]$, divided by u (for normalization purposes [131]).

Since (p)AUC measures are sample-based estimates, one typically reports estimates of standard deviations (for example jackknife estimates; Section 3.2.2) to be able to compare the (p)AUC values [81].

K.2 Precision-recall curve

If there is a large skew in the class distribution, as is typically the case when inferring gene regulatory networks (because of their sparseness), *precision-recall* (PR) *curves* give a more accurate picture of an algorithm's performance than ROC curves [47]. The PR curve is a diagram which plots *precision*, defined as the fraction of true positives among those inferred as positive:

$$\text{prec} = \begin{cases} \frac{\text{TP}}{\text{TP+FP}} & \text{if TP} + \text{FP} > 0 \ , \\ 0 & \text{otherwise} \ , \end{cases}$$

versus *recall*, defined as the fraction of true positives among all true labels:

$$\text{rec} = \begin{cases} \frac{\text{TP}}{\text{TP+FN}} & \text{if TP} + \text{FN} > 0 \ , \\ 0 & \text{otherwise} \ , \end{cases}$$

for different values of the threshold on a two-dimensional coordinate system [251].

These quantities depend on the threshold chosen to return a binary decision. The quality of an algorithm is measured by the *area under the PR curve* (AUPRC) [47]. Alternatively, one can compute the *F-measure*, which is defined as the harmonic mean of the precision and recall quantities:

$$F\,(\text{prec}, \text{rec}) = \begin{cases} \frac{2 \cdot \text{prec} \cdot \text{rec}}{\text{prec+rec}} & \text{if prec} + \text{rec} > 0 \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{K.1}$$

L

Nitrogen Catabolite Repression

L.1 Sets of genes

The set of 4 NCR regulators (RNCR) [107, 146, 147] is given in Table L.1.

Table L.1: NCR regulators (RNCR).

Standard name(s)	Systematic name
DAL80/UGA43	YKR034W
GAT1/NIL1/MEP80	YFL021W
GLN3	YER040W
GZF3/DEH1/NIL2	YJL110C

The set of 41 annotated NCR genes (ANCR) [107, 146, 147] are given in Table L.2. The four genes that were not identified as NCR-responding in any of the three genome-wide experimental and bioinformatics studies described in Bar-Joseph et al. [11], Godard et al. [107], Scherens et al. [214] are marked with an asterisk (*).

The set of 90 manually-selected genes known to be insensitive to NCR (NNCR) [146, 147] is given in Tables L.3 and L.4.

L.2 NCR related motifs

The 65 NCR related motifs used in Godard et al. [107] are given in Table L.5.

Table L.2: Annotated NCR genes (ANCR).

Standard name	Systematic name
AGP1	YCL025C
ASP3-1	YLR155C
ASP3-2	YLR157C
ASP3-3	YLR158C
ASP3-4	YLR160C
ATG14	YBR128C
BAT2	YJR148W
CAN1	YEL063C
CAR1	YPL111W
CPS1	YJL172W
DAL1	YIR027C
DAL2	YIR029W
DAL3	YIR032C
DAL4	YIR028W
DAL5	YJR152W
DAL7	YIR031C
DAL80	YKR034W
DCG1	YIR030C
DUR1,2	YBR208C
DUR3	YHL016C
ECM38	YLR299W
GAP1	YKR039W
GAT1	YFL021W
GDH2	YDL215C
GDH3	YAL062W
GLN1	YPR035W
GZF3	YJL110C
LAP4	YKL103C
MEP1	YGR121C
MEP2	YNL142W
MEP3	YPR138C
PEP4	YEL060C
PRB1	YLR142W
PUT1	YHR037W
PUT2	YOR348C
PUT4	YGR019W
UGA1	YDL210W
UGA4	YGL227W
VID30	YHR029C
YHI9	YGR125W
	YPL154C

Table L.3: Genes known to be insensitive to NCR (NNCR). Part 1 of 2.

Standard name	Systematic name
ABC1	YOR237W
ACT1	YML102W
AGP2	YPR016C
ALG6	YML099C
ALK1	YJR126C
APM1	YKL068W
ARG81	YGR253C
ATR1	YNL261W
AVT3	YNL268W
CAC2	YLR362W
CCT2	YGL030W
CDC23	YML116W
CIT3	YDR310C
CKI1	YLR188W
CMK1	YMR058W
COQ2	YIR006C
CRP1	YGR020C
DAN2	YLR055C
DOT5	YPL131W
ECM10	YEL030W
ERP5	YGR074W
FET3	YKR055W
GCD6	YLR133W
GEA1	YDR409W
GUT1	YFL039C
HEM13	YHR073W
HES1	YLL040C
HLJ1	YIL142W
HOS3	YDR005C
HRT1	YER115C
INP54	YPR051W
ISY1	YHR110W
LAC1	YOR368W
LAT1	YOR221C
LEU3	YKR106W
LOT6	YOL133W
LYP1	YOR002W
MAF1	YPR001W
MAK3	YOR187W
MCT1	YIL010W
MDL1	YGR270W
MET6	YNL071W
MKK2	YGL021W
MLH1	YBR132C
MNN4	YJR031C

Table L.4: Genes known to be insensitive to NCR (NNCR). Part 2 of 2.

Standard name	Systematic name
MSS51	YLR426W
NBP1	YGR229C
NGG1	YHR146W
NIC96	YLR457C
NUP100	YDR211W
ORC5	YGL112C
OSH3	YPL259C
PAN1	YDR176W
PBN1	YLR011W
PEP8	YLR090W
PET8	YMR284W
PUP2	YKL201C
RAD17	YFR014C
RHO4	YLR037C
RPA190	YMR167W
RPL30	YDR044W
RPL5	YHR106W
RSE1	YFR002W
SIZ1	YLR203C
SMD1	YJL053W
SMI1	YCL052C
SPR6	YGL119W
SPT10	YPL140C
SPT8	YER091C
STE11	YMR161W
SUM1	YPL128C
TAF6	YKL008C
TBF1	YHR166C
TIF6	YML049C
TOK1	YNL003C
TPS1	YER005W
TRR2	YPL116W
TUF1	YOR341W
VMA7	YKL146W
VPS13	YOR229W
VPS70	YJR050W
WTM2	YBR126C
XDJ1	YNR041C
YFL063W	YLR451W
YKU70	YHL032C
YND1	YJL093C
YTA7	YPR004C
	YJL127C
	YFL063W
	YOL065C

Table L.5: NCR related motifs used in Godard et al. [107] and Kontos et al. [142].

AAGATA	CGATAAGA
AAGATAA	CGCCG
AAGATAAG	CGCTTATC
AAGCG	CTGATA
ACCTTATC	CTGATAA
AGATA	CTGATAAG
AGATAA	CTTATC
AGATAAG	CTTATCA
AGATAAGA	CTTATCAA
AGATAAGC	CTTATCGC
AGCCG	CTTATCNn{0,60}GATAAG
AGCCTA	GATAA
ATAAG	GATAAC
ATAAGA	GATAACA
ATAAGAT	GATAACAA
ATAAGATA	GATAAGA
ATAAGC	GATAAGC
ATAAGCG	GATAAGNn{0,60}CTTATC
ATAAGG	GATAAGNn{0,60}GATAAG
ATAAGGG	GATAANn{0,60}GATAA
ATCAG	GATAANn{0,60}TTATC
ATCTTA	GCACC
ATCTTATC	GCCGC
CAGATAAG	GCGATAA
CCCGG	GGCAC
CCCTTA	TAAGA
CCTTA	TAAGATA
CCTTATC	TAAGATAA
CCTTATCA	TATCA
CGATAA	TGATAA
CGATAAG	TTATCNn{0,60}GATAA
GATAAG	GATAAH
GATTA	

Gram-Schmidt Orthogonalization

The *Gram-Schmidt orthogonalization* (GSO) procedure can be used to rank p input variables $\mathbf{x}_1, \ldots, \mathbf{x}_p$ given an output variable $\mathbf{y}$ [32]. If we let $\mathcal{V} = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ denote the set of input variables, it first selects the variable which exhibits the highest correlation with $\mathbf{y}$:

$$\mathbf{x}_{s_1} = \max_{\mathbf{x}_i \in \mathcal{V}} \rho_{(\mathbf{x}_i, \mathbf{y})} \ .$$

Next, the method selects the variable whose partial correlation with the output given the already selected variables is the highest. Let $\mathcal{S}_k$ denote the set of k selected variables after step k, $k = 1, \ldots, p-2$. At step $k+1$, the GSO procedure will thus select the following variable:

$$\mathbf{x}_{s_{k+1}} = \max_{\mathbf{x}_i \in \mathcal{V} \setminus \mathcal{S}_k} \rho_{(\mathbf{x}_i, \mathbf{y} | \mathcal{S}_k)} \ .$$

Once $p-1$ variables have been ranked, the procedure stop (since the remaining variable is of course the last in the ranking).

To use GSO as a variable selection procedure, we assess at each step the predictive power of the selected variables with a linear model through leave-one-out cross-validation. The subset having the lowest prediction error is the selected subset.

List of Tables

List of Figures

Bibliography

[1] Akutsu, T., Kuhara, S., Maruyama, O., and Miyano, S. (1998a). Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. In Karloff, H., editor, *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, volume 3, pages 695–702. Society for Industrial and Applied Mathematics.

[2] Akutsu, T., Kuhara, S., Maruyama, O., and Miyano, S. (1998b). A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions. *Genome Informatics*, 9:151–160.

[3] Akutsu, T., Miyano, S., and Kuhara, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In Altman, R. B., Lauderdale, K., Dunker, A. K., Hunter, L., and Klein, T. E., editors, *Pacific Symposium on Biocomputing*, volume 4, pages 17–28. World Scientific Publishing.

[4] Alter, O., Brown, P. O., and Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106.

[5] Anderson, T. W. (1984). *An Introduction to Multivariate Statistical Analysis*. Wiley series in probability and mathematical statistics. Wiley, second edition.

[6] Anderson, T. W. (1996). R. A. Fisher and multivariate analysis. *Statistical Science*, 11(1):20–34.

[7] Arnold, V. I. (2008). *Ordinary Differential Equations*. Springer.

[8] Baba, K., Shibataand, R., and Sibuya, M. (2004). Partial correlation and conditional correlation as measures of conditional independence. *Australian and New Zealand Journal of Statistics*, 46(4):657–664.

[9] Banerjee, O., Ghaoui, L. E., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516.

[10] Banerjee, O., Ghaoui, L. E., d'Aspremont, A., and Natsoulis, G. (2006). Convex optimization techniques for fitting sparse Gaussian graphical models. In *Proceedings of the 23rd International Conference on Machine Learning*.

[11] Bar-Joseph, Z., Gerber, G., Lee, T., Rinaldi, N., Yoo, J., Robert, F., Gordon, D., Fraenkel, E., Jaakkola, T., Young, R., et al. (2003). Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11):1337–1342.

[12] Barabási, A.-L. and Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews*, 5:101–114.

[13] Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour.* Princeton University Press.

[14] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 57(1):289–300.

[15] Bickel, P. and Levina, E. (2008). Regularized estimation of large covariance matrices. *Annals of Statistics*, 36(1):199–227.

[16] Billingsley, P. (1995). *Probability and Measure.* Wiley, third edition.

[17] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

[18] Blum, A. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271.

[19] Bontempi, G. (1999). *Local Learning Techniques for Modeling, Prediction and Control.* PhD thesis, Université Libre de Bruxelles.

[20] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152.

[21] Boulouri, H. and Davidson, E. H. (2002a). Modeling DNA sequence-based cis-regulatory gene networks. *Developmental Biology*, 246(1):2–13.

[22] Boulouri, H. and Davidson, E. H. (2002b). Modeling transcriptional regulatory networks. *BioEssays*, 24:1118–1129.

[23] Brazhnik, P., de la Fuente, A., and Mendes, P. (2002). Gene networks: how to put the function in genomics. *TRENDS in Biotechnology*, 20(11):467–472.

[24] Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., Gaasterland, T., Glenisson, P., Holstege, F. C. P., Kim, I. F., Markowitz, V., Matese, J. C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., and Vingron, M. (2001). Minimum information about a microarray experiment (MIAME)–toward standards for microarray data. *Nature Genetics*, 29(4):365–371.

[25] Breiman, L. (1996a). Bagging predictors. *Machine Learning*, 24:123–140.

[26] Breiman, L. (1996b). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6):2350–2383.

[27] Brohée, S., Faust, K., Lima-Mendez, G., Sand, O., Janky, R., Vanderstocken, G., Deville, Y., and van Helden, J. (2008). NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways. *Nucleic Acids Research*, 36.

[28] Bryant, P. (1984). Geometry, statistics, probability: variations on a common theme. *The American Statistician*, 38(1):38–48.

[29] Butte, A., Tamayo, P., Slonim, D., Golub, T., and Kohane, I. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences*, 97(22):12182–12186.

[30] Castelo, R. and Roverato, A. (2006). A robust procedure for Gaussian graphical model search from microarray data with p larger than n. *Journal of Machine Learning Research*, 7:2621–2650.

[31] Chawla, N. V. (2005). Data mining for imbalanced datasets: An overview. In Maimon, O. and Rokach, L., editors, *The Data Mining and Knowledge Discovery Handbook*, chapter 40, pages 853–867. Springer.

[32] Chen, S., Billings, S. A., and Luo, W. (1989). Orthogonal least squares methods and their application to non-linear system identification. *Proceedings of the National Academy of Sciences*, 50:1873–1896.

[33] Chen, T., He, H. L., and Church, G. M. (1999). Modeling gene expression with differential equations. In *Pacific Symposium on Biocomputing*, volume 4, pages 29–40. World Scientific Publishing.

[34] Chernick, M. R. (1999). *Bootstrap Methods: A Practitioner's Guide*. Wiley Series in Probability and Statistics. Wiley.

[35] Chickering, D. M. (1996). Learning bayesian networks is NP-complete. In Fisher, D. and Lenz, H.-J., editors, *Learning from Data: Artificial Intelligence and Statistics V*. Springer.

[36] Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1):65–73.

[37] Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. O., and Herskowitz, I. (1998). The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705.

[38] Cooper, T. G. (2002). Transmitting the signal of excess nitrogen in *Saccharomyces cerevisiae* from the Tor proteins to the GATA factors: connecting the dots. *FEMS Microbiology Reviews*, 26(3):223–238.

[39] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms.* MIT Press, second edition.

[40] Cover, T. M. and Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

[41] Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems.* Springer.

[42] Cox, D. R. and Wermuth, N. (1993). Linear dependencies represented by chain graphs. *Statistical Science*, 8(3):204–218.

[43] Cox, D. R. and Wermuth, N. (1996). *Multivariate Dependencies.* Chapman & Hall/CRC.

[44] Daniels, M. J. and Kass, R. E. (2001). Shrinkage estimators for covariance matrices. *Biometrics*, 57:1173–1184.

[45] d'Aspremont, A., Banerjee, O., and Ghaoui, L. E. (2008). First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and its Applications*, 30(1):56–66.

[46] Davidson, E. H., McClay, D. R., and Hood, L. (2003). Regulatory gene networks and the properties of the developmental process. In *Proceedings of the National Academy of Science*, volume 100, pages 1475–1480.

[47] Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240.

[48] Dawid, A. P. (1979). Conditional independence in statistical theory. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 41(1):1–31.

[49] Dawid, A. P. (1980). Conditional independence for statistical operations. *The Annals of Statistics*, 8(3):598–617.

[50] de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103.

[51] de la Fuente, A., Bing, N., Hoeschele, I., and Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18):3565–3574.

[52] Dempster, A. P. (1972). Covariance selection. *Biometrics*, 28(1):157–175.

[53] Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition.* Springer.

[54] D'haeseleer, P. (2000). *Reconstructing Gene Networks from Large Scale Gene Expression Data.* PhD thesis, The University of New Mexico.

[55] D'haeseleer, P., Liang, S., and Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726.

[56] D'haeseleer, P., Wen, X., Fuhrman, S., and Somogyi, R. (1999). Linear modeling of mRNA expression levels during CNS development and injury. In *Pacific Symposium on Biocomputing*, volume 4, pages 41–52. World Scientific Publishing.

[57] Diestel, R. (2005). *Graph Theory*, volume 173 of *Graduate Texts in Mathematics.* Springer, third edition.

[58] Dietterich, T. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

[59] Dobra, A., Hans, C., Jones, B., Nevins, J., Yao, G., and West, M. (2004). Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212.

[60] Domingos, P. and Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2):103–130.

[61] Draper, N. R. and Smith, H. (1981). *Applied Regression Analysis.* Wiley series in probability and mathematical statistics. Wiley, second edition.

[62] Drton, M. and Perlman, M. D. (2004). Model selection for Gaussian concentration graphs. *Biometrika*, 91(3):591–602.

[63] Drton, M. and Perlman, M. D. (2007). Multiple testing and error control in Gaussian graphical model selection. *Statistical Science*, 22(3):430–449.

[64] Drton, M. and Perlman, M. D. (2008). A SINful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, 138:1179–1200.

[65] Drăghici, S. (2003). *Data Analysis Tools for DNA Microarrays.* Chapman & Hall/CRC.

[66] Duda, R. O., Hart, P. E., and Stork., D. G. (2001). *Pattern Classification.* Wiley, second edition.

[67] Dudoit, S. and van der Laan, M. J. (2008). *Multiple Testing Procedures with Applications to Genomics.* Springer.

[68] Dutilh, B. E. and Hogeweg, P. (1999). Gene networks from microarray data. Technical Report Binf.1999.11.01, Bioinformatics, Utrecht University.

[69] Dykstra, R. (1970). Establishing the positive definiteness of the sample covariance matrix. *The Annals of Mathematical Statistics*, 41(6):2153–2154.

[70] Edwards, D. (2000). *Introduction to Graphical Modelling.* Springer Texts in Statistics. Springer, second edition.

[71] Efron, B. (1994). *The Jackknife, the Bootstrap and Other Resampling Plans.* CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics.

[72] Efron, B. (2003). Robbins, empirical Bayes and microarrays. *The Annals of Statistics*, 31(2):366–378.

[73] Efron, B. (2005). Local false discovery rates. Preprint available from `http://www-stat.stanford.edu/~ckirby/brad/papers/2005LocalFDR.pdf`, Department of Statistics, Stanford.

[74] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, 32(2):407–451.

[75] Efron, B., Tibshirani, R., Storey, J. D., and Tusher, V. (2001). Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96(456):1151–1160.

[76] Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap.* Chapman & Hall/CRC.

[77] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95:14863–14868.

[78] Embrechts, P., McNeil, A., and Straumann, D. (2002). Correlation and dependence in risk management: properties and pitfalls. In Dempster, M. A. H., editor, *Risk Management: Value at Risk and Beyond*, pages 176–223. Cambridge University Press.

[79] Endy, D. and Brent, R. (2001). Modelling cellular behaviour. *Nature*, 409:391–395.

[80] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

[81] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874.

[82] Feller, W. (1968). *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, third edition.

[83] Feller, W. (1971). *An Introduction to Probability Theory and Its Applications*, volume 2. Wiley, second edition.

[84] Fields, S. and Johnston, M. (2005). Whither model organism research? *Science*, 307:1885–1886.

[85] Filkov, V. (2006). Identifying gene regulatory networks from gene expression data. In Aluru, S., editor, *Handbook of Computational Molecular Biology*, Computer and Information Science Series, chapter 27. Chapman & Hall/CRC.

[86] Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron*, 1:3–32.

[87] Fisher, R. A. (1924). The distribution of the partial correlation coefficient. *Metron*, 3:329–332.

[88] Fitch, A. M. and Jones, M. B. (2009). Shortest path analysis using partial correlations for classifying gene functions from gene expression data. *Bioinformatics*, 25(1):42–47.

[89] Frank, I. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35(2).

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

[91] Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175.

[92] Friedman, J. H. (1997). On bias, variance, 0/1–loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77.

[93] Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, 303:799–805.

[94] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29:131–163.

[95] Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620.

[96] Friedman, N. and Pe'er, I. N. D. (1999). Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. In Dubios, H. and Laskey, K., editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 206–215. Morgan Kaufmann.

[97] Fu, W. J. (1998). Penalized regressions: the bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416.

[98] Gardner, T. and Faith, J. (2005). Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2(1):65–88.

[99] Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., and Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11:4241–4257.

[100] Geard, N. (2004). Modelling gene regulatory networks: Systems biology to complex systems. Technical report, Australian Centre for Complex Systems, The University of Queensland. ACCS Draft Technical Report available from `http://www.itee.uq.edu.au/~nic/_accs-grn/modelling-grns.pdf`.

[101] Geurts, P. (2005). Bias vs. variance decomposition for regression and classification. In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*. Kluwer Academic Publisher.

[102] Geurts, P., Touleimat, N., Dutreix, M., and d'Alché Buc, F. (2007). Inferring biological networks with output kernel trees. *BMC Bioinformatics (PMSB06 special issue)*, 8(Suppl. 2):S4.

[103] Giles, P. J. and Kipling, D. (2003). Normality of oligonucleotide microarray data and implications for parametric statistical analyses. *Bioinformatics*, 19(17):2254–2262.

[104] Glass, L. (1977). Global analysis of nonlinear chemical kinetics. In Berne, B., editor, *Statistical Mechanics, Part B: Time Dependent Processes*, pages 311–349. Plenum Press, New York.

[105] Godard, P. (2003). Étude du contrôle par l'azote du transcriptome de la levure *Saccharomyces cerevisiae*. Journée des doctorants, Institut de biologie et de médecine moléculaires – Université Libre de Bruxelles.

[106] Godard, P. (2006). *Analyse systématique de l'influence de la source d'azote sur le transcriptome de la levure Saccharomyces cerevisiae*. PhD thesis, Université Libre de Bruxelles, Brussels, Belgium.

[107] Godard, P., Urrestarazu, A., Vissers, S., Kontos, K., Bontempi, G., van Helden, J., and André, B. (2007). Effect of 21 different nitrogen sources on global gene expression in the yeast *Saccharomyces cerevisiae*. *Molecular and Cellular Biology*, 27(8):3065–3086.

[108] Golub, G. H. and Loan, C. F. V. (1989). *Matrix Computations*. The Johns Hopkins University Press, second edition.

[109] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

[110] Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L., editors (2006). *Feature Extraction: Foundations and Applications*. Springer.

[111] Haff, L. R. (1980). Empirical Bayes estimation of the multivariate normal covariance matrix. *Annals of Statistics*, 8(3):586–597.

[112] Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In Langley, P., editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pages 359–366, Stanford University, Standord, CA, USA.

[113] Hamilton, D. (1987). Sometimes $R^2 > r^2_{yx_1} + r^2_{yx_2}$: correlated variables are not always redundant. *The American Statistician*, 41(2):129–132.

[114] Hastie, T. and Tibshirani, R. (2004). Efficient quadratic regularization for expression arrays. *Biostatistics*, 5(3):329–340.

[115] Hastie, T., Tibshirani, R., Eisen, M. B., and Alizadeh, A. (2000). 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2).

[116] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer.

[117] Hasty, J., McMillen, D., Isaacs, F., and Collins, J. J. (2001). Computational studies of gene regulatory networks: *In numero* molecular biology. *Nature Reviews Genetics*, 2:268–279.

[118] Heckerman, D. (1995). A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research.

[119] Herr, D. G. (1980). On the history of the use of geometry in the general linear model. *The American Statistician*, 34(1):43–47.

[120] Hoerl, A. E. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67.

[121] Holloway, D. T., Kon, M., and DeLisi, C. (2006). Machine learning for regulatory analysis and transcription factor target prediction in yeast. *Systems and Synthetic Biology*, 1:25–46.

[122] Holstege, F. C., Jennings, E. G., Wyrick, J. J., Lee, T. I., Hengartner, C. J., Green, M. R., Golub, T. R., Lander, E. S., and Young, R. A. (1998). Dissecting the regulatory circuitry of a eukaryotic genome. *Cell*, 95(5):717–728.

[123] Hoste, V. (2005). *Optimization Issues in Machine Learning of Coreference Resolution.* PhD thesis, Universiteit Antwerpen.

[124] Hotelling, H. (1953). New light on the correlation coefficient and its transforms. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 15(2):193–232.

[125] Huang, S. (1999). Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery. *Journal of Molecular Medicine*, 77:469–480.

[126] Ideker, T., Galitski, T., and Hood, L. (2001). A new approach to decoding life: Systems biology. *Annual Review of Genomics and Human Genetics*, 2:343–372.

[127] Ideker, T. E., Thorsson, V., and Karp, R. M. (2000). Discovery of regulatory interactions through perturbation: Inference and experimental design. In Altman, R. B., Lauderdale, K., Dunker, A. K., Hunter, L., and Klein, T. E., editors, *Pacific Symposium on Biocomputing*, volume 5, pages 302–313, Singapore. World Scientific Publishing.

[128] Imoto, S., Goto, T., and Miyano, S. (2002). Estimation of genetic networks and functional structures between genes by using bayesian network and nonparametric regression. In *Pacific Symposium on Biocomputing*, volume 7, pages 175–186. World Scientific Publishing.

[129] Jacod, J. and Protter, P. (2000). *Probability Essentials.* Springer.

[130] James, W. and Stein, C. (1961). Estimation with quadratic loss. In LeCam, L. M. and Neyman, J., editors, *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, pages 361–379, Berkeley, California. University of California Press.

[131] Jiang, Y. L., Metz, C. E., and Nishikawa, R. M. (1996). A receiver operating characteristic partial area index for highly sensitive diagnostic tests. *Radiology*, 201:745–750.

[132] John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In Cohen, W. W. and Hirsh, H., editors, *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–129.

[133] Jordan, M. I., editor (1999). *Learning in Graphical Models.* The MIT Press, New York.

[134] Karp, R. M., Stoughton, R., and Yeung, K. Y. (1999). Algorithms for choosing differential gene expression experiments. In *3rd Annual International Conference on Computational Molecular Biology (RECOMB'99)*, pages 208–217, New York. ACM Press.

[135] Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467.

[136] Kauffman, S. A. (1971). Gene regulation networks: a theory for their global structure and behaviours. *Current Topics in Development Biology*, 6:145–182.

[137] Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, New York.

[138] Kishino, H. and Waddell, P. (2000). Correspondence analysis of genes and tissue types and finding genetic links from microarray data. *Genome Informatics*, 11:83–95.

[139] Kleywegt, A. J., Shapiro, A., and Homem-De-Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502.

[140] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324.

[141] Kontos, K. (2005). *Machine Learning Methods for Network Inference from Microarray Data.* Master's thesis, Université Libre de Bruxelles, Belgium.

[142] Kontos, K., André, B., van Helden, J., and Bontempi, G. (2009). Gaussian graphical models to infer putative genes involved in nitrogen catabolite repression in *S. cerevisiae.* In Pizzuti, C., Ritchie, M. D., and Giacobini, M., editors, *Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBIO 2009)*, volume 5483 of *Lecture Notes in Computer Science*, pages 13–24. Springer.

[143] Kontos, K. and Bontempi, G. (2008a). Nested q-partial graphs for genetic network inference from "small n, large p" microarray data. In Elloumi, M., Küng, J., Linial, M., Murphy, R., Schneider, K., and Toma, C., editors, *Proceedings of the 2nd International Conference on Bioinformatics Research and Development (BIRD 2008)*, volume 13 of *Communications in Computer and Information Science (CCIS)*, pages 273–287. Springer.

[144] Kontos, K. and Bontempi, G. (2008b). Nested q-partial graphs for genetic network inference from "small n, large p" microarray data. In *Proceedings of Journées Ouvertes Biologie Informatique Mathématiques (JOBIM 2008).*

[145] Kontos, K. and Bontempi, G. (2009). An improved shrinkage estimator to infer regulatory networks with Gaussian graphical models. In *Proceedings of the 24th Annual ACM Symposium on Applied Computing (ACM SAC 2009).*

[146] Kontos, K., Godard, P., André, B., van Helden, J., and Bontempi, G. (2007). Machine learning techniques to identify putative genes involved in nitrogen catabolite repression in the yeast *Saccharomyces cerevisiae.* In *Proceedings of the First International Workshop on Machine Learning in Systems Biology (MLSB 2007)*, pages 21–26.

[147] Kontos, K., Godard, P., André, B., van Helden, J., and Bontempi, G. (2008). Machine learning techniques to identify putative genes involved in nitrogen catabolite repression in the yeast *Saccharomyces cerevisiae. BMC Proceedings*, 2(Suppl 4):S5.

[148] Krämer, N., Schäfer, J., and Boulesteix, A.-L. (2009). Regularized estimation of large scale gene regulatory networks. Submitted.

[149] Lambert, J. D. (1991). *Numerical Methods for Ordinary Differential Equations.* Wiley.

[150] Lauritzen, S. L. (1996). *Graphical Models.* Oxford Statistical Science Series. Clarendon Press, Oxford.

[151] Ledoit, O. and Wolf, M. (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621.

[152] Ledoit, O. and Wolf, M. (2004a). Honey, I shrunk the sample covariance matrix. *Journal of Portfolio Management*, 31(1):110–119.

[153] Ledoit, O. and Wolf, M. (2004b). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411.

[154] Lehmann, E. L. (1986). *Testing statistical hypotheses.* Wiley, second edition.

[155] Lehmann, E. L. and Casella, G. (1998). *Theory of Point Estimation.* Springer, second edition.

[156] Levina, E., Rothman, A., and Zhu, J. (2008). Sparse estimation of large covariance matrices via a nested lasso penalty. *The Annals of Applied Statistics*, 2(1):245–263.

[157] Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24(9):1175–1182.

[158] Li, H. (2008). Statistical methods for inference of genetic networks and regulatory modules. In Emmert-Streib, F. and Dehmer, M., editors, *Analysis of Microarray Data: A Network-Based Approach*, chapter 6. Wiley.

[159] Li, H. and Gui, J. (2006). Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302–317.

[160] Li, J. and Wang, Z. J. (2009). Controlling the false discovery rate of the association/causality structure learned with the PC algorithm. *Journal of Machine Learning Research*, 10:475–514.

[161] Liang, K.-C. and Wang, X. (2008). Gene regulatory network reconstruction using conditional mutual information. *EURASIP Journal on Bioinformatics and Systems Biology*, Article ID 253894:14 pages.

[162] Liang, S., Fuhrman, S., and Somogyi, R. (1998). REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In Altman, R. B., Dunker, A. K., Hunter, L., and Klein, T. E., editors, *Pacific Symposium on Biocomputing*, volume 3, pages 18–29, Singapore. World Scientific Publishing.

[163] Lodish, H., Berk, A., Matsudaira, P., Kaiser, C. A., Krieger, M., Scott, M. P., Zipursky, L., and Darnell, J. (2003). *Molecular Cell Biology.* W. H. Freeman, fifth edition.

[164] Magwene, P. and Kim, J. (2004). Estimating genomic coexpression networks using first-order conditional independence. *Genome Biology*, 5:R100.

[165] Maki, Y., Tominaga, D., Okamoto, M., Watanabe, S., and Eguch, Y. (2001). Development of a system for the inference of large scale genetic networks. In Altman, R. B., Dunker, A. K., Hunter, L., Lauderdale, K., and Klein, T. E., editors, *Pacific Symposium on Biocomputing*, volume 6, pages 446–458, Singapore. World Scientific Publishing.

[166] Maloof, M. A. (2003). Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 Workshop on Learning from Imbalanced Data Sets II.*

[167] Maniatis, T. and Tasic, B. (2002). Alternative pre-mRNA splicing and proteome expansion in metazoans. *Nature*, 418:236–243.

[168] Markowetz, F. (2007). A bibliography on learning causal networks of gene interactions available from `http://genomics.princeton.edu/~florian/`.

[169] Markowetz, F. and Spang, R. (2007). Inferring cellular networks – a review. *BMC Bioinformatics*, 8(Suppl. 6):S5.

[170] Marquardt, D. W. (1970). Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation. *Technometrics*, 12(3).

[171] McAdams, H. H. and Arkin, A. (1998). Simulation of prokaryotic genetic circuits. *Annual Review of Biophysics and Biomolecular Structure*, 27:199–224.

[172] McClish, R. J. (1989). Analyzing a portion of the ROC curve. *Medical Decision Making*, 9:190–195.

[173] McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques and Tools.* Princeton University Press.

[174] Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462.

[175] Mendes, P. (1993). GEPASI: A software package for modelling the dynamics, steady states and control of biochemical and other systems. *Computer Applications in the Biosciences*, 9(5):563–571.

[176] Mendoza, L. and Alvarez-Buylla, E. R. (1998). Dynamics of the genetic regulatory network for *Arabidopsis thaliana. Journal of Theoretical Biology*, 193:307–319.

[177] Mendoza, L., Thieffry, D., and Alvarez-Buylla, E. R. (1999). Genetic control of flower morphogenesis in *Arabidopsis thaliana*: A logical analysis. *Bioinformatics*, 15(7/8):593–606.

[178] Mitchell, T. (1997). *Machine Learning.* McGraw Hill.

[179] Mjolsness, E., Sharp, D. H., and Reinitz, J. (1991). A connectionist model of development. *Journal of Theoretical Biology*, 152:429–454.

[180] Muirhead, R. J. (2005). *Aspects of Multivariate Statistical Theory.* Wiley Series in Probability and Statistics. Wiley.

[181] Nilsson, R., Pena, J. M., Björkegren, J., and Tegnér, J. (2007). Consistent feature selection for pattern recognition in polynomial time. *Journal of Machine Learning Research*, 8:589–612.

[182] Noda, K., Shinohara, A., Takeda, M., Matsumoto, S., Miyano, S., and Kuhara, S. (1998). Finding genetic network from experiments by weighted network model. *Genome Informatics*, 9:141–150.

[183] Ogawa, N., DeRisi, J., and Brown, P. O. (2000). New components of a system for phosphate accumulation and polyphosphate metabolism in *Saccharomyces cerevisiae* revealed by genomic expression analysis. *Molecular Biology of the Cell*, 11:4309–4321.

[184] Opgen-Rhein, R., Schäfer, J., and Strimmer., K. (2007). *GeneNet: Modeling and Inferring Gene Networks.* R package version 1.2.1.

[185] Orphanides, G. and Reinberg, D. (2002). A unified theory of gene expression. *Cell*, 108:439–451.

[186] Ott, S., Imoto, S., and Miyano, S. (2004). Finding optimal models for small gene networks. In *Pacific Symposium on Biocomputing*, volume 9, pages 557–567. World Scientific Publishing.

[187] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Publishers, San Fransisco, CA.

[188] Pearl, J. (2000). *Causality.* Cambridge University Press.

[189] Pearl, J. and Paz, A. (1987). Graphoids: a graph based logic for reasoning about relevancy relations. In Boulay, B. D., Hogg, D., and Steel, L., editors, *Advances in Artificial Intelligence—II*, pages 357–363, Amsterdam. North-Holland.

[190] Pe'er, D., Regev, A., Elidan, G., and Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(S1):S215–S224.

[191] Pellet, J.-P. and Elisseeff, A. (2008). Using markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9:1295–1342.

[192] Pena, J. M. (2008). Learning Gaussian graphical models of gene networks with false discovery rate control. In Marchiori, E. and Moore, J., editors, *6th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, volume 4973 of *Lecture Notes in Computer Science*, pages 165–176. Springer.

[193] Provost, F., Fawcett, T., and Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–453, San Francisco. Morgan Kaufmann.

[194] Quenouille, M. H. (1956a). Approximate tests of correlation in time-series. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 11(68-84).

[195] Quenouille, M. H. (1956b). Notes on bias in estimation. *Biometrika*, 43(353-360).

[196] R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

[197] Rao, C. R. and Toutenburg, H. (1995). *Linear Models: Least Squares and Alternatives*. Springer.

[198] Reichhardt, T. (1999). It's sink or swim as a tidal wave of data approaches. *Nature*, 399(6736):517–520.

[199] Reverter, A. and Chan, E. K. F. (2008). Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks. *Bioinformatics*, 24(21):2491–2497.

[200] Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

[201] Rosen, R. (1968). Recent developments in the theory of control and regulation of cellular processes. *International Review of Cytology*, 23:25–88.

[202] Rosenberg, A. L. and Heath, L. S. (2001). *Graph separators, with applications*. Kluwer Academic Publishers.

[203] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychology Review*, 65(6):386–408.

[204] Saerens, M., Latinne, P., and Decaestecker, C. (2002). Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural Computation*, 14:21–41.

[205] Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

[206] Sancetta, A. (2008). Sample covariance shrinkage for high dimensional dependent data. *Journal of Multivariate Analysis*, 99:949–967.

[207] Sánchez, L. and Thieffry, D. (2001). A logical analysis of the *Drosophila* gap genes. *Journal of Theoretical Biology*, 211:115–141.

[208] Sánchez, L., van Helden, J., and Thieffry, D. (1997). Establishment of the dorso-ventral pattern during embryonic development of *Drosophila melanogaster*: A logical analysis. *Journal of Theoretical Biology*, 189:377–389.

[209] Sanford, J. R. and Caceres, J. F. (2004). Pre-mRNA splicing: life at the centre of the central dogma. *Journal of Cell Science*, 117:6261–6263.

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

[211] Schäfer, J. and Strimmer, K. (2005b). Learning large-scale graphical gaussian models from genomic data. *AIP Conference Proceedings*, 776(1):263–276.

[212] Schäfer, J. and Strimmer, K. (2005c). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1):32.

[213] Schena, M. (2003). *Microarray Analysis.* Wiley.

[214] Scherens, B., Feller, A., Vierendeels, F., Messenguy, F., and Dubois, E. (2006). Identification of direct and indirect targets of the Gln3 and Gat1 activators by transcriptional profiling in response to nitrogen availability in the short and long term. *FEMS Yeast Research*, 6(5):777–791.

[215] Schmidt-Heck, W., Guthke, R., Toepfer, S., Reischer, H., Dürrchmid, K., and Bayer, K. (2004). Reverse engineering of the stress response during expression of a recombinant protein. In *European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems (EUNITE)*, pages 407–412, Aachen, Germany.

[216] Schölkopf, B., Tsuda, K., and Vert, J.-P., editors (2004). *Kernel Methods in Computational Biology.* The MIT Press.

[217] Schuchhardt, J., Beule, D., Malik, A., Wolski, E., Eickhoff, H., Lehrach, H., and Herzel, H. (2000). Normalization strategies for cDNA microarrays. *Nucleic Acids Research*, 28(10):e47i–e47v.

[218] Shmulevich, I., Dougherty, E., Kim, S., and Zhang, W. (2002). Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274.

[219] Simonis, N., Wodak, S. J., Cohen, G. N., and van Helden, J. (2004). Combining pattern discovery and discriminant analysis to predict gene co-regulation. *Bioinformatics*, 20(15):2370–2379.

[220] Smith, C. W. and Valcarel, J. (2000). Alternative pre-mrna splicing: the logic of combinatorial control. *Trends in Biochemical Science*, 25:381–388.

[221] Smolen, P., Baxter, D. A., and Byrne, J. H. (2000). Modeling transcriptional control in gene networks – methods, recent results, and future directions. *Bulletin of Mathematical Biology*, 62:247–292.

[222] Snoussi, E. H. (1989). Qualitative dynamics of piecewise-linear differential equations: A discrete mapping approach. *Dynamics and Stability of Systems*, 4(3/4):189–207.

[223] Somogyi, R. and Sniegoski, C. A. (1996). Modeling the complexity of genetic networks: understanding multigenic and pleiotropic regulation. *Complexity*, 1:45–63.

[224] Soussi-Boudekou, S., Vissers, S., Urrestarazu, A., Jauniaux, J. C., and André, B. (1997). Gzf3p, a fourth GATA factor involved in nitrogen-regulated transcription in *Saccharomyces cerevisiae*. *Molecular Microbiology*, 23(6):1157–1168.

[225] Speed, T., editor (2003). *Statistical Analysis of Gene Expression Microarray Data*. Chapman & Hall/CRC.

[226] Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Brown, M. B. E. P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle–regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297.

[227] Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*. The MIT Press, second edition.

[228] Stein, C. (1956). Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 197–206, Berkeley, California. University of California Press.

[229] Stoppiglia, H., Dreyfus, G., Dubois, R., and Oussar, Y. (2003). Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research*, 3(7-8):1399–1414.

[230] Struhl, K. (1999). Fundamentally different logic of gene regulation in eukaryotes and prokaryotes. *Cell*, 98:1–4.

[231] Stuart, A., Ord, J. K., and Arnold, S. (1999). *Kendall's Advanced Theory of Statistics: Volume 2A—Classical Inference and the Linear Model*. Hodder Arnold Publication, 6th edition.

[232] Szallasi, Z. and Liang, S. (1998). Modeling the normal and neoplastic cell cycle with 'realistic boolean genetic networks': Their application for understanding carcinogenesis and assessing therapeutic strategies. In Altman, R. B., Dunker, A. K., Hunter, L., and Klein, T. E., editors, *Pacific Symposium on Biocomputing*, volume 3, pages 66–76, Singapore. World Scientific Publishing.

[233] Tax, D. M. J. (2001). *One-class classification; Concept-learning in the absence of counter-examples*. PhD thesis, Delft University of Technology, Delft, The Netherlands.

[234] Thieffry, D. and Thomas, R. (1995). Dynamical behaviour of biological networks—II. Immunity control in bacteriophage lambda. *Bulletin of Mathematical Biology*, 57(2):277–297.

[235] Thomas, G. and O'Quigley, J. (1993). A geometric interpretation of partial correlation using spherical triangles. *The American Statistician*, 47(1):30–32.

[236] Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42:563–585.

[237] Thomas, R. (1979). *Kinetic Logic: A Boolean Approach to the Analysis of Complex Regulatory Systems*, volume 29 of *Lecture Notes in Biomathematics.* Springer.

[238] Thomas, R. (1991). Regulatory networks seen as asynchronous automata: A logical description. *Journal of Theoretical Biology*, 153:1–23.

[239] Thomas, R. and d'Ari, R. (1990). *Biological Feedback.* CRC Press, Boca Raton, FL.

[240] Thomas, R., Gathoye, A.-M., and Lambert, L. (1976). A complex control circuit: Regulation of immunity in temperate bacteriophages. *European Journal of Biochemistry*, 71:211–227.

[241] Thomas, R., Thieffry, D., and Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks—I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, 57(2):247–276.

[242] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1):267–288.

[243] Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2003). Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, 18(1):104–117.

[244] Tikhonov, A. N., Goncharsky, A., Stepanov, V. V., and Yagola, A. G. (1995). *Numerical Methods for the Solution of Ill-Posed Problems.* Springer.

[245] Toh, H. and Horimoto, K. (2002a). Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, 18(2):287–297.

[246] Toh, H. and Horimoto, K. (2002b). System for automatically inferring a genetic network from expression profiles. *Journal of Biological Physics*, 28(3):449–464.

[247] Tukey, J. W. (1958). Bias and confidence in not quite large samples. *The Annals of Mathematical Statistics*, 29(614).

[248] van der Putten, P. and van Someren, M. (2004). A bias-variance analysis of a real world learning problem: The CoIL challenge 2000. *Machine Learning*, 57:177–195.

[249] van Ham, P. (1979). How to deal with more than two levels. In Thomas, R., editor, *Kinetic Logic: A Boolean Approach to the Analysis of Complex Regulatory Systems*, volume 29 of *Lecture Notes in Biomathematics*, pages 326–343. Springer.

[250] van Helden, J. (2003). Regulatory sequence analysis tools. *Nucleic Acids Research*, 31(13):3593–3596.

[251] van Rijsbergen, C. J. (1979). *Information Retrieval.* Buttersworth, London.

[252] van Someren, E. P., Wessels, L. F. A., Backer, E., and Reinders, M. J. T. (2002). Genetic network modeling. *Pharmacogenomics*, 3(4):507–525.

[253] Vapnik, V. N. (1998). *Statistical Learning Theory.* Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. Wiley.

[254] Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory.* Statistics for Engineering and Information Science. Springer, second edition.

[255] Villers, F., Schaeffer, B., Bertin, C., and Huet, S. (2008). Assessing the validity domains of graphical Gaussian models in order to infer relationships among components of complex biological systems. *Statistical Applications in Genetics and Molecular Biology*, 7(2):14.

[256] Waddell, P. and Kishino, H. (2000). Cluster inference methods and graphical models evaluated on NCI60 microarray gene expression data. *Genome Informatics*, 11:129–140.

[257] Wald, P. W. and Kronmal, R. A. (1977). Discriminant functions when covariances are unequal and sample sizes are moderate. *Biometrics*, 33:479–484.

[258] Wasserman, L. (2004). *All of statistics.* Springer.

[259] Wasserman, L. (2006). *All of nonparametric statistics.* Springer.

[260] Weaver, D. C., Workman, C. T., and Stormo, G. D. (1999). Modeling regulatory networks with weight matrices. In *Pacific Symposium on Biocomputing*, volume 4, pages 112–123. World Scientific Publishing.

[261] Webb, A. R. (2002). *Statistical Pattern Recognition.* Wiley, second edition.

[262] Webb, G. I., Boughton, J. R., and Wang, Z. (2005). Not so naive Bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24.

[263] Weiss, G. and Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19(315-354).

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

[265] Westerhoff, H. V. and van Workum, M. (1990). Control of DNA structure and gene expression. *Biomedica Biochimica ACTA*, 49:839–853.

[266] Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*. Wiley.

[267] Wille, A. and Bühlmann, P. (2006). Low-order conditional independence graphs for inferring genetic networks. *Statistical Applications in Genetics and Molecular Biology*, 5(1):Article 1.

[268] Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., Prelić, A., von Rohr, P., Thiele, L., Zitzler, E., Gruissem, W., and Bühlmann, P. (2004). Sparse graphical Gaussian modeling of the isoprenoid gene network in *Arabidopsis thaliana*. *Genome Biology*, 5:R92.

[269] Wit, E. and McClure, J. (2004). *Statistics for Microarrays: Design, Analysis and Inference*. Wiley.

[270] Wold, S., Ruhe, A., Wold, H., and Dunn III, W. J. (1984). The collinearity problem in linear regression: The partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific Computing*, 5(3):735–743.

[271] Wolf, M. (2007). Resampling vs. shrinkage for benchmarked managers. *WILMOTT magazine*, pages 76–81.

[272] Wong, F., Carter, C. K., and Kohn, R. (2003). Efficient estimation of covariance selection models. *Biometrika*, 90(4):809–830.

[273] Wu, X., Ye, Y., and Subramanian, K. R. (2003). Interactive analysis of gene interactions using graphical Gaussian model. *BIOKDD03: 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics*, pages 63–69.

[274] Yu, H., Luscombei, N. M., Qian, J., and Gerstein, M. (2003). Genomic analysis of gene expression relationships in transcriptional regulatory networks. *TRENDS in Genetics*, 19(8):422–427.

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

[276] Zhao, W., Serpedin, E., and Dougherty, E. R. (2008). Inferring connectivity of genetic regulatory networks using information-theoretic criteria. *IEEE/ACM Transactions On Computational Biology and Bioinformatics*, 5(2):262–274.

[277] Zheng, Z. and Webb, G. I. (2000). Lazy learning of bayesian rules. *Machine Learning*, 41(53-87).

[278] Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

[279] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Index