Meta-analysis of Incomplete Microarray Studies

THÈSE N^{O} 6371 (2014)

PRÉSENTÉE LE 17 OCTOBRE 2014

À LA FACULTÉ DES SCIENCES DE BASE

CHAIRE DE STATISTIQUE

PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Alix LEBOUCQ

acceptée sur proposition du jury:

Prof. T. Mountford, président du jury
Prof. A. C. Davison, Dr D. Goldstein, directeurs de thèse
Dr M. Delorenzi, rapporteur
Prof. S. Morgenthaler, rapporteur
Dr L. Wernisch, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2014

Acknowledgements

There are many people I would like to thank, who either helped me directly or indirectly in producing this thesis over these four years. Of course, my first thanks go to my advisors, who both helped and supported me at different times of this thesis work and in different ways. I feel very lucky that I had two thesis advisors. This is a huge opportunity to learn from two different people whom I admire. So I would like to thank Darlene for her patience, understanding and endless motivation. She is an incredibly rich person to learn from and to work with. I also learned a lot during the exercise sessions for her classes. She gave me much responsibility and I felt trusted, which is very empowering. I thank Anthony for his amazing availability (=always!), his endless patience, great advice, presence, motivation and his communicative optimism. I still cannot understand how he could manage to always be available whenever I had a question to ask or needed to see him. He really does take the time to fully understand the problem and think about a solution, and always comes with some ideas. I certainly learned a lot from these two people and cannot thank them enough.

I thank the jury for reading my thesis and accepting to be members of my committee: Dr. Mauro Delorenzi, Dr. Lorenz Wernisch, Prof. Stephan Morgenthaler and Prof. Thomas Mountford.

During this work, I met several people who helped me a lot in making progress in my research, either through discussions or by providing useful material. I would therefore like to thank Asa for much advice and for the R packages grema, nclust and nclust2, useful for producing the nice heatmaps in this thesis; Viola Heinzelman and Francis Jacob for their expertise on the list of differentially expressed genes, propositions of further interesting analyses on ovarian cancer data, and for the great and interesting opportunity to present my results on real data in front of biologists in Basel, leading to very fruitful discussions; Elisabeth Boggis for helpful discussions and comments about the implementation of the normal gamma prior; the people from BCF and more particularly, Mauro Delorenzi and Eduardo Massiglia for their help on the gene set enrichment analysis; Terry Speed, who I am very glad to have met in person, who very kindly proof-read the article corresponding to Chapters 3 to 5 of this thesis, and who I thank for helpful suggestions and interesting discussions about this work; Irina Irincheeva for interesting discussions about the computation of the latent variables part in the Gibbs sampler, and helpful suggestions to lower the computational complexity of the program.

Teaching was a big part of my work too. Although I did most of my teaching with Darlene, I would also like to thank Sahar Hosseinian, with whom I had the opportunity to work in my

last year. It felt great to have so much responsibility and to be able to discuss the content of the course. She is also a very friendly person and it was a pleasure to work with her.

A thesis would be almost impossible to complete if done alone. I am very lucky to have met so many wonderful people at EPFL, who made every working days so pleasant. Therefore I thank my coworkers, PhD students and postdocs of the chair of statistics that I met at some point during my thesis: Linda, Claudio (special thank for the code leading to Figure 7.8), Raphaël, Emeric, Yousra, Jenny, Thomas, David, Sebastian, Jacques, Juliette, Miguel and Vanda, but also all the coworkers from the SMAT and STAP chairs Yoav, Mikael, Marie-Hélène, Andrea, David, Shahin, Kjell, Susan, Valentina, Daria, Laurence, Nico and Maya. I also want to thank coworkers from CUSO, from all Swiss universities with whom I had a lot of fun and fruitful discussions during the PhD days or summer and winter schools. As the PhD student representative, I also thank Sylvain and Valérie, and the members of the CUSO committee for taking my opinion into account and for giving me the opportunity to organize events for CUSO PhD students. I also was involved in the SMA as an assistant representative, and I especially thank the members of "Le conseil des Sages": Claudio, Rosalie, Varvara and Shahin, with whom I organized many fun events for PhD students and had great lunch meetings, pizza talks, cookie breaks and BBQs! All of this would not have been possible without the support of Kathryn Hess, who encourages us to organize activities, and is a person I admire, she also is an amazing doctoral program director.

I am very grateful to Nadia Kaiser, Anna Dietler and Valerie Kormann who do an awesome job regarding administration, and were always available when I had questions regarding PhD, conferences, or organisations of events.

Four years is a very long road toward the PhD, and I don't think I would have been able to survive it without my friends and family with whom I could relax and talk about anything but stats! So a huge thanks to my friends, who were there from the first year: Muriel, Greg and Aline, Aurélie and Michele, Laura and Andrei, François and Claire, Anaïs, Gwenol, Karine and Sébastien, Nath, Linda, and my friends from a (much) longer time ago, Lou and Nico, Isa and Stéphane, Sophie and Francesco, or from very far away: Aurélie, Aina, Henriikka and Celia.

Sport is also a big part of my life, and was even more important to unwind after a long day at the office, I thank Céline and all my friends from the Académie de danse classique de Pully, who made Monday evenings incredibly fun, Carlos for his funny salsa classes, and my discofit teachers at the centre sportif.

Of course a thesis could not be completed without the love and support of my entire family, and more particularly my sister and brother, Isaure and Tristan, and my parents, who always supported me and encouraged me during my studies. And also from my family in law, Anouk, Chantal and Patrik, who welcomed me into their family with open arms .

And finally a very special thanks to the love of my life, Vincent, for his love, patience, understanding, support, encouragement and so much more.

Lausanne, 25th July 2014 A. L.

Abstract

Meta-analysis of microarray studies to produce an overall gene list is relatively straightforward when complete data are available. When some studies lack information, providing only a ranked list of genes, for example, it is common to reduce all studies to ranked lists prior to combining them. Since this entails a loss of information, we consider a hierarchical Bayes approach to meta-analysis using different types of information from different studies: the full data matrix, summary statistics or ranks. The model uses an informative prior for the parameter of interest to aid the detection of differentially expressed genes. Simulations show that the new approach can give substantial power gains compared to classical meta analysis and list aggregation methods. A meta-analysis of 11 published ovarian cancer studies with different data types identifies genes known to be involved in ovarian cancer, shows significant enrichment, while controlling the number of false positives.

Independence of genes is a common assumption in microarray data analysis, and in the previous model, although it is not true in practice. Indeed, genes are activated in groups called modules: sets of co-regulated genes. These modules are usually defined by biologists, based on the position of the genes on the chromosome or known biological pathways (KEGG, GO for example). Our goal in the second part of this work is to be able to define modules common to several studies, in an automatic way. We use an empirical Bayes approach to estimate a sparse correlation matrix common to all studies, and identify modules by clustering. Simulations show that our approach performs as well or better than existing methods in terms of detection of modules across several datasets. We also develop a method based on extreme value theory to detect scattered genes, which do not belong to any module. This automatic module detection is very fast and produces accurate modules in our simulation studies. Application to real data results in a huge dimension reduction, which allows us to fit the hierarchical Bayesian model to modules, without the computational burden. Differentially expressed modules identified by this analysis present significant enrichment, indicating promising results of the method for future applications.

Key-words: clustering, empirical Bayes estimation, hierarchical Bayesian model, high-dimensional data, large covariance matrix estimation, MCMC, meta-analysis, microarray gene expression data, modules.

Résumé

Les méta-analyses de puces à ADN utilisées dans le but d'identifier une liste globale de gènes sont relativement faciles à conduire lorsque les données complètes sont disponibles. Cependant, de nombreuses études omettent de mettre leurs jeux de données à disposition, ne publiant que la liste résultante de l'analyse. Il est donc courant d'obtenir des listes ordonnées de gènes pour chaque étude qui sont ensuite combinées. Cette étape provoque inévitablement une perte d'information. Afin d'y remédier, nous avons développé un modèle hiérarchique Bayesien pour la méta-analyse utilisant différents types d'information provenant de diverses études. Ainsi nous considérons que des matrices d'expression de gènes, des statistiques ou des rangs des différentes études considérées peuvent être collectés. Le modèle utilise une densité à priori pour le paramètre d'intérêt qui facilite la détection des gènes exprimés différemment. En comparaison aux méthodes classiques de méta-analyse et de combinaison de rangs, les simulations montrent que cette nouvelle approche aboutit à une augmentation de la puissance statistique. Une méta-analyse de 11 études publiées sur le cancer ovarien mettant à disposition les différents types de données considérés dans ce travail a permis d'identifier des gènes connus pour être actifs dans le cancer ovarien tout en contôlant le nombre de faux positifs. De plus, les gènes détectés présentent un enrichissement biologique significatif.

Alors que l'indépendance des gènes est souvent supposée dans les analyses de données de puces à ADN, ainsi que dans le modèle précédent, l'hypothèse n'est pas vérifiée en pratique. En effet, les gènes tendent à être activés par groupes de gènes co-régulés, que l'on nomme modules. Ceux-ci sont généralement définis en collaboration avec des biologistes. Ils peuvent être basés sur la position des gènes sur le chromosome ou d'après des voies biologiques (KEGG, GO par exemple). Dans la seconde partie de ce travail, nous nous appliquons à définir de manière automatique des modules communs à plusieurs études. Pour cela, nous utilisons une approche Bayesienne empirique afin d'estimer une matrice de corrélation éparse commune à un ensemble d'études. Les modules sont ensuite identifiés par clustering. En termes de détection de modules communs à plusieurs jeux de données, nous montrons par des simulations que notre approche obtient des performances similaires ou supérieures aux méthodes existantes. En outre, une méthode basée sur la théorie des valeurs extrêmes permet d'identifier les gènes épars, qui n'appartiennent à aucun module.
Notre étude de simulations a montré que cette détection automatique est particulièrement rapide, simple et efficace. En effet, les modules définis sont précis et justes. Appliquée à des données réelles, il en résulte une réduction de la dimension permettant d'ajuster plus facile-

ment et rapidement le modèle développé dans la première partie de ce travail, en utilisant cette fois les modules au lieu des gènes. Cette méthode nous a permis d'identifier des modules exprimés différemment entre les individus atteints de cancer et les individus sains qui présentent un enrichissement biologique significatif. Les résultats obtenus suggèrent donc qu'elle est prometteuse pour d'éventuelles applications à d'autres types d'études.

Mots-clés : clustering, estimation Bayesienne empirique, estimation de matrices de covariance en grandes dimensions, expression de gènes, MCMC, méta-analyse, modèles hiérarchiques Bayesiens, modules, puces à ADN.

Contents

List of Figures

List of Tables

1 Preliminaries

1.1 Introduction

Understanding the role and identifying the genes involved in the mechanisms of a given biological process is a central topic in biomedical research. Identifying the genes related to serious diseases such as cancer allows better understanding and helps in finding suitable treatments or better methods for early detection. By providing a way to look at all genes of the genome simultaneously, microarrays are particularly appropriate for this kind of study. They also allow the comparison of different tissues, thus enabling the identification of genes specific to a given tissue. Microarray data are not new: according to Chon and Lancaster (2011), the first time a collection of distinct DNA in arrays was used for expression profiling was in Kulesh *et al.* (1987), whereas the first complete eukaryotic genome on a microarray was published in 1997 (Lashkari *et al.*, 1997). More and more studies use microarray data in order to detect genes differentially expressed between several groups of tissues, but such data are highly variable and the results obtained are often non-reproducible or not robust to even the mildest perturbations (Ramasamy *et al.*, 2008). This large variability may be due to improper analysis or inadequate control of false discoveries, and is exacerbated by a small sample size compared to the number of genes, which is usually in the tens of thousands. It is therefore valuable to combine the results of different microarray studies that address the same question.

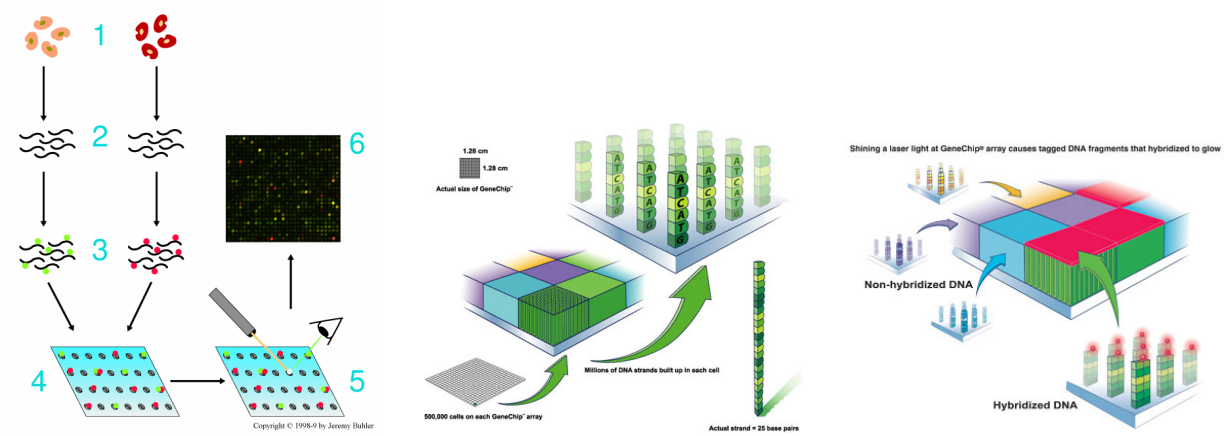
Combination of study results has been applied in a great variety of fields and especially in biomedicine, where sample sizes are usually quite small. Such methods are generally called meta-analysis, and are defined as statistical analyses that combine similar study results in an automatic way (Sutton *et al.*, 2000). Meta-analysis is different from literature review, where one simply extracts and compares results from different studies through discussion, or pooled analysis, where one treats the data as if they came from the same study. Meta-analysis combines study results in a quantitative way, increasing power and reinforcing the results. Since the first meta-analysis of microarray data (Rhodes *et al.*, 2002) was performed, many methods have been developed for this purpose, as we shall see in Section 2.1. Ramasamy *et al.* (2008) provide guidelines to conduct a meta-analysis on microarray data, while Stevens and

Doerge (2005) give indications for Affymetrix data. Meta-analysis requires that studies address the same question, so study selection is one of the most important steps in a meta-analysis. It is made easier for microarray data through the use of public repositories, as we will explain in Section 1.4. Data, especially raw data, are not always easy to collect, which is a major difficulty in meta-analyses. When study results are not of the same type, a common way of integrating information consists in transforming the data to the least informative summary, usually ranks, which are then combined using list aggregation methods, presented in Section 2.2. This reduction results in a clear loss of power, as most of the available information is not used. Being able to combine heterogenous data types, while using all the information available and maintaining high power, is one of the main motivations for the work in this thesis, as we will further explain in Section 1.5.

Microarray data are often used in cancer studies, where the aim is to find genes differentially expressed between cancer and normal tissues, or genetic profiles of patients more responsive to some treatment. In this thesis, we are interested in ovarian cancer, which is common and has a high mortality rate. Some biological background about microarray data and ovarian cancer is presented in Sections 1.2 and 1.3.

1.2 Microarray data

A microarray is an array of thousands of spots on which specific portions of genes, called probes, are deposited. A solution of DNA from a particular biological sample, called the target, is prepared and labeled, and then the array is washed with it. The idea is that the target solution contains sequences of DNA which are complementary to those deposited on the array and will therefore hybridize, or match, with the complementary sequences (Draghici, 2003). It is then possible to measure the amount of target DNA hybridized on the array. The higher the amount of hybridized DNA, the higher the gene expression in the sample. Depending on the technology used, the protocol can change slightly. Here, we describe two of the most common arrays, cDNA arrays and oligonucleotide arrays. The first, also called spotted DNA array or two-color arrays (Figure 1.1a), takes mRNA from two different biological samples and reverse transcribes them to cDNA, which is then labeled using dyes of two different colors, one for each sample. The array is a glass slide, which is composed of hundreds of thousands of spots, each containing a different DNA sequence. The array is then washed with the labeled cDNA from the two samples. If a cDNA probe is complementary to the DNA, they will hybridize. With the help of a scanner, fluorescence is quantified, indicating the amount of gene expression from each sample at each spot. Two samples are hybridized on the same array using different dyes (red and green). Each entry of the data matrix is the $\log_2$ ratio of the two intensities, allowing comparison of the quantity of expression of the gene in each sample. In order for these ratios to be comparable between arrays, they need to be normalized, generally through scaling and median centering, or loess normalization (Guerra and Goldstein, 2009). The normalized log ratios are then ready for analysis.



(a) cDNA array (b) Oligonucleotide array (c) Oligonucleotide array

Figure 1.1 – cDNA and oligonucleotide arrays: (a): protocol for cDNA array[1]. 1: choice of cell population, 2: mRNA extraction and reverse transcription, 3: fluorescent labeling of cDNA, 4: hybridization to a DNA microarray, 5: scanning the hybridized array, 6: scanned image. (b): oligonucleotide array[2]: actual size of GeneChip is 1.28×1.28 cm, 500 000 cells on each array, millions of DNA strands built up in each cell, actual strand is 25 base pairs.(c): oligonucleotide arrays, non-hybridized and hybridized DN Shining a laser light at gene chip arrays causes tagged DNA fragments that hybridized to glow.

Oligonucleotide arrays (Figures 1.1b-1.1c) use mRNA, which is reverse transcribed to cDNA and transformed to labeled cRNA. The most commonly used oligonucleotide array is the Affymetrix GeneChip, which holds million of spots, each composed of oligonucleotides of 25 bases called probes. Each gene is represented by a set of 11 to 20 pairs of probes on the array. The first type of each pair is called a perfect match (PM), and is taken from a gene sequence. The second type of the pair is called a mismatch (MM), and is similar to the PM but with the middle base changed (Parmigiani *et al.*, 2003). The cRNA contained in the sample solution hybridizes with the probes on the array, and scanning the chips gives a measure of the intensity of each probe, which is recorded in a CEL file. The role of MM is to measure non-specific hybridizations and background noise, so the intensity of interest is only that from the PM. Contrary to cDNA arrays, only one sample is hybridized on each chip. There exist many methods for normalizing Affymetrix data, but the most common, and the one we will use in this thesis, is robust multi-chip analysis (Irizarry *et al.*, 2003), which we call RMA. It uses only the perfect matches and ignores the mismatches, corrects the perfect match background (BG) on the raw scale and performs quantile normalization of $\log_2(\text{PM}^\star) = \log_2(\text{PM} - \text{BG})$. Then it assumes an additive model

$$\log_2(\text{PM}^\star_{ij}) = \alpha_i + \beta_j + e_{ij},$$

where α_i is the effect of chip i, β_j is the effect of probe j and e_{ij} are the independent and identically distributed errors with zero mean (Irizarry *et al.*, 2003). The parameters α and β are estimated robustly, using median polish or robust linear modeling.

1.2.1 Enrichment analysis

The most studied problem in the analysis of microarray gene expression data is to identify genes that are differentially expressed between several conditions. The differential expression of each gene is usually quantified by some statistic, for which significance is declared if the corresponding p-value, corrected for multiple testing, is below a pre-chosen threshold. This procedure is fairly automatic and does not require any biological knowledge. However the results obtained, i.e., a list of differentially expressed genes, should have biological meaning. The analysis performed to assess the biological meaning of a candidate list of genes is called gene set enrichment analysis, and is reviewed in Hung *et al.* (2012), Irizarry *et al.* (2009) and Goeman and Bühlmann (2007), among others. The authors describe two sets of methods for including biological knowledge into data analyses. In both, gene sets are defined prior to the analysis, based on biological processes. The Gene Ontology (The Gene Ontology Consortium, 2008) is one such database, and groups genes according to their biological role under three ontologies: biological process, molecular function and cellular component. KEGG pathways (Kanehisa and Goto, 2000, Kyoto Encyclopedia for Genes and Genomes) is another database, and consists of sixteen different parts categorized into system information, perturbed systems information, genomic information, and chemical information. The goal of KEGG is to link genes to higher level functions. Information from several such databases are accessible through the MSig repositories (Liberzon *et al.*, 2011).

The first enrichment method presented by Goeman and Bühlmann (2007) tests significance of over representation of biologically meaningful genes in the candidate list compared to the pre-defined list. One starts by creating a 2×2 table, where each cell counts the number of genes present in both lists, only one list, or none, and tests the null hypothesis of no enrichment using a hypergeometric distribution. This method has been used by Falcon and Gentleman (2007), through the GOstats Bioconductor R package, and has been made very accessible and easy to use by the development of DAVID (Huang *et al.*, 2008), which is an online bioinformatics tool to systematically extract biological information from large gene lists. The candidate gene list submitted requires the choice of a cutoff to separate differentially expressed genes from the rest of the list. If we denote by G the gene universe (the entire set of genes), S_{GO} a set of genes from, say, GO ontologies, and L a list of interesting genes, then we can obtain the distribution of the observed genes as presented in Table 1.1. Significance of the gene set can be assessed by a χ^2 test, a hypergeometric test, or Fisher's exact test. This method has the advantage of being simple and does not require a particular score for each gene: an ordered list of gene names is sufficient. However, it does require a fixed list, meaning that one needs to define a threshold above which genes are declared differentially expressed.

The second method, which Irizarry *et al.* (2009) called the aggregate score approach, assigns a score to each gene set based on the scores of the genes belonging to that set. It assumes that each gene is attributed a score indicating its differential expression, like a z-score or a

[1] http://www.cs.wustl.edu/~jbuhler/research/array/
[2] http://images.yourdictionary.com/micro-array

Table 1.1 – Distribution of the genes under the hypergeometric setup. G is the gene universe, L is the list of interesting genes and S_{GO} is the set of genes from a GO ontology.

	in L	not in L	
in S_{GO}	n_{11}	n_{12}	$\lvert S_{\text{GO}}\rvert$
not in S_{GO}	n_{21}	n_{22}	$\lvert G\rvert - \lvert S_{\text{GO}}\rvert$
	$\lvert L\rvert$	$\lvert G\rvert - \lvert L\rvert$	$\lvert G\rvert$

t-statistic, based on two sample t-test for instance. One of the most common methods in this category is gene set enrichment analysis (Mootha *et al.*, 2003; Subramanian *et al.*, 2005, GSEA), which focuses on groups of genes that share biological functions like chromosomal location or regulation. GSEA is based on the Kolmogorov–Smirnov test and can be summarized as follows (Hung *et al.*, 2012):

1. rank all genes and select a set of contiguous genes, which is the candidate list;

2. calculate the enrichment score for each gene, which is defined as a weighted Kolmogorov–Smirnov statistic. The score is increased if the gene is in the pre-defined set and decreased otherwise, by a magnitude proportional to the correlation of the gene with the phenotype;

3. estimate the significance level of the enrichment score by permutation of the sample labels, which keeps the gene correlation structure;

4. repeat the procedure for another candidate gene list;

5. correct for multiple testing (Benjamini and Hochberg, 1995).

Due to the permutations required to estimate the null distribution, i.e., to assess significance of the enrichment score, GSEA is computationally intensive. Irizarry *et al.* (2009) circumvent this by using an aggregate score which has a known distribution under the null hypothesis (either a normal or a chi-squared, depending on the version), so p-values are easy to obtain. Their method assumes that each gene is attributed a score which is normally distributed under the null hypothesis of no differential expression.

In Chapter 5, when analysing enrichment of the gene list obtained from our model, we will use the first type of approach, as it would be too computationally intensive to apply GSEA and we don't know the distribution of the scores attributed to each gene in our list. We therefore compare the enrichment of our list of genes with gene sets defined from the Gene Ontology (The Gene Ontology Consortium, 2008) or from KEGG (Kanehisa and Goto, 2000).

Table 1.2 – Surgical staging of ovarian carcinomas. Stages presented here are FIGO (Federation of International Gynecology and Obstetrics) stages (Merck, 2014).

Staging	
Stage I	Tumor limited to the ovaries
IA	Tumor limited to one ovary; no tumor on the external surface and capsule intact.
IB	Tumor limited to both ovaries; no tumor on the external surface and capsules intact.
IC	Stage IA or IB but with tumor on the surface of one or both ovaries, with capsule ruptured, or with malignant cells in ascites or in peritoneal washings.
Stage II	Tumor involving one or both ovaries with pelvic extension or metastases.
IIA	Extension and/or metastases to the uterus, fallopian tube, or both.
IIB	Extension to other pelvic tissues.
IIC	Stage IIA or IIB but with malignant cells in ascites or in peritoneal washings
Stage III	Tumor involving one or both ovaries with histologically confirmed peritoneal matastases outside the pelvis
IIIA	Microscopic peritoneal metastases outside the pelvis and negative lymph nodes.
IIIB	Macroscopic peritoneal metastases outside the pelvis that are ≤2 cm in diameter and negative lymph nodes.
IIIC	Abdominal peritoneal metastases that extend beyond the pelvis and are > 2 cm in diameter and/or regional lymph node metastases.
Stage IV	Distant metastases to the liver or outside the peritoneal cavity.

1.3 Ovarian Cancer

According to Merck (2014), ovarian cancer is the fifth most frequent cause of death from cancer in women and the second most frequent from gynecological cancers. In 2013, about 22 200 new cases were discovered and 14 000 women died from it (Merck, 2014). Ovarian cancer is called a “silent killer” as it generates no, or only non-specific symptoms, and therefore is hard to detect at an early stage. The 5-year survival rate is as low as 10–20% when the cancer is detected at an advanced stage, whereas it reaches 70–100% for early stage detection.

The most common type of ovarian cancer is epithelial ovarian cancer, which arises on the surface (or epithelium) of the ovary. This type represents about 80% of all ovarian cancers, other types arising in germ cells or in sex cord and stromal cells. However, the epithelial type can itself be divided into five subtypes based on biological behavior, response to treatment and overall prognosis (Jacob *et al.*, 2009). Staging can also be useful to distinguish cancers within a given subtype. The different characteristics for each stage are described in Table 1.2. Even with this large heterogeneity among cancer types and subtypes, all patients are treated the same way. First a hysterectomy and a bilateral salpingo-oophorectomy is applied and then a post-operative chemotherapy composed of carboplatin and paclitaxel follows (Merck, 2014).

As seen above, ovarian cancer is very heterogenous, and due to its high mortality rate it has been extensively studied. A large number of studies concerning ovarian cancer have been published, many of which use high-throughput technologies such as genomics and proteomics.

According to Jacob *et al.* (2009), 237 studies using various -omic technologies were published between 1999 and 2007. Indeed, those techniques allow analysis of the expression of thousands of genes in only one experiment and thus are really useful for analyzing the specificities of ovarian cancer. These publications identified genes or molecular profiles specific to ovarian cancers. However, these results are often highly variable and may not be directly comparable. It is even possible that the final lists of differentially expressed genes provided by different studies do not overlap at all, which raises the problem of the reproducibility and reliability of such lists. All these criteria are a great motivation to study this cancer more deeply, by integrating the results of studies about ovarian cancer in a fairly automatic way.

1.4 Data collection

In order to integrate results from different sources, it is necessary to select studies looking at the question of interest. The more information available, the more reproducible the results, which makes access to raw data particularly useful. Genomic data are more and more easily accessible with the development of public databases, which bring together studies (often with a link to the article), raw data, and sometimes information about the patients involved in the study. The Gene Expression Omnibus (GEO), for instance, created in 2000, aims to cover the broadest spectrum of high-throughput experimental methods (Edgar *et al.*, 2002). Each dataset is accessible via a unique number and is classified under three categories: the platform (list of probes indicating which genes are present in the experiment), the series (a set of samples which form a dataset), and samples (measures of gene expression). In 2013, GEO hosted over 32 000 public series, comprising more than 800 000 samples from 1600 different organisms, with a constantly increasing submission rate (Barrett *et al.*, 2013). Another public repository that we will use for collecting data is Array Express (Brazma *et al.*, 2003), which is a database for microarray gene expression data that uses the Minimum Information About a Microarray Experiment (Brazma *et al.*, 2001, MIAME). The MIAME criteria were introduced in 2001 by the Microarray Gene Expression Database Group (MGED), and consists in describing the information necessary to ensure interpretability of the experimental results obtained using microarrays, and to allow independent verification. It consists of six points (MGED, 2010):

1. raw data for each hybridization;
2. final processed data for the set of hybridizations in the experiment;
3. essential sample annotation including experimental factors and their values;
4. experimental design including sample data relationships;
5. sufficient annotation of the array; and
6. essential laboratory and data processing protocols.

The MIAME criteria aids integration of datasets and ensures reproducibility.

There also exist other public repositories or more specific data collections, such as the Stanford Microarray Database (Stanford, 2010) or Oncomine (Rhodes *et al.*, 2004). Sometimes, datasets are provided directly by the authors via their webpage, so articles need also to be looked at when recording datasets. One can look in Pubmed (NCBI, 2010), Cochrane (Cochrane, 2010) or Medline for example, in order to find publications relevant to the question of interest.

When looking for datasets to be included in an analysis, it is first important that they address the same scientific question. Then data need to be extracted from the studies, and can be of different types: image files, already preprocessed matrices of gene expression values, or final lists of summary statistics or ranks. Of course, access to raw data allows one to preprocess and analyze all the datasets in a similar way, thus reducing yet another source of variability. It also gives access to the largest amount of information one can obtain from the data. However, publication of raw data is far from systematic. Larsson and Sandberg (2006) noted that raw data were available for only 34% of the samples on GEO, many of which did not meet the necessary quality standards. Even if since 2006 the proportion of available raw data is very likely to have increased, it is far from always being the case that raw data are available, implying a huge loss of information. Jacob *et al.* (2009) point out that authors frequently publish only lists of significant genes, which are of limited utility for meta-analysis.

After data collection, data annotation must be harmonized across studies. Patients can sometimes appear several times under different IDs, or variables with the same name may describe different quantities and vice versa. Some studies give little or no information about inclusion-exclusion criteria concerning the patients or the tissues involved. As data are built on different platforms, probe-to-gene mapping may also be problematic, as it should be consistent across studies. To get rid of these problems, the strategy usually consists in considering each study separately and harmonizing between studies by hand as much as possible, or setting some possibly arbitrary assumptions.

1.5 Purpose of the thesis

The first motivation of this thesis was to study ovarian cancer; more precisely, to detect genes differentially expressed between serous ovarian cancer and normal samples. As outlined in Section 1.3, this cancer has a high mortality rate and it therefore seems important to better understand the genes involved. Even with many published studies analyzing ovarian cancer data, we saw in Section 1.4 that it was not always easy to find datasets that provide the raw matrix of gene expression values. Studies publish their results in a form which may vary from the ordered list of top k genes to the values of some statistic for each gene. When first looking for datasets comparing serous ovarian cancer and normal control samples, we could not find many, due to the fact that we were only interested in one particular subtype, and only a small proportion provided raw data. The rest had either lists of summary statistics, or published ordered lists of gene names. Integrating heterogeneous data types is not straightforward, as current methods requires homogeneous data types. Meta-analysis (Section 2.1) combines

similar study results in an automatic way and therefore discards results that are not in the appropriate format. List aggregation reduces all data to the least informative support (usually ranks) before combination (Section 2.2). However either discarding or transforming to the least informative support may lead to a huge loss of information, and we want to integrate all study results, exploiting the information they provide. Jacob *et al.* (2009) highlight the fact that meta-analysis methods require access to raw data, and that extension to combination of heterogeneous data would make fuller use of the available data.

Our goal in this work is to combine four heterogeneous types of information: matrices of gene expression values, full or partial lists of z-scores, and partial lists of ranks. We develop a hierarchical Bayesian model, presented in Chapter 3, where each type of data is modeled separately and then combined through a single parameter γ. A prior on this parameter is then carefully chosen to aid the detection of differentially expressed genes. Model efficiency is assessed using simulated microarray data in Chapter 4. A real data illustration is presented in Chapter 5, where our model is applied to serous ovarian cancer, in order to detect differentially expressed genes between cancer and normal samples.

Even if our model seems to perform well on both simulated and real data, it assumes independence of the genes, which is not true in practice, and it cannot handle the entire gene set, owing to computational restrictions. To overcome these problems, we consider modules, i.e., groups of genes that are correlated and usually have similar functions. We develop a method for automatic module detection which is based on the estimation of a common sparse correlation matrix followed by clustering. Using modules instead of genes considerably decreases the dimension of the problem, thus increasing power, and as modules are independent, they are better suited for our model. Estimation of the common correlation matrix will be discussed in Chapter 6, while module selection and application of the model to the detection of differentially expressed modules are presented in Chapter 7. Conclusions are given in Chapter 8, and calculations, additional results and lists of notations and parameters used in the models and the simulations are in Appendix A.

2 Data combination

2.1 Meta-analysis

High dimensional data are more and more common, especially in biomedicine. However, even if we are able to measure more and more variables, sample sizes do not usually follow the same growth, and are much smaller than the number of features, typically due to costs, logistic or ethical issues. It is thus common that datasets have a number of variables in the order of the tens of thousands, while the number of samples remains in the hundreds. In such setting, a simple analysis is not likely to have high power, as it will certainly be highly variable and not reproducible. When several studies of the same question are available, it may therefore be interesting to look at their results and identify the similarities. Combination of study results increases power and provides more reliable and reproducible results. Such statistical analysis, called meta-analysis, combines the research findings of the studies included, in a systematic way (Sutton *et al.*, 2000). Meta-analysis is very easy to use due to its systematic nature, and it can handle a large number of studies, which is why it has been so much used. It allows the discovery of effects that may have been missed in single study analysis, while preventing over-interpretation of the differences across studies. Selection of studies to include in a meta-analysis remains an open question. Indeed the selection can be intentional, if one selects only studies with a particular design for example, or due to publication bias, where studies having positive results are more likely to be published, and subsequently selected. In both cases selection can bias the results.

Microarray data are one example of high dimensional data, where meta-analysis seems to be especially appropriate to reduce the variability of the results. Although it seems quite simple, pooling the raw data and treating them as a single study is often a bad idea because of Simpson's paradox, which can lead to false or contradictory results. Combining final decisions, as whether a gene is differentially expressed or not, by taking the intersection of Venn diagrams, for example, might lead to a very small or even an empty set of genes.

In this section, we first review some of the common methods of meta-analysis, mostly following Sutton *et al.* (2000), and we then focus on meta-analysis for microarray data, for

which a good reference is Guerra and Goldstein (2009). Other references for meta-analysis are Kulinskaya *et al.* (2008) and Whitehead (1997).

2.1.1 Methods for meta-analysis

Before combining study results, it is important to define measures of outcome to be calculated for each of the studies separately. They depend on the type of data at hand and on the type of question one wishes to answer. They can be odds, summary statistics, p-values or ordinal outcomes.

Fixed and random effects models

Fixed and random effects models can use most of these types. The fixed effects model assumes that all population effect sizes are equal, in other words, there is no heterogeneity between the studies. This is a strong assumption, which can be tested through a heterogeneity test. When the test is rejected, a random effects model should be preferred, as it assumes that the studies estimate different effect sizes, and therefore takes the between-study, as well as the within-study, variance into account.

The general fixed effects model uses an inverse variance weighted combination of the measures of outcome. The idea is that each study estimate is given a weight which is inversely proportional to its variance. Suppose that $b_1, \ldots, b_n$ are the observed effect sizes of the n independent studies to be combined, and v_i denotes the variance of b_i. Then the combined effect is given by

$$\overline{b} = \frac{\sum_{i=1}^{n} w_i b_i}{\sum_{i=1}^{n} w_i}, \quad w_i = \frac{1}{v_i}.$$

An estimate of the variance of $\overline{b}$ and a $(1-\alpha)\%$ confidence interval can be easily obtained as

$$\mathrm{var}\left(\overline{b}\right) = \frac{1}{\sum_{i=1}^{n} w_i}, \quad b \in \left[\overline{b} - z_{1-\alpha/2}\sqrt{\mathrm{var}\left(\overline{b}\right)}; \overline{b} + z_{1-\alpha/2}\sqrt{\mathrm{var}\left(\overline{b}\right)}\right],$$

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution. The fixed effects model assumes homogeneity between studies, which can be tested using Cochran's test (Cochran, 1950). Under the null hypothesis of homogeneity,

$$Q = \sum_{i=1}^{n} w_i \left(b_i - \overline{b}\right)^2 \underset{H_0}{\sim} \chi^2_{n-1}.$$

If homogeneity cannot be assumed, a random effects model, which adds a variation term to the fixed effects model to take heterogeneity into account, is used. The random effects model assumes that $b_i = \theta_i + \epsilon_i$, where b_i is the estimate of the effect size, θ_i represents the true effect

size, and ϵ_i is the estimation error. We define

$$s_w^2 = \frac{1}{n-1}\sum_{i=1}^{n} w_i^2 - n\overline{w}^2, \quad \overline{w} = \frac{1}{n}\sum_{i=1}^{n} w_i, \quad U = (n-1)\left(\overline{w} - \frac{s_w^2}{n\overline{w}}\right),$$
$$\tau^2 = \begin{cases} 0, & Q \le n-1, \\ \frac{Q-(n-1)}{U}, & Q > n-1, \end{cases} \qquad w_i^\star = \frac{1}{1/w_i + \tau^2}.$$

The combined estimate $\overline{b}_{RAN}$ and its corresponding variance are

$$\overline{b}_{RAN} = \frac{\sum_{i=1}^n w_i^\star b_i}{\sum_{i=1}^n w_i^\star}, \quad \text{var}\left(\overline{b}_{RAN}\right) = \frac{1}{\sum_{i=1}^n w_i^\star}.$$

Bayesian methods

Meta-analysis can also be performed in a Bayesian way. The model usually depends on the context but hierarchical models (to be defined in Section 2.3.1) are very useful. Taking the setting of the random effects model, one could construct the Bayesian model

$$b_i \mid \theta_i, \sigma_i^2 \sim \mathcal{N}(\theta_i, \sigma_i^2), \quad \theta_i \sim \pi(\theta_i), \quad \sigma_i^2 \sim \pi(\sigma_i^2),$$

where $\pi(x)$ is the prior distribution of x. MCMC methods (see Section 2.3) are then used to sample from the posterior densities of the parameters. Priors are chosen either to reflect prior knowledge, or to give the largest flexibility in case of non-informative priors . Bayesian meta-analysis borrows strength from all studies to update estimates of single studies, which leads to precise point estimates and shrinks the confidence intervals by reducing variability. Moreover, it is highly flexible and is therefore easily modified to include other types of information or covariates, while taking parameter uncertainty into account. However, this method is usually computationally intensive, which makes it often less attractive than frequentist approaches. There is also the recurrent problem of the choice of the prior.

Combining different types of data

When the results provided by the different studies are not of the same type, they need to be made comparable. Transformation to binary outcomes (for example whether a gene is differential expressed or not) prior to combination usually leads to uninteresting results, as the intersection is likely to be empty or to contain very few genes. Chang *et al.* (2013) describe several methods to convert study results to the same scale. For example if study results consist of summary statistics not arising from the same distribution, they can be transformed to normal z-scores. To this end, we obtain the p-values, p_{gi}, $i = 1,\ldots,n$, for each summary statistic and then transform them to z-scores, $Z_{gi} = \Phi^{-1}\left(p_{gi}\right)$, for variable g in study i, where Φ denotes the standard normal distribution. The z-scores, $Z_{g1},\ldots,Z_{gn}$, $Z_{gi} \sim \mathcal{N}(0,1)$, are then combined by the inverse normal method, also known as Stouffer's method (Stouffer *et al.*,

1949),

$$\overline{Z}_g = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} Z_{gi} \underset{H_0}{\sim} \mathcal{N}(0,1),$$

where $\underset{H_0}{\sim}$ means that, under the null hypothesis H_0 that variable g has no effect across studies, the test statistic follows a standard normal distribution. Rejection of the null hypothesis indicates an effect from variable g across studies.

Another possibility is to directly combine the p-values. They can be obtained from any test statistics, and are therefore easily combined. Several methods exist for combining p-values:

- Fisher's method (Fisher, 1932),

$$X_g^2 = -2 \sum_{i=1}^{n} \log(p_{gi}) \underset{H_0}{\sim} \chi^2_{2n};$$

- the logit method developed by George (1977),

$$T = \frac{-\sum_{i=1}^{n} \log\{p_i/(1-p_i)\}}{\sqrt{n\pi^2(5n+2)/\{3(5n+4)\}}} \underset{H_0}{\sim} t_{5n+4},$$

  where t_d is the Student distribution with d degrees of freedom;

- the minimum p-value, which takes the minimum p-value among all studies as test statistic,

$$T = \min_i p_{gi} \underset{H_0}{\sim} \text{Beta}(1,n);$$

- the maximum p-value, which takes the maximum p-value among all studies as test statistic,

$$T = \max_i p_{gi} \underset{H_0}{\sim} \text{Beta}(n,1);$$

- the rth ordered p-value, which takes the rth ordered p-value as a test statistic,

$$T = p_{g,(r)} \underset{H_0}{\sim} \text{Beta}(r, n-r+1);$$

where $\underset{H_0}{\sim} F$ means "follows the distribution F under the null hypothesis". If none of the previous methods is applicable, study results may be transformed to ranks. This will be studied in Section 2.2.

2.1.2 Meta-analysis of genomics data

Microarray data are good candidates for meta-analysis, as the number of variables (genes) is much larger than the number of samples (patients), and results from single studies tend to be highly variable. Meta-analysis, by combining results from several studies looking at the same question, may increase power in detecting differentially expressed genes between groups of tissues. However, care should be taken, as microarray data need special treatment in order to perform good meta-analyses. Guerra and Goldstein (2009, Chapters 1 and 9) provide information and guidelines to combine data from genomic studies, while Ramasamy *et al.* (2008) give seven steps for meta-analysis for gene expression data, and likewise Stevens and Doerge (2005) for Affymetrix data. Because microarrays can be built on different platforms, or analyzed using very different techniques, one needs to ensure that the studies to be included in the meta-analysis are comparable. It is first important to identify suitable microarray datasets, which should of course address the same scientific question. Then data need to be extracted from the studies, and can be of different types, including image files, also called raw data, already preprocessed matrices of gene expression values, final lists of summary statistics, or ranks. Suppose that raw data, or at least gene expression matrices, have been collected for each of the selected studies. The datasets need to be prepared in order to be comparable. Covariates of the clinical data must be consistently named and encoded (same binary variable indicating cancer or control for example). The probes need to be matched to their corresponding gene, and when several probes match to the same gene, a common procedure should be applied, e.g., averaging or selecting the most variable probe. Gene annotation is also a recurrent problem, especially if the datasets are built on different platforms. In this case one has to make sure that the annotation is the same for every dataset. Once the datasets have been prepared, a suitable method for meta-analysis is selected.

Tseng *et al.* (2012) performed a literature review recording all meta-analyses performed on genomics data. They identified the methods, the type of data and the question of interest for each study. Most of the papers used genome-wide data in order to discover differentially expressed genes between two groups of tissues. The authors identified five methods, or classes of methods, used to combine results from genomics data. The first consists in combining p-values, which is the method used in the very first meta-analysis of microarray data (Rhodes *et al.*, 2002). p-values are easy to combine as they are on the same scale, and can be easily obtained from all studies. Methods for combining p-values were described in Section 2.1.1. However, using p-values loses the information on the direction of the differential expressions, i.e., over- or under-expression. Second possibility is to combine effect size changes, as in Choi *et al.* (2003). These can be combined through fixed or random effects models (see Section 2.1.1 for more details) or by Bayesian analysis, which Choi *et al.* (2003) were the first to apply. This method, unlike to p-value or decision combinations, does not lose information regarding the magnitude or the direction of the differential expressions. However, it usually requires access to raw data, or less preferably to the gene expression matrix, which we saw in Section 1.4 can be a problem in practice. The third method identified was the combination of ranks, which we will present in more detail in Section 2.2 and which was used in a meta-analysis of

gene expression data by DeConde *et al.* (2006). Ranks are interesting, because they are not driven by outliers and tend to be more robust and therefore more reliable than effect sizes or p-values. However, using ranks loses information on the magnitude and direction of the differential expressions. Some studies merge the data, in what is called a mega-analysis, and as stated previously this can be dangerous if not performed carefully, because of Simpson's paradox. In the case of gene expression data, one could merge all the raw data (CEL files for example) and normalize them all together. This procedure is very restrictive as it requires all the data to be built on the same or very similar platforms. However, even with normalization, it may not remove cross-study discrepancies or batch effects, as demonstrated in Goldstein *et al.* (2009). Meta-analysis, by considering studies to be different, and therefore taking into account their possible heterogeneity, deals with Simpson's paradox. The last group of methods identified by Tseng *et al.* (2012) concerns the latent variable approach, including hierarchical Bayesian models. These models are more complex than the others described in this section and usually use Markov chain Monte Carlo, which can be quite computationally demanding, which may be the reason why they have not been much used. Conlon *et al.* (2007) combine probabilities of differential expression, without combining expression values, through a hierarchical Bayesian model. By eliminating the need to model between-study variability, they claim to be able to increase the number of integration-driven discoveries, i.e., genes that are found to be differentially expressed by the combination of studies but not in single study analyses. Their model can combine several studies providing full information. Hierarchical models for microarray data were also used by Ishwaran and Rao (2003) for a single study. Shen *et al.* (2004) used them to obtain the probability of expression for each gene and each study prior to combination, and finally Chen *et al.* (2013) propose to model gene sets and gene expression data from multiple studies using hierarchical Bayesian framework. In this last paper, raw data along with pathway information have to be available. Bayesian methods allow the fitting of complicated models, while being very flexible, as described in Section 2.1.1. However, up to now, they were only designed to combine the same type of information from different studies.

2.2 List aggregation

Analysis of microarray data often aims at finding differentially expressed genes, in which case the published results consist of a list of ordered genes, from the most to the least differentially expressed. It may therefore be interesting to combine ranks, which are usually easy to find as published study results in articles or supplementary material. Ranks can also be obtained by ordering the absolute values of z-scores or the p-values for example. Therefore, even if a study provides more information about its data, it is always possible to extract an ordered list of genes. Transformation to ranks avoids discarding studies, which commonly arises when restricting the analysis to those providing raw data. Ranks also have the valuable property of being robust to outliers, invariant under normalization or transformation, and scale free. However, as stated in the previous section, they lose information about the magnitude and

the direction of the differential expressions. In this section, we present a brief review of the methods used for list aggregation. We consider either full or partial lists, composed of the top k most interesting genes, with k much smaller than the number of genes G. We assume that top k lists are subsets of the same common set of genes $\mathcal{G}$. Thus if an element does not appear in a partial list, it means it was included in the analysis, but ranked lower than k, and therefore not observed in the top k list. This section is mainly inspired by Lin (2010) and DeConde *et al.* (2006).

Let $\mathcal{G}$ be a discrete space of size G, with elements $\mathcal{G} = \{1, \ldots, G\}$. The rank of an element g of $\mathcal{G}$ is denoted $R(g)$. The complete ranking of the elements of $\mathcal{G}$ is $\tau(\mathcal{G}) = \left(g_1, \ldots, g_G\right)$, $g_i \in \mathcal{G}$, such that $R(g_i) < R(g_j)$, for $i < j$, obtained by applying the permutation τ to all the elements of $\mathcal{G}$. A partial list is based on a subset $\mathcal{S} \subset \mathcal{G}$, $\mathcal{S} = (s_1, \ldots, s_S)$, and is supposed to be already ranked, i.e., $R(s_1) < R(s_S)$, and a particularly interesting case is when $\mathcal{S}$ consists of the top k list, $\mathcal{S} = \left(g_1, \ldots, g_k\right)$, the k first elements of $\tau(\mathcal{G})$.

We are interested in aggregating L lists, $S_1, \ldots, S_L$, where list S_l is of length n_l, either full or partial (top n_l), obtained from applying a permutation τ_l to the elements of interest. We aim to find the resulting aggregated list of elements in $\mathcal{S} = \cup_{l=1}^{L} S_l$. If an element is in the union of all lists but not in S_l, i.e., $i \in \mathcal{S} \cap S_l^c$, we set $R_{\tau_l}(i) = n_l + 1$. Methods for list aggregation can be classified into three categories, according to Lin (2010): distribution-based, heuristic and based on stochastic optimization.

2.2.1 Distribution-based methods

Distribution-based methods were first developed for aggregating many short lists, such as consumer rankings of products, or internet search results (Lin, 2010).

Thurstone's model

This model was first used in the context of marketing, where the interest is in combining many short lists. For full ranked lists, it assumes that the underlying vector of values $X = (X_1, \ldots, X_G)$ follows a multivariate normal distribution with mean vector $\mu = (\mu_1, \ldots, \mu_G)$ and $G \times G$ covariance matrix Σ. Each pair is distributed as a bivariate normal

$$(X_u, X_v) = \mathcal{N}\left[\begin{pmatrix} \mu_u \\ \mu_v \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & \sigma_u\sigma_v\rho_{uv} \\ \sigma_u\sigma_v\rho_{uv} & \sigma_v^2 \end{pmatrix}\right].$$

It is therefore possible to compute the probability that an element X_u of X is ranked above another, X_v,

$$P(X_u > X_v) = \Phi\left(\frac{\mu_u - \mu_v}{\sqrt{\sigma_u^2 + \sigma_v^2 - 2\rho_{uv}\sigma_u\sigma_v}}\right), \tag{2.1}$$

where Φ is the standard normal distribution function.

If $\tau_1, \ldots, \tau_L$ are the full rankings of each of the L lists that we want to aggregate, Thurstone's score for two elements u and v of a ranked list l is

$$H_{\tau_l}(u,v) = I_{\left\{R_{\tau_l}(u) < R_{\tau_l}(v)\right\}}.$$

We estimate the probability (2.1) by

$$P(X_u > X_v) = \frac{1}{L}\sum_{l=1}^{L} H_{\tau_l}(u,v).$$

The final aggregated list is obtained by ranking the entries of the estimated mean parameters μ. In practice, the elements of the covariance matrix are hard to estimate if we consider all pairs. To simplify the problem and reduce the number of unknown parameters, the variances σ_u^2 can be fixed to 1, while the correlations are set to $\rho_{uv} = 0$ (DeConde *et al.*, 2006). The mean parameters are estimated by least squares.

For partial lists, we want to aggregate $S_1, \ldots, S_L$, lists containing the top n_l rankings of the elements of $\mathcal{G}$, with underlying ranking functions $\tau_1, \ldots, \tau_L$. If $u \in S_l^c$, then $R_{\tau_l}(u) = n_l + 1$. For $u, v \in \mathcal{S} = \cup_{l=1}^{L} S_l$, Thurstone's score is

$$H_{\tau_l}(u,v) = \begin{cases} p, & R_{\tau_l}(u) = R_{\tau_l}(v) = n_l + 1, \\ I_{\left\{R_{\tau_l}(u) < R_{\tau_l}(v)\right\}}, & \text{otherwise,} \end{cases}$$

where p is a parameter to be set between 0 and 1, and which represents the probability that u is ranked before v when their rankings are unknown. Usually we choose $p = 0.5$, indicating an equal chance for u to be ranked higher than v.

This method is quite simple, under the assumption of normality. However, as pointed out previously, the parameters are hard to estimate due to the large number of pairwise comparisons, and setting $\sigma_u^2 = 1$ and $\rho_{uv} = 0$ may be unrealistic. Thurstone's method is more appropriate for many short lists for two reasons. The first concerns the number of pairwise comparisons, which is much smaller if the lists are small, and the second concerns the estimation of the probabilities (2.1), which are well estimated if L is large.

2.2.2 Heuristic algorithms

In this section, we describe two types of algorithms, Borda's method and some other related rank combination methods, as well as several Markov chain-based methods.

Borda's method

This method is simple and quite intuitive. It consists in attributing a score to each element of each list and aggregate them through a function. Suppose we have L full lists $S_1, \ldots, S_L$, and let $\mathscr{S} = \cup_{i=1}^{L} S_i$. For an element g having rank $R_{\tau_l}(g)$ in list S_l, Borda's score is

$$B_l(g) = f\left\{R_{\tau_1}(g), \ldots, R_{\tau_L}(g)\right\},$$

where f can be one of the following functions:

- the median, $f\left\{R_{\tau_1}(g), \ldots, R_{\tau_L}(g)\right\} = \text{median}\left\{R_{\tau_1}(g), \ldots, R_{\tau_L}(g)\right\}$;
- the arithmetic mean, $f\left\{R_{\tau_1}(g), \ldots, R_{\tau_L}(g)\right\} = L^{-1}\sum_{l=1}^{L} R_{\tau_l}(g)$;
- the geometric mean, $f\left\{R_{\tau_1}(g), \ldots, R_{\tau_L}(g)\right\} = \left(\prod_{l=1}^{L} R_{\tau_l}(g)\right)^{1/L}$;
- the p-norm, $f\left\{R_{\tau_1}(g), \ldots, R_{\tau_L}(g)\right\} = L^{-1}\sum_{l=1}^{L} R_{\tau_l}(g)^p$.

For partial lists, one can either use $R_{\tau_l}(g) = n_l + 1$ or ignore the lists that do not include element g. The final list is then based on the Borda scores, where a small score corresponds to a small rank.

The product of ranks (PR) and sum of ranks(SR) methods (Chang *et al.*, 2013) are both based on similar ideas to Borda's method, as they attribute a score to each rank, defined as:

$$\text{PR}_g = \prod_{i=1}^{n} R_{\tau_i,g}, \quad \text{SR}_g = \sum_{i=1}^{n} R_{\tau_i,g},$$

respectively. p-values for these methods may be obtained by permutation of the variable labels.

Markov Chain methods

These methods only use pairwise ranking information and consist in constructing an ergodic Markov chain, whose transition matrix converges to a stationary distribution that will assign a larger probability to the elements/states that are ranked higher. Details about Markov chains and their construction are presented in Section 2.3. There exist several methods for constructing the transition matrix $\mathscr{P}$, with elements $\mathscr{P}_{uv}$ being the probability of going from state u to state v, with $u, v \in \mathscr{S}$. Dwork *et al.* (2001) introduced the method in the context of spam fighting, Lin (2010) present three algorithms, which we will denote MC1, MC2 and MC3. The MC1 and MC2 algorithms are slight variations of the algorithms developed by Dwork *et al.* (2001), while MC3 is inspired by the algorithm developed by DeConde *et al.* (2006), which is more appropriate for combining lists of genes from microarray data analysis (Lin, 2010).

- **MC1**: the transition probability matrix is

$$\mathscr{P}_{uv} = \begin{cases} 1/S, & \text{if } R_{\tau_l}(u) > R_{\tau_l}(v) \text{ for at least one of the input lists,} \\ 0.5/S, & \text{if } u \text{ and } v \text{ are not compared,} \\ 0, & \text{otherwise,} \end{cases}$$
$$\mathscr{P}_{uu} = 1 - \sum_{v \neq u} \mathscr{P}_{uv}.$$

- **MC2**: in this case, the probability transition matrix is

$$\mathscr{P}_{uv} = \begin{cases} 1/S, & \text{if } R_{\tau_l}(u) > R_{\tau_l}(v) \text{ for a majority of the input lists,} \\ 0.5/S, & \text{if } u \text{ and } v \text{ are not compared,} \\ 0, & \text{otherwise,} \end{cases}$$
$$\mathscr{P}_{uu} = 1 - \sum_{v \neq u} \mathscr{P}_{uv}.$$

  This algorithm is also called the MC4 algorithm in DeConde *et al.* (2006).

- **MC3**: for $u, v \in \mathscr{S}$,

$$\mathscr{P}_{uv} = \begin{cases} (LS)^{-1} \sum_{l=1}^{L} I_{\left\{R_{\tau_l}(u) > R_{\tau_l}(v)\right\}}, \\ 0.5/S, \text{ if } u \text{ and } v \text{ are not compared in any list,} \end{cases}$$
$$\mathscr{P}_{uu} = 1 - \sum_{v \neq u} \mathscr{P}_{uv}.$$

  This algorithm is also called the MCT algorithm in DeConde *et al.* (2006).

In order to make the transition matrix ergodic, we apply the following transformation

$$\mathscr{P}' = \mathscr{P}(1-\epsilon) + \frac{\epsilon}{S},$$

where ϵ is a small number (we choose $\epsilon = 0.05$ when using these methods). Once the transition matrix is defined, we can find the stationary distribution of the Markov chain, which will determine the aggregate ranking, by attributing a higher probability to the states (the genes) that are ranked higher.

2.2.3 Stochastic optimization methods

Stochastic optimization methods optimize a criterion based on a dissimilarity measure between the ranked list and the aggregated list. Here we define two distance measures, the Kendall's tau and Spearman's footrule distances.

Kendall's tau

Let S_1 and S_2 be two lists of elements of $\mathscr{S}$, with ranking functions τ_1 and τ_2 respectively. We

define

$$B = \left\{(u,v) : R_{\tau_1}(u) = R_{\tau_1}(v) = n_1 + 1, \text{ or } R_{\tau_2}(u) = R_{\tau_2}(v) = n_2 + 1\right\},$$

i.e., the set of pairs of elements that do not appear in at least one of the two lists. The Kendall's tau measure between τ_1 and τ_2 is defined as

$$K(\tau_1,\tau_2) = \begin{cases} \sum_{u,v\in\mathscr{S}} I_{\{(R_{\tau_1}(u)-R_{\tau_1}(v))(R_{\tau_2}(u)-R_{\tau_2}(v))<0\}}, & (u,v) \in B^c, \\ Sp, & \text{otherwise,} \end{cases}$$

where p is a parameter between 0 and 1, usually set to 0.5.

Spearman's footrule distance
Spearman's footrule distance is defined as

$$S(\tau_1,\tau_2) = \sum_{u\in\mathscr{S}} |R_{\tau_1}(u) - R_{\tau_2}(u)|.$$

In the stochastic optimization setting one wants to minimize a criterion following the generalized Kemeny guidelines (Lin, 2010), which, in the case of full ranked lists of elements of $\mathscr{G}$, assumes that the observed τ_l are some noisy realizations of the true list τ and that they differ from τ by swapping two elements with a given probability $p < 0.5$. The Kemeny optimal aggregation is the maximum likelihood estimate of τ,

$$\widehat{\tau} = \underset{\tau}{\operatorname{argmin}} \frac{1}{L} \sum_{l=1}^{L} K(\tau,\tau_l),$$

where $K(\tau,\tau_l)$ is Kendall's tau distance.

The assumption that each observed list differs from the true one by simply swapping two elements is not realistic but the idea can be extended to any ranked list, also partial, of the elements of $\mathscr{G}$, under the generalized Kemeny guidelines. If $S_1,\ldots,S_L$ are partial lists with underlying ranking functions $\tau_1,\ldots,\tau_L$ and common element space $\mathscr{S} = \cup_{l=1}^{L} S_l$, we want to find τ, a list of elements in $\mathscr{S}$, which minimizes the distance with each of the observed lists. This is equivalent to minimizing the function f, defined as the weighted sum of distances between τ and each of the S_l,

$$\widehat{\tau} = \underset{\tau}{\operatorname{argmin}} \left\{f(\tau), \tau \subset S\right\} = \underset{\tau}{\operatorname{argmin}} \left\{\sum_{l=1}^{L} w_l d(\tau, S_l)\right\}, \tag{2.2}$$

where w_l represents the confidence we put on the list S_l and $d(\cdot,\cdot)$ is either Kendall's tau or Spearman's footrule.

Cross-Entropy Monte Carlo method

In order to optimize criterion (2.2), one could try every possible list τ and take the one that gives the lowest value of the objective function. However, since the set $\mathscr{S}$ can contain thousands of elements, such a method is not tractable. The Cross-Entropy Monte Carlo (CEMC) method searches for the optimal list that minimizes (2.2), by using an iterative method (Lin and Ding, 2009). Suppose we have L partial lists $S_1,\ldots,S_L$ of different lengths, but truncated to have length k. As previously, we denote by $\mathscr{S} = \cup_{l=1}^{L} S_l$ of size S the list of all elements. We wish to find $\tau \subset \mathscr{S}$ of length k, which satisfies (2.2). First, we define a matrix X of size $S \times k$ with elements taking values 0 or 1, $X_{ij} = 1$ meaning that $R_{\tau_l}(i) = j$, and satisfying

$$\sum_{i=1}^{n} X_{ij} = 1, \quad \sum_{j=1}^{k} X_{ij} \leq 1.$$

A probability matrix $v = (v_{ij})$, of size $S \times k$, with columns summing to 1, is associated to the matrix X. Therefore, each column X_j of X follows a multinomial distribution with sample size 1 and probability vector $v_j = (v_{1j},\ldots,v_{nj})$. Then, the probability mass function can be computed as

$$P_v\left(X = x = (x_{ij})_{S\times k}\right) \propto \prod_{j=1}^{k}\prod_{i=1}^{S}(v_{ij})^{x_{ij}} I_{\left\{\sum_{i=1}^{S} x_{ij}=1,\, i=1,\ldots,k;\, \sum_{j=1}^{k} x_{ij}\leq 1,\, j=1,\ldots S\right\}}.$$

In CEMC, the candidate top k list is $\tau = \left\{x_{ij} : x_{ij} = 1;\, i = 1,\ldots,n;\, j = 1,\ldots,k\right\}$, so finding the optimal τ is equivalent to finding the optimal x, $\widehat{x}$, by iteratively updating the parameter matrix v, such that $P_v(x)$ places more and more probability mass on the x's in the neighborhood of $\widehat{x}$. We describe the order explicit algorithm (OEA), as presented in Lin and Ding (2009):

1. set v^0, with each $v_{ij}^0 = 1/S$ for example, if no prior information is available;
2. draw a sample $X_r = (x_{rij})$ from $P_{v^t}(x)$, $r = N_1+1,\ldots,N$. The N_1 first values, $X_1,\ldots,X_{N_1}$, are the realizations from $P_{v^{t-1}}$ leading to the smallest values of the objective function;
3. we obtain a sample of size N, $(X_1,\ldots,X_{N_1},X_{N_1+1},\ldots,X_N)$, from which we find the corresponding top k list candidates, $\tau_1,\ldots,\tau_N$, and their objective values, $y_1 = f(\tau_1),\ldots,y_N = f(\tau_N)$;
4. sort the objective values in increasing order $f_{(1)} \leq \cdots \leq f_{(N)}$ and define $y^t = f_{\lfloor \rho N \rfloor}$, $0 < \rho < 1$;
5. using the same sample, update the parameter vector v^{t+1} as follows

$$v^{t+1} = (1-\pi)\,v^t + \pi\, v_{\text{new}}, \quad (v_{\text{new}})_{ij} = \frac{\sum_{r=1}^{N} I_{\{f(\tau_i)\leq y^t\}} x_{rij}}{\sum_{r=1}^{N} I_{\{f(\tau_i)\leq y^t\}}},$$

where π is the weight parameter;

6. if $||v^{t+1} - v^t|| < \epsilon$, $0 < \epsilon < 0.01$, go to step 7, otherwise, set $t = t + 1$ and go back to step 2;

7. set $y = f_{(1)}$ and the corresponding ordered subset is the aggregated list.

Rank aggregation methods have been used in several studies to combine results from microarray studies. DeConde *et al.* (2006) combine five prostate cancer studies using two Markov chain algorithms and compare the results with Thurstone's method. Pihur *et al.* (2009) developped an R package to compute the CEMC algorithm and applied it to the meta-analysis of five prostate cancer studies. Lin (2010) reviews existing methods and compares the results of all methods presented above based on three long fictional lists and five prostate cancer gene expression studies. She found that the CEMC usually performs slightly better than other methods. However, it needs quite a lot of tuning and results are sensitive to the choice of the tuning parameters, which was also found by Pihur *et al.* (2009). Borda's method, despite its simplicity and absence of tuning parameters, performs surprisingly well, giving results similar to the more complicated algorithms.

2.3 Bayesian approaches

2.3.1 Hierarchical models

Our model, presented in Chapter 3, is a hierarchical Bayesian model. We here review the definition and motivation for using such models. Material on Bayesian hierarchical models mainly comes from Gelman *et al.* (2013, Chapter 5), Congdon (2010, Chapter 1) and Davison (2003, Chapter 11). Hierarchical models allow the fitting of complex models to data in a relatively straightforward way. They are particularly suited for data having several levels of variation, or that depend on parameters connected in some way by the structure of the problem. In the context of microarray data, for example, studying the differential expression of a gene g across several patients from different studies, we may assume that the differential expression parameters of gene g, in each study l, $\beta_g^{(l)}$, are related. We can see different $\beta_g^{(l)}$ as coming from the same underlying distribution. The expression value of a gene g, for the jth patient, in the lth study, $Y_{gj}^{(l)}$, can be used to estimate aspects of the unobserved $\beta_g^{(l)}$. A hierarchical structure in this case seems natural: the observations are modeled conditioned on the parameter $\beta_g^{(l)}$, which comes from a common population distribution, which may also depend on another set of parameters, called hyperparameters. Hierarchical models help to understand the dependence structure among multiple parameters. They rely on using the observations to update knowledge about unknown parameters, obtained by revised knowledge through the posterior density of the parameter given the data. The prior density of a parameter represents knowledge collected before observing the data, while the posterior density uses the information collected in the observations to update the parameter.

A crucial assumption in this context is infinite exchangeability.

Definition 2.3.1. (Davison, 2003, p. 619) The random variables $X_1, \ldots, X_n$ are called *finitely*

exchangeable if their density $f(x_1,\dots,x_n)$ can be written as $f(x_1,\dots,x_n) = f\left(x_{\xi(1)},\dots,x_{\xi(n)}\right)$, for any permutation $\xi : \{1,\dots,n\} \to \{1,\dots,n\}$.
An infinite sequence $X_1, X_2,\dots$ is *infinitely exchangeable* if any finite sequence of it is finitely exchangeable.

Infinite exchangeability may be assumed if there is no way to identify the studies in the microarray data example presented previously, i.e., the joint distribution of $\beta_g^{(1)},\dots,\beta_g^{(L)}$ would be exactly the same if we permute the indices $1,\dots,L$. Assuming exchangeability of the random variables, hierarchical models are motivated by the following theorem.

Theorem 2.3.2. ***De Finetti's theorem:*** *(Davison, 2003, p. 619) If $X_1, X_2,\dots$ is an infinitely exchangeable sequence of binary variables, taking values 0 or 1, then for any n, there is a distribution G such that*

$$f(x_1,\dots,x_n) = \int_0^1 \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i}\, dG(\theta),$$

where

$$G(\theta) = \lim_{m\to\infty} P\left(\frac{1}{m}\sum_{i=1}^{m} X_i \le \theta\right), \quad \theta = \lim_{m\to\infty} \frac{1}{m}\sum_{i=1}^{m} X_i.$$

For the proof, we refer to Davison (2003) (Section 11.4). This theorem implies that any set of exchangeable binary variables may be modeled as independent Bernoulli variables, conditional on success probability θ, which has a distribution G. Even if the theorem is stated for binary variables for simplicity, it also holds for continuous variables.

As the parameters on which the observations rely are not observed ($\beta_g^{(l)}$ in our example) we put a prior distribution on them, to reflect prior knowledge, which may depend on hyperparameters, φ. The joint posterior distribution is given by

$$\pi(\beta,\varphi \mid y) \propto f(y \mid \beta)\pi(\beta \mid \varphi)\pi(\varphi),$$

using the dependence structure to simplify the calculations (y only depends on β and not on φ). To obtain a sample from the posterior distribution, one can follow the algorithm described in Gelman *et al.* (2013, p. 126), assuming that the posterior $\pi(\beta \mid \varphi)$ is conjugate with the likelihood $f(y \mid \beta)$. To simplify the notation and following the arguments of the authors, we denote the observations by y, which depend on a parameter β, itself depending on a hyperparameter φ. Then the algorithm is

- write the joint posterior density $\pi(\beta,\varphi \mid y) = f(y \mid \beta)\pi(\beta \mid \varphi)\pi(\varphi)/f(y)$. The marginal likelihood appearing in the denominator $f(y)$ does not depend on β nor φ, so can be treated as a constant, in which case we could write $\pi(\beta,\varphi \mid y) \propto f(y \mid \beta)\pi(\beta \mid \varphi)\pi(\varphi)$, or it could be computed by integrating the likelihood over β: $f(y \mid \varphi) = \int f(y \mid \beta)\pi(\beta \mid \varphi)\mathrm{d}\beta$;

- for a fixed observation y, obtain the explicit form of the conditional posterior density $\pi(\beta \mid \varphi, y)$;
- use the marginal posterior density of φ, $\pi(\varphi \mid y)$ to estimate φ, by integrating over β, $\pi(\varphi \mid y) = \int_\beta \pi(\varphi, \beta \mid y) \mathrm{d}\beta$, or using the conditional probability formula $\pi(\varphi \mid y) = f(\beta, \varphi \mid y) / \pi(\beta \mid \varphi, y)$.

In the last step, to simulate from the posterior distribution $\pi(\beta, \varphi \mid y)$, one can use another algorithm from Gelman *et al.* (2013, Chapter 5, page 127):

- simulate $\varphi \sim \pi(\varphi \mid y)$;
- simulate β from its conditional posterior distribution $\pi(\beta \mid \varphi, y)$, given the value of φ generated in the previous step;
- if necessary, draw $\tilde{y}$ from its posterior predictive distribution given the value of θ generated in the previous step.

These steps are performed until a sample of acceptable size is reached. This allows the estimation or prediction of any quantity of interest.

To better visualize and represent hierarchical models, directed acyclic graphs allow the identification of the different levels of the hierarchy and the relation between the different variables and parameters. A directed acyclic graph is a graph having oriented edges and no loop. Fixed hyperparameters can be represented by squares, and parameters by circles. The relation between two variables is represented by an arrow going from the parameter (the parent) to the variable (the descendent). The directed acyclic graph gives a representation of the decomposition of the joint distribution into conditional distributions, implying that variables are conditionally independent from the non-descendents given the parents (Green, 2001). As an illustration, we give here a very simple example, inspired by Green (2001, p. 5), for modeling a normal random sample $Y_1, \ldots, Y_n$, having mean β, with a normal prior depending on two hyperparameters, and variance σ^2, with an inverse gamma distribution also depending on two hyperparameters, a and b. The model is

$$Y_i \mid \beta, \sigma^2 \overset{\text{iid}}{\sim} \mathcal{N}(\beta, \sigma^2), \quad \beta \mid \theta, \tau^2 \sim \mathcal{N}(\theta, \tau^2), \quad \sigma^{-2} \mid a, b \sim \text{Gamma}(a, b), \tag{2.3}$$

and is represented as a directed acyclic graph in Figure 2.1.

Meta-analysis, being a method for combining results from several indistinguishable sources (i.e., exchangeable), has a natural hierarchical structure and is therefore a perfect candidate for hierarchical modeling. We need to assume that the studies are exchangeable, i.e., we do not assume that they are identical replications, they may show differences, but these differences are not expected to have any impact on the outcome of interest a priori. In other words, even if study results vary, no study is favored and variation is at random. We also need to assume that

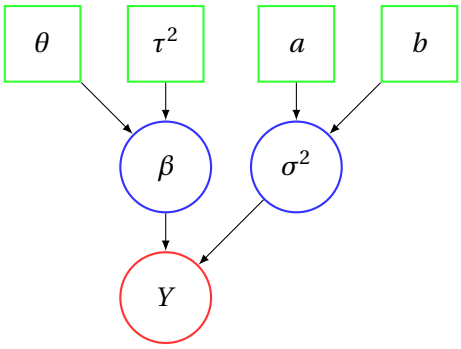


Figure 2.1 – Directed acyclic graph (DAG) of the example model (2.3)

the individuals enrolled in each study are independent samples from a common population, conditional on model parameters. The idea, as already mentioned in the illustration at the beginning of this section, is to attribute a parameter to each study, assuming that it comes from the same underlying population. A combined estimate is obtained through the estimation of the population distribution parameters.

An important type of hierarchical model, which we will use in Chapter 3, is the latent hierarchical Bayesian model. It assumes that the observed data rely on unobserved latent variables, intermediate between the data and the parameters. We let y denote the observed data, u the latent variables and β the parameter. This structure defines a three-stage hierarchical Bayesian model whose joint density may be written

$$f(y,u,\beta) = \pi(y \mid u,\beta)\pi(u \mid \beta)\pi(\beta).$$

Using Bayes' theorem, one can write the joint posterior density for β and u and the marginal posterior density for β as

$$\pi(\beta,u \mid y) = \frac{\pi(y \mid u,\beta)\pi(u \mid \beta)\pi(\beta)}{f(y)}, \quad \pi(\beta \mid y) = \frac{\pi(y \mid \beta)\pi(\beta)}{f(y)} = \frac{\pi(\beta)\int f(y \mid u,\beta)\pi(u \mid \beta)\mathrm{d}u}{f(y)},$$

where $f(y \mid \beta)$ is called the observed data likelihood or the integrated likelihood. Often, this quantity is not explicitly available and classical likelihood estimation relies on numerical approximations. MCMC methods circumvent this by sampling directly from the posterior distribution $\pi(\beta,u \mid y)$, which only requires $\pi(\beta \mid u,y)$ and $\pi(u \mid \beta,y)$ (see Congdon, 2010, p. 8, for more details). If the latent data exist for each observation, there is a parameter β_u corresponding to each latent variable. This model was applied by Albert and Chib (1993) for modeling latent variables $u_1,\ldots,u_n$ underlying binary observations $y_1,\ldots,y_n$.

Hierarchical Bayesian models have become increasingly used due to the development of Markov chain Monte Carlo (MCMC), a computing tool for parameter estimation, which draws multiple parameter samples from the posterior distributions of interest (Congdon, 2010, p.

2). MCMC methods can handle models of increased complexity, and ease inference by not requiring any normality assumption to compute confidence intervals for example. We describe Markov chains in the next sections, MCMC methods in Section 2.3.4, and the best-known algorithms in Sections 2.3.5 and 2.3.6.

2.3.2 Markov chains

This section is mainly inspired by Brooks *et al.* (2011) and Gilks *et al.* (1996).

Definition 2.3.3. A sequence of random variables $X_1, X_2, \ldots$ is a *Markov chain*, if the conditional probability of the next value given the present and the past only depends on the present:

$$P(X_{n+1} \mid X_n, X_{n-1}, \ldots X_1) = P(X_{n+1} \mid X_n), \quad n \in \mathbb{N}.$$

We denote the chain by $\{X_n\}$. If the conditional probability of X_{n+1} given X_n does not depend on n, the chain is time-homogeneous. In this work we only consider time-homogeneous Markov chains.

The set of possible values for $\{X_n\}$ is called the state space and is denoted by $\mathscr{S}$. If the state space is discrete and finite, the stationary transition probability may be written as a matrix indicating the probability of going from x_i to x_j in one step, $x_i, x_j \in \mathscr{S}$,

$$\mathscr{P}_{ij}^{(1)} = P\left(X_n = x_j \mid X_{n-1} = x_i\right).$$

The probability of going from x_i to x_j in k steps is $\mathscr{P}_{ij}^{(k)}$. The marginal distribution of the nth stage, $\pi^{(n)}$, takes values in $\mathscr{S}$, and $\pi^{(1)}$, the marginal distribution of X_1, gives the initial distribution of the chain.

When the state space is uncountable, which is generally the case in applications, the transition probability matrix becomes a conditional probability distribution. For simplicity, we will present results for countable state spaces, keeping in mind that they can be generalized. A Markov chain is stationary when the distribution of consecutive variables at different time lags does not depend on the lag, i.e., for any positive integer k, the distribution of $(X_{n+1}, \ldots, X_{n+k})$, does not depend on n.

Our goal in the next sections will be to construct a homogeneous Markov chain that converges to a given stationary distribution. To obtain this property, the chain must satisfy three criteria: irreducibility, aperiodicity and positive recurrence. An aperiodic and positive recurrent Markov chain is called ergodic. We give the definitions of each of the terms, as stated by Gilks *et al.* (1996, Chapter 3, p.46-47).

Definition 2.3.4. A Markov chain $\{X_n\}$ is *irreducible* if for all i, j there exists a k such that $\mathscr{P}_{ij}^{(k)} > 0$.

Definition 2.3.5. An irreducible Markov chain $\{X_n\}$ is *recurrent* if $P(\tau_{ii} < \infty) = 1$ for any i, where τ_{ii} is the return time to state $x_i \in \mathscr{S}$. Otherwise the chain is *transient*.

Definition 2.3.6. An irreducible and recurrent Markov chain $\left\{X^{(n)}\right\}$ is *positive recurrent* if the expected return time is finite for all i, $\mathbb{E}(\tau_{ii}) < \infty$. Otherwise, it is called *null-recurrent*.

Definition 2.3.7. An irreducible Markov chain is *aperiodic* if the greatest common divisor of the number of steps to return to state x_i from x_i is 1,

$$\gcd\left(n > 0 : \mathscr{P}_{ii}^{(n)} > 0\right) = 1.$$

An irreducible and positive-recurrent Markov chain admits the existence of a stationary or invariant distribution $\pi(\cdot)$, which satisfies

$$\sum_i \pi(x_i)\mathscr{P}_{ij} = \pi(x_j), \quad x_i, x_j \in \mathscr{S}. \tag{2.4}$$

In matrix notation, equation (2.4) can be written $\pi\mathscr{P} = \pi$. The stationary distribution of an irreducible, aperiodic and positive definite Markov chain is the limiting distribution, whatever the starting value, in the sense that, if there exists a stationary distribution π, then $\lim_{n\to\infty} \mathscr{P}_{ij}^{(n)} = \pi(x_j)$. It can also be shown that if $\pi^{(n)}$ is the marginal distribution at stage n, then $\pi^{(n)} \underset{n\to\infty}{\to} \pi$. This is stated more formally by the following theorem (Gilks *et al.*, 1996, Chapter 3, p.47),

Theorem 2.3.8. *If $\{X^{(n)}\}$ is ergodic, its stationary distribution $\pi(\cdot)$ is the unique stationary probability distribution. The chain also satisfies the following properties,*

- $\mathscr{P}_{ij}^{(n)} \to \pi(x_j)$, $n \to \infty$, *for all* i, j.
- ***Ergodic theorem:*** *If $\mathbb{E}_\pi\left(|f(X)|\right) < \infty$, then $P\left[\overline{f}_N \to \mathbb{E}_\pi\left\{f(X)\right\}\right] = 1$,*

where $\mathbb{E}_\pi\left[f(X)\right] = \sum_i f(x_i)\pi(x_i)$ is the expectation of $f(X)$ with respect to $\pi(\cdot)$, and $\overline{f}_N = N^{-1}\sum_{i=1}^N f(X_i)$.

The last part of Theorem 2.3.8 is the Markov equivalent of the law of large numbers and ensures consistency of the estimators of the parameters of the limiting distribution (Gamerman, 1997, p.105). There exists a version of the central limit theorem for ergodic Markov chains which satisfies a property called geometric ergodicity. Having an equivalent of the central limit theorem will be very useful in the next section, and especially in Section 2.3.7 for inference.

Definition 2.3.9. An ergodic Markov chain is *geometrically ergodic* if there exist $0 \leq \lambda < 1$ and a real integrable function $V(x)$ such that

$$\sum_j |\mathscr{P}_{ij}^{(n)} - \pi(x_j)| \leq V(x_i)\lambda^n, \quad i = 1, \ldots, n.$$

The value $\lambda^\star = \inf\left\{\lambda : \exists V \text{such that } \sum_j |\mathscr{P}_{ij}^{(n)} - \pi(x_j)| \leq V(x_i)\lambda^n\right\}$ is called the rate of convergence.

If $\{X^{(n)}\}$ is geometrically ergodic, then we can give a version of the central limit theorem for Markov chains:

$$\frac{1}{\sqrt{n}}\left[\overline{f}_n - \mathbb{E}_\pi\{f(X)\}\right] \xrightarrow{d} \mathscr{N}(0,\tau^2),$$

where $\xrightarrow{d}$ means convergence in distribution, and τ^2 is the limiting sampling variance of $\sqrt{n}\overline{f}_n$, which is obtained as the limit of the sampling variance $\text{var}_\pi\left(\overline{f}_n\right) = \tau_n^2$

$$\frac{\tau_n^2}{n} = \underset{n\to\infty}{\to} \tau^2,$$

Another property that is useful for construction is reversibility, which holds when the detailed balance equation

$$\pi(x_i)\mathscr{P}_{ij} = \pi(x_j)\mathscr{P}_{ji}, \tag{2.5}$$

is satisfied. Detailed balance is a sufficient condition for invariance (Gamerman, 1997, Section 4.7).

2.3.3 Empirical Bayes and Bayes models

In Bayesian settings, all parameters are attributed prior distributions, which may themselves depend on hyperparameters, on which we may put second stage priors, either parametric or non-parametric. Priors for hyperparameters are often difficult to estimate and their effect on subsequent inferences is hard to assess (Davison, 2003, Section 11.5). Empirical Bayes methods solve these issues by using the information contained in the data to estimate the hyperparameters. The Bayesian model is then applied using the estimated value of the hyperparameters rather than a prior distribution. The name empirical Bayes comes from the fact that we are using the data to estimate the hyperparameters (Carlin and Louis, 2000, Chapter 3). If an explicit expression can be obtained for the likelihood, we can estimate the hyperparameters of the model by maximum likelihood. We consider a simple example, which will be the basis of our empirical Bayes model from Chapter 6. Suppose we observe data Y which are a mixture of two normals with different means, a priori

$$\begin{aligned} Y \mid \theta, \sigma &\sim \mathscr{N}(\theta, \sigma^2), \\ \theta \mid p, \tau &\sim (1-p)\delta_0(\theta) + p\mathscr{N}(0,\tau^2), \end{aligned}$$

where $\delta_0(\theta)$ puts mass 1 at $\theta = 0$. The likelihood based on observations $y_1, \ldots, y_n$ is

$$\begin{aligned} L(p, \sigma, \tau) = \prod_{i=1}^{n} f(y_i) &= \prod_{i=1}^{n} \int_\theta \frac{1}{\sigma} \varphi\left(\frac{y_i - \theta_i}{\sigma}\right) \left\{(1-p)\delta_\theta + p\frac{1}{\tau}\varphi\left(\frac{\theta_i}{\tau}\right)\right\} \mathrm{d}\theta, \\ &= \prod_{i=1}^{n} \left\{(1-p)\frac{1}{\sigma}\varphi\left(\frac{y_i}{\sigma}\right) + \frac{p}{\sqrt{\sigma^2+\tau^2}}\varphi\left(\frac{y_i}{\sqrt{\sigma^2+\tau^2}}\right)\right\}, \end{aligned}$$

where φ is the standard normal density. Thus, we can estimate parameters p, σ and τ by maximizing the log likelihood, and use Bayesian methods to obtain an estimate for θ_i, with the hyperparameters replaced by their estimate. Inference is based on the estimated posterior distribution $\pi(\theta \mid y, \widehat{\sigma}, \widehat{\tau}, \widehat{p})$, and with $\widehat{\phi} = (\widehat{\sigma}, \widehat{\tau}, \widehat{p})$,

$$\pi(\theta_i \mid y_i, \widehat{\phi}) = \frac{f(y_i \mid \theta_i)\pi(\theta \mid \widehat{\phi})}{f(y_i)}.$$

In our previous example, the parameter θ depends on a parametric distribution and therefore the approach is called parametric empirical Bayes (Carlin and Louis, 2000, Chapter 3). Nonparametric empirical Bayes methods offer more flexibility, and are applied when $\pi(\theta)$ has an unknown form. Returning to the previous example, and following Carlin and Louis (2000, Chapter 3, Section 2.1), we now consider the model

$$\begin{aligned} &Y_i \mid \theta_i \sim f(y_i \mid \theta_i), \quad i = 1, \ldots, n, \\ &\theta_i \sim G(\cdot). \end{aligned}$$

Robbins (1985) developed a non-parametric estimate for θ using empirical frequencies to estimate the marginal probabilities, in the case where $y_i \mid \theta_i \sim \text{Poisson}(\theta_i)$. However, Carlin and Louis (2000, Section 3.2.3) show that this estimator performs poorly, and they prefer the approach which consists in estimating G, before using the method of Robbins.

2.3.4 Markov Chain Monte Carlo

As seen in Section 2.3.1, posterior distributions from hierarchical Bayesian models may be quite complicated to sample from. In simpler nonhierarchical models, especially when conjugate prior distributions are assumed, the sampling process may be done directly (Gelman *et al.*, 2013, chapter 11). For hierarchical Bayesian or other complicated models, one may factor the posterior distribution of interest and sample by parts. For example, one may sample from the marginal posterior distribution of the hyperparameters, and then sample from the conditional posterior distribution of the other parameters given the value generated for the hyperparameters and the data. This procedure is usually simple as it involves sampling from known distributions. Obtaining a sample from a distribution f without sampling directly from it is the goal of Markov chain Monte Carlo methods. As their name suggests, these methods aim at constructing Markov chains, defined in Section 2.3.2, which are ergodic and whose stationary distribution is the posterior distribution of interest. As defined by Robert and

Casella (2005, Chapter7, p. 268) "a Markov chain Monte Carlo method for the simulation of a distribution f is any method producing an ergodic Markov chain $\{X^{(n)}\}$ whose stationary distribution is f". The samples are drawn successively with the sampled draws depending on the last value drawn, which defines a Markov chain. Starting from an initial value $x^{(0)}$, at each t, we draw $x^{(t)}$ from the transition distribution $\pi\left(x^{(t)} \mid x^{(t-1)}\right)$, which depends on the previous draw, and the transition distribution must be constructed so that the Markov chain converges to the stationary distribution, which is the posterior distribution of interest (Gelman *et al.*, 2013, Chapter 11, p. 286). The chain must therefore be run for long enough to generate samples from the target distribution. There exist several algorithms to construct such a Markov chain. We present the two best-known and most-used algorithms in Sections 2.3.5 and 2.3.6.

We must distinguish Markov chain Monte Carlo (MCMC) methods from general Monte Carlo methods to generate independent simulations from a target density. Indeed MCMC methods lead to successive sampled parameters that are not independent (Congdon, 2010, p. 4).

2.3.5 Gibbs sampling

In this section and the next, we present two widely used methods for stochastic simulation using Markov chains. As said in the previous section, we would like to generate draws from the posterior distribution. To this end, we construct a Markov chain whose stationary distribution is the target distribution. We denote the target distribution by $\pi(x)$, with $x = (x_1, \ldots, x_n)$, and we suppose that the full conditional distributions $\pi_i(x_i \mid x_{-i})$, $i = 1, \ldots, n$, and $x_{-i} = \{x_g, g \neq i\}$, are available and easy to sample from. Gibbs sampling was named by Geman and Geman (1984), but the algorithm was known much earlier. It is based on successive draws from the full conditional distributions (Green, 2001, p.15). As a first step, initial values are arbitrarily chosen, $x^{(0)} = \left(x_1^{(0)}, \ldots, x_n^{(0)}\right)$, and at each step of the Gibbs sampler, each element is updated by sampling from the full conditional distributions. At step r of the Gibbs sampler, we generate

$$
\begin{aligned}
x_1^{(r)} &\sim \pi\left(x_1 \mid x_2^{(r-1)}, \ldots, x_n^{(r-1)}\right), \\
x_2^{(r)} &\sim \pi\left(x_2 \mid x_1^{(r)}, x_3^{(r-1)}, \ldots, x_n^{(r-1)}\right), \\
&\vdots \\
x_n^{(r)} &\sim \pi\left(x_n \mid x_1^{(r)}, \ldots x_{n-1}^{(r)}\right).
\end{aligned}
$$

The process is repeated until convergence, i.e., when $x^{(R)} = \left(x_1^{(R)}, \ldots, x_n^{(R)}\right)$ is a sample from the target distribution. In more details, the algorithm constructs a Markov chain, with transition kernel

$$\mathscr{P}(x, y) = \pi\left(y_i \mid x_{-i}\right) I_{\{x_{-i} = y_{-i}\}},$$

where I_A is the indicator function, and equals 1 if condition A is satisfied and 0 otherwise. We can easily verify that detailed balance (2.5) holds for the Gibbs sampler, since

$$\begin{aligned}
\pi(x)\mathscr{P}(x,y) &= \pi(x_i, x_{-i})\pi(y_i \mid x_{-i}) I_{\{x_{-i}=y_{-i}\}} \\
&= \pi(x_i \mid x_{-i})\pi(x_{-i})\pi(y_i \mid x_{-i}) I_{\{x_{-i}=y_{-i}\}} \\
&= \pi(x_i \mid y_{-i})\pi(y_{-i})\pi(y_i \mid y_{-i}) I_{\{x_{-i}=y_{-i}\}} \\
&= \pi(y_i, y_{-i})\mathscr{P}(y,x) \\
&= \pi(y)\mathscr{P}(y,x).
\end{aligned}$$

Parts of the Gibbs sampler can be done several times without changing the convergence of the chain, the effect of multiple updates being the same as the effect of just one (Brooks *et al.*, 2011, Chapter 1, p.25). We can easily check the detailed balance equation (2.5) for going from x to y in two steps, and it generalizes to n steps similarly. We want to show that $\pi(x)\mathscr{P}^2(x,y) = \pi(y)\mathscr{P}^2(y,x)$, by applying the detailed balance equation (2.5) for one step, twice,

$$\begin{aligned}
\pi(x)\mathscr{P}^2(x,y) &= \underbrace{\pi(x)\mathscr{P}(x,\tilde{x})}_{=\pi(\tilde{x})\mathscr{P}(\tilde{x},x)}\mathscr{P}(\tilde{x},y) \\
&= \mathscr{P}(\tilde{x},x)\underbrace{\pi(\tilde{x})\mathscr{P}(\tilde{x},y)}_{=\pi(y)\mathscr{P}(y,\tilde{x})} \\
&= \pi(y)\mathscr{P}^2(y,x),
\end{aligned}$$

we obtain the desired results, for any intermediate state $\tilde{x}$. This remark will be useful when applying our model in Chapter 3. Indeed, part of our model involves a group of highly correlated latent variables. Running this part of the chain more than the rest will help reduce the correlation between the draws, without changing the target distribution $\pi(x)$.

2.3.6 Metropolis–Hastings algorithm

In this Section, we present two algorithms, which consist in proposing a new candidate from a proposal distribution q_i to update the current value, and accept it with some probability $a(x,y)$. The first algorithm, called the Metropolis algorithm, was introduced by Metropolis *et al.* (1953). It consists in constructing a candidate y by drawing y_i from a proposal distribution $q(y_i \mid x)$, parametrized by x, and setting $y_{-i} = x_{-i}$. We write $q_i(y_i \mid x) = q_i(x,y)$ and we require symmetry, $q(x,y) = q(y,x)$. The proposal is accepted, and therefore x_i is updated, with probability

$$a(x,y) = \min\left\{1, \frac{\pi(y)}{\pi(x)}\right\} = \min\left\{1, \frac{\pi(y_i \mid x_{-i})}{\pi(x_i \mid x_{-i})}\right\}.$$

The Metropolis algorithm was generalized by Hastings (1970) to the case where q_i is not

symmetric. The acceptance probability is

$$a(x,y) = \min\left\{1, \frac{\pi(y)q_i(y,x)}{\pi(x)q_i(x,y)}\right\} = \min\left\{1, \frac{\pi(y_i \mid y_{-i})q_i(x_i \mid y)}{\pi(x_i \mid x_{-i})q_i(y_i \mid x)}\right\}.$$

The Metropolis–Hastings algorithm constructs a Markov chain with kernel

$$\mathscr{P}(x,y) = q_i(y_i \mid x)a(x,y) + I_{\{y=x\}}r(x), \quad r(x) = 1 - \int q_i(y \mid x)a(x,y)\,dy,$$

where $r(x)$ is the probability of rejecting the proposal.

Detailed balance (2.5) for the Metropolis–Hastings algorithm is also true, since

$$\begin{aligned}
\pi(x)\mathscr{P}(x,y) &= \pi(x)\left[q_i(y_i \mid x)a(x,y) + I_{\{y=x\}}r(x)\right] \\
&= \pi(x)q_i(y_i \mid x)\min\left\{1, \frac{\pi(y)q_i(x \mid y)}{\pi(x)q_i(y \mid x)}\right\} + \pi(x)r(x)I_{\{y=x\}} \\
&= \min\left\{\pi(x)q_i(y \mid x), \pi(y)q_i(x \mid y)\right\} + \pi(y)r(y)I_{\{y=x\}} \\
&= \pi(y)q_i(x \mid y)\min\left\{1, \frac{\pi(x)q_i(y \mid x)}{\pi(y)q_i(x \mid y)}\right\} + \pi(y)r(y)I_{\{y=x\}} \\
&= \pi(y)\left[q_i(x \mid y)a(y,x) + r(y)I_{\{y=x\}}\right] \\
&= \pi(y)\mathscr{P}(y,x).
\end{aligned}$$

The Metropolis–Hastings algorithm reduces to the Metropolis algorithm by imposing symmetry on q and to the Gibbs sampler when $q_i(y_i \mid x) = \pi(y_i \mid x_{-i}) = \pi(y_i \mid y_{-i})$, leading to an acceptance probability of 1. The main difference between the Gibbs and Metropolis–Hastings algorithms lies in the fact that for the first, one needs to generate from the full conditionals, whereas for the latter, one simply needs to evaluate the ratio $\pi(y)/\pi(x)$.

Concerning the proposal distribution, there are several possibilities. The best choice usually depends on the context. Here, we give some examples (Green, 2001, p.18):

- the independence proposal, whereby $q(y)$ is unrelated to x;
- the random walk, whereby $q_i(x,y) = q_i(y_i - x_i)$ is symmetric about 0 and can also be written $y_i = x_i + \epsilon$, with $\epsilon \sim q_i$;
- the random walk on the log scale, when $x_i > 0$, the random walk is applied to $\log(x_i)$.

Finally, the Metropolis–Hastings algorithm needs some tuning, aiming for an acceptance probability rate of about 25% (Roberts *et al.*, 1997; Robert and Casella, 2005, Chapter 7). When a Metropolis–Hastings step is included within a Gibbs framework, it is possible to perform the sampling several times before moving on to sampling from the posterior density of another parameter. The important fact here is that the number of times that we perform the Metropolis–Hastings step should be independent of the rejection or acceptance of the

proposal. We show the detailed balance equation for going from x to y in two steps, which is enough to generalize to n steps. Starting from x there are three solutions for the first move before heading to state y. We either stay at x, and therefore we reject the proposal, say $\tilde{x}$, only moving to y on the second step. We can move to an intermediate proposed state $\tilde{x} \neq y$ and then accept the second move from $\tilde{x}$ to y, or we can directly move to y and reject the proposal the second time. We want to show that $\pi(x)\mathscr{P}^2(x,y) = \pi(y)\mathscr{P}^2(y,x)$, keeping in mind that detailed balance (2.5) is already satisfied for one step.

$$\begin{aligned}
\pi(x)\mathscr{P}^2(x,y) &= \pi(x)\mathscr{P}(x,\tilde{x})\mathscr{P}(\tilde{x},y) + \pi(x)\mathscr{P}(x,x)\mathscr{P}(x,y) + \pi(x)\mathscr{P}(x,y)\mathscr{P}(y,y) \\
&= \pi(\tilde{x})\mathscr{P}(\tilde{x},x)\mathscr{P}(\tilde{x},y) + \pi(y)\mathscr{P}(y,x)\mathscr{P}(x,x) + \pi(y)\mathscr{P}(y,x)\mathscr{P}(y,y) \\
&= \pi(y)\mathscr{P}(y,\tilde{x})\mathscr{P}(\tilde{x},x) + \pi(y)\mathscr{P}(y,x)\mathscr{P}(x,x) + \pi(y)\mathscr{P}(y,y)\mathscr{P}(y,x) \\
&= \pi(y)\mathscr{P}^2(y,x),
\end{aligned}$$

where we apply the detailed balance equation (2.5) for one step each time and for any proposal $\tilde{x}$. This remark will be useful in Section 3.3 for the model under the normal-gamma prior, as one parameter requires a Metropolis–Hastings update, and tends to have high correlation and move slowly if only one step is performed at each stage of the Gibbs sampler. Increasing the number of times we perform this part of the algorithm gives better convergence and reduces correlation.

2.3.7 MCMC in practice

Now that we know how to construct a Markov chain whose stationary distribution is the distribution of interest, we want to assess the convergence of the chain and identify what kind of information is useful for further analyses from the output. Convergence of MCMC algorithms has been widely studied (see Gilks *et al.*, 1996; Robert and Casella, 2005, Chapter 8, and Chapter 12 respectively, for instance). It can be based on rigorous statistical tests or on graphical output. We will follow Brooks *et al.* (2011, Chapter 6) to give some ideas on monitoring convergence and performing inference with MCMC output. One of the simplest ways to see whether the constructed Markov chain has converged is to generate multiple parallel chains, with very different starting values, and see whether they converge to the same distribution. Indeed, as stated in Section 2.3.2, the stationary distribution of the Markov chain does not depend on the starting value. From parallel chains, one can also compare convergence within and between chains by analysing the variance. The idea is to detect when the variance between the different runs is larger than the variance within each chain, which would indicate a lack of convergence. Suppose that I chains were run in parallel and that we have output information about $f = f(x)$. We denote each chain i of length M by f_i, with elements $f_i^{(m)}$ ($i = 1,\ldots,I$; $m = 1,\ldots,M$). The between- and within-sequence variances are defined respectively by (Gilks *et al.*, 1996, p.137)

$$B = \frac{M}{I-1}\sum_{i=1}^{I}\left(\overline{f}_i - \overline{f}\right)^2, \quad W = \frac{1}{I}\sum_{i=1}^{I}\frac{1}{M-1}\sum_{m=1}^{M}\left(f_i^{(m)} - \overline{f}_i\right)^2,$$

where $\overline{f}_i = M^{-1}\sum_{m=1}^{M} f_i^{(m)}$ and $\overline{f} = I^{-1}\sum_{i=1}^{I}\overline{f}_i$. The quantity of interest, $\widehat{R}$, defined by

$$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}(f)}{W}}, \quad \widehat{\text{var}}(f) = \frac{M-1}{M}W + \frac{1}{M}B,$$

is the ratio between the estimates of the variance of f and the within-chain variance. The estimate of the variance of f tends to be an overestimate, and the estimate of the within-chain variance tends to be an underestimate, because the individual chains probably did not explore the entire target distributions, thus reducing the variance. In case of convergence, we therefore expect $\widehat{R}$ to tend to 1. We usually reject convergence for values of $\widehat{R}$ larger than 1.1 or 1.2. Other tests of convergence exist, such as the Geweke, Raftery and Lewis, and Heidelberg and Welch diagnostics, all available in the `coda` package in R (see also Chapter 6 of Brooks *et al.*, 2011).

We often discard the early iterations of the chain, as they may be too strongly influenced by starting values, or not probable under the target distribution. Time series tools are useful to check whether the sequence obtained is a quasi-independent sample from the target distribution. Autocorrelation and partial autocorrelation plots help to check the correlation within the chain, while traceplots give information about convergence. In case of correlation or slow mixing, a common method consists in performing thinning, where one only keeps observations separated by a lag of length k, where the choice of k is based on the autocorrelation plot for example.

Finally, one could also generate data from the model and use MCMC to re-estimate the parameters. The chains should converge to distributions that represent the true distribution of the parameters.

Once the output of the chain is satisfactory, the required quantities can be estimated from the simulations. We suppose we have a chain $\{X^{(R)}\}$ of length R, with appropriate burn-in and thinning already performed. Then we can estimate several quantities from the output chain:

- the expectation $\mathbb{E}_\pi\{f(X)\}$, where f is a real-valued function on the state space, is estimated by the sample average, as justified by the ergodic theorem (Theorem 2.3.8), i.e.,

$$\overline{f}_R = \frac{1}{R}\sum_{r=1}^{R} f\left(x^{(r)}\right);$$

- the posterior variance is estimated by

$$\text{var}_\pi\left(f\right) = \frac{1}{R-1}\sum_{r=1}^{R}\left(f\left(x^{(r)}\right) - \overline{f}_R\right)^2;$$

- the qth quantile ϕ_q can be estimated by the inverse empirical distribution,

$$\widehat{\phi}_q = X_{(r+1)}, \quad \frac{r}{R} \leq q \leq \frac{r+1}{R},$$

where $X_{(1)} < \cdots < X_{(R)}$ are the order statistics of $\{X^{(R)}\}$. Using the quantiles, one can construct posterior credible intervals for the quantity of interest;

- the Monte Carlo standard errors are obtained through

$$\text{var}\left(\overline{f}_R\right) = \frac{1}{R^2} \sum_{t=-R+1}^{R-1} (R-|t|)\gamma_t \approx \frac{1}{R} \sum_{t=-\infty}^{\infty} \gamma_t,$$

where γ_t is the autocovariance, defined as

Definition 2.3.10. The *autocovariance* of lag $k \geq 0$ of the chain $\{f^{(n)}\} = \{f\left(X^{(n)}\right)\}$ is

$$\gamma_k = \text{cov}_\pi\left(f^{(n)}, f^{(n+k)}\right),$$

and the variance is $\text{var}_\pi\left\{f(X)\right\} = \sigma^2 = \gamma_0$.

Several methods are available to estimate $\text{var}\left(\overline{f}_R\right)$, we refer to Green (2001) for more details;

- the marginal density of X_1 can be estimated by one of

$$\sum_{r=1}^{R} \frac{1}{h} K\left(\frac{X_1 - X_1^{(r)}}{h}\right), \quad \frac{1}{R}\sum_{r=1}^{R} \pi\left(X_1 \mid X_{-1}^{(r)}\right),$$

where h is the bandwidth and K is a kernel function.

2.4 Conclusion

Data combination is relatively straightforward when similar studies provide data of the same type. We saw in this section that there exist many data combination methods in this case, depending on the type of data available. Meta-analysis of gene expression microarray data requires particular attention in order for studies to be comparable. To perform meta-analysis, one can combine summary statistics (p-values, z-scores), or perform list aggregation, using frequentist or Bayesian methods. Bayesian and empirical Bayesian methods are the most flexible ways to incorporate several levels of variation. Hierarchical Bayesian models provide a simple way to model complex data. Moreover the fitting of such models is made easier with the use of MCMC methods, consisting in sampling from full conditional distributions, which are usually easier to obtain and to sample from.

When the set of studies is heterogeneous, one needs to transform the study results to meet the requirements for combination. As study results need to be of the same type in order

to be combined, the best solution usually consists in transforming everything to the least informative support, which does not fully exploit all the information at hand.

In the next chapter, we present a model to combine datasets of different types without requiring all the data to be transformed before combination. To this end, we develop a hierarchical Bayesian model for differential expression of a gene in each study. The parameters of each study are assumed to be samples from a common underlying distribution, whose mean parameter aids in detecting differentially expressed genes across all studies.

3 Hierarchical Bayesian modeling for incomplete microarray studies

We saw in Section 1.4 that finding studies that provide raw data for meta-analysis is not an easy task. We then saw in Section 2.1 that up to now, there does not exist a meta-analysis method to combine heterogeneous types of data. Existing methods either select only those studies providing the same type of information, discarding possibly many others, or they transform all studies to the least informative support, usually ranks, not benefiting from all the extra information at hand. The purpose of this thesis is to combine results from studies providing different levels of information, which are described in Section 3.1. We describe a hierarchical Bayesian model, under which each type of data is modeled using a single parameter to measure the effect of each gene. A prior on this parameter is then chosen to aid the detection of differentially expressed genes.

3.1 Data types

We aim to detect genes differentially expressed between two groups of samples, say cancer and normal for concreteness, among L studies. We assume that each gene g included in the analysis has a parameter γ_g representing its differential expression, but that γ_g is not directly observed; instead, a noisy realization, $\beta_g^{(l)}$ influences study l ($l = 1, \ldots, L$), where $\beta_g^{(l)}$ follows a normal distribution centered at γ_g,

$$\beta_g^{(l)} \sim \mathcal{N}(\gamma_g, \sigma^2_{\beta}), \quad l = 1, \ldots, L. \tag{3.1}$$

We discuss some possible priors for the parameter γ_g in Section 3.3, but first, we describe the four types of data that we consider:

1. *Raw data*: the most complete data one can obtain from a study, usually consisting of a matrix of gene expression values along with some clinical information about the study patients. If values are missing for technical reasons, we consider them to be missing completely at random, as the missing values depend neither on the observed nor the missing data. The patients are assumed to be independent.

2. *Full lists of z-scores or other statistics*: a list recording a value of a statistic for each gene. The statistic can be any result of a statistical analysis based on a gene expression matrix, most often a t-statistic or a p-value. Without loss of generality, we assume that the values are standardized and can be therefore considered to be z-scores.

3. *Partial z-scores*: the value of some statistic for each of the top k genes. Here missingness is not at random, as only the most significant genes are observed. As for Type 2, we assume that the values can be transformed to z-scores.

4. *Partial list of ranks*: a list, often incomplete, of the "best" genes detected by a statistical analysis based on a gene expression matrix. This list usually gives the top k most significant genes, with k usually between 20 and 100.

We develop a hierarchical model that can handle each of the previously defined data types. In the development of the model, we consider one study of each type to simplify the representation. However, note that we can model any reasonable number of studies of each data type.

A Type 1 study consists of a $G \times N$ data matrix Y, containing the gene expressions for the G genes, for n_1 cancer samples and $N - n_1$ normal control samples. We suppose that the first n_1 columns of the matrix contain the cancer samples. The intensities for each gene are denoted by Y_{gj} and are modeled using a parameter μ_g representing the baseline mean and a parameter $\beta_g^{(1)}$ which represents the differential expression of gene g in the Type 1 study,

$$Y_{gj} \sim \mathcal{N}\left(\mu_g + \beta_g^{(1)} I_{\{j \le n_1\}}, \sigma_g^2\right), \quad \sigma_g^{-2} \sim \text{Gamma}\,(b_1, b_2), \quad g = 1, \ldots, G,\ j = 1, \ldots, N.$$

Type 2 provides a vector of z-scores, $Z = (Z_1, \ldots, Z_G)$, which results from performing gene-by-gene two-sample t-tests, or other similar tests, on a full data matrix, in order to compare the gene expression of cancer and control samples. In what follows the calculations are the same for each gene, so we omit the index g. The setting is the same as for the full data, i.e., a full data matrix Y of size $G \times N$, containing the gene intensities for the G genes and the n_1 cancer and $n_2 = N - n_1$ control samples, is available. Let, $\overline{Y}_1$, S_1^2 and $\overline{Y}_2$, S_2^2 denote the sample mean and sample variance of the cancer and control groups, respectively,

$$\overline{Y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} Y_j, \quad S_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} \left(Y_j - \overline{Y}_i\right)^2, \quad i = 1, 2.$$

We can easily find the distribution of the sample variance,

$$\frac{n_i - 1}{\sigma^2} S_i^2 \sim \chi^2_{n_i - 1}, \quad i = 1, 2,$$

where σ^2 is the variance of a gene in a Type 1 study. The two-sample t-statistic is given by

$$T_{\text{obs}} = \frac{\overline{Y}_1 - \overline{Y}_2}{\sqrt{\left(\frac{1}{n_1} + \frac{1}{N-n_1}\right) S_p^2}}, \quad S_p^2 = \frac{(n_1-1)S_1^2 + (N-n_1-1)S_2^2}{N-2}.$$

Defining $X = (N-2)S_p^2\delta^2$, we note that $X \sim \chi^2_{N-2}$, with $\delta^2 = \sigma^{-2}$ and $\nu = (N-n_1)n_1/N$. We first notice that, if N is large enough, $X/(N-2) \approx 1$, and then we obtain

$$T_{\text{obs}} \sim \beta^{(2)}\sqrt{\nu\delta^2} + t_{N-2} \doteq \mathcal{N}\left(\beta^{(2)}\sqrt{\nu\delta^2}, 1\right),$$

where $\beta^{(2)}$ represents differential expression of a gene in a Type 2 study. The z-statistic for a gene g in the Type 2 study can thus be modeled as

$$Z_g \mid \beta_g^{(2)}, \delta_g^2 \sim \mathcal{N}\left(\beta_g^{(2)}\sqrt{\delta_g^2\nu}, 1\right), \quad \delta_g^2 \sim \text{Gamma}\left(f_1, f_2\right).$$

Notice that we use a two sample t-test, which assumes equality of the variances of the two groups, though this is untrue in real datasets. However, this assumption greatly simplifies the calculations, allowing the derivation of a closed form prior, and hence simplifies the expression for the posterior distribution. As this assumption is only made for the prior, we believe it will only have a minor impact on the posterior results because information from the data should counterweight this assumption, and using a more complicated statistic would not make a huge difference. Usually, assumptions on variances are of secondary importance compared to those on means.

A Type 3 study consists of a partial lists of z-scores, for which only the k largest absolute values are observed. For observed genes, we use the same model as for the Type 2 data, and we impute data for missing genes, using information from the other studies. The z-scores decompose into $Z = (Z^{\text{o}}, Z^{\text{m}})$, for the observed genes and for the missing ones. We know that the observed Z^{o} are the largest k in absolute value of the entire vector. Let $|z_{\text{ref}}|$ be the smallest observed score. Then, for any missing gene g belonging to the set $\{Z^{\text{m}}\}$, we have $-|z_{\text{ref}}| < Z_g < |z_{\text{ref}}|$, and

$$Z_g \mid \beta_g^{(3)}, \delta_g \sim \mathcal{N}_{-|z_{\text{ref}}|}^{|z_{\text{ref}}|}\left(\beta_g^{(3)}\sqrt{\delta_g^2\nu}, 1\right), \quad \delta_g^2 \sim \text{Gamma}(f_1, f_2),$$

where $\mathcal{N}_a^b$ is the truncated normal distribution on the interval $[a, b]$, and $\beta_g^{(3)}$ is the differential expression parameter for a gene g in the Type 3 study.

A Type 4 study consists of a partial list of ranks. For each gene g, a rank R_g is attributed by sorting some unobserved statistics, such as z-scores. We introduce a latent variable u_g to model the summary statistic that leads to R_g, i.e., $R_g = \text{rank}(|u_g|)$. Using the parameter $\beta_g^{(4)}$ to represent the differential expression of gene g in the Type 4 study, we have

$$u_g \sim \mathcal{N}\left(\beta_g^{(4)}\sqrt{\nu}, \sigma_{u,g}^2\right),$$

with $v = (N - n_1)n_1/N$. The variance $\sigma^2_{u,g}$ is gene-dependent, taking into account uncertainty about the methods used to obtain the ranks. Since the ranks are incompletely observed, missing ranks are imputed by borrowing information from the other studies, as done for the Type 3 data. We write $R = (R^{\text{o}}, R^{\text{m}})$, with corresponding $u = (u^{\text{o}}, u^{\text{m}})$, where the superscripts o and m represent observed and missing genes respectively. To impute the missing ranks, we first impute the missing $u_g \in \{u^{\text{m}}\}$. Let R_{ref} be the smallest rank observed and let $|u_{\text{ref}}|$ denote the corresponding latent variable. Any missing gene g would have a higher rank than R_{ref}, if it were observed, or said differently, $|u_g| \leq |u_{\text{ref}}|$. Thus u_g is imputed according to

$$u_g \sim \mathcal{N}^{|u_{\text{ref}}|}_{-|u_{\text{ref}}|}\left(\beta^{(4)}_g \sqrt{v}, \sigma^2_{u,g}\right),$$

and the missing ranks are then imputed by ordering the vector $|u|$.

3.2 Combining all data types

Now that we have modeled each type of data separately we can combine them using relation (3.1), where each differential expression parameter for each gene g in each study l, β^l_g, is assumed to be a noisy realization of a common normal distribution centered at γ_g. The full model combining one study of each type of information is, for gene $g = 1, \ldots, G$, and sample $j = 1, \ldots, N$,

$$\begin{aligned}
Y_{gj} \mid \mu_g, \beta^{(1)}_g, \sigma^2_g &\sim \mathcal{N}\left(\mu_g + \beta^{(1)}_g I_{j \leq n_1}, \sigma^2_g\right), \quad \sigma^{-2}_g \mid b_1, b_2 \sim \text{Gamma}(b_1, b_2), \\
Z^{(l)}_g \mid \beta^{(l)}_g, \delta^2_g &\sim \mathcal{N}\left(\beta^{(l)}_g \sqrt{v_l \delta^2_g}, 1\right), \quad l = 2,3, \quad \delta^2_g \sim \text{Gamma}(f_1, f_2), \\
u_g \mid \beta^{(4)}_g, \sigma^2_{u,g} &\sim \mathcal{N}\left(\beta^{(4)}_g \sqrt{v_4}, \sigma^2_{u,g}\right), \quad \sigma^{-2}_{u,g} \mid d_1, d_2 \sim \text{Gamma}(d_1, d_2), \\
\beta^{(l)}_g \mid \gamma_g, \sigma^2_\beta &\sim \mathcal{N}(\gamma_g, \sigma^2_\beta), \quad l = 1, \ldots, 4, \quad \sigma^{-2}_\beta \mid e_1, e_2 \sim \text{Gamma}(e_1, e_2).
\end{aligned} \tag{3.2}$$

The model is also represented as a directed acyclic graph in Figure 3.1, where the part concerning the Type 3 study, being identical to that for the Type 2 study, was omitted for better readability. Full conditional densities of all parameters of model (3.2), along with detailed calculations, are available in Section A.1 of the Appendix. Most of the posterior densities are conjugate. A summary of the parameters used for the model and their meaning is also presented in Section A.5 of the Appendix.

3.3 Priors for the parameter of interest, γ

The parameter of interest γ_g indicates differential expression of the corresponding gene. A value close to zero indicates no differential expression, while a value far from zero indicates that the gene is differentially expressed between the two groups of samples, among all studies. As this parameter is of central importance, we need to choose its prior carefully. Most of the genes included in the analysis are not significant, and therefore a large proportion of the γ's

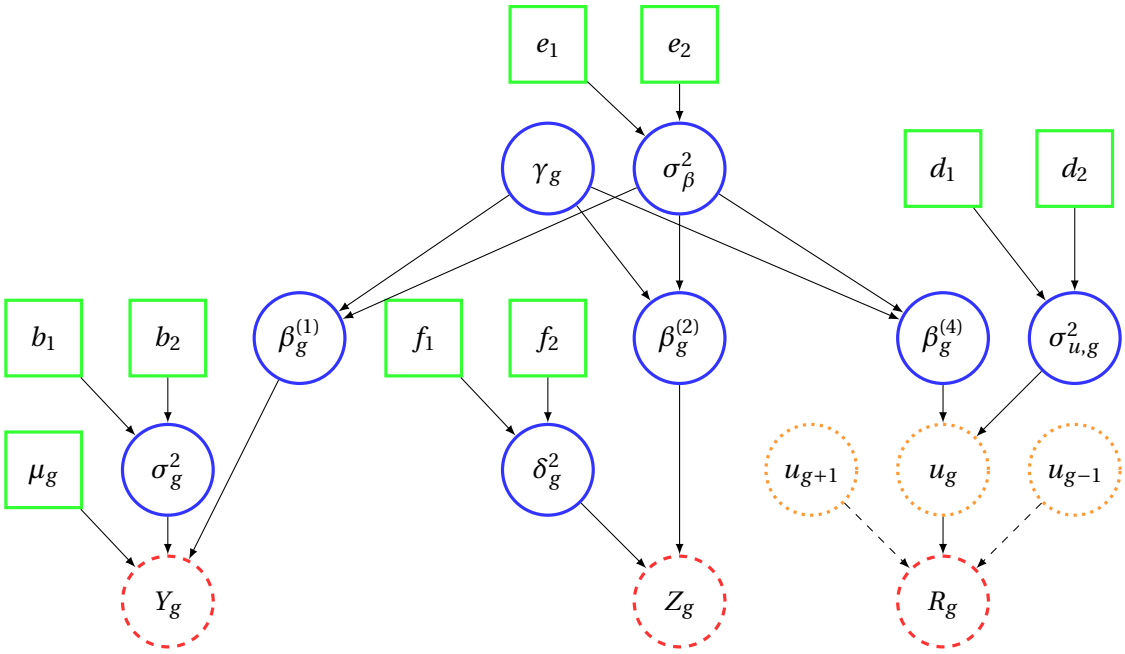
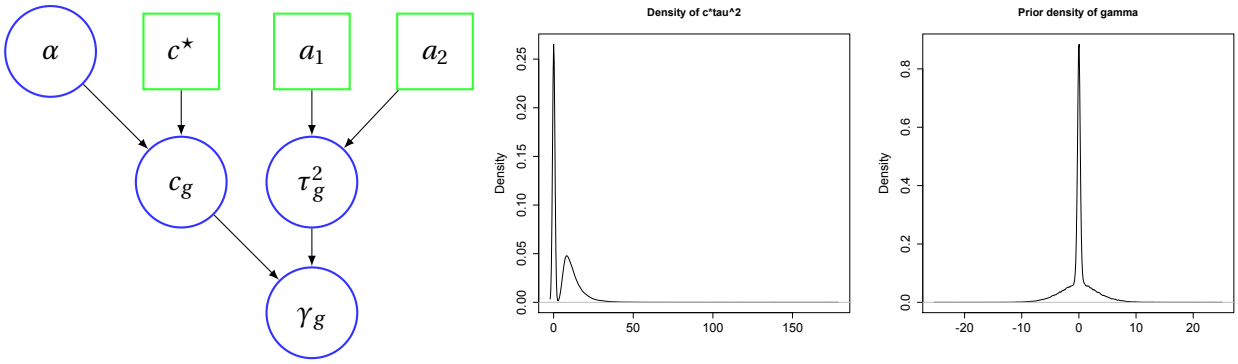


Figure 3.1 – Directed acyclic graph of the hierarchical model representing one study of each of Types 1, 2 and 4. Type 3 studies are omitted as they are modeled and represented as Type 2 studies. Red dashed circles are data; Y denotes a full data matrix (Type 1), Z, a z-score (Type 2 or Type 3) and R, a list of ranks (Type 4). The variables u, in the orange dotted circles, are latent. Blue circles represent parameters, while green squares denote hyperparameters.

should be null. On the other hand, some of the genes are differentially expressed, and in order to be able to detect them easily, the corresponding γ should be large. Three different priors for the parameter γ are considered, all based on the idea of shrinking the parameters of the uninteresting genes towards zero, while leaving the parameters of the differentially expressed genes large enough to be easily detected. The first is the spike and slab prior (Mitchell and Beauchamp, 1988; George and McCulloch, 1993), which has been used in the context of the detection of differentially expressed genes in microarrays by Ishwaran and Rao (2003, 2005), and of multiple hypothesis testing by Pang and Gill (2011). The second is the horseshoe prior, introduced by Carvalho *et al.* (2010), and used in several contexts (Carvalho *et al.*, 2009; Polson and Scott, 2010; Datta and Ghosh, 2012). The third is the normal-gamma prior (Griffin and Brown, 2010). Computations of the posterior densities for each of the models can be found in Section A.1 of the Appendix.

The spike and slab prior (Mitchell and Beauchamp, 1988) uses a mixture of two normal distributions centered at zero, one with very small variance for the non-differentially expressed genes, and the other with possibly large variance for differentially expressed genes,

$$\gamma_g \mid c_g, \tau_g^2 \sim \mathcal{N}\left(0, c_g\tau_g^2\right), \quad \tau_g^2 \mid a_1, a_2 \sim \text{Gamma}(a_1, a_2),$$
$$c_g \mid \alpha \sim (1-\alpha)\delta_{c^\star} + \alpha\delta_1, \quad \alpha \sim \mathcal{U}(0,1),$$



(a) Graphical model (b) Density of the variance of γ_g, $c_g\tau_g^2$ (c) Density of γ

Figure 3.2 – Spike and slab prior. (a): Directed acyclic graph. Blue circles represent parameters, while green squares are hyperparameters. (b): Density of the variance of γ for the spike and slab prior. (c): Density of the mean parameter γ for the spike and slab prior.

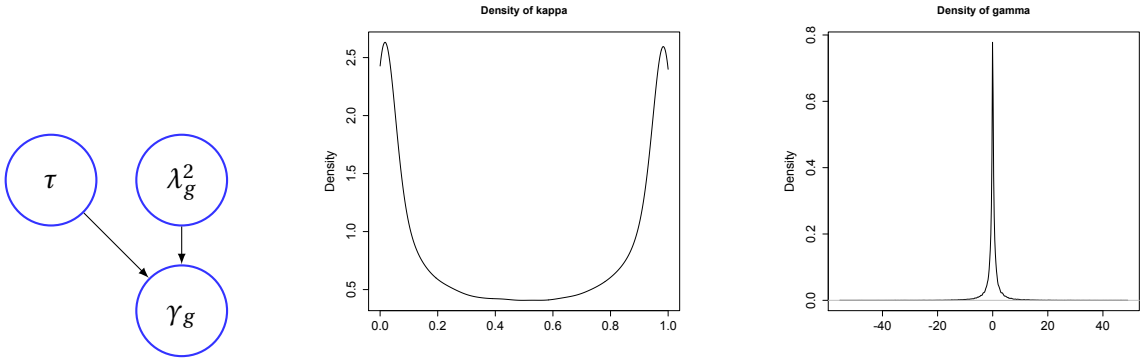
where δ_a is the Dirac function putting all mass at a. The parameter c is binary and can either take the value of the hyperparameter $c^\star = 0.005$, which describes the spike of the distribution, and shrinks most of the γ's to zero; or 1, leading to possibly large values of the variance of γ_g through the parameter τ. A graphical representation of this part of the model is shown in Figure 3.2a. Figure 3.2b presents the bimodal density of the variance of γ_g; it is clear that a lot of weight is attributed to very small values, and a small weight to large values. Figure 3.2c, which presents the prior distribution of γ_g, illustrates the "spike and slab" shape of this density. All posterior densities are obtained by conjugacy of the priors, as detailed in Section A.1.8 of the Appendix.

The horseshoe prior (Carvalho *et al.*, 2010) uses half-Cauchy distributions, C^+, as follows:

$$\gamma_g \mid \lambda_g \sim \mathcal{N}(0, \lambda_g^2\tau^2), \quad \lambda_g \sim C^+(0,1), \quad \tau \sim C^+(0,1).$$

It is represented as a directed acyclic graph in Figure 3.3a. The name of the prior stems from the shape of the distribution of the parameter $\kappa_g = (1+\lambda_g^2\tau^2)^{-1}$, which follows a Beta$(1/2,1/2)$ distribution when $\lambda_g \sim C^+(0,1)$, as illustrated in Figure 3.3b. The global shrinkage parameter τ gives a measure of the underlying sparsity of the data: a small τ indicates large shrinkage. The parameter λ_g is gene-dependent, yielding a highly adaptive prior. From Figure 3.3c, we see that the density of γ_g under the horseshoe has a huge spike at 0, but can also take very large values. The posterior density of γ_g is obtained by conjugacy of the prior. However, sampling from the posterior densities of λ_g and τ is not straightforward and we use the algorithm described in Scott (2011) to avoid a Metropolis–Hastings step (see details in Section A.1.9 of the Appendix).

The normal-gamma prior, introduced by Griffin and Brown (2010) and represented as a



(a) Graphical model (b) Density of κ_g (c) Density of γ_g

Figure 3.3 – Hierarchical model for the horseshoe prior. (a) Directed acyclic graph. (b): Density of the parameter κ_g. (c) Density of the mean parameter γ_g.

directed acyclic graph in Figure 3.4a, is defined as follows:

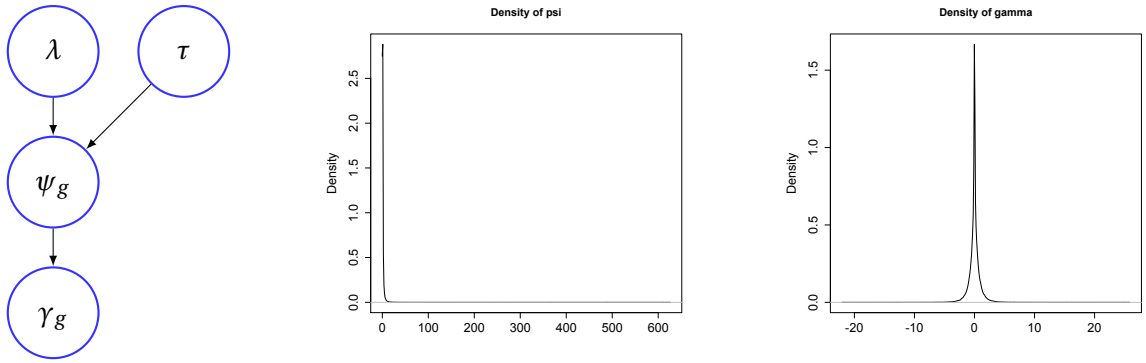
$$\gamma_g \mid \psi_g \sim \mathcal{N}(0, \psi_g), \quad \psi_g \mid \lambda, \tau \sim \text{Gamma}\left(\lambda, \frac{1}{2\tau^2}\right), \quad \lambda \sim \mathcal{E}(1),$$
$$\tau^{-2} \mid \lambda \sim \text{Gamma}\left(2, \frac{M}{2\lambda}\right), \quad M = \frac{1}{p}\sum_{g=1}^{p} \widehat{\gamma}_g^2, \quad \widehat{\gamma}_g = \frac{1}{L}\sum_{i=1}^{L} \beta_g^{(i)}.$$

Large shrinkage is obtained for small values of the parameter λ, whose prior is motivated by the fact that $\lambda = 1$ corresponds to the Bayesian lasso; the exponential prior gives more variability and flexibility around this value. The prior for τ^2 is chosen by noticing that the variance of ψ_g is $v_\psi = 2\lambda\tau^2$, for which we choose an inverse gamma prior. The distribution of ψ, is very concentrated at zero, while also allowing for large values (Figure 3.4b), which leads to a prior distribution for γ as desired (Figure 3.4c). Posterior densities are obtained by conjugacy of the priors for all parameters, except λ, which requires a Metropolis–Hastings step (see Section A.1.10 for detailed calculations).

Figure 3.5 compares the three priors presented previously. While all three achieve the goal of shrinking the non differentially expressed genes to zero, the figure shows how different they are, by comparing the prior density of the parameter γ_g under each of them.

3.4 Finding differentially expressed genes

Once values have been sampled from the posterior densities, we need a cutoff or a posterior quantity from which to decide if a gene is differentially expressed. This problem was discussed in Carvalho *et al.* (2010) and Scott (2009) for the horseshoe prior in the context of linear regression. Their simulations suggest that a simple weight-based thresholding rule leads to strong control of the false positives by automatically penalizing for multiple hypothesis



(a) Graphical model (b) Density of ψ_g (c) Density of γ_g

Figure 3.4 – The normal-gamma prior (a): directed acyclic graph for the normal-gamma prior for γ. (b): Density of the variance of γ_g under the normal-gamma prior. (c): Density of the mean parameter γ under the normal-gamma prior.

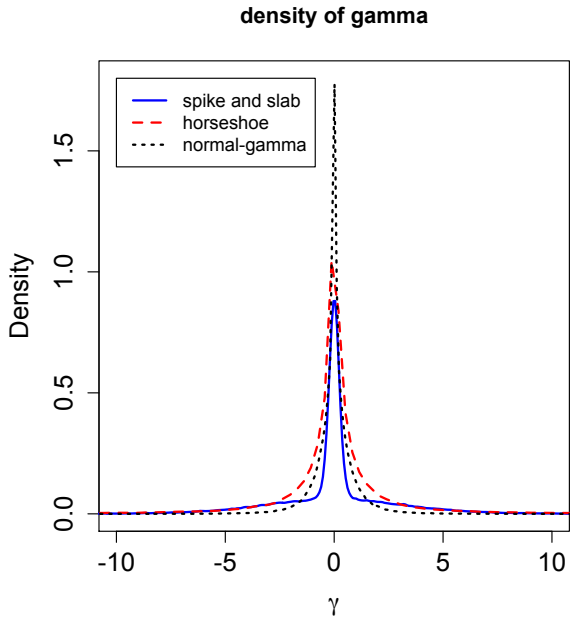


Figure 3.5 – Densities of the three priors considered for the parameter γ: the plain blue line is the spike and slab prior, the dashed red line is the horseshoe prior, and the dotted black line is the normal-gamma prior.

testing (Scott, 2009), while maintaining high power. They make a parallel between the signal selection in the simple two-group discrete model and the horseshoe prior. Suppose we have data $y_i \sim \mathcal{N}(\theta_i, 1)$ and we put a two-group discrete prior on θ_i, $\theta_i \sim (1-w)\delta_0 + w\mathcal{N}(0, \tau^2)$, i.e., θ_i will be non zero with some probability w. Then,

$$\mathbb{E}\left(\theta_i \mid y_i\right) = \mathbb{E}\left(\theta_i \mid \theta_i \neq 0, y_i\right) \mathrm{P}\left(\theta_i \neq 0 \mid y_i\right). \tag{3.3}$$

The quantity $w_i = \mathrm{P}\left(\theta_i \neq 0 \mid y_i\right)$ is the posterior probability of inclusion. If instead of choosing a discrete prior for θ_i, we choose a horseshoe prior,

$$y_i \mid \theta_i \sim \mathcal{N}(\theta_i, 1), \quad \theta_i \mid \lambda_i, \tau \sim \mathcal{N}\left(0, \lambda_i^2\tau^2\right), \quad \lambda_i \sim C^+(0,1),$$

and denoting $\kappa_i = \left(1+\lambda_i^2\tau^2\right)^{-1}$, we can obtain, similarly to (3.3), a link between the posterior mean and the data,

$$\mathbb{E}\left(\theta_i \mid y_i, \kappa_i\right) = (1-\kappa_i)\, y_i.$$

The quantity $w_i = (1-\kappa_i)$, even if not formally a probability of inclusion, can be interpreted as such. The shrinkage parameter, κ_i, gives an indication of how much observation i was shrunk to 0. Indeed, $\kappa_i \approx 0$ indicates almost no shrinkage and identifies signals whereas $\kappa_i \approx 1$ indicates almost complete shrinkage and identifies noise. Carvalho *et al.* (2010) proposed to call i a signal if the corresponding weight w_i is larger than 0.5. Datta and Ghosh (2012) show that this simple thresholding rule has the same risk as the Bayes oracle, i.e., the decision rule under the discrete mixture model, when using an additive loss, where the total loss is the number of incorrect decisions (0-1 loss function). In our context, we can use these results to construct a thresholding rule for our priors. In each case, our parameter of interest is γ_g, whose posterior mean is

$$\widehat{\gamma}_g = \mathbb{E}\left(\gamma_g \mid \lambda_g^2, \tau^2, \sigma_\beta^2, \beta_g\right) = \frac{\lambda_g^2\tau^2}{\sigma_\beta^2 + L_g\lambda_g^2\tau^2} \sum_{l=1}^{L_g} \beta_g^{(l)} = \left(\frac{L_g\lambda_g^2\tau^2}{\sigma_\beta^2 + L_g\lambda_g^2\tau^2}\right) \frac{1}{L_g} \sum_{l=1}^{L_g} \beta_g^{(l)}.$$

where L_g is the number of studies that contain gene g. The quantity $w_{g,HS} = L_g\lambda_g^2\tau^2 / \left(\sigma_\beta^2 + L_g\lambda_g^2\tau^2\right)$ is a signal to noise variance ratio, $\lambda_g^2\tau^2$ being the prior variance of γ_g and σ_β^2/L_g being the variance of the average signal. The thresholding rule, inspired by Carvalho *et al.* (2010), consists in calling gene g differentially expressed if $w_{g,HS} > 0.5$ and noise otherwise.

This rule can be applied for the other priors, using the appropriate form of the posterior weights, $w_{g,NG} = L_g\psi_g / \left(L_g\psi_g + \sigma_\beta^2\right)$, for the normal-gamma prior, and $w_{g,SAS} = L_g c_g \tau_g^2 / \left(L_g c_g \tau_g^2 + \sigma_\beta^2\right)$, for the spike and slab prior.

For the spike and slab prior, the posterior probability of inclusion is readily available via the parameter c_g, which indicates whether the gene is differentially expressed or not, by calculating $\mathrm{P}(c_g = 1)$. This last quantity can be roughly estimated as the number of times $c_g = 1$ over the total number of replications. Therefore, if the number of realizations in

which $c_g = 1$ is larger than the number of realizations where $c_g = c^\star$, we can call the gene differentially expressed, which is equivalent to taking a threshold of 0.5 for the posterior probability of inclusion.

3.5 Practical considerations

The model presented in the previous sections is fully Bayesian. However, some adjustments are needed for practical or computational reasons. First, the parameter μ_g representing the baseline mean in the Type 1 data, is directly estimated from the data for each gene. Indeed, Type 1 data are usually quite precise and they bring a lot of information, so we assume that this parameter can be quite well estimated by

$$\widehat{\mu}_g = \frac{1}{N - n_1} \sum_{j=n_1+1}^{N} Y_{gj}.$$

This empirical step also reduces the computational cost. The second parameter that is empirically estimated is δ_g^2, which appears in Type 2 and 3 data. Taking $\delta_g^2 \sim \text{Gamma}(f_1, f_2)$ leads to identifiability issues and considerably increases the computational cost, as it requires a Metropolis–Hastings step to sample from the posterior distribution. The parameter δ_g^2 is therefore estimated from the L_1 full data ($L_1 \geq 2$), as the inverse of the average gene variance over all Type 1 studies:

$$\widehat{\delta}_g^2 = \left(\frac{1}{L_1} \sum_{l=1}^{L_1} \widehat{\sigma}_g^{2,(l)} \right)^{-1}.$$

A version of the model, where a single δ^2 was sampled for all genes through a Metropolis–Hastings step, was also tried, but, it appears to be less efficient than the version where δ_g^2 is estimated from the Type 1 studies for each gene, while also increasing the computational cost, which is why we prefer to estimate δ_g^2 from the data.

Values of the hyperparameters for the spike and slab prior (Section 3.3) are taken to be $a_1 = 5$ and $a_2 = 50$ as suggested by Ishwaran and Rao (2003). The values of the other hyperparameters are $b_1 = d_1 = e_1 = 5$ and $b_2 = d_2 = e_2 = 50$. Different values of the hyperparameters were tried but they did not affect the results.

The model was coded in R using C code for the Gibbs sampler part to decrease computational time. We performed 31500 iterations, discarding the 1500 first iterations and using a thinning of 10 to reduce correlation. The part concerning Type 4 data (ranks) is run 40 times more at each step to reduce the correlation between the draws. Indeed, for this particular type of data, each gene depends on its neighbors, making consecutive draws highly correlated. We thus obtain 3000 roughly independent realizations from the posterior density of the parameters. Computational time for $G = 200$ genes, $N = 50$ samples, is of the order of 900 seconds for the spike and slab prior and the horseshoe prior and 1200 seconds for the normal-gamma prior.

This last prior requires a larger computational time as one of its parameters is updated using a Metropolis–Hastings step.

Running the code without the extra lagging for the ranks does not change the results concerning the variables of interest. However the hidden variables tend to be highly correlated, especially when the number of genes is high, as the number of latent variables is larger in this case. Only looking at the variable of interest, lagging is not required and the computational time is reduced to 45 seconds instead of 900 seconds for $G = 200$. We believe the variables of interest are not affected by the lack of convergence of the latent variables because the magnitude of the parameter $\beta_g^{(4)}$ does not depend on the latent variables but this information rather comes from the other study parameters, through the parameter γ. Indeed, this information is not present in the ranks and therefore not present in the latent variables. The latent variables u influence the order of the $\beta^{(4)}$, so we just require order stability of the latent variable, which seems to appear quite early in our simulations, in order to have proper convergence for the variables of interest.

However, for the calculations presented in this thesis, we chose to obtain roughly independent realizations for the latent variables, i.e., we perform the extra lagging of 40, which increases the computational time.

3.6 Conclusion

We developed a hierarchical Bayesian model to combine different types of microarray data. The detection of differentially expressed genes is aided by choosing an appropriate prior for the parameter of interest, which puts most of its mass at zero, but allowing large values for interesting genes. We also developed a thresholding-based criterion, which can be interpreted almost as a posterior probability of differential expression, to identify genes of interest.

In the next Chapter, we assess the performance of this model through simulation. We compare our model with existing meta-analysis methods in terms of the number of true differentially expressed genes identified, and power. We also show how much we gain from integrating all information, compared to reducing everything to ranks, or combining only full information data. To this end, we develop a simulation design which is intended to mimic true gene expression data.

4 Simulations

4.1 Numerical examples

In this chapter, model (3.2) is applied to simulated microarray data. Several studies are generated according to the simulation design presented in Section 4.1.1. We consider five studies, two of Type 1 and one each of the other types. Each study consists of a set of $G = 200$ genes, of which the first 10 are differentially expressed, and $N = 50$ patients, of which $n_1 = 40$ are cancer samples. The ratio 10/40 for the two groups of patients was chosen to mimic real data. Indeed, it is often difficult to obtain control samples in ovarian cancer, as the procedure is quite invasive (see Chapter 5 for examples of real data). The gene ratio 10/200 was chosen in order to have only a small proportion of genes differentially expressed, as is often the case in real data. One could argue that 200 genes is a very small number compared to what is usually encountered in real data. However, as simulations require to run our MCMC algorithm many times, the number of genes had to be kept small for computational reasons. A summary of the parameters used for simulations with their usual value is available in Table A.5 of the Appendix. We compare the performance of our model using the three priors presented in Section 3.3 with several meta-analysis and rank combination methods in Section 4.2.1, and assess the power of different combination schemes in Section 4.2.2.

4.1.1 Simulation design for microarray data

The simulation design we use comes from DeConde *et al.* (2006), who took the idea from Kooperberg *et al.* (2005). The authors wanted to confirm the results obtained from a real dataset on some simulated data, and their design seemed more realistic than other existing designs. We further modify the design to make it even more realistic, as we will explain later. We consider G genes, of which k are differentially expressed, with N samples belonging to $M = 2$ groups (cancer or normal) and L such studies. Let x_{ijml} denote the expression level of the ith gene for the jth sample belonging to the mth group (either cancer or control) and for study l ($i = 1,\ldots,G$, $j = 1,\ldots,N$, $m = 1,2$ and $l = 1,\ldots,L$). The expression level x_{ijml} is generated by decomposition into a mean expression parameter μ_i, a differential expression parameter δ_{im}

and noise Z_{ijml},

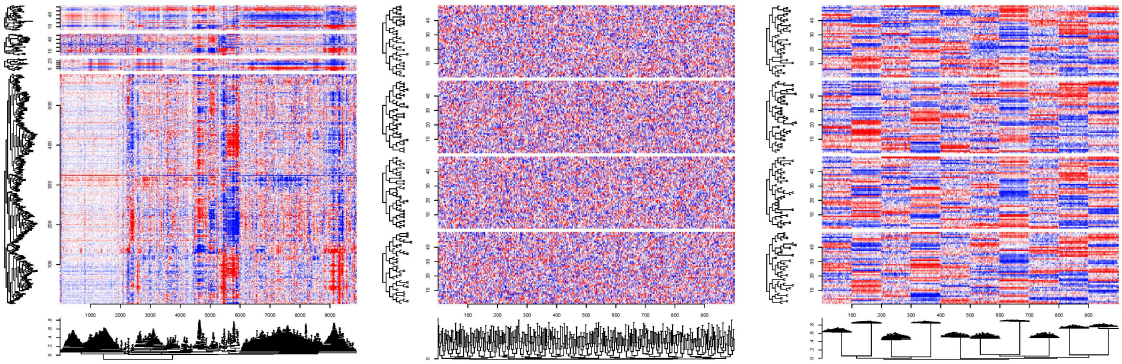
$$\begin{aligned} &x_{ijml} = \mu_i + \delta_{im} + Z_{ijml}, \quad \mu_i \sim \mathcal{U}(0,1), \\ &\delta_{im} = \begin{cases} a(2B_i - 1)G_i, & m = 1,\ i = 1,\ldots,k, \\ 0, & \text{otherwise.} \end{cases} \\ &B_i \sim Be(p), \quad G_i \sim \Gamma(5,1), \quad Z_{jl} \sim \mathcal{N}(0, \Sigma_l), \end{aligned} \tag{4.1}$$

where a is a differential expression parameter, usually chosen equal to 0.5, and Σ_l is a $G \times G$ block matrix with B blocks (in the simulations, we take $B = 1$), diagonal elements σ_{il}^2 and off-diagonal elements $\sigma_i \sigma_j \rho_b$, where $\rho_b \sim \mathcal{U}(0.5,1)$, $b = 1,\ldots,B$. The variance σ_{il}^2 is generated as $\sigma_{il}^2 = (0.3 - 0.02\mu_i)G_{il}$, where $G_{il} \sim \Gamma(5,0.1)$. In DeConde *et al.* (2006), the authors defined $G_{il} \sim \Gamma(5,1)$, which we modified to $G_{il} \sim \Gamma(5,0.1)$ to better mimic real datasets. The variance parameter of a gene thus depends on the mean in such a way that genes with smaller mean expression have a larger variance, as is often encountered in real data (Kooperberg *et al.*, 2005), and the variance varies across studies. We also modify the design from Kooperberg *et al.* (2005) by including a block diagonal covariance matrix. This modification makes the simulation design more realistic by introducing correlated genes. Indeed, even if we model genes independently in model (3.2), the independence assumption is not satisfied for real data, as genes tend to have correlated expression in groups (see Chapter 6 for more details), as may also be seen from Figure 4.1.

The data matrix obtained for study l is:

$$X = \left(\begin{array}{ccc|ccc} \mu_1 + \delta_{1,1} + Z_{1,1,1,l} & \cdots & \mu_1 + \delta_{1,1} + Z_{1,n_1,1,l} & \mu_1 + Z_{1,n_1+1,2,l} & \cdots & \mu_1 + Z_{1,N,2,l} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \mu_k + \delta_{k,1} + Z_{k,1,1,l} & \cdots & \mu_k + \delta_{k,1} + Z_{k,n_1,1,l} & \mu_k + Z_{k,n_1+1,2,l} & \cdots & \mu_k + Z_{k,N,2,l} \\ \hline \mu_{k+1} + Z_{k+1,1,1,l} & \cdots & \mu_{k+1} + Z_{k+1,n_1,1,l} & \mu_{k+1} + Z_{k+1,n_1+1,2,l} & \cdots & \mu_{k+1} + Z_{k+1,N,2,l} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \mu_G + Z_{G,1,1,l} & \cdots & \mu_G + Z_{G,n_1,1,l} & \mu_G + Z_{G,n_1+1,2,l} & \cdots & \mu_G + Z_{G,N,2,l} \end{array}\right).$$

We illustrate the simulation design in the heatmaps of Figure 4.1, where four studies were generated under the design presented previously, with a ten-block diagonal covariance matrix, either with no correlation, or with large correlations ($\rho \sim \mathcal{U}(0.5,1)$). We also show the heatmaps of four real datasets, which will be presented in detail in Chapter 5, and from which we conclude that assuming gene independence does not seem realistic. Our simulation design mimics the behaviour of real microarray data and is more realistic than other simulation designs, which consist in generating from independent normal distributions. To illustrate this, we present some plots in Figure 4.2 that compare a simulated dataset from our design and a real dataset, labelled YOS (Yoshihara *et al.*, 2009, presented in Chapter 5). Figure 4.2 suggests that $a = 0.5$ is a sensible value for the differential expression parameter, if we want to mimic real data. This choice is also motivated by looking at the gene mean and variance for a simulated data and for YOS. Mean gene expression for YOS is around 8, while it is smaller,

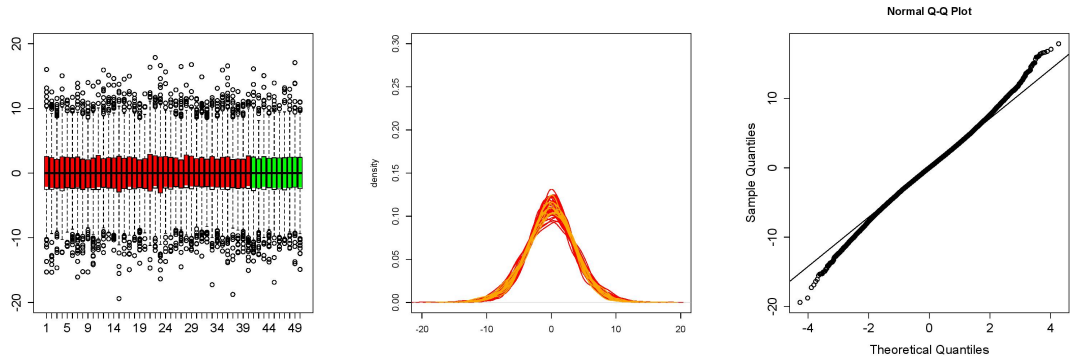


(a) Real datasets (b) without correlation (c) with correlation

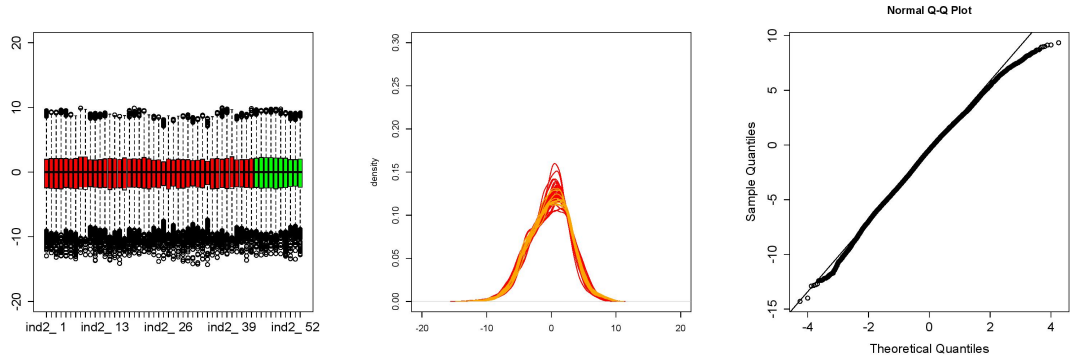
Figure 4.1 – Heatmaps of the four Type 1 real datasets presented in Chapter 5 (left), and four simulated studies under a design with no correlation (center) and with correlation, $\rho \sim \mathcal{U}(0.5, 1)$ (right). The trees on the y axis are hierarchical clustering trees for the individuals, while the tree on the x axis is a hierarchical clustering tree for the genes. We clearly identify the block structure in the right-hand panel.

around 6, for simulated data with $a = 0.5$. Genes are however more variable in our simulated dataset, making differentially expressed genes a little easier to detect than in YOS, which is why we don't increase a further to obtain the same range of mean gene expression values. On the one hand, our simulation design is different from our model in many ways, as it generates dependent genes with variance depending on the mean expression, for example. On the other hand, we assume normality of gene expression, suppose that there are no subgroups other than the two main cancer and control groups, and assume that the same genes are differentially expressed across studies. In that sense, it may be easier to detect differentially expressed genes in our simulated data than in real data. However, we believe that our design does not favour any particular model, so that it should provide a fair way to compare our method with the other meta-analysis and rank aggregation procedures presented in Section 4.2.1. In the next sections, the simulation design can be used as such to obtain a Type 1 study. For Type 2 and 3 studies, a matrix X is generated and z-scores are obtained by applying limma (Smyth, 2005) from the bioconductor package in R. For Type 4 data, after generating another matrix X and applying limma, the absolute values of the z-scores are ordered to obtain the ranks.

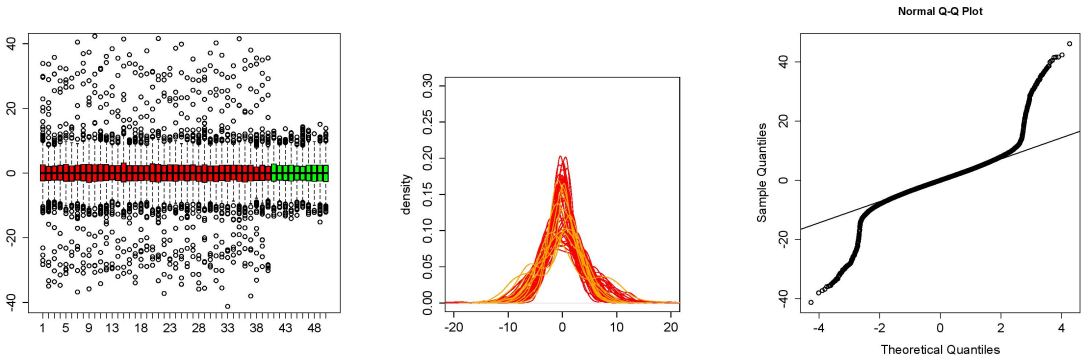
To make the simulations even more realistic, missingness at random is also introduced. In Type 1 studies, some genes are completely missing while some only have missing values for a few patients. In Type 2 studies, missingness is introduced by randomly selecting genes with missing information at random. In Types 3 and 4, missingness is present as only the tops of the lists are observed.



(a) Boxplot, $a = 0.5$ (b) Density, $a = 0.5$ (c) QQ plot, $a = 0.5$

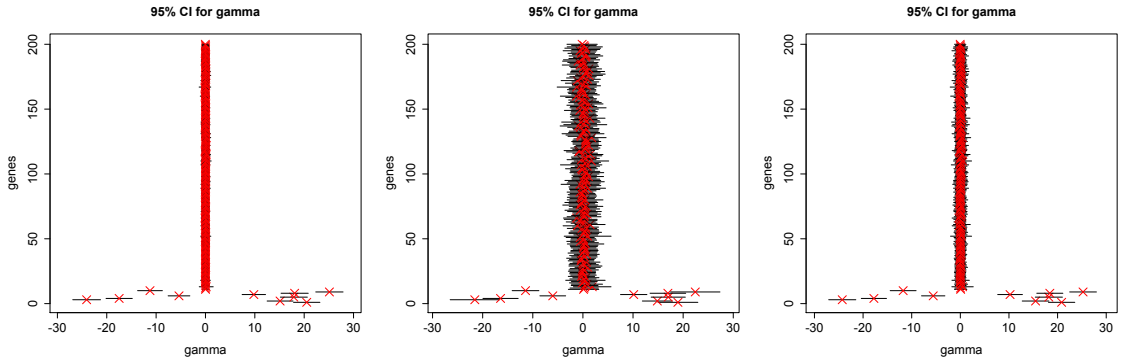


(d) boxplot of YOS (e) density of YOS (f) QQ plot of YOS



(g) boxplot, $a = 5$ (h) density, $a = 5$ (i) QQ plot, $a = 5$

Figure 4.2 – Comparison of a simulated dataset with different values of a and a real dataset. Rows: *Top*: plots for simulated data with $a = 0.5$; *center*: plots for real data; *bottom*: plots for simulated data with $a = 5$. Columns: *Left*: boxplots for each sample, with red indicating cancer samples and green indicating normal controls; *center*: density plot of each sample, where red is for cancer and orange for controls; *Right*: QQ plots of gene expressions for each dataset.



(a) spike and slab (b) horseshoe (c) normal-gamma

Figure 4.3 – 95% posterior credible intervals for the parameter γ for each of the three priors. The first 10 genes (at the bottom of each plot) were set to be differentially expressed with a large value of the differential expression parameter.

4.1.2 Basic simulations for each prior

In this section, we are interested in seeing whether our method, using each of the three priors, can properly detect the 10 first genes as being differentially expressed, when the differential expression parameter is as large as $a = 2$. We also present the kind of output one can obtain from the model.

In Figure 4.3, the 95% confidence intervals for the parameter γ_g are presented for each of the three priors. If R is the number of iterations recorded for each parameter, then a 95% confidence interval for a parameter β, say, is given by

$$\beta \in \left[\widehat{\beta} \pm z_{0.975}\sqrt{\text{var}(\widehat{\beta})}\right], \quad \widehat{\beta} = \frac{1}{R}\sum_{r=1}^{R}\beta^{(r)}, \quad \text{var}(\widehat{\beta}) = \frac{1}{R-1}\sum_{r=1}^{R}\left(\beta^{(r)} - \widehat{\beta}\right)^2,$$

where $z_{0.975}$ is the 97.5% quantile of the standard normal distribution.

The true differentially expressed genes are clearly correctly identified by all three priors, as expected under such a large value of the differential expression parameter ($a = 2$). The main difference concerns the width of the posterior confidence intervals, indicating that the estimation is less precise when using the horseshoe than the spike and slab or the normal-gamma prior, whose intervals both have about the same width.

In the previous simulations, and in the following, we choose to discard some of the early iterations and perform thinning to reduce correlations between the draws. Figure 4.4 shows diagnostic plots for the parameter γ, before and after adjusting for convergence and correlation. Each time, 31500 iterations are performed. Data are simulated using the simulation design presented in Section 4.1.1, for $a = 0.5$. The model is fitted with the spike and slab prior, but the

same results apply for the other priors. Based on these plots, we chose to discard the 1500 first draws to ensure convergence and to use a thinning of 10 to reduce correlation between the draws. Therefore we obtain 3000 quasi-independent draws for each parameter. In both plots the extra lagging of 40 for the latent variables was already applied.

4.2 Simulations for model assessment

4.2.1 Comparison with other methods

We saw in Section 4.1.2 that the model with any of the priors seems to detect differentially expressed genes quite well when they are easy to detect. We want to assess the performance of the different models in terms of detection of differentially expressed genes, for different degrees of differential expression. We compare the performance of the models with Borda's method (de Borda, 1781), the Markov chain methods MC4 and MCT (DeConde *et al.*, 2006), product and sum of ranks (Chang *et al.*, 2013), which reduce all information to lists of ranks prior to combining (see Section 2.2 for details), and some classical meta-analysis methods for combining p-values— Fisher, Stouffer, minimum p-values (MinP), maximum p-values (MaxP), rth ordered p-values (rOP, with r=0.7) (see Section 2.1.1 for details)— as described in Chang *et al.* (2013) and coded in the R package MetaDE (Wang *et al.*, 2012). We apply Borda's method with the arithmetic mean; the median was also used to combine the scores but it did not change the results. We also considered paired comparisons, using the Bradley and Terry (1952) model, but as its performance was worse than Borda's method, these results are not presented.

For values of the parameter a from 0.2 to 2, we generated 100 datasets using the simulation design presented in Section 4.1.1. For each dataset, we fitted our model using each of the three priors, and applied the other methods just mentioned. We recorded the values of the parameters $\widehat{\gamma}_g$ and $\widehat{w}_g$ for each prior and each gene g, along with scores corresponding to the other methods. Figure 4.5 compares all methods based on the detection of differentially expressed genes; the absolute values of the posterior means of γ are ranked for each of the three priors and the number of true differentially expressed genes present in the top 10 is recorded. The same thing is done for the other methods, using the appropriate score. Our method detects the differentially expressed genes more easily than any other method, no matter which prior is chosen, and for all values of the differential expression parameter a.

It is also interesting to see how the priors perform in terms of false positives. For this analysis, we obtain permutation p-values for the Borda and Markov chain methods based on 1000 random permutations. This step is computationally intensive, especially in very large datasets. The p-values for all the methods were then corrected by the Benjamini and Hochberg (1995) correction to account for multiple testing. Figure 4.6 shows the receiver operating characteristic (ROC) curves, which plot the false positive rate against the true positive rate, for the three priors and the other methods, for different values of a and for 500 simulations. Our models

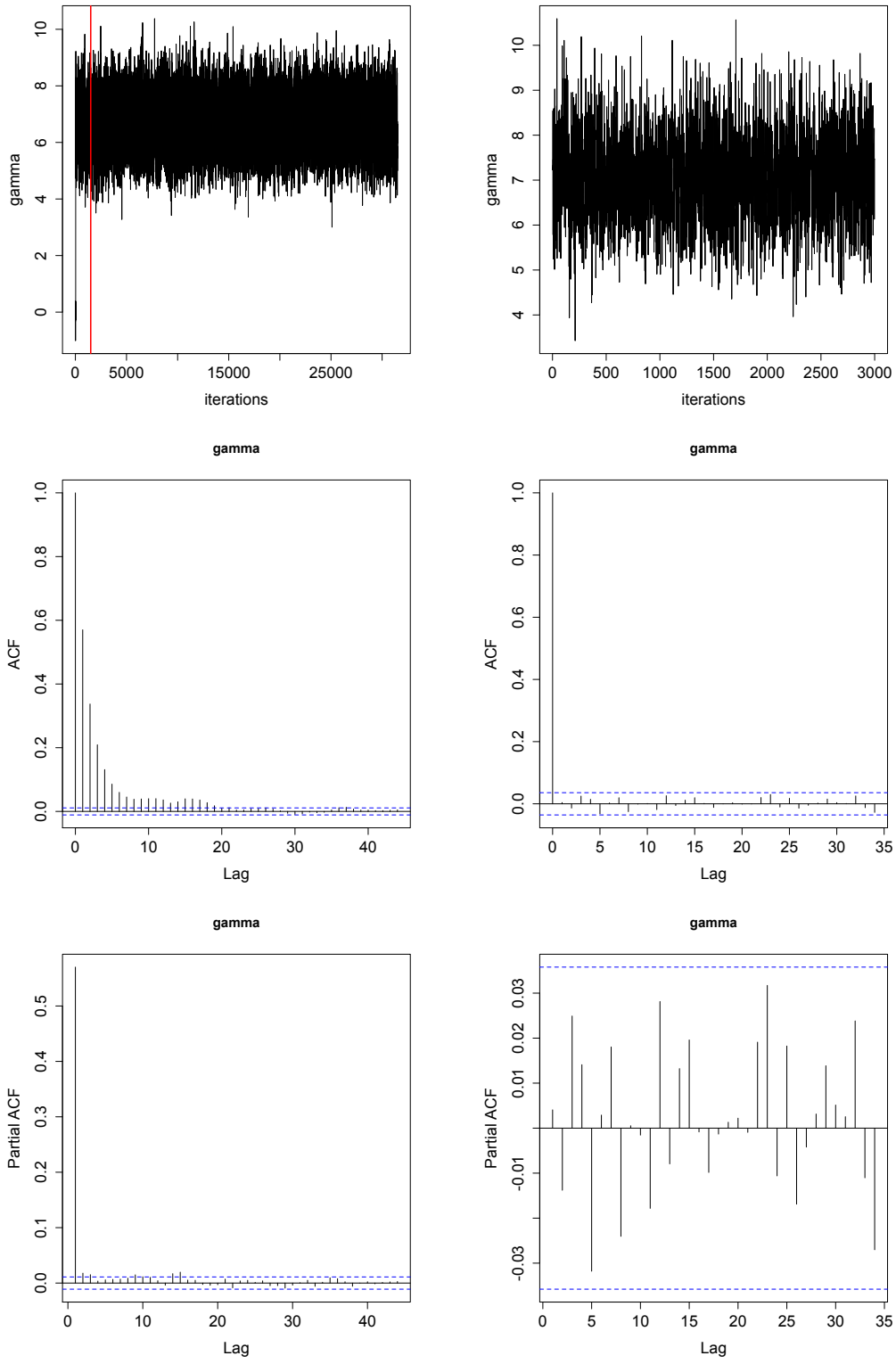


Figure 4.4 – Diagnostic plots (trace plot, ACF and PACF) for one parameter γ in one of the simulations, with $a = 0.5$ for the spike and slab prior. *left column:* 31500 iterations were performed without any thinning or burn-in. *Right column*: 31500 iterations were performed with a burn-in of 1500 and a thinning of 10.

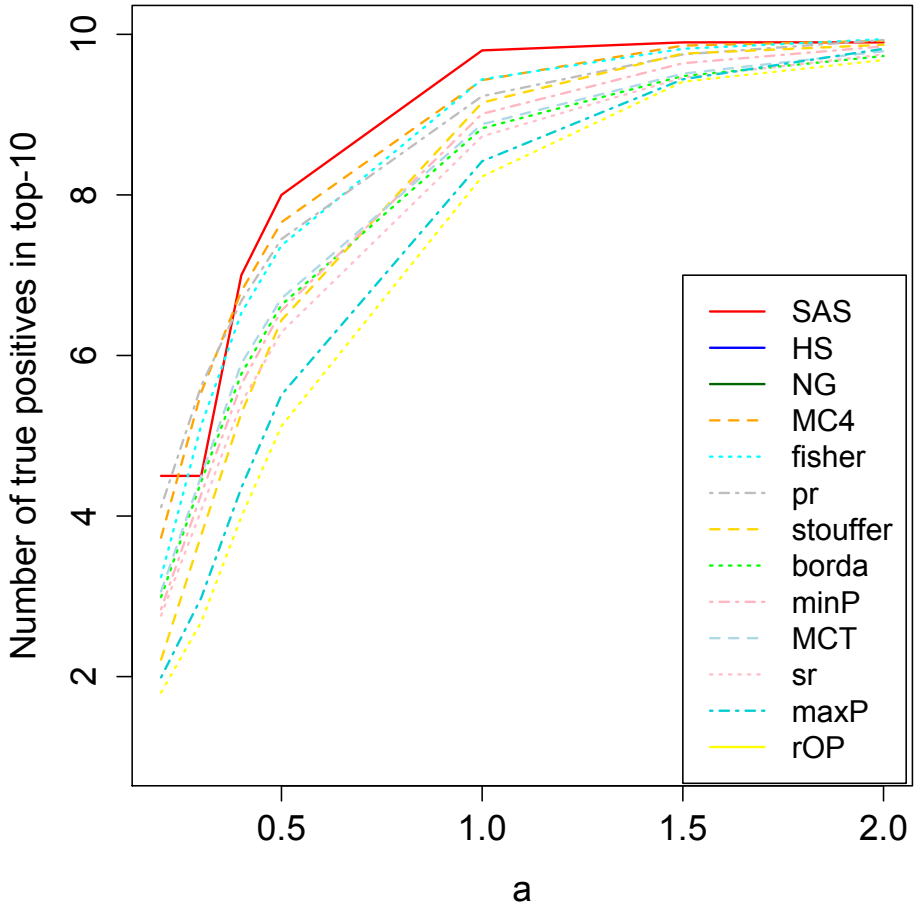


Figure 4.5 – Comparison of the model with the three different priors: horseshoe (HS), spike and slab (SAS) and normal gamma (NG) priors (all superimposed under the plain red curve), and several other meta-analysis and aggregation methods: Fisher, Stouffer, Borda, product of ranks (PR), minimum p-value (minP), sum of ranks (SR), maximum p-values (maxP), rth ordered p-value (rOP with r=0.7), MC4 and MCT which are the two Markov chains methods described in Section 2.2.2. Here we compare the number of true differentially expressed genes in the top 10 genes based on the value of $\widehat{\gamma}$ for our model, or the corresponding scores for other methods. In the simulation design, 10 genes out of 200 were differentially expressed. We therefore provide the number of truly differentially expressed genes to the methods and count how many they can identify by looking at their top 10.

perform well in terms of ROC for values of $a \geq 0.3$, which corresponds to a realistic setting for real microarray data; see Figure 4.2.

All priors detect differentially expressed genes equally well in Figure 4.6, especially for small values of a, and they all perform almost perfectly for larger values. The spike and slab prior seems to be best in each panel of Figure 4.6, closely followed by the normal-gamma prior, particularly when $a = 0.5$. The horseshoe prior performs less well, but it remains competitive and has high power. Since the normal-gamma prior requires a Metropolis–Hastings step to sample one of its parameters, we prefer the spike and slab prior, which performs at least as well, and is much faster to run. All these priors give more power to detect differentially expressed genes than any other methods with which they were compared.

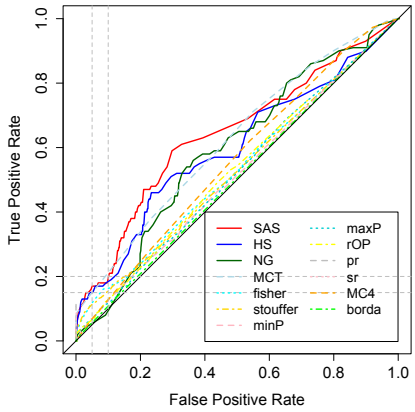
4.2.2 What do we gain from including all studies?

In order to highlight the gain of power from the inclusion of several types of data rather than only considering full raw datasets, we perform 500 simulations, generating two Type 1 datasets, and one dataset of each of the other types, and include the different studies as follows:

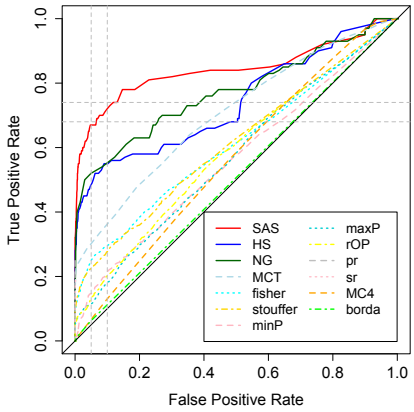
1. only the two Type 1 studies (full raw data);
2. the two Type 1 studies and the full list of z-scores;
3. the two Type 1 studies and the partial list of ranks;
4. the two Type 1 studies and the partial list of z-scores;
5. the two Type 1 studies, the full list of z-scores and the partial list of ranks; and
6. all the studies.

We fit the model using the spike and slab prior to these data and plot the ROC curves for several values of the differential expression parameter a in Figure 4.7. Using all the information available results in a clear increase of power compared to when only Type 1 data are combined.

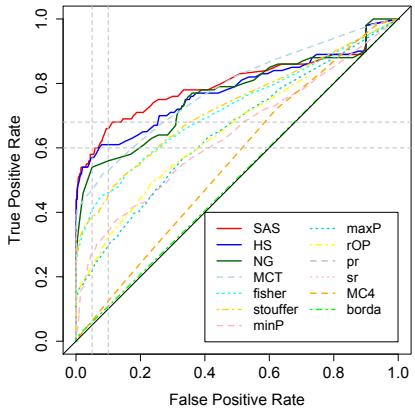
More formally, we test whether the differences between the black (full studies only) and the pink (all data included) ROC curves are significantly different, using a non-parametric test introduced by DeLong *et al.* (1988). It is based on the comparison of the AUCs (Area Under the Curve) corresponding to the ROC curves. The authors notice that estimating the AUC by the trapezoidal rule corresponds to the Mann–Whitney U-statistic. They then use a jacknife estimator for the variance to test the null hypothesis that the two areas are equal. Suppose that we have G genes on which a test was conducted to decide whether they are differentially expressed. We also assume that higher values of the test statistic are associated with disease. In our case the quantity to decide from corresponds to $\widehat{w}_i$ for each gene i averaged over all simulations, as defined in Section 3.4. We know that k genes are truly



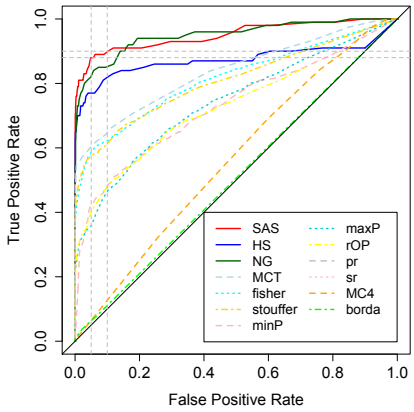
(a) $a = 0.2$



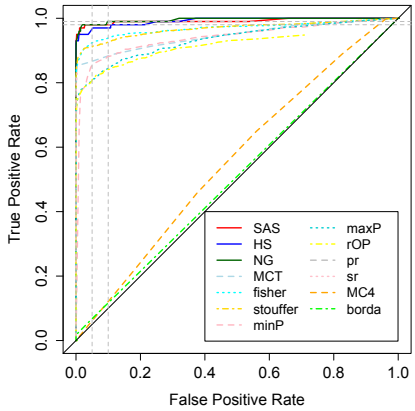
(b) $a = 0.3$



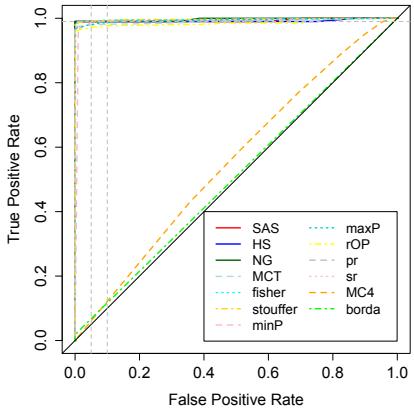
(c) $a = 0.4$



(d) $a = 0.5$

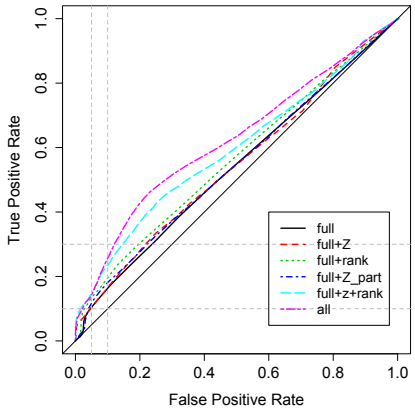


(e) $a = 1$

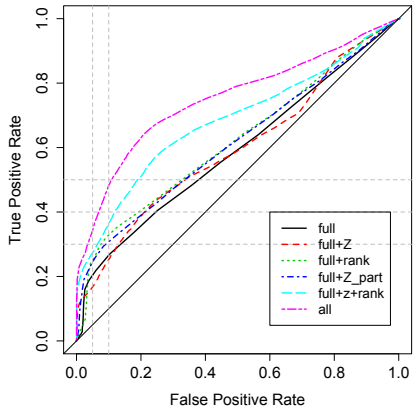


(f) $a = 2$

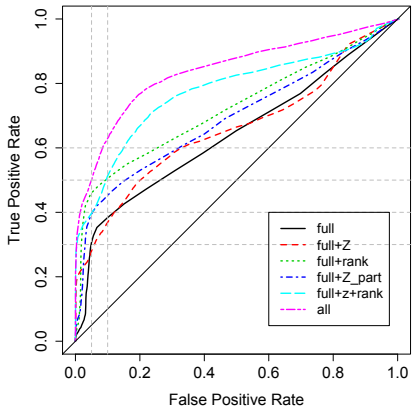
Figure 4.6 – ROC curves for prior comparisons with other meta-analysis and rank aggregation methods. Power of the model for each of the three priors: spike and slab (SAS), horseshoe (HS) and normal-gamma (NG) priors, for several values of the differential expression parameter a along with corresponding power for other meta-analysis methods: Fisher, Stouffer, minimum p-value (minP), maximum p-value (maxP), rth ordered p-value (rOP), product of ranks (PR), sum of ranks (SR), Borda and the two Markov chain methods described in DeConde *et al.* (2006), MC4 and MCT. The vertical dashed grey lines indicate 5% and 10% false positive rate and the horizontal dashed grey lines indicates the corresponding true positive rate for our best method.



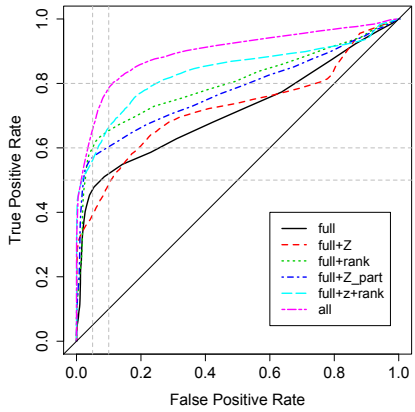
(a) $a = 0.2$



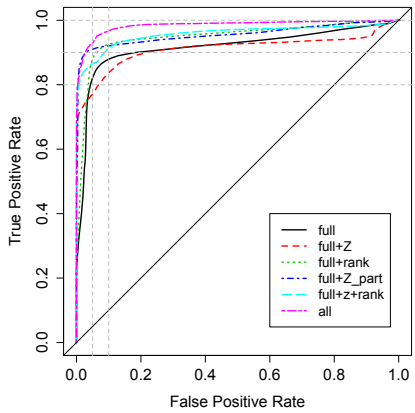
(b) $a = 0.3$



(c) $a = 0.4$



(d) $a = 0.5$



(e) $a = 1$

Figure 4.7 – Comparison of the power of several data combinations among two full studies, a full list of z-scores, a partial list of ranks and a partial list of z-scores. The different combinations are as follow (in the same order as the legend): *black*: only full studies, *red*: the full studies and the full list of z-scores, *green*: the full studies and the partial list of ranks, *blue*: the full studies and the partial list of z-scores, *cyan*: the full studies, the full z-scores and the partial list of ranks, *pink*: all the studies. The ROC curves are plotted for several values of the differential expression parameter a. The vertical dashed grey lines indicate 5% and 10% false positive rate and the horizontal dashed grey lines indicates the corresponding true positive rate for our best method.

differentially expressed while the other $G-k$ are not. The group of differentially expressed genes, C_1 contains the variables X_i, $i = 1,\ldots,k$, of the test variable in this group, whereas the group of non-differentially expressed genes, C_2, contains the variables Y_j, $j = 1,\ldots,G-k$, which denotes the values of the test variable for this other group. In this notation, the ROC curve can be obtained by plotting the sensitivity (the true positive rate) against 1− the specificity (the false positive rate) at different values of a threshold t,

$$\text{sens}(t) = \frac{1}{k}\sum_{i=1}^{k} I_{\{w_i \geq t\}}, \quad \text{spec}(t) = \frac{1}{G-k}\sum_{j=1}^{G-k} I_{\{Y_j < t\}}.$$

The estimated AUC is equal to the Mann–Whitney statistic, which estimates the probability θ that a randomly selected observation from the non-differentially expressed genes will be less than or equal to a randomly selected observation from the differentially expressed population. An estimator of θ can be computed as:

$$\widehat{\theta} = \frac{1}{k(G-k)}\sum_{j=1}^{G-k}\sum_{i=1}^{k}\Psi(X_i, Y_j), \quad \Psi(X,Y) = \begin{cases} 1, & Y < X, \\ 1/2, & Y = X, \\ 0, & Y > X. \end{cases}$$

Suppose we want to compare two ROC curves. We thus have a vector $\widehat{\theta} = (\widehat{\theta}^1, \widehat{\theta}^2)$ representing the areas related to the ROC curves, and the measures $\left\{X_i^r\right\}$, $\left\{Y_j^r\right\}$ $(r = 1,2)$. The elements of the variance covariance matrix are obtained by using jackknife estimator,

$$V_{10}^r(X_i) = \frac{1}{G-k}\sum_{j=1}^{G-k}\Psi(X_i^r, Y_j^r), \quad V_{01}^r(Y_j) = \frac{1}{k}\sum_{i=1}^{k}\Psi(X_i^r, Y_j^r).$$

Then, the 2×2 matrices S_{10} and S_{01} are defined using

$$s_{10}^{rs} = \frac{1}{k-1}\sum_{i=1}^{k}\left[V_{10}^r(X_i) - \widehat{\theta}^r\right]\left[V_{10}^s(X_i) - \widehat{\theta}^s\right], \quad s_{01}^{rs} = \frac{1}{(G-k)-1}\sum_{j=1}^{G-k}\left[V_{01}^r(Y_j) - \widehat{\theta}^r\right]\left[V_{01}^s(Y_j) - \widehat{\theta}^s\right].$$

Finally, the covariance matrix of the parameter estimates $\widehat{\theta}$ is

$$S = \frac{S_{10}}{k} + \frac{S_{01}}{G-k}.$$

Assuming asymptotic normality, a test statistic of the null hypothesis $H_0 : \theta^1 = \theta^2$, can be easily constructed, and is

$$Z = \frac{L\left(\widehat{\theta} - \theta\right)}{\left(LSL^T\right)^{1/2}} \underset{H_0}{\sim} \mathcal{N}(0,1),$$

where $L = (1,-1)$ is the contrast matrix. There exist other tests for comparing two ROC curves, but they require permutation to find the null distribution (Venkatraman, 2000; Bandos *et al.*, 2004; Braun and Alonzo, 2008) or bootstrapping (Pepe *et al.*, 2009), and are computationally

Table 4.1 – Comparison of the area under the ROC curves of Figure 4.7 for different values of the parameter a. We compare the combination of all datasets and the one including only full studies.

		AUC	Z	p-value
a=0.2	design 1	0.54	-13.23	$< 2.2.10^{-16}$
	design 6	0.62		
a=0.3	design1	0.59	-26.36	$< 2.2.10^{-16}$
	design 6	0.75		
a=0.4	design 1	0.64	-33.97	$< 2.2.10^{-16}$
	design 6	0.84		
a=0.5	design 1	0.71	-32.97	$< 2.2.10^{-16}$
	design 6	0.90		
a=1	design 1	0.92	-22.28	$< 2.2.10^{-16}$
	design 6	0.98		

intensive, especially when dealing with high-dimensional data. We therefore apply the test developed by DeLong *et al.* (1988) and implemented in the `pROC` R package (Robin *et al.*, 2011), through the `roc.test` function. Only two ROC curves are compared: the one corresponding to full studies only (design 1, in black in Figure 4.7) and the one corresponding to the analysis with all data included (design 6, in magenta in Figure 4.7).

All tests are strongly rejected, as presented in Table 4.1 for different values of a. This indicates a significant improvement of the power of the method when all studies are included.

4.3 Conclusion

In this chapter, we assessed the performance of our method. By developing a simulation design independent of our model, and which attempts to mimic real microarray gene expression data, we saw that we can detect true differentially expressed genes quite well, no matter which prior is used. Moreover, we attain high power, which suggests that our procedure has a high true positive rate associated with a low false positive rate. We also made comparisons with other meta-analysis methods and show that our methods have more power than any of the methods with which they were compared. The spike and slab prior being the fastest to fit and obtaining slightly better results on simulated data, we apply it in the real data application of the next chapter.

5 Real data application

5.1 Selection of datasets

We now combine results from studies of the four types defined in Section 3.1, in order to find genes differentially expressed between serous ovarian cancer and normal control samples. Electronic search of the online databases PubMed, GEO and ArrayExpress identified 11 such studies. They were selected based on previous work of Jacob *et al.* (2009), for studies before 2007, and from systematic search of online data and articles bases for studies published after 2007. In their paper, Jacob *et al.* (2009) identified 237 studies of ovarian cancer using different -omics technologies. From these studies we extracted those using microarray gene expression data to compare serous ovarian cancer and normal samples. Studies using cell lines were removed, retaining only studies using human tissues. As ovarian cancer is very heterogeneous, we focus our attention on the serous subtype, discarding all other subtypes. For the online search, databases such as GEO, ArrayExpress and Pubmed were screened, looking for studies with the characteristics detailed previously: microarrays, serous ovarian cancer, human tissues and comparison between normal and cancer samples. We first selected 13 studies published before 2007 and described by Jacob *et al.* (2009). Among these, three did not have free access to the articles and were discarded, one did not provide the raw data and the analysis performed in the article was not that of interest, and two provided a gene list in alphabetical order and not ranked according to the differential expression of the genes, which was of little use for our problem. This led to seven studies retained for further analysis, one of which provides a full list of z-scores (Welsh *et al.*, 2001), four provide partial ranks (Warrenfeltz *et al.*, 2004; Meinhold-Heerlein *et al.*, 2007; Zhang *et al.*, 2005; Bignotti *et al.*, 2006), and two provide partial z-scores (Martoglio *et al.*, 2000; Donninger *et al.*, 2004). We will denote these studies by the three first letters of the first author's name: WEL, WAR, MEI, ZHA, BIG, MAR and DON.

We then systematically searched the public repositories GEO, ArrayExpress and PubMed in order to find relevant studies published after 2007. We identified three studies whose raw datasets were all published on GEO (Mok *et al.*, 2009; Yoshihara *et al.*, 2009; Lili *et al.*, 2013), and denoted by the first three letters of the first author, MOK, YOS and LIL. The main difficulty

when looking for microarray data comparing serous ovarian cancer and normal samples is to find studies providing normal samples. Normal ovarian tissues are difficult to obtain, as the surgery is quite invasive and is rarely performed on healthy women. This also explains why in our datasets the number of normal samples is usually lower than the number of cancer cases.

Recently, Ganzfried *et al.* (2013) developed an R package grouping published microarray datasets with available raw data on ovarian cancer. The R package `curatedOvarianData`, provides the gene expression matrices of 21 published studies measuring gene expression of primary ovarian tumors on human tissues. They obtained their data from GEO, ArrayExpress or the supplementary material of publications. We screened their selection of studies and, apart from studies already included in our analysis (MOK and YOS), only one dataset fulfilled the requirements of our analysis, the TCGA dataset (Cancer Genome Atlas Research Network, 2011), which we thus downloaded directly from the package. Information about all the studies is summarized in Table 5.1.

5.2 Preprocessing and normalization

In this section, we present the datasets selected for our analysis. We describe the main preprocessing and normalization steps that were applied in order to have comparable data.

- **MOK** is an Affymetrix dataset with .CEL files available. We performed quality control of the chips using the R package `aroma.affymetrix` (Bengtsson *et al.*, 2008). Some chips were detected with NUSE (Nomalized Unscaled Standard Error) larger than 1.05, but analysis of the data with and without the unusual chips showing no significant differences, and NUSE being really close to 1.05, we decided to keep all chips in the analysis. We normalized the data using RMA (Irizarry *et al.*, 2003), as described in Section 1.2. We matched probes to gene symbols using the probe with the largest mean absolute deviation (MAD) when several probes matched the same gene. This led to a gene expression matrix containing 20 827 unique genes.
- **YOS** is built on an Agilent platform and raw data are not available. Instead we have a gene expression matrix returned by the Agilent Feature extraction software. Following the paper (Yoshihara *et al.*, 2009), we removed intensities smaller than 0.01, treating them as missing values, performed a per chip median normalization and took the $\log_2$ values. Probes were then matched to a unique gene name using the process described for the MOK dataset.
- **LIL**: As .CEL files are available, the procedure is similar to that used for MOK dataset. This dataset also comprises stroma samples which were discarted prior to normalization and analysis. The dataset comprises 21 049 unique genes.
- **TCGA**: This dataset was loaded from the `curatedOvarianData` package, which provides a gene expression matrix already normalized by RMA and where probes were also already matched to the correct gene symbol.

- **WEL**: A full list of z-scores is available as supplementary material from the authors' website, and contains information about 5280 unique genes.
- **WAR**: The analysis described in the paper provides results that were of little use for us. Therefore, we extracted the top 15 list of genes from the paper.
- **MEI**: We extracted the top 43 genes found differentially expressed between serous ovarian cancer and control samples from the paper.
- **ZHA**: They provide intensity ratios for some genes, which we cannot use as such for an analysis. We thus decided to transform the results to ranks, giving a top list of 39 genes.
- **BIG**: The fold changes are available for some genes, but are of little use for us. We used rank information for 117 unique genes.
- **DON**: The authors provide a list of p-values along with a fold change sign, which allows us to transform back to obtain z-scores for the 995 genes published.
- **MAR**: sign information and p-values were transformed into signed z-scores for the top 33 genes published in the paper.

We chose to use gene symbols in our analysis because they are often what is reported in Type 3 and 4 studies, even if we know that using them is not optimal because of the numerous synonyms that may appear. Although dataset selection and preparation only represents a very small written part of this thesis, it was long and tedious and required several months of work.

5.3 Gene selection

As seen in Table 5.1, the studies were conducted on different platforms, so they include very different probes, and therefore genes. Moreover, as some studies only provide partial lists, only a fraction of these genes are visible. The distribution of the genes is presented in Table 5.2, where the number of genes belonging to k studies, $k = 1, \ldots, 11$ is given. We see in the table that no gene appears in all studies, and some genes appear in very few studies.

Since the union of all genes for all the studies includes 27063 unique genes, some gene selection is essential before fitting the hierarchical model. To have a better idea of the proportion of differentially expressed genes in our data, we fit the hierarchical model using the spike and slab prior to full data only (i.e., Types 1 and 2), selecting the 4343 genes common to the five studies. The differential expression proportion is estimated to be $\widehat{\alpha} = 0.009$, which indicates that aiming to reduce the dataset to include more than 2000 genes seems reasonable, as it will preserve many non-differentially expressed genes as well as those of interest, which is needed for the priors to be valid. Gene selection is conducted in two steps, aiming to reduce the gene set to about 5000 genes. The data are separated into full (Types 1 and 2) and partial (Types 3 and 4) groups, and gene selection is performed on these two groups separately. Based

Table 5.1 – Summary of the 11 studies included in the meta-analysis. SOC denotes serous ovarian cancer.

Abbreviation	Study	Data source	Platform	Samples
Type 1: Full data				
MOK	Mok *et al.* (2009)	GEO GSE18520	Affymetrix U133 plus 2.0	53 SOC+10 controls 20827 unique genes
YOS	Yoshihara *et al.* (2009)	GEO GSE12470	Agilent Human 1A	43 SOC+10 controls 16546 unique genes
LIL	Lili *et al.* (2013)	GEO GSE38666	Affymetrix U133 plus 2.0	18 SOC+12 controls 21049 unique genes
TCGA	Cancer Genome Atlas Research Network (2011)	`curatedOvarianData` (R package)	Affymetrix U133A	570 SOC+8 controls 12981 unique genes
Type 2: z-scores				
WEL	Welsh *et al.* (2001)	Website	Affymetrix HuGene FL	27 SOC+4 controls 5280 unique genes
Type 3: partial z-scores				
MAR	Martoglio *et al.* (2000)	Article	Human Genome mapping	4 SOC+5 controls 33 unique genes
DON	Donninger *et al.* (2004)	Supplementary file	Affymetrix U133 Plus 2.0	37SOC+6 controls 995 unique genes
Type 4: partial list of ranks				
WAR	Warrenfeltz *et al.* (2004)	Article	CMT-GAPS slide	31 SOC+5 controls 15 unique genes
MEI	Meinhold-Heerlein *et al.* (2007)	Article	Affymetrix U133A	67 SOC+9 controls 43 unique genes
ZHA	Zhang *et al.* (2005)	Article	united genes holding	4 SOC+13 controls 39 unique genes
BIG	Bignotti *et al.* (2006)	Supplementary file	Affymetrix U133A	19SOC+15 controls 117 unique genes

Table 5.2 – Distribution of the genes by studies for the entire gene set and after gene selection. The table shows the number of genes appearing in k studies, $k = 1, \ldots, 11$, before and after gene selection.

Number of studies	1	2	3	4	5	6	7	≥ 8	Total
Number of genes (before)	7228	4655	4623	5936	4208	400	12	1	27063
Number of genes (after)	778	19	1	0	3940	390	12	1	5141

on the idea that the partial data include only genes that were found interesting at the study level, it is important to include all the genes from these studies, as they provide good potential candidates for differential expression. The union of all genes provided by partial studies comprised 1201 genes. For the set of full data, we selected genes in the intersection (4343 genes). The final set of genes consisted of the union of the genes from the partial studies and the full studies and comprised 5141 genes distributed as in Table 5.2.

5.4 Hierarchical Bayesian model

We fitted the hierarchical Bayesian model using the spike and slab prior to the 11 studies on the reduced gene set, with 31500 iterations performed in order to obtain 3000 roughly independent samples from the posterior distribution. This took 25 days, due to the large number of iterations of the Gibbs sampler and the number of genes. Thinning by a factor 40 was used to reduce the correlation of the hidden variables u_g to an acceptable level, but using the unthinned chain does not affect the output for the parameters of interest and reduces the running time to less than a day. We found a total of 296 differentially expressed genes ($\widehat{w} > 0.5$), presented in Figure 5.1a, 68 having values of $\widehat{w}$, defined in Section 3.4, whose confidence intervals do not contain $w = 0.5$. In Figure 5.1a, we notice a jump in the values of $\widehat{w}$, which is due to the fact that the first 1201 genes were included based on the selection performed on the partial studies (types 3 and 4), and the rest are present or imputed in all studies, as shown in Figure 5.1b. Table 5.3 presents the names of the top 100 genes, the corresponding values of $\widehat{\gamma}$, which gives information about the direction of the differential expression, and $\widehat{w}$. A more complete list of differentially expressed genes is presented in Section A.2 of the Appendix. Bold genes from Table 5.3 are known to be active in ovarian cancer (Jacob *et al.*, 2009, for instance), which shows that our model produces a credible list of differentially expressed genes.

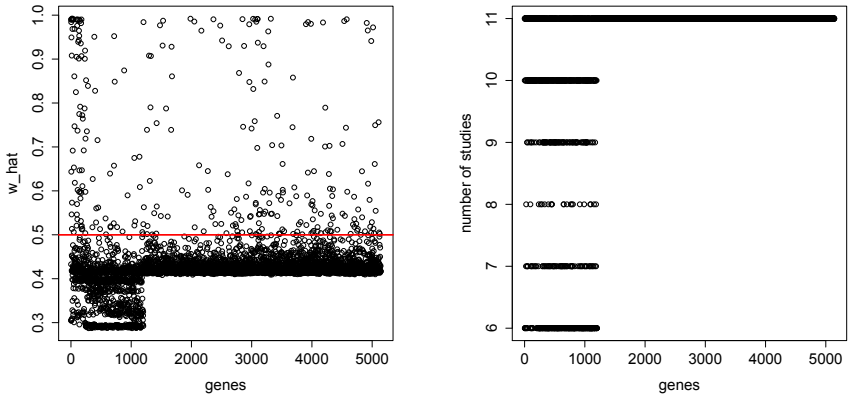
5.4.1 Enrichment analysis

In order to further assess the credibility of the gene list found, we performed a gene set enrichment analysis based on GO terms (The Gene Ontology Consortium, 2008). We started by identifying the GO terms that are associated with the genes selected by our model. We calculated the GO enrichment for our selected gene list, and present the 20 most enriched

Table 5.3 – Top 100 list of differentially expressed genes. Genes are ordered according to the value of $\widehat{w}$, from the most to the least differentially expressed. The estimates $\widehat{w}$ are obtained from the fit of our model to the 11 studies selected for the analysis. The estimate of $\widehat{\gamma}$ is also given for each gene and indicates the direction of differential expression. Bold genes are known to have a role in ovarian cancer.

Rank	Genes	$\widehat{w}$	sd($\widehat{w}$)	$\widehat{\gamma}$	sd($\widehat{\gamma}$)
1	CP	0.99	$< 10^{-2}$	3.07	0.38
2	**TOP2A**	0.99	$< 10^{-2}$	2.60	0.38
3	BCHE	0.99	0.01	-2.49	0.41
4	NEK2	0.99	0.01	2.48	0.38
5	TTK	0.99	0.01	2.48	0.38
6	CENPA	0.99	0.02	2.45	0.38
7	SPP1	0.99	0.02	2.31	0.39
8	**MELK**	0.99	0.02	2.35	0.39
9	PRAME	0.99	0.02	2.41	0.39
10	ADH1B	0.99	0.02	-2.33	0.41
11	KIAA0101	0.99	0.02	2.41	0.40
12	NMU	0.99	0.02	2.30	0.39
13	IGFBP6	0.99	0.03	-2.22	0.39
14	**KLK6**	0.99	0.02	2.24	0.39
15	EVI1	0.99	0.01	2.67	0.45
16	**CLDN3**	0.99	0.02	2.38	0.41
17	SST	0.99	0.02	2.27	0.41
18	**FOLR1**	0.99	0.03	2.22	0.39
19	**WFDC2**	0.99	0.03	2.30	0.40
20	UBE2C	0.99	0.03	2.22	0.38
21	CD24	0.99	0.03	2.47	0.45
22	HMGA2	0.99	0.04	2.21	0.43
23	SPARCL1	0.99	0.04	-2.00	0.38
24	KIF2C	0.99	0.04	2.08	0.40
25	ELF3	0.99	0.05	2.11	0.41
26	**MAL**	0.99	0.05	2.07	0.42
27	CDKN2A	0.98	0.05	2.09	0.41
28	**PAX8**	0.98	0.05	2.05	0.41
29	FOXM1	0.98	0.05	2.05	0.40
30	CENPF	0.98	0.06	2.02	0.40
31	TNNT1	0.98	0.06	2.01	0.42
32	**CLDN4**	0.98	0.06	2.05	0.42
33	HMMR	0.98	0.07	1.99	0.42
34	LCT	0.98	0.07	1.94	0.41
35	KIF11	0.98	0.07	1.92	0.41
36	TRIM31	0.98	0.08	1.92	0.43
37	CDC20	0.98	0.08	1.88	0.42
38	**TACSTD1**	0.98	0.08	2.61	0.63
39	**CCNB1**	0.97	0.09	1.87	0.44
40	PTTG1	0.97	0.10	1.93	0.48
41	PRSS8	0.97	0.10	1.87	0.46
42	CCNE1	0.97	0.11	1.80	0.45
43	RRM2	0.96	0.11	1.84	0.47
44	EHF	0.96	0.11	1.90	0.48
45	**S100A1**	0.96	0.11	1.82	0.47
46	ATP6V1B1	0.95	0.13	1.75	0.50
47	KLK7	0.95	0.13	1.72	0.48
48	ABCA8	0.95	0.13	-1.77	0.49
49	ALDH1A1	0.95	0.13	-1.70	0.47
50	SCNN1A	0.94	0.14	1.73	0.51

Rank	Genes	$\widehat{w}$	sd($\widehat{w}$)	$\widehat{\gamma}$	sd($\widehat{\gamma}$)
51	C7	0.94	0.14	-1.69	0.52
52	PDHA2	0.94	0.15	1.70	0.52
53	ANXA8	0.94	0.17	2.22	0.74
54	GLDC	0.94	0.15	1.66	0.49
55	LAMP3	0.94	0.16	1.72	0.54
56	KRT7	0.93	0.16	1.57	0.50
57	CKS2	0.93	0.16	1.61	0.50
58	AOX1	0.93	0.16	-1.63	0.53
59	EZH2	0.93	0.16	1.60	0.49
60	MYH11	0.93	0.16	-1.63	0.51
61	NY-REN-7	0.92	0.20	2.41	0.94
62	CCNA2	0.91	0.18	1.51	0.53
63	TK1	0.91	0.18	1.50	0.52
64	MAD2L1	0.91	0.18	1.51	0.54
65	ACTG2	0.91	0.18	-1.53	0.56
66	SCGB2A1	0.91	0.19	1.65	0.64
67	**MUC1**	0.90	0.19	1.52	0.57
68	SLC2A1	0.90	0.19	1.46	0.53
69	SULT1C2	0.89	0.20	1.47	0.57
70	MGP	0.87	0.21	-1.37	0.55
71	THBD	0.87	0.21	-1.38	0.57
72	IGF2BP3	0.86	0.22	1.36	0.58
73	SPINT2	0.86	0.22	1.36	0.58
74	ZIC1	0.86	0.22	1.38	0.60
75	CGN	0.85	0.24	1.44	0.66
76	RNASE4	0.85	0.22	-1.31	0.57
77	EFEMP1	0.85	0.22	-1.33	0.60
78	APOA1	0.85	0.23	1.31	0.59
79	DXYS155E	0.84	0.28	2.22	1.21
80	TFAP2A	0.83	0.23	1.25	0.57
81	NDP52	0.83	0.29	2.21	1.27
82	FRY	0.82	0.24	-1.25	0.62
83	DEFB1	0.79	0.25	1.12	0.59
84	TYMS	0.79	0.25	1.12	0.56
85	GRPR	0.79	0.25	1.15	0.60
86	MYBL2	0.79	0.25	1.13	0.58
87	MAOB	0.79	0.25	-1.11	0.56
88	CNN1	0.77	0.25	-1.08	0.57
89	CLDN7	0.77	0.26	1.14	0.64
90	BLM	0.77	0.25	1.09	0.59
91	CXCL10	0.76	0.25	1.07	0.61
92	KIF23	0.76	0.25	1.04	0.56
93	KLF4	0.76	0.25	-1.03	0.56
94	CDKN3	0.75	0.26	1.03	0.57
95	HTR3A	0.75	0.25	1.02	0.57
96	EPCAM	0.75	0.27	1.09	0.67
97	GNG11	0.75	0.26	-1.01	0.55
98	KIF14	0.74	0.25	1.00	0.55
99	NDN	0.74	0.25	-0.99	0.54
100	RAD54L	0.74	0.25	0.98	0.55



(a) $\widehat{w}$ (b) number of studies per gene

Figure 5.1 – Results for the real data analysis. *Left*: Posterior mean of w for all genes included in the analysis, with the 0.5 threshold discriminating between differentially expressed and non-differentially expressed genes in red. *Right*: number of studies each gene belongs to.

terms in Table 5.4.

Top entries from Table 5.4 suggest using cell-related GO terms as candidates for significant enrichment. As we study differences between cancer and normal samples, which are very different samples, it seems reasonable to use such terms. We used two reference GO terms "cell cycle" and "cell proliferation". We prepared two lists of genes containing the genes from the two GO terms and their GO offspring and that were also present in our analysis. The two gene lists have length 570 for cell cycle and 854 for cell proliferation. 281 out of our 296 selected genes could be mapped to a gene ID and were included in the analysis. The enrichment for our selected list of DE genes in these two GO terms was significant, with p-values of $1.6e^{-4}$ for each GO term. The number of genes present in our data and in each GO category is presented in Table 5.5.

We also compare the enrichment obtained by our method and by three other methods, MCT, MC4 and Borda. To this end, we used the Molecular Signature database (MSig DB, Liberzon *et al.*, 2011), which provides pathways from several repositories. For our analysis, we kept the pathways from KEGG, Biocarta, Reactome and GO available to download from MSig under the c2 and c5 collections. After downloading all the pathways, we removed those too large and too small, keeping only pathways containing between 5 and 200 genes, as suggested in Tseng *et al.* (2012). We also only selected those having common genes with our gene set. The enrichment analysis included 4024 genes and 2037 pathways. Enrichment of the top k differentially expressed genes in each pathway, with $k = 100, 500, 1500$, was tested using Fisher's exact test, and the r tests with the smallest p-values, or the r most enriched terms, were recorded for each module. Results are presented in Table 5.6, for the top 100 genes

Table 5.4 – Top 20 GO enrichment of the genes selected by our model. p-values are corrected using Bonferroni's correction.

Rank	p-value	Fraction	Ontology	TermName
1	$< 10^{-5}$	0.10	BP	mitotic cell cycle phase transition
2	$< 10^{-5}$	0.03	BP	sister chromatid segregation
3	$< 10^{-5}$	0.15	BP	mitotic cell cycle
4	$< 10^{-5}$	0.07	BP	mitosis
5	$< 10^{-5}$	0.09	BP	cell division
6	$< 10^{-5}$	0.02	BP	mitotic sister chromatid segregation
7	$< 10^{-5}$	0.03	CC	condensed chromosome kinetochore
8	$< 10^{-5}$	0.03	CC	condensed chromosome, centromeric region
9	$< 10^{-5}$	0.03	BP	spindle organization
10	$< 10^{-5}$	0.27	CC	extracellular region
11	$< 10^{-5}$	0.17	BP	cell cycle process
12	$< 10^{-5}$	0.05	BP	digestion
13	$< 10^{-5}$	0.17	CC	extracellular space
14	$< 10^{-5}$	0.03	BP	mitotic spindle checkpoint
15	$< 10^{-5}$	0.02	CC	kinesin complex
16	$< 10^{-5}$	0.04	BP	chromosome segregation
17	$< 10^{-5}$	0.03	BP	metaphase/anaphase transition of mitotic cell cycle
18	$< 10^{-5}$	0.03	BP	regulation of mitotic metaphase/anaphase transition
19	$< 10^{-5}$	0.03	BP	spindle checkpoint
20	$< 10^{-5}$	0.04	BP	regulation of cyclin-dependent protein serine/threonine kinase activity

Table 5.5 – Distribution of the genes from the GO term (in reference) and from the differentially expressed gene list (selected); *Left*: GO term "cell cycle". *Right*: GO term "cell proliferation". Both terms are enriched (p-value=10^{-4}).

		selected	
		FALSE	TRUE
in Reference	FALSE	3905	227
	TRUE	516	54

		selected	
		FALSE	TRUE
in Reference	FALSE	3642	206
	TRUE	779	75

identified by four methods mentioned previously, and for r=25. We see that our method generally has higher or equivalent enrichment than obtained from MCT and MC4. Borda's method on the other hand shows poor enrichment. The most enriched pathway is different for each method, making it difficult to compare the results. We also present the results in Figure 5.2, where we compare the enrichment of the top 100 enriched pathways for all methods and for k = 100, 500 and 1500. In the left part of this figure, the 100 pathways may not be the same for all methods, but in the right part of the figure, we compare the enrichment of the same pathways for all methods. For the latter, we first select pathways in the union of the top 100 enriched pathways for each method. Then we plot the corresponding enrichment of each pathway for each method, which is the p-value on the log10 scale. In Figure 5.2, our method has higher enrichment than competing methods for top 100 and top 1500 genes, but the top 500 genes are slightly less enriched in our method compared to others. If we compare the same pathways across all methods, we notice that our procedure identifies a list of genes which is more or equally enriched than those detected by other methods. Once again, the difference is larger for top 100 and 1500 genes than top 500 genes.

We saw in Figure 4.6 that our model tends to include few false positives compared to other methods. To illustrate this, we studied the number of known housekeeping genes that are not selected by our model. The list of housekeeping genes for human samples was obtained from Eisenberg and Levanon (2013) and comprised 1059 genes that were included in our analysis. Out of these genes, only one was found in our top list of 296 differentially expressed genes, which is strong evidence that our model does not select housekeeping genes. The significance was further assessed by permuting the list 10^6 times and the resulting p-value was 10^{-6}, which means that finding one or less housekeeping gene in a top list at random never happened in any of the permutations performed. As a comparison, other meta-analysis methods were applied to the set of real data, and the number of differentially expressed genes identified by these methods is reported in Table 5.7. We took a threshold of 0.05 on the p-values adjusted by the Benjamini and Hochberg (1995) correction method for multiple testing. All these methods identify many genes and therefore, if real data behave similarly to simulated data, this may imply that these other methods tend to include many false positives.

To our knowledge, there does not exist any list of ovarian cancer markers that we could use to assess whether the 13 genes that were identified by biologists in our differentially expressed gene list comprises a significant enrichment. Moreover, as, to our knowledge, we have included all studies comparing serous ovarian cancer and normal samples, we cannot use another study to validate the results.

5.5 Conclusion

Application of our model to a set of 11 microarray studies looking for genes differentially expressed between serous ovarian cancer and normal control samples identified 296 potentially interesting genes. Among those, 13 were recognized as being known to have a role in

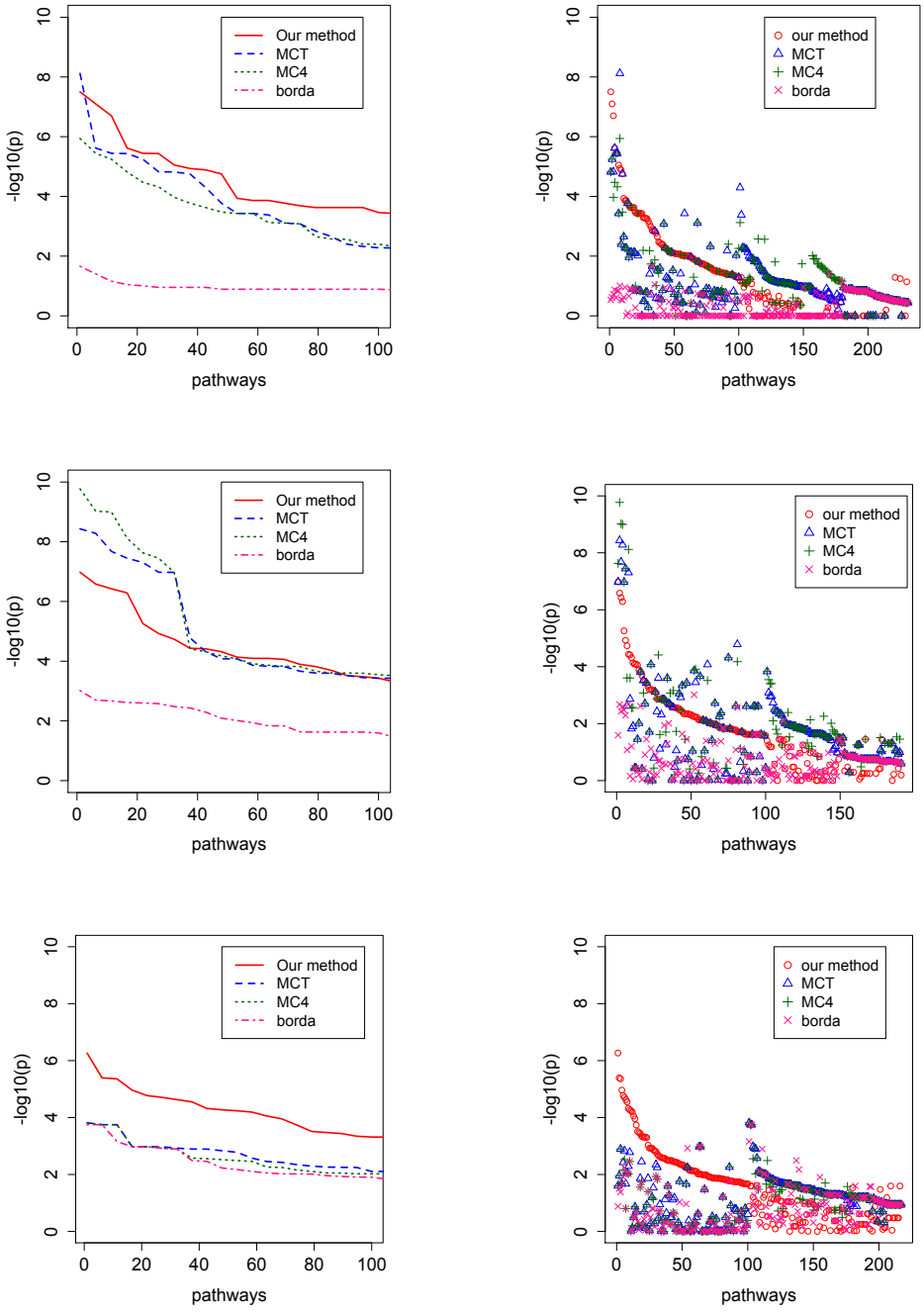


Figure 5.2 – Comparisons of the enrichments obtained for our method, MCT, MC4 and Borda. The left column shows the enrichment of the first 100 most enriched pathways for each method. The right column represents the enrichment of each method on the same pathways, selected as the union of the top 100 most enriched pathways of each method.

Table 5.6 – 25 most enriched pathways for the top 100 differentially expressed genes for our method (top left), MCT (top right), MC4 (bottom left) and Borda (bottom right).

Pathway	p-value
cell cycle process	3.13e-08
cell cycle phase	7.98e-08
mitotic cell cycle	1.99e-07
mitosis	2.41e-06
M phase	3.61e-06
M phase of mitotic cell cycle	3.67e-06
Reactome apc CDC20 mediated degradation of nek2a	8.89e-06
microtubule cytoskeleton	1.16e-05
regulation of mitosis	1.29e-05
spindle	1.77e-05
microtubule cytoskeleton organization and biogenesis	1.16e-04
kinesin complex	1.37e-04
spindle organization and biogenesis	1.37e-04
KEGG cell cycle	1.68e-04
Reactome regulation of mitotic cell cycle	2.05e-04
Reactome inhibition of the proteolytic activity of apc c required for the onset of anaphase by mitotic spindle checkpoint components	2.36e-04
Reactome apc c cdc20 mediated degradation of cyclin b	2.36e-04
calcium independent cell cell adhesion	2.36e-04
microtubule motor activity	2.36e-04
regulation of cell cycle	3.47e-04
Reactome tight junction interactions	3.73e-04
Reactome kinesins	3.73e-04
tight junction	3.73e-04
mitotic cell cycle checkpoint	3.73e-04

Pathway	p-value
microtubule cytoskeleton	7.60e-09
mitosis	2.41e-06
m phase	3.61e-06
m phase of mitotic cell cycle	3.67e-06
cell cycle phase	5.69e-06
mitotic cell cycle	1.50e-05
cell cycle process	1.52e-05
spindle	1.77e-05
microtubule organizing center	5.10e-05
KEGG cell cycle	1.68e-04
Reactome apc cdc20 mediated degradation of nek2a	3.73e-04
spindle pole	3.73e-04
centrosome	4.17e-04
chromosome segregation	7.76e-04
KEGG oocyte meiosis	8.32e-04
Reactome mitotic prometaphase	1.45e-03
microtubule cytoskeleton organization and biogenesis	2.22e-03
regulation of mitosis	3.98e-03
chromosome	4.74e-03
regulation of protein kinase activity	5.25e-03
Reactome metal ion slc transporters	5.34e-03
Reactome conversion from apc c cdc20 to apc c cdh1 in late anaphase	5.34e-03
kinesin complex	5.34e-03
spindle organization and biogenesis	5.34e-03
regulation of kinase activity	5.90e-03

Pathway	p-value
microtubule cytoskeleton	1.15e-06
m phase	3.61e-06
cell cycle phase	5.69e-06
cell cycle process	1.52e-05
mitosis	3.37e-05
m phase of mitotic cell cycle	4.79e-05
mitotic cell cycle	1.08e-04
KEGG cell cycle	1.68e-04
Reactome apc c cdc20 mediated degradation of cyclin b	2.36e-04
spindle	3.34e-04
Reactome kinesins	3.73e-04
Reactome apc cdc20 mediated degradation of nek2a	3.73e-04
microtubule organizing center	7.53e-04
chromosome segregation	7.76e-04
KEGG oocyte meiosis	8.32e-04
microtubule cytoskeleton organization and biogenesis	2.22e-03
protein c terminus binding	2.67e-03
Biocarta p53 pathway	2.74e-03
KEGG retinol metabolism	3.98e-03
regulation of mitosis	3.98e-03
chromosome	4.74e-03
Reactome metal ion slc transporters	5.34e-03
Reactome conversion from apc c cdc20 to apc c cdh1 in late anaphase	5.34e-03
kinesin complex	5.34e-03
spindle organization and biogenesis	5.34e-03
KEGG homologous recombination	5.52e-03

Pathway	p-value
chromosomepericentric region	2.18e-02
gamete generation	3.94e-02
sexual reproduction	6.92e-02
Reactome mitotic prometaphase	9.00e-02
microtubule cytoskeleton	9.85e-02
Reactome sphingolipid de novo biosynthesis	1.11e-01
Reactome conversion from apc c cdc20 to apc c cdh1 in late anaphase	1.11e-01
kinesin complex	1.11e-01
voltage gated sodium channel activity	1.11e-01
Biocarta lectin pathway	1.29e-01
Reactome inhibition of the proteolytic activity of apc c required for the onset of anaphase by mitotic spindle checkpoint components	1.29e-01
Reactome regulation of complement cascade	1.29e-01
Reactome glycoprotein hormones	1.29e-01
Reactome apc c cdc20 mediated degradation of cyclin b	1.29e-01
Reactome phosphorylation of the apc c	1.29e-01
establishment of organelle localization	1.29e-01
cytokine and chemokine mediated signaling pathway	1.29e-01
sodium channel activity	1.29e-01
microtubule motor activity	1.29e-01
kinase activator activity	1.29e-01
mitosis	1.34e-01
Biocarta ahsp pathway	1.46e-01
Reactome kinesins	1.46e-01
Reactome apc cdc20 mediated degradation of nek2a	1.46e-01

Table 5.7 – Number of differentially expressed genes identified by meta-analysis methods, based on Benjamini and Hochberg (1995) corrected p-values, taking a threshold of 0.05.

Method	Fisher	Stouffer	minP	maxP	Ours
Number of DE genes	3858	3510	3784	1548	296

ovarian cancer by our biologist, Dr. Viola Heinzelmann. Enrichment analysis of our gene set showed strong enrichment in GO terms related to cell functions (cell cycle and cell proliferation). This last result was to be expected, as we are studying two sets of samples that are biologically and genetically very different, so anything related to cells is likely to be enriched. However it confirms that our model does not produce a random list of genes, but a biologically meaningful list. Comparison of pathway enrichment of the top k differentially expressed genes from our method, MCT, MC4 and Borda's method also shows that we tend to detect genes that are equally or more enriched than those of other methods, when $k = 100$ or 1500. Again, as we are comparing such different tissues, we do expect other methods to also find biologically enriched lists of differentially expressed genes. We also studied the robustness of our model regarding false positives, by looking at the "non detection" of housekeeping genes. Our results were highly significant, with only one housekeeping gene present in our top list of differentially expressed genes, which is an evidence in favour of the robustness of our model regarding false positives. This is also supported by comparing the number of differentially expressed genes identified by other methods, which is much larger, indicating that other methods, while finding many true differentially expressed genes, may also identify many false positives. However, without the existence of a gold standard, we cannot know whether we find many false negatives or whether we are robust against false positives. We can only collect evidence to support the latter.

Even though our model seems to produce promising results, the computational time and the necessity for gene selection are two drawbacks that we try to alleviate in the next chapter.

6 Correlation matrix estimation

6.1 Introduction and motivation

One drawback of our model is its large computational requirements. It is infeasible to fit the model to the union of all genes from all studies, so gene selection is essential. Even if our selection of genes is not based on differential expression, which implies it is independent of our model, we might discard important genes. Including only the best candidate genes is unrealistic and is not appropriate for the chosen priors, which require the inclusion of a large proportion of uninteresting genes. The real data example of Chapter 5 performs a gene selection independent of the model, as we only select genes corresponding to unions or intersections of groups of genes. However, it would be interesting to be able to fit the model to the full set of genes.

We also assume that the genes are independent, which is not true, though it is a common assumption in the literature even for single study analysis. Correlations between the genes are included in the simulation design of Section 4.1.1, and do not seem to affect the efficiency of the model, but we believe that our procedure should perform even better if the elements on which it is applied are independent. One could define modules, sets of genes that are correlated or related in some sense, and apply the model on these elements, to solve the gene selection and dependence problems simultaneously. Indeed, using modules results in a huge dimension reduction by grouping genes together, and also results in independence, as genes that do not belong to the same module may be considered to be independent. With the computational time thus reduced, our model would be even more competitive with others.

This idea was applied in Wirapati *et al.* (2008), who used coexpression modules in breast cancer meta-analysis. They selected a few prototype genes having specific roles in the problem under study, and grouped genes based on the correlation of their expression values with the prototypes, thus defining coexpression modules and module scores to be used in a meta-analysis. This method is particularly interesting in the context of microarray data for several reasons:

- it is biologically meaningful—genes do not act alone, but they are usually activated in groups;
- it is statistically interesting—considering only a small group of coexpression modules rather than 10000 genes reduces the number of hypotheses to be tested, thereby increasing the power of the analysis;
- detecting interesting groups of genes is more reliable than identifying genes, since differentially expressed genes can vary greatly from study to study while their common modes of action may not (Ein-Dor *et al.*, 2005, 2006). Therefore there is a better chance of identifying a common differentially expressed module among studies than a particular gene.

Up to now, identification of modules has been based on prior knowledge on specific genes, implying collaboration with biologists on a case-by-case basis. This motivates the development of an automatic prototype identification procedure that would not require prior knowledge and could be applied to any dataset, with the goal of reducing its dimension.

In order to define modules, we base our approach on the correlation matrix. We define modules to be groups of genes with high correlation. In this chapter, we discuss estimation of large covariance matrices. We assume that the covariance matrix is sparse, as probably many genes have no or weak correlation with the others, while only some of them are very correlated, defining modules. Estimating a sparse covariance matrix will aid the detection of modules by highlighting groups of genes that are highly correlated. We first review estimation methods for a single high dimensional covariance matrix in Section 6.2, then present some tests for assessing the equality of two large covariance matrices in Section 6.3. In the early sections of this chapter, we focus on estimating a single covariance matrix. However, in the meta-analysis context, we wish to define modules common to all studies, so in Section 6.4 we investigate an empirical Bayes method for estimating a sparse high dimensional correlation matrix common to several studies. Identification of modules based on clustering and this common correlation matrix is described in Chapter 7.

6.2 Large covariance matrix estimation

When the number of individuals N is much smaller than the number of covariates G, $N \ll G$, the sample covariance matrix given by

$$\Sigma_N = \frac{1}{N-1}\sum_{k=1}^{N}(X_k - \overline{X})(X_k - \overline{X})^T, \quad \overline{X} = \frac{1}{N}\sum_{k=1}^{N} X_k,$$

has rank at most $N-1$, is not a consistent estimator of the population covariance matrix, and is often ill-conditioned. The elements of Σ_n are denoted by $\widehat{\sigma}_{ij}$. Several methods exist to estimate the covariance matrix of a population in the high-dimensional case, many of

which require extra assumptions, such as sparsity, in order to construct a consistent estimator. Bai and Shi (2011) identify four classes of methods for covariance estimation when $G \gg N$ and in the context of asset returns: shrinkage, factor model, Bayesian approach and random matrix theory methods. Pourahmadi (2011) presents a review of several methods to estimate covariance matrices from two perspectives: generalized linear models and regularization. We discuss some of these in the following sections.

6.2.1 Thresholding based methods

In the context of large covariance matrix estimation, thresholding methods are applied to the sample covariance matrix in order to obtain a better estimate. Ledoit and Wolf (2004) were among the first to use this technique. Their proposed estimator is both well-conditioned and more accurate than the sample covariance matrix, with respect to a quadratic loss function. They obtain a well-conditioned estimator by imposing a diagonal structure, which they average with the sample covariance matrix, using appropriate weights, estimated to optimize a loss function. Ledoit and Wolf (2012) state that shrinkage is one of the simplest ways to estimate the covariance matrix, because it is distribution free, and usually has a simple and explicit formula, which is easy to compute and interpret. They extend their previous linear shrinkage estimator to nonlinear transformations of the sample eigenvalues.

Bickel and Levina (2008) developed the universal thresholding method, which fixes a common threshold for all the entries of the matrix and sets to zero any value that is below that pre-determined threshold. Their hard thresholded estimator is shown to be consistent in the operator norm, as long as the true covariance matrix is sparse in a suitable sense and $(\log G)/N \to 0$ as $G, N \to \infty$. Cai and Liu (2011) extended this idea by defining an adaptive threshold for each entry of the sample covariance matrix. By making an analogy with mean estimation, and noticing that

$$\frac{1}{N}\sum_{k=1}^{N}(X_{ik}-\mu_i)(X_{jk}-\mu_j) = \sigma_{ij} + \sqrt{\frac{\theta_{ij}}{N}}z_{ij}, \quad z_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1),\ 1 \le i,j \le G,$$

they find that if the θ_{ij} were known, then a good estimator for σ_{ij} would be $\widehat{\sigma}^0_{ij} = s_{\lambda^0_{ij}}(\widehat{\sigma}_{ij})$, with s_λ a thresholding function and $\lambda^0_{ij} = 2\sqrt{\theta_{ij}\log G/N}$. As θ_{ij} is generally unknown, they estimate it, by $\widehat{\theta}_{ij} = N^{-1}\sum_{k=1}^{N}\left\{(X_{ik}-\overline{X}_i)(X_{jk}-\overline{X_j}) - \widehat{\sigma}_{ij}\right\}^2$.

Their final estimator of the covariance matrix is

$$\widehat{\sigma}^\star_{ij} = s_{\lambda_{ij}}(\widehat{\sigma}_{ij}), \quad \lambda_{ij} = \delta\sqrt{\frac{\widehat{\theta}_{ij}\log G}{N}}. \tag{6.1}$$

The thresholding function s_λ is any function satisfying the conditions

- $|s_\lambda(z)| \le c|y|$, for all y, z satisfying $|z-y| \le \lambda$;

- $s_\lambda(z) = 0$, for $|z| \leq \lambda$;
- $|s_\lambda(z) - z| \leq \lambda, \quad z \in \mathbb{R}$.

The soft thresholding function, $s_\lambda(z) = \text{sign}(z)(z-\lambda)_+$, and the adaptive lasso, $s_\lambda(z) = z(1 - |\lambda/z|)_+^\eta$, $\eta \geq 1$, satisfy these conditions. The authors also notice that the hard thresholding function $s_\lambda(z) = zI_{\{|z|>\lambda\}}$, although not satisfying all the conditions, can be used anyway. The parameter $\delta > 0$ can be chosen equal to 2, as suggested by Cai and Liu (2011), or through cross-validation. The adaptive thresholding estimator attains the optimal rate of convergence in the spectral norm.

Thresholded matrices may have negative eigenvalues. This is avoided by lasso-type penalized normal likelihood optimization, as detailed in Section 6.2.3.

6.2.2 Factor models to estimate large covariance matrices

A lot of methods such as Carvalho *et al.* (2008) and Fan *et al.* (2011, 2008), use factor modeling, where the idea is to consider two types of variables: known variables and latent factors, to represent known modules, defined by biologists, and unknown modules, that one wishes to retrieve. Carvalho *et al.* (2008) base their method on sparsity modeling of multivariate regression, ANOVA, and latent factor models to identify biological subpathway structure. They assume that the matrix of factor loadings is sparse. However, they recognize that fitting models with a very large number of variables and factors is computationally intensive, due to the MCMC calculations involved. They aim to identify new genes and pathways linked to a small number of known biological pathways. They recommend starting with genes of known relevance, and then gradually including related genes. Runcie and Mukherjee (2013) develop a Bayesian genetic sparse factor model for estimating the covariance matrix of high dimensional traits in a mixed effects model. They only consider biologically plausible matrices, thus imposing structure. They assume sparsity, and suppose that only a small number of factors control the variation in the high dimensional phenotype, so that they are more interpretable.

Fan *et al.* (2008) develop a factor model for large covariance matrix estimation in economics, where they assume that the factors are observable. Fan *et al.* (2011) relax the idiosyncratic and independence assumptions often made on the components of a factor model. They assume a sparse covariance matrix for the error term, which they estimate using the method of Cai and Liu (2011). They then use factor models to estimate the covariance matrix of the observations.

6.2.3 Methods based on graphical models or penalized likelihood

In graphical models, conditional independence is assessed through the precision matrix, which is the inverse of the covariance matrix. Zeros in the precision matrix correspond to pairs of genes that are conditionally independent given the other variables, or certain subsets

of them. The precision matrix can be estimated by maximum likelihood. Maximizing

$$l(\Sigma^{-1}) = \frac{N}{2}\left\{\log|\Sigma^{-1}| - \mathrm{tr}\left(S\Sigma^{-1}\right)\right\},$$

where S is the sample covariance matrix, leads to an estimate $\widehat{\Sigma}^{-1} = S^{-1}$. However, when $G \gg N$, S is singular and therefore not invertible. This problem can be circumvented by using the graphical lasso (Friedman *et al.*, 2008), which maximizes

$$\log|\theta| - \mathrm{tr}(S\theta) - \lambda||\theta||_1,$$

with respect to θ. In order to estimate the precision matrix or the covariance matrix, one can impose a penalty either on the entries of the matrix or on parts of some decomposition of the matrix, like the Cholesky or the spectral decomposition (Pourahmadi, 2011). Stein (1975) noticed that the sample covariance matrix distorts the eigenstructure of the covariance matrix. By shrinking the eigenvalues to some central values, he obtains

$$\widehat{\Sigma} = Pf(\lambda)P^T,$$

where λ represents the ordered eigenvalues and P the normalized eigenvectors of the sample covariance matrix, and f is some shrinking function.

By imposing lasso-type constraints on the precision matrix, one can obtain sparse estimators (Friedman *et al.*, 2008; Rothman *et al.*, 2008), while a logarithmic barrier function ensures positive definiteness, as developed by Rothman (2012), who focuses on estimators invariant under permutation. The fastest algorithm available to date is the glasso, proposed by Friedman *et al.* (2008). Guo *et al.* (2011) describe a likelihood-based method for estimating precision matrices across data from several graphical models that share the same variables and some of the dependence structure. They employ a hierarchical penalty that forces similar patterns of sparsity in the inverse covariance matrices across classes. This method is similar to the group graphical lasso of Danaher *et al.* (2014). Banerjee and Ghosal (2013) use a Bayesian model to estimate the precision matrix of a Gaussian graphical model. They impose a banding structure on the matrix and require the user to set the order of the graph a priori. They show that their estimator performs better than the maximum likelihood estimator in terms of the L_2 operator norm.

6.2.4 Bayesian, empirical Bayes and other methods

In Bayesian methods the covariance matrix is often modeled through either a Jeffreys non-informative prior or an inverse Wishart prior. Pourahmadi (2011) identifies several methods in Bayes and empirical Bayes covariance matrix estimation. First, one can model the logarithm of the covariance matrix by using the log matrix prior, which introduces a multivariate normal prior with many hyperparameters. While this is more flexible than an inverse Wishart prior, it is not easily interpretable. The generalized inverse Wishart prior is obtained by partitioning a

multivariate normal vector $Y = (Z_1, \cdots, Z_k) \sim \mathcal{N}(0, \Sigma)$ into k subsets, writing the joint density as

$$f(y) = f(z_1) f(z_2 \mid z_1) f(z_3 \mid z_2) \cdots f(z_k, z_{k-1} \ldots z_1),$$

then placing a normal prior on the regression coefficients and an inverse Wishart prior on the prediction variances in each conditional distribution (Pourahmadi, 2011). Carvalho *et al.* (2007) propose an efficient method for direct sampling from the hyper-Wishart distribution. Conlon *et al.* (2012) developed a Bayesian model to incorporate information of co-regulated genes, that are close on the chromosome and co-transcribed. They construct a hierarchical Bayesian model to obtain the gene-specific posterior probability of differential expression including co-regulation information. They show that this supplementary information improves the independence model that they developed in a previous paper, which assumes genes to be independent. Zhao *et al.* (2012) introduced a novel Bayesian model to integrate microarray data with pathways obtained from KEGG. Corander *et al.* (2013) developed a predictive supervised Bayesian classifier for several classes, assumed to be represented by multivariate Gaussian distributions and known in advance, while a block diagonal structure is assumed on the covariance matrix.

One can also model the correlation matrices, as done by Barnard *et al.* (2000), who use log normal priors on variances independent of a prior on the whole matrix of correlations R.

Finally, Abadir *et al.* (2012) noticed that orthogonal matrices from the orthogonal decomposition of the sample covariance matrix are never ill-conditioned, and therefore they focus on estimating the eigenvalues. Fan *et al.* (2013) apply the adaptive thresholding estimator of Cai and Liu (2011) to principal orthogonal complements, in order to construct a sparse covariance matrix.

Estimation of large covariance matrices is thus well-studied. Some methods focus on the estimation of a covariance matrix common to several studies, either by estimating a single covariance matrix for all groups, or by independently estimating the covariance matrices of each group and identify their similarities (Gaskins and Daniels, 2013; Guo *et al.*, 2011, for example). Our model presented in Section 6.4 is of the former type and gives an empirical Bayes procedure which estimates a sparse correlation matrix common to several studies. Before presenting our model in Section 6.4, we describe some work on testing the equality of several covariance matrices in Section 6.3.

6.3 Test for the equality of two covariance matrices

6.3.1 Likelihood ratio tests and other tests

To test the hypothesis $H_0 : \Sigma_1 = \Sigma_2$ against $H_1 : \Sigma_1 \neq \Sigma_2$, one usually uses the modified likelihood ratio test introduced by Bartlett (1937). Suppose there are two independent and normally

distributed samples X_1 and X_2 of size n_1 and n_2 respectively, having measurements for G variables. Denoting by S_1 and S_2 the sample covariance matrices of each sample respectively, the likelihood ratio statistic is given by

$$L = \frac{|S_1|^{(n_1-1)/2}|S_2|^{(n_2-1)/2}}{|S|^{N/2}}, \quad S = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1+n_2-2}, \quad N = n_1+n_2,$$

where $|S|$ is the determinant of S, and under the null hypothesis, $-2\log L \underset{H_0}{\sim} \chi^2_{G(G+1)/2}$, where $\underset{H_0}{\sim} F$ means 'follows the distribution F under the null hypothesis'. However, when $G > N$, the sample covariance matrix is singular, and the statistic is not well defined.

Several tests have been developed to test the equality of covariance matrices when $G \gg N$. Chaipitak and Chongcharoen (2013) review existing methods that can handle this case. The test developed by Schott (2007) is based on an estimator of the square Frobenius norm of $\Sigma_1 - \Sigma_2$, and defines

$$\begin{aligned} T_S &= \frac{(n_1-1)(n_2-1)}{2(n_1+n_2-2)}\left(\widehat{a}_{21} + \widehat{a}_{22} - \frac{2}{G}\mathrm{tr}(S_1S_2)\right), \\ \widehat{a}_{2i} &= \frac{(n_i-1)^2}{G(n_i-2)(n_i+1)}\left\{tr(S_i^2) - \frac{1}{n_i-1}\mathrm{tr}(S_i)^2\right\}, \quad i = 1,2, \end{aligned}$$

which, under the null hypothesis asymptotically follows a standard normal distribution, $T_S \xrightarrow{d} \mathcal{N}(0,1)$, as $(G, n_1, n_2) \to \infty$.

Srivastava *et al.* (2011) propose a test based on a lower bound of the Frobenius norm, given by

$$\begin{aligned} T_{Sr} &= \frac{\widehat{a}_{21} - \widehat{a}_{22}}{\sqrt{\widehat{\eta}_1^2 + \widehat{\eta}_2^2}}, \quad \widehat{\eta}_i^2 = \frac{4}{(n_i-1)^2}\widehat{a}_2^2\left(1 + \frac{2(n_i-1)\widehat{a}_4}{G\widehat{a}_2^2}\right), \\ \widehat{a}_4 &= \frac{1}{c_0}\left(\frac{1}{G}\mathrm{tr}(A^4) - Gc_1\widehat{a}_1 - G^2c_2\widehat{a}_1^2\widehat{a}_2 - Gc_3\widehat{a}_2^2 - NG^3\widehat{a}_1^4\right), \\ c_0 &= N(N^3+6N^2+21N+18), \quad c_1 = 2N(2N^2+6N+9), \quad c_2 = 2N(3N+2). \end{aligned}$$

Under the null hypothesis, the test statistic, T_{Sr}, asymptotically follows a standard normal distribution, $T_{Sr} \xrightarrow{d} \mathcal{N}(0,1)$, $(G,N) \to \infty$.

Srivastava and Yanagihara (2010) compare the traces of the squared matrices,

$$\begin{aligned} T_{SY} &= \frac{\widehat{\gamma}_1 - \widehat{\gamma}_2}{\sqrt{\widehat{\xi}_1^2 + \widehat{\xi}_2^2}} \underset{H_0}{\sim} \mathcal{N}(0,1), \\ \widehat{\gamma}_i &= \frac{\widehat{a}_{2i}}{\widehat{a}_{1i}^2}, \quad \widehat{\xi}_i^2 = \frac{4}{(n_i-1)^2}\left\{\frac{\widehat{a}_2^2}{\widehat{a}_1^4} + \frac{2(n_i-1)}{G}\left(\frac{\widehat{a}_2^3}{\widehat{a}_1^6} - \frac{2\widehat{a}_2\widehat{a}_3}{\widehat{a}_1^5} + \frac{\widehat{a}_4}{\widehat{a}_1^4}\right)\right\}, \\ \widehat{a}_3 &= \frac{1}{N(N^2+3N+4)}\left(\frac{1}{G}\mathrm{tr}(A^3) - 3N(N+1)G\widehat{a}_2\widehat{a}_1 - NG^2\widehat{a}_1^3\right). \end{aligned}$$

Under the null hypothesis, T_{SY} also follows a standard normal distribution asymptotically.

Finally, Chaipitak and Chongcharoen (2013) develop a test based on the ratio of the traces of the squared matrices, $b = \text{tr}(\Sigma_1^2)/\text{tr}(\Sigma_2^2)$, whose estimator, $\widehat{b}$ follows asymptotically ($G, N \to \infty$) a normal distribution under the null hypothesis,

$$\begin{aligned}
&\frac{\widehat{b}-1}{\widehat{\delta}} \xrightarrow[H0]{d} \mathcal{N}(0,1),\\
&\quad \widehat{b} = \frac{\widehat{a}_{21}}{\widehat{a}_2 2}, \quad \widehat{\delta}^2 = \frac{4}{G}\left\{\frac{2\widehat{a}_4^\star}{\widehat{a}_2^2}\sum_{i=1}^{2}\frac{1}{n_i-1} + \sum_{i=1}^{1}\frac{G}{(n_i-1)^2}\right\},\\
&\quad \widehat{a}_4^\star = \frac{\tau}{G}\left\{\text{tr}(S^4) + \beta^\star \text{tr}(S^3)\text{tr}(S) + c^\star(\text{tr}(S^2))^2 + d\,\text{tr}(S^2)(\text{tr}(S))^2 + e(\text{tr}(S))^4\right\},\\
&\quad \tau = \frac{N^5(N^2+N+2)}{(N+1)(N+2)(N+4)(N+6)(N-1)(N-2)(N-3)},\\
&\quad b^\star = \frac{-4}{N}, \quad c^\star = \frac{-(2N^2+3N-6)}{N(N^2+N+2)}, \quad d = \frac{2(5N+6)}{N(N^2+N+2)}, \quad e = \frac{-(5N+6)}{N^2(N^2+N+2)}.
\end{aligned}$$

Chaipitak and Chongcharoen (2013) performed simulations to assess the attained significance level (ASL) of each of the above tests. The ASL is expected to be close to the nominal significance level setting, usually 0.05. They also compare their test with others in terms of power and found that overall, they attain the best power while keeping the ASL close to $\alpha = 0.05$.

Motivated by the fact that the accuracy of the likelihood ratio test can be poor, Davison *et al.* (2014) developed a method based on computing a p-value using a one-dimensional numerical integration, conditioning on the direction. The authors use directional p-values to test the hypothesis $\Lambda = \Lambda_0$, where Λ is the inverse covariance matrix, and Λ_0 is equal to Λ, apart from some zero entries. We extend the theory to the comparison of two covariance matrices and therefore test the hypothesis $\Lambda_1 = \Lambda_2 = \Lambda_0$ against the alternative $\Lambda_1 \neq \Lambda_2$, in the hope that this test is more accurate than the likelihood ratio test, and has potentially more power in settings where the likelihood ratio test fails, i.e., when the number of samples is too small. Suppose we have two independent populations, Y_1 and Y_2, of size respectively n_1 and n_2, having G parameters and following a multivariate normal distribution with parameter μ_1, Σ_1 and μ_2, Σ_2 respectively. Then we can write the log likelihood of the parameter $\theta = (\mu_1, \mu_2, \Lambda_1, \Lambda_2)$, with $\Lambda_i = \Sigma_i^{-1}$, $i = 1, 2$, as

$$l(\theta; y) = \sum_{i=1}^{2}\frac{n_i}{2}\log|\Lambda_i| - \frac{1}{2}\text{tr}(\Lambda_i Y_i^T Y_i) + 1_{n_i}^T Y_i \Lambda_i \mu_i - \frac{n_i}{2}\mu_i^T \Lambda_i \mu_i.$$

The maximum likelihood estimator is $\widehat{\theta} = (\widehat{\mu}_1, \widehat{\mu}_2, \widehat{\Lambda}_1, \widehat{\Lambda}_2)$, with $\widehat{\mu}_i = Y_i^T 1_{n_i} n_i^{-1}$, and $\widehat{\Lambda}^{-1} = Y_i^T Y_i n_i^{-1} - Y_i^T 1_{n_i} 1_{n_i}^T Y_i n_i^{-2}$.

We can now compute the likelihood ratio test corresponding to the null hypothesis, using

$\widehat{\Lambda}_0^{-1} = (n_1\widehat{\Sigma}_1 + n_2\widehat{\Sigma}_2)/(n_1+n_2)$,

$$\begin{aligned} w = -2(l_0 - l_1) =& 2\left(\sum_{i=1}^{2}\frac{n_i}{2}\log(|\widehat{\Lambda}_i|) - \frac{1}{2}\mathrm{tr}(\widehat{\Lambda}_i Y_i^T Y_i) + 1_{n_i}^T Y_i\widehat{\Lambda}_i\widehat{\mu}_i\right. \\ & \left. - \frac{n_i}{2}\log(|\widehat{\Lambda}_0|) + \frac{1}{2}\mathrm{tr}(\widehat{\Lambda}_0 Y_i^T Y_i) + 1_{n_i}^T Y_i\widehat{\Lambda}_0\widehat{\mu}_i - \frac{n_i}{2}\widehat{\mu}_i^T\widehat{\Lambda}_0\widehat{\mu}_i\right) \\ =& -N\log(|\widehat{\Lambda}_0|) + n_1\log(|\widehat{\Lambda}_1|) + n_2\log(|\widehat{\Lambda}_2|) \\ =& -n_1\log(|\widehat{\Lambda}_0\widehat{\Lambda}_1^{-1}|) - n_2\log(|\widehat{\Lambda}_0\widehat{\Lambda}_2^{-1}|). \end{aligned}$$

We introduce $\varphi = (\xi_1, \xi_2, \Lambda_1, \Lambda_2) = (\Lambda_1\mu_1, \Lambda_2\mu_2, \Lambda_1, \Lambda_2)$, and the log likelihood becomes

$$l(\varphi; y) = \sum_{i=1}^{2}\frac{n_i}{2}\log|\Lambda_i| - \frac{1}{2}\mathrm{tr}(\Lambda_i Y_i^T Y_i) + 1_{n_i}^T Y_i\xi_i - \frac{n_i}{2}\xi_i^T\Lambda_i^{-1}\xi_i. \tag{6.2}$$

Following Davison *et al.* (2014), we compute s_ψ, which is the value of $s = -\partial l^0(\varphi)/\partial\varphi$ evaluated at $\varphi = \widehat{\varphi}_\psi^0$, the maximum likelihood estimate from (6.2). We can easily show that $\widehat{\varphi} = (\widehat{\Lambda}_1\widehat{\mu}_1, \widehat{\Lambda}_2\widehat{\mu}_2, \Lambda_1, \Lambda_2)$, and from the first order derivatives,

$$\begin{aligned} \frac{\partial l(\varphi)}{\partial\xi_i}\Big|_{\varphi=\widehat{\varphi}^0} &= Y_i^T 1_{n_i} - n_i\Lambda_i^{-1}\xi_i\Big|_{\varphi=\widehat{\varphi}^0} = 0 \\ \frac{\partial l(\varphi)}{\partial\Lambda_i}\Big|_{\varphi=\widehat{\varphi}^0} &= \frac{n_i}{2}\Lambda_i^{-1} - \frac{1}{2}Y_i^T Y_i + \frac{n_i}{2}\Lambda_i^{-1}\xi\xi^T\Lambda_i^{-1}\Big|_{\varphi=\widehat{\varphi}^0} \\ &= \frac{n_i}{2}\widehat{\Lambda}_0^{-1} - \frac{1}{2}Y_i^T Y_i + \frac{n_i}{2}\widehat{\mu}_i\widehat{\mu}_i^T = \frac{n_i}{2}(\widehat{\Lambda}_0^{-1} - \widehat{\Lambda}_i^{-1}), \end{aligned} \tag{6.3}$$

we obtain, $s_\psi = \left\{0, 0, -(n_1/2)\left(\widehat{\Lambda}_0^{-1} - \widehat{\Lambda}_1^{-1}\right), -(n_2/2)\left(\widehat{\Lambda}_0^{-1} - \widehat{\Lambda}_2^{-1}\right)\right\}$. The tilted log likelihood along the line $s(t) = (1-t)s_\psi$ is

$$l(\varphi; y) = \sum_{i=1}^{2}\frac{n_i}{2}\log|\Lambda_i| - \frac{1}{2}\mathrm{tr}(\Lambda_i Y_i^T Y_i) + 1_{n_i}^T Y_i\xi_i - \frac{n_i}{2}\xi_i^T\Lambda_i^{-1}\xi_i - \frac{n_i}{2}(1-t)\Lambda_i(\widehat{\Lambda}_0^{-1} - \widehat{\Lambda}_i^{-1}). \tag{6.4}$$

The maximization of (6.4) gives $\widehat{\varphi}(s(t)) = \left\{\widehat{\xi}_1, \widehat{\xi}_2, \widehat{\Lambda}_1(t), \widehat{\Lambda}_2(t)\right\}$, where $\widehat{\Lambda}_i^{-1}(t) = t\widehat{\Lambda}_i^{-1} + (1-t)\widehat{\Lambda}_0^{-1}$. The directional p-value is defined as

$$p(\psi) = \frac{\int_1^{t_{max}} t^{d-1}h(t;\psi)\mathrm{d}t}{\int_0^{t_{max}} t^{d-1}h(t;\psi)\mathrm{d}t}, \quad h(t;\psi) = \frac{c\exp\left[l\left\{\widehat{\varphi}_\psi^0(s(t)); s(t)\right\} - l\left\{\widehat{\varphi}(s(t)); s(t)\right\}\right]}{\sqrt{|J_{\varphi\varphi}(\widehat{\varphi}(s(t)); s(t))|}}.$$

We therefore need to compute the double derivatives of the tilted likelihood function, for which we use the formulae in Fackler (2005). From the single derivatives (6.3), we see that second order derivatives involving parameters from two different populations are 0, and we

get the other second order derivatives, which evaluated at $\varphi = \widehat{\varphi}$ lead to

$$\begin{aligned}
\frac{-\partial^2 l}{\partial \xi_i \partial \xi_i^T}\Big|_{\varphi=\widehat{\varphi}} &= n_i \widehat{\Lambda}_i^{-1}, \\
\frac{-\partial^2 l}{\partial \xi_i \Lambda_i}\Big|_{\varphi=\widehat{\varphi}} &= -n_i(\widehat{\Lambda}^{-1} \otimes \widehat{\mu}^T), \\
\frac{-\partial^2 l}{\partial \Lambda_i \xi_i}\Big|_{\varphi=\widehat{\varphi}} &= \frac{-n_i}{2}(\widehat{\mu}_i \otimes I_p) - \frac{n_i}{2}(I_p \otimes \widehat{\mu}_i), \\
\frac{-\partial^2 l}{\partial \Lambda_i \Lambda_i}\Big|_{\varphi=\widehat{\varphi}} &= \frac{n_i}{2}(\widehat{\Lambda}_i^{-1} \otimes \widehat{\Lambda}_i^{-1}) + \frac{n_i}{2}(\widehat{\mu}_i \mu_i^T \otimes \widehat{\Lambda}_i^{-1}) + \frac{n_i}{2}(\widehat{\Lambda}_i^{-1} \otimes \widehat{\mu}_i \widehat{\mu}_i^T).
\end{aligned}$$

The determinant of $J_{\varphi\varphi}(\widehat{\varphi})$, is $|J_{\varphi\varphi}(\widehat{\varphi})| = |\widehat{\Lambda}_1^{-1}(t)|^{G+2}|\widehat{\Lambda}_2^{-1}(t)|^{G+2}$. It is now possible to compute $h(t;\psi)$, which leads to the directional p-value, after integration, given by

$$\begin{aligned}
h(t;\psi) &= \left(\frac{|\widehat{\Lambda}_1^{-1}(t)|}{|\widehat{\Lambda}_0^{-1}|}\right)^{(n_1-G-2)/2} \left(\frac{|\widehat{\Lambda}_2^{-1}(t)|}{|\widehat{\Lambda}_0^{-1}|}\right)^{(n_2-G-2)/2}, \\
p(\psi) &= \frac{\int_1^{t_{\max}} t^{d-1} h(t;\psi)\mathrm{d}t}{\int_0^{t_{\max}} t^{d-1} h(t;\psi)\mathrm{d}t}, \quad d = \frac{G(G+1)}{2}.
\end{aligned}$$

6.3.2 Simulations to compare tests of equality of large covariance matrices

As the likelihood ratio test is not well defined for $G > N$, we develop a new test for assessing the equality of two covariance matrices, Σ_1 and Σ_2 of size $G \times G$, with $G \gg N$. From our gene set of size G, we first randomly select a subset of genes of size $q < N$. Then we extract the corresponding $q \times q$ covariance matrices, σ_1 and σ_2, from Σ_1 and Σ_2, and test the equality of these two submatrices, $H_0 : \sigma_1 = \sigma_2$, which results in a p-value. The likelihood ratio test is valid in this case, as $q < N$. We repeat the process B times, resulting in a set of p-values $p^{(1)}, \ldots, p^{(B)}$, which we correct for multiple testing using the Benjamini and Hochberg (1995) method. If any of the tests is rejected, we reject the null hypothesis that the two matrices are equal.

Before assessing this method directly, we want to see how the different methods compare in the case where the number of variables G is smaller than the number of samples N. This would correspond to one test of the submatrices method described previously. We perform simulations based on the five simulation designs presented in Chaipitak and Chongcharoen (2013). Each design is based on the same structure; under H_0, we generate $\Sigma_{10} = \Sigma_{20} = M_0$ and under H_1, $\Sigma_{11} = M_0$ and $\Sigma_{21} = M_1$, where M_0 and M_1 are defined as follows:

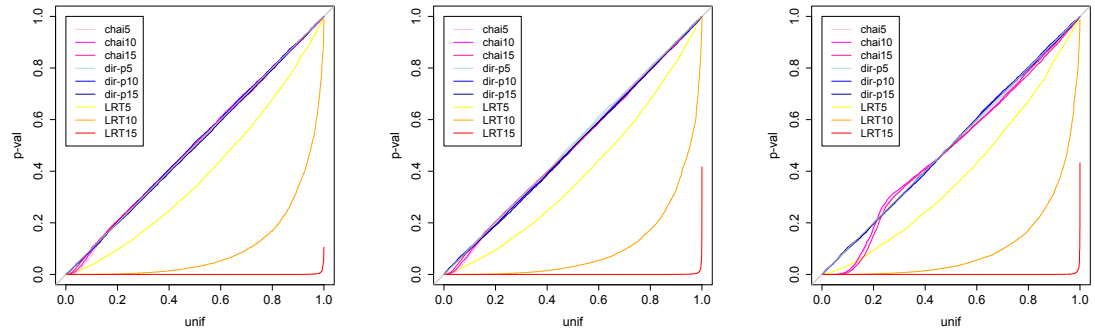
- unstructured pattern (UN): for M_0, $m_{ij} = 1$ if $i = j$ and $m_{ij} = (-1)^{i+j} 0.1\, i/j$ otherwise;
- compound symmetry pattern (CS): $M_0 = 0.99 I_G + (0.01) 1_G 1_G^T$, $M_1 = 0.95 I_G + (0.05) 1_G 1_G^T$;
- heterogeneous compound symmetry pattern (CSH), $m_{ii} = m_i^2$ and $m_{ij} = m_i m_j \rho$, where $m_i \sim \mathcal{U}(5,6)$, $\rho = 0.5$ for H_0, and $m_i \sim \mathcal{U}(4,5)$, and $\rho = 0.4$ for H_1;

- simple pattern 1 (SIM1) $M_0 = 2I_G$, $M_1 = 1.5I_G$;
- simple pattern 2 (SIM2) $M_0 = I_G$, $M_1 = \text{diag}(1,1,1,2,\ldots,1,1,1,2)$.

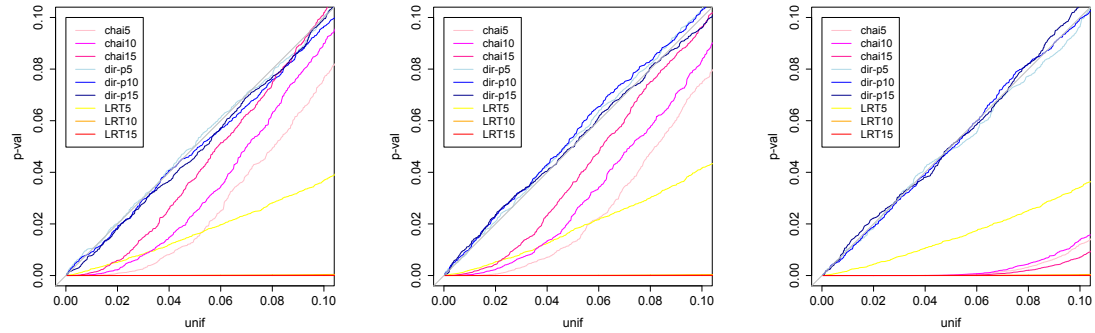
We compare three methods: the likelihood ratio test (LRT), the directional p-value (dirp) and the Chaipitak and Chongcharoen (2013) method (chai), based on the attained significance level (ASL), which gives the proportion of p-values that fall below the level α ($\alpha = 0.05$ in our case) under the null hypothesis, and power in rejecting the null, which counts the number of rejected hypotheses when simulating under the alternative. Results of the comparisons are presented in Tables 6.1 and 6.2 for $N = 20$ and 100 respectively, and for G =5, 10 and 15 genes, based on 1000 simulations. In these tables, we give the ASL in the first row. This value should be close to the nominal value of 0.05. When the ASL is larger than 0.05, the power of the test is overestimated. We therefore computed the corrected α which is the significance level required to test at a 5% level, and the corresponding power, computed using this corrected value of α. For $n_1 = n_2 = 20$, we observe that the directional p-value has a value of the ASL which is almost perfectly equal to 0.05, whereas the LRT and the method of Chaipitak and Chongcharoen (2013) tend to overestimate it. However, the power of all three methods is very low. In the case of the LRT for $G = 15$, we see how important it is to correct the significance level of the test. Indeed, the power of LRT attains 99% without correcting the significance level, whereas it is only 8% after correction. In general, all tests have low power when $n_1 = n_2 = 20$. Increasing the sample sizes leads to better results in terms of power, even after correcting the significance level for the SIM1 and SIM2 designs. However, the power under the other simulation designs is still low. Moreover, recall we are in a favorable setting in these simulations, as the number of variables is smaller than the number of samples.

Figures 6.1, 6.2, 6.3 and 6.4 present the sorted p-values obtained from the simulations under the different tests, with a zoom on the smallest values. Under the null hypothesis, where the two matrices are identical, we should obtain roughly uniformly distributed p-values. We notice that this is always the case for directional p-values for all simulation designs, which indicates that the ASL is well controlled. p-values obtained from the Chaipitak and Chongcharoen (2013) test follow approximately a uniform distribution when $N = 200$, only departing a bit from the theoretical uniform distribution for small p-values. However, for smaller N there is a clear departure from the uniform distribution, indicating that the ASL is not as expected. Indeed, in order to test at a 5% level, one would need to take a much smaller value for α. This departure from the uniform distribution is even more pronounced for the likelihood ratio test, for which observed p-values tend to be much smaller than they should, even for large sample sizes.

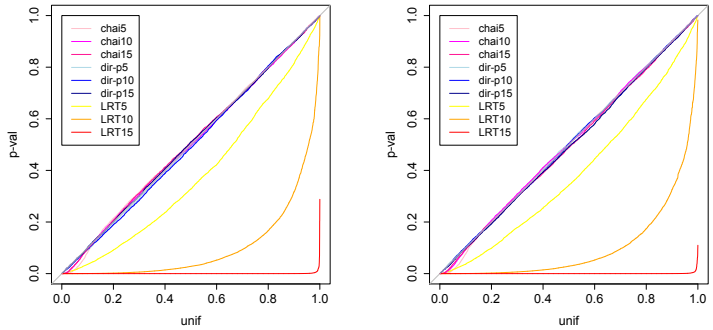
The results of the simulations show that the significance level is often overestimated in the LRT and the test from Chaipitak and Chongcharoen (2013), but is well controlled under the directional p-value setting. However the power of the latter is lower than or comparable to the corrected power of the other methods. We also noticed that when the sample sizes are small, the power of all tests is very low. Therefore, the idea of splitting large covariance matrices into



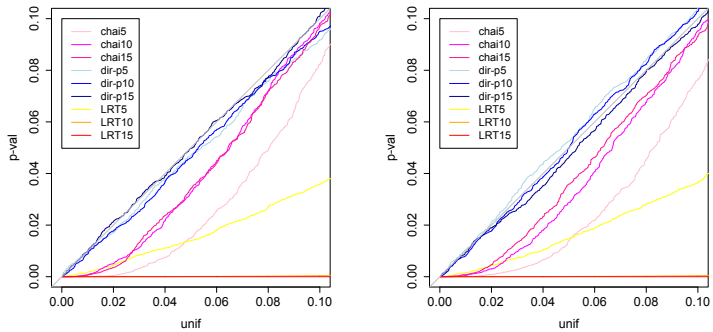
(a) model UN (b) model CS (c) model CSH



(d) model UN (e) model CS (f) model CSH

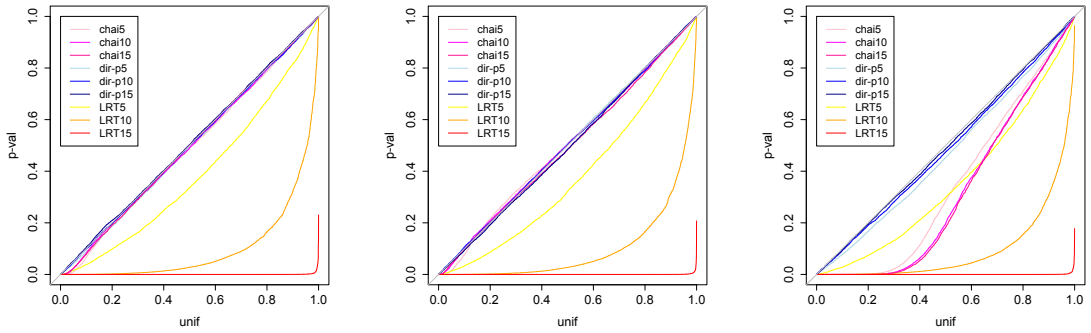


(g) model sim1 (h) model sim2

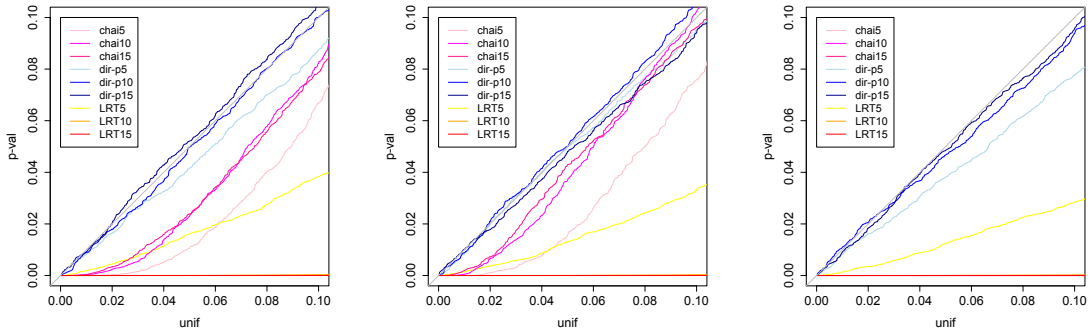


(i) model sim1 (j) model sim2

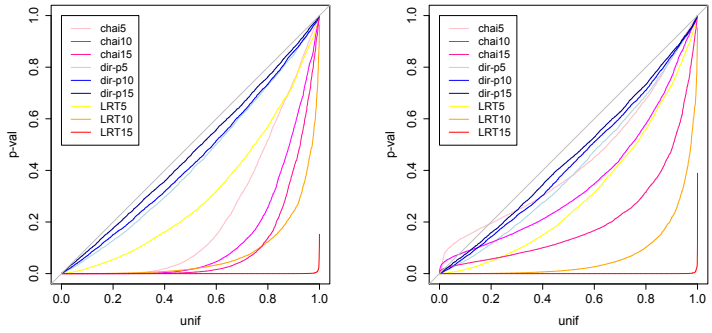
Figure 6.1 – p-values under H_0 for the different simulation designs, for $n_1 = n_2 = 20$, and for the likelihood ratio test (LRT), the directional p-value (dirp) and the test from Chaipitak and Chongcharoen (2013)(chai). Under the null hypothesis, p-values are assumed to follow a uniform distribution (grey line). The second and fourth rows are a zoom on the small p-values (p<0.1).



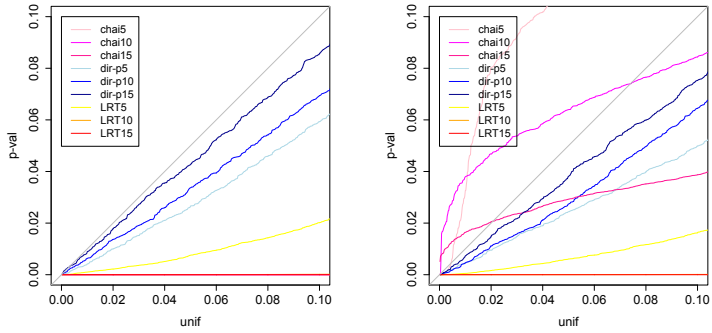
(a) model UN (b) model CS (c) model CSH



(d) model UN (e) model CS (f) model CSH



(g) model sim1 (h) model sim2

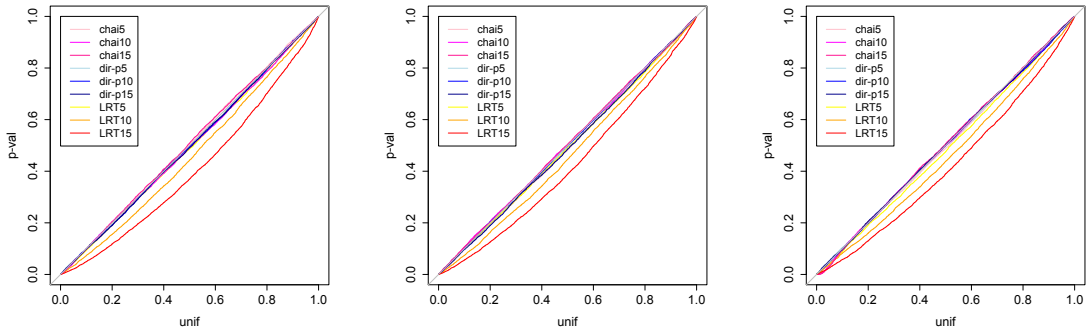


(i) model sim1 (j) model sim2

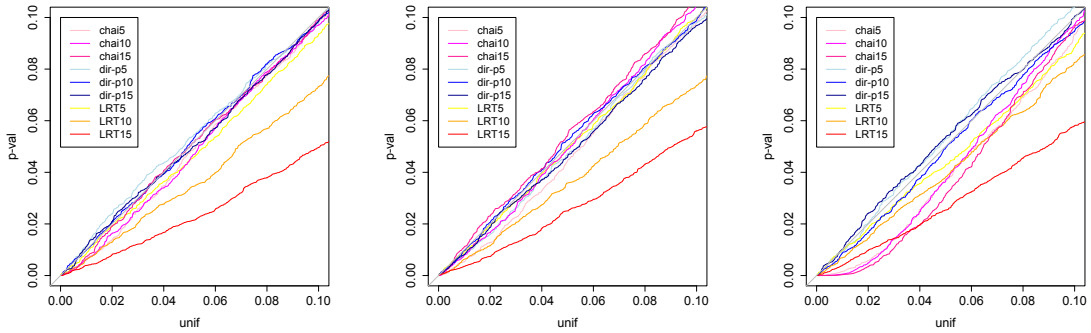
Figure 6.2 – p-values under H_1 for the different simulation designs, for $n_1 = n_2 = 20$, comparing three tests: the likelihood ratio test (LRT), the directional p-value (dirp) and the test from Chaipitak and Chongcharoen (2013)(chai). The second and fourth rows are is a zoom on the small p-values (p<0.1)

Table 6.1 – Results from the simulations comparing three tests of equality of covariance matrices of size $G \times G$, for sample sizes $n_1 = n_2 = 20$, and for five simulation designs. The first row of the table gives the ASL, which should be close to 5%. The second row is the power of the test using a significance level of $\alpha = 5\%$. The third and fourth rows give the corrected α, a corrected significance level required to test at a 5% level, and the corresponding power.

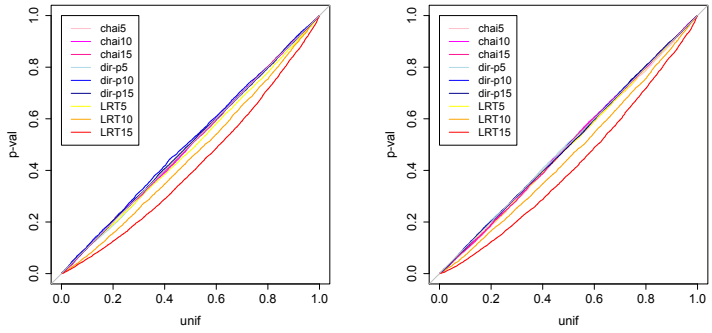
ASL (%)																
	UN			CS			CSH			SIM1			SIM2			
G	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	
5	8.1	4.9	12.3	8.2	4.7	11.8	13.9	5.3	13.0	8.0	5.4	13.0	8.3	4.7	12.3	
10	7.0	5.2	58.9	7.2	4.7	58.5	13.4	5.1	58.2	6.5	5.3	59.02	6.9	5.0	59.3	
15	5.9	5.4	99.8	6.1	5.0	99.8	14.6	5.0	99.8	6.4	5.1	99.8	6.3	5.3	99.7	
power (%)																
	UN			CS			CSH			SIM1			SIM2			
G	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	
5	8.9	5.9	12.3	8.0	5.0	13.6	34.3	6.6	14.7	48.1	8.6	18.8	1.2	10.0	20.6	
10	7.2	5.0	59.7	6.0	5.0	59.8	38.3	5.5	61.8	60.8	7.4	66.5	2.5	8.0	69.3	
15	7.5	4.7	99.8	5.8	5.2	99.8	39.5	5.2	99.8	70.4	5.8	99.8	16.2	6.5	99.8	
corrected α (%)																
	UN			CS			CSH			SIM1			SIM2			
G	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	
5	1.6	5.0	1.6	1.3	5.0	1.7	0.01	5.0	1.4	1.5	5.0	1.4	1.4	5.0	1.5	
10	2.5	5.0	0.001	2.3	5.0	0.001	0.01	5.0	0.001	3.0	5.0	0.001	3.0	5.0	0.001	
15	4.0	5.0	10^{-12}	4.0	5.0	10^{-12}	0.001	5.0	10^{-12}	3.0	5.0	10^{-12}	4.0	5.0	10^{-12}	
corrected power (%)																
	UN			CS			CSH			SIM1			SIM2			
G	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	
5	5.5	5.8	4.9	4.6	5.4	6.1	16.	6.6	5.	38.2	8.2	7.9	0.6	10.0	9.4	
10	5.1	4.9	6.0	4.0	5.3	5.2	21.0	5.9	5.1	56.0	6.9	7.0	0.5	8.0	9.0	
15	6.0	4.3	4.6	5.0	5.2	6.3	19.0	5.2	6.0	67.0	5.8	6.1	8.0	6.1	8.2	



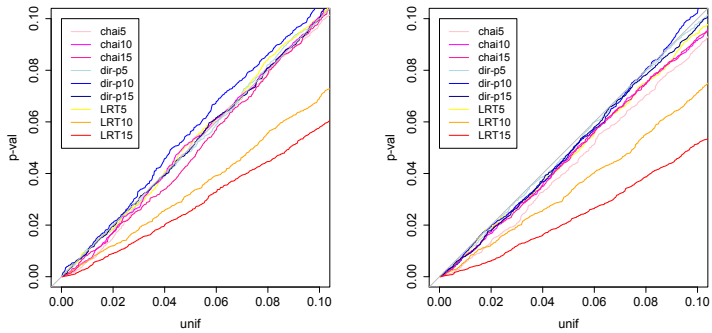
(a) model UN (b) model CS (c) model CSH



(d) model UN (e) model CS (f) model CSH

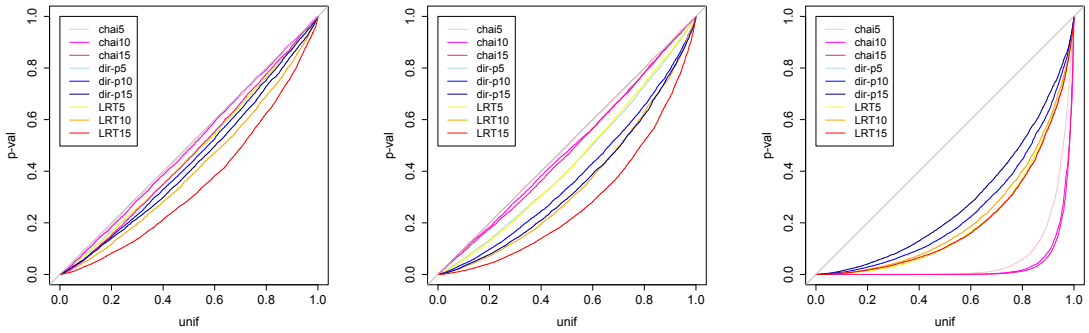


(g) model sim1 (h) model sim2

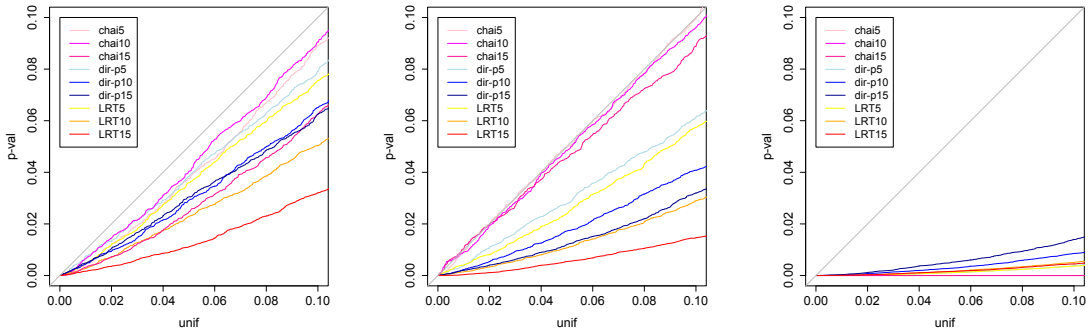


(i) model sim1 (j) model sim2

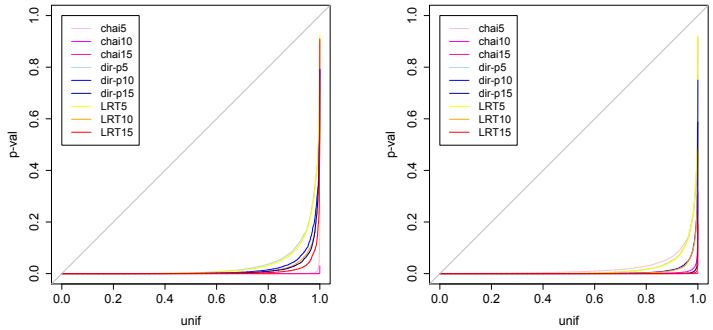
Figure 6.3 – p-values under H_0 for the different simulation designs, for $n_1 = n_2 = 200$, and for the likelihood ratio test (LRT), the directional p-value (dirp) and the test from Chaipitak and Chongcharoen (2013)(chai). Under the null hypothesis, p-values are assumed to follow a uniform distribution (grey line). The second and fourth rows are a zoom on the small p-values (p<0.1).



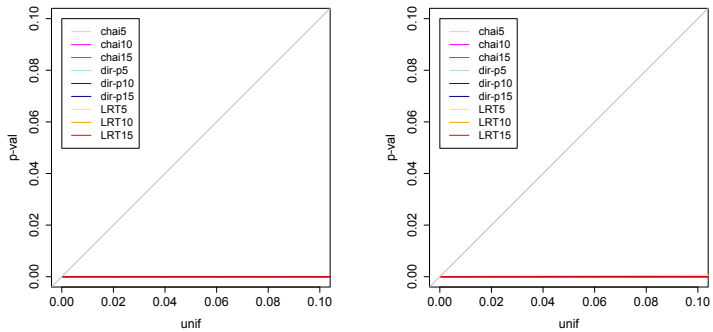
(a) model UN (b) model CS (c) model CSH



(d) model UN (e) model CS (f) model CSH



(g) model sim1 (h) model sim2



(i) model sim1 (j) model sim2

Figure 6.4 – p-values under H_1 for the different simulation designs, for $n_1 = n_2 = 200$, comparing three tests: the likelihood ratio test (LRT), the directional p-value (dirp) and the test from Chaipitak and Chongcharoen (2013)(chai). The second and fourth rows are a zoom on the small p-values (p<0.1)

Table 6.2 – Results from the simulations comparing three tests of equality of covariance matrices of size $G \times G$, for sample sizes $n_1 = n_2 = 200$, and for five simulation designs. The first row of the table gives the ASL, which should be close to 5%. The second row is the power of the test using a significance level of $\alpha = 5\%$. The third and fourth rows give the corrected α, a corrected significance level required to test at a 5% level, and the corresponding power.

ASL (%)																				
	UN			CS			CSH			SIM1			SIM2							
G	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT					
5	5.2	4.7	5.6	5.6	5.4	5.0	6.1	4.8	5.9	4.9	5.1	4.8	5.8	5.3	5.5					
10	5.3	4.9	6.9	5.1	4.7	6.9	6.1	5.0	6.2	4.7	4.3	7.4	5.3	5.2	7.5					
15	5.0	5.1	10.0	4.6	5.5	9.0	6.5	4.6	8.9	5.4	5.0	8.8	5.4	5.2	9.7					
power (%)																				
	UN			CS			CSH			SIM1			SIM2							
G	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT					
5	6.4	6.4	6.6	5.0	8.3	8.9	79.0	37.0	37.0	99.0	84.0	85.0	87.0	89.0	90.0					
10	5.8	8.0	9.9	5.1	12.0	15.0	87.0	27.0	33.0	100	90.0	93.0	99.0	96.0	96.0					
15	8.6	8.3	14.0	5.6	14.0	22.0	88.0	22.0	35.0	100	93.0	96.0	100	100	100					
corrected α (%)																				
	UN			CS			CSH			SIM1			SIM2							
G	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT					
5	4.7	5.3	4.5	4.4	4.7	4.9	3.7	5.2	4.2	5.0	4.8	5.1	4.1	4.7	4.6					
10	4.4	5.1	3.4	4.9	5.2	3.6	3.6	4.9	3.8	5.2	5.6	3.2	4.6	4.7	3.2					
15	5.0	4.8	2.1	5.4	4.6	2.5	3.1	5.4	2.5	4.4	4.9	2.5	4.5	4.7	2.1					
corrected power (%)																				
	UN			CS			CSH			SIM1			SIM2							
G	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT	Chai	dir-p	LRT					
5	6.2	6.7	6.1	4.4	7.8	8.9	76.0	38.0	35.0	99.0	83.0	85.0	84.0	89.0	89.0					
10	5.0	8.0	7.0	5.0	13.0	12.0	85.0	28.0	28.0	100	91.0	89.0	99.0	96.0	95.0					
15	9.0	8.0	7.0	6.0	13.0	14.0	86.0	24.0	25.0	100	93.0	93.0	100	99.0	99.0					

many smaller submatrices to test equality is not promising: each individual test having low power, combining them will certainly not be very powerful.

In the next section, we present a method to estimate a large sparse correlation matrix common to several studies.

6.4 Empirical Bayes estimation of sparse correlation matrices

6.4.1 The model

As shown in Section 6.1, the estimation of large covariance or correlation matrices has been well studied, and is eased by assuming that the matrix is sparse, which also makes sense biologically. If several studies are available, estimation of common covariance features is done either by estimating a single covariance matrix, or by independently estimating the covariance matrices of each study and identifying their similarities afterwards. We use the first approach. Starting from several studies, we estimate a common sparse correlation matrix, which we can then analyze to discover modules. The idea is to use the Fisher transformation on the pairwise correlation coefficients and assume that they are normally distributed. This transformation was also used by Hafdahl (2007) to combine correlation matrices in a fixed effects meta-analysis. The idea of the model comes from Johnstone and Silverman (2005), who develop a Bayesian approach to wavelet regression, also used by Davison (2008) to model nuclear magnetic resonance imaging.

We start from the sample correlation matrices of the studies providing a gene expression matrix (Type 1 studies in Chapter 3), denoted by $R^{(1)},\dots,R^{(L_1)}$, and we obtain the vectors $r^{(1)},\dots,r^{(L_1)}$, the vectorized upper diagonal part of the correlation matrices. Applying Fisher's transformation to these vectors leads to a set of vectors $\left\{Z^{(1)},\dots,Z^{(L_1)}\right\}$, with elements

$$Z_g^{(l)} = \frac{1}{2}\log\left(\frac{1+r_g^{(l)}}{1-r_g^{(l)}}\right), \quad g = 1,\dots,G(G-1)/2, \quad l = 1,\dots,L_1.$$

We now assume the following model on Z, for each of the $G(G-1)/2$ different pairwise correlations:

$$Z_g^{(l)} \sim \mathcal{N}(\theta_g,\sigma_l^2), \quad \theta_g \sim (1-p)\delta_0(\theta) + p\mathcal{N}(0,\tau^2), \quad g = 1,\dots,G(G-1)/2, \tag{6.5}$$

where δ_θ is the Dirac function putting all mass at 0. The prior we choose for θ allows some of the correlations to be exactly zero, making the resulting correlation matrix sparse. We can easily obtain the posterior density for θ_g, and, the calculations being the same for all g, we

omit the subscript in what follows:

$$\begin{aligned}\pi\left(\theta \mid Z^{(1)},\ldots,Z^{(L_1)}\right) &= \frac{\pi(\theta) f\left(Z^{(1)},\ldots,Z^{(L_1)} \mid \theta\right)}{f\left(Z^{(1)},\ldots,Z^{(L_1)}\right)} \\ &= \frac{\pi(\theta) f\left(Z^{(1)},\ldots,Z^{(L_1)} \mid \theta\right)}{\int_\theta f\left(Z^{(1)},\ldots,Z^{(L_1)} \mid \theta\right)\pi(\theta)\mathrm{d}\theta} \\ &= p_z\sqrt{b}\varphi\left(\frac{\theta - b^{-1}\sum_{l=1}^{L_1} Z^{(l)}/\sigma_l^2}{b^{-1/2}}\right) + (1-p_z)\delta_0,\end{aligned} \tag{6.6}$$

where

$$\begin{aligned}&b = \frac{1}{\tau^2} + \sum_{l=1}^{L_1}\frac{1}{\sigma_l^2}, \quad w_1 = \frac{p}{\sqrt{b}\tau}\exp\left\{\frac{1}{2b}\left(\sum_{l=1}^{L_1}\frac{Z^{(l)}}{\sigma_l^2}\right)^2\right\}\prod_{l=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{Z^{(l)}}{\sigma_l}\right), \\ &w_2 = (1-p)\prod_{l=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{z^{(l)}}{\sigma_l}\right), \quad p_z = \frac{w_1}{w_1+w_2}.\end{aligned}$$

Detailed calculations are provided in Section A.3 with a summary of the parameters used for model (6.5) in Table A.6 of the Appendix, and an illustration of the prior and posterior distributions of θ is presented in Figure 6.5.

As it is possible to obtain an explicit expression for the likelihood, we adopt an empirical Bayes approach, and estimate the parameters $\sigma_1,\ldots,\sigma_{L_1}$, p, τ by maximum likelihood. The log likelihood is given by

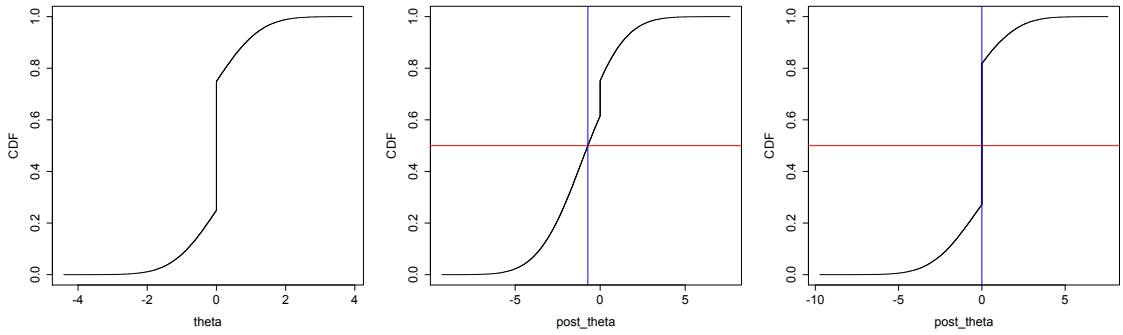
$$l(\sigma_1,\ldots,\sigma_{L_1},p,\tau) = \sum_{g=1}^{G}\log\left[(1-p)\tilde{w}_2 + \frac{p\tilde{w}_2}{\tau\sqrt{b}}\exp\left\{\frac{1}{2b}\left(\sum_{l=1}^{L_1}\frac{Z_g^{(l)}}{\sigma_l^2}\right)^2\right\}\right], \quad \tilde{w}_2 = \prod_{l=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{Z_g^{(l)}}{\sigma_l}\right),$$

which we optimize with respect to $q = \log\{p/(1-p)\}$, $\sigma_1,\ldots,\sigma_{L_1}$ and τ.

A natural summary for θ is the posterior mean. However, as we seek a sparse correlation matrix, we rather choose the posterior median $\tilde{\theta}$ of θ, which performs a form of soft shrinkage (Johnstone and Silverman, 2005; Davison, 2008). We denote by $p_z = w_1/(w_1+w_2)$ the posterior probability that θ is non-null. We also need to compute the posterior distribution of θ, at $\theta = 0$,

$$\begin{aligned}&\tilde{H}(0) = \int_{-\infty}^{0}\pi\left(\theta \mid Z^{(1)},\ldots,Z^{(L_1)}\right)d\theta = \int_{-\infty}^{0} p_z\sqrt{b}\varphi\left((\theta-\mu_\theta)\sqrt{b}\right)d\theta = p_z\sqrt{b}\Phi\left(-\mu_\theta\sqrt{b}\right) = p_z H(0), \\ &\mu_\theta = \frac{1}{b}\sum_{l=1}^{L_1}\frac{Z^{(l)}}{\sigma_l^2}.\end{aligned}$$

The quantity $\tilde{H}(0)$ identifies lower point of the jump in the posterior distribution of θ and is useful to distinguish the different cases which can appear in the calculation of the median, as



(a) prior, $p = 0.5$, $\tau = \sigma_i = 1$ (b) posterior (c) posterior

Figure 6.5 – Prior (left) and posterior (center and right) distributions for θ for one gene present in three studies, with, *center*: $z = (-1.5, -1.8, -1.2)$, *right*: $z = (-0.5, -0.8, -0.2)$. The blue line represents the median.

illustrated in Figure 6.6:

- if $p_z < 0.5$, then $0.5 \in \left[H(0)p_z\,;\, p_z H(0) + (1-p_z)\right]$ and therefore the median $\tilde{\theta} = 0$;
- if $p_z > 0.5$;
  - if $0.5 \in \left[H(0)p_z\,;\, p_z H(0) + (1-p_z)\right]$, $\tilde{\theta} = 0$;
  - if $H(0)p_z > 0.5$, we want to solve

$$p_z \Phi\left(\frac{\tilde{\theta} - \mu}{\sigma}\right) = 0.5 \Rightarrow \tilde{\theta} = \Phi^{-1}\left(\frac{1}{2p_z}\right)\frac{1}{\sqrt{b}} + \mu_\theta;$$

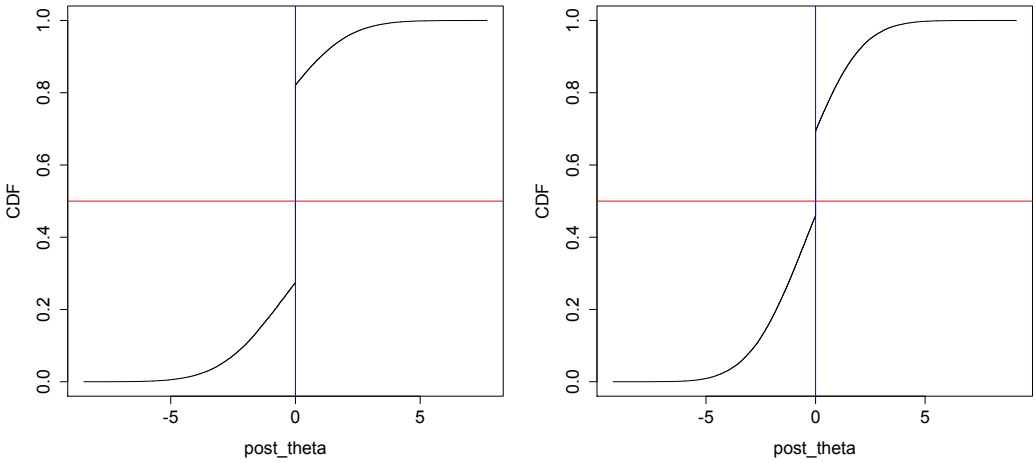
  - if $H(0)p_z + (1-p_z) < 0.5$, then

$$p_z \Phi\left((\tilde{\theta} - \mu_\theta)\sqrt{b}\right) + (1-p_z) = 0.5 \Rightarrow \tilde{\theta} = \Phi^{-1}\left(\frac{2p_z - 1}{2p_z}\right)\frac{1}{\sqrt{b}} + \mu_\theta.$$

The common correlation matrix is then defined by applying the inverse of Fisher's transformation to the matrix having entries $\tilde{\theta}$ in the upper diagonal part, and filling the rest of the estimated matrix to make it symmetric. We denote the estimated common correlation matrix by $\tilde{R}$.

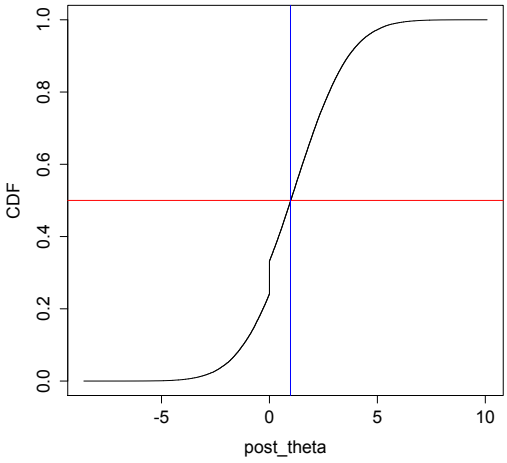
6.4.2 Simulations

We perform some simulations to ensure that the estimation of the hyperparameters is appropriate. To this end, we generate from model (6.5) with $L = 2$, $G = 100$, $p = 0.06$, $\sigma = (0.5, 1.5)$, $\tau = 2.52$, for 100 replications. The boxplots of the estimated parameters are presented in Figure 6.7, where it can be observed that the estimation seems to be really good.

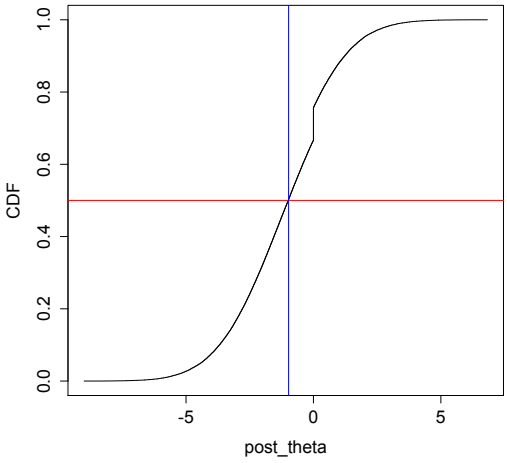


(a) $p_z < 0.5$

(b) $p_z > 0.5$, and $\tilde{\theta} = 0$



(c) $p_z > 0.5$, $p_z\Phi(0) + 1 - p_z < 0.5$



(d) $p_z > 0.5$, $\Phi(0)p_z > 0.5$

Figure 6.6 – Posterior distribution of θ and the corresponding median in blue, for different situations.

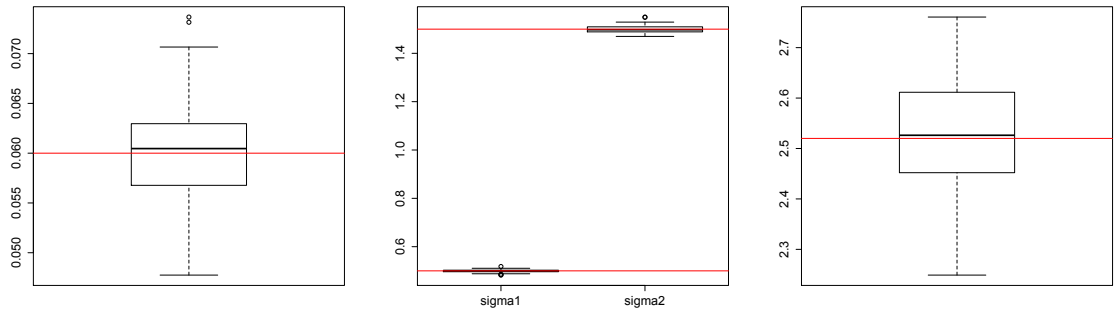


Figure 6.7 – Boxplots of the estimated parameters over 100 simulations from the model.

We now turn to more interesting simulations, and present results for two simulation designs. The first generates a block diagonal correlation matrix R, using the model presented in Section 4.1.1, which gives

$$R = \begin{pmatrix} A_1 & & 0 \\ & \ddots & \\ 0 & & A_B \end{pmatrix}, \quad A_b = \begin{pmatrix} 1 & \rho_b & \cdots & \rho_b \\ \rho_b & & & \vdots \\ \vdots & & \ddots & \rho_b \\ \rho_b & \cdots & \rho_b & 1 \end{pmatrix}, \quad b = 1, \ldots, B, \tag{6.7}$$

where the correlation ρ_b inside each block A_b, is generated according to a uniform distribution between 0.5 and 1, $b = 1, \ldots, B = 5$. Then we transform R to the Fisher scale, add white noise with variance σ_ϵ^2, and transform back to obtain several noisy correlation matrices as illustrated by Figure 6.8. A summary of the parameters used for the simulations is presented in Table A.7 of the Appendix. We performed 100 replications for several values of the noise σ_ϵ^2. Boxplots of the estimated parameters can be found in Figure 6.9, while model performance is shown in Table 6.3. The estimates of σ_l and τ tend to be very close to the predefined values, even when the noise added is large. However, the probability of being non null, p, tends to be overestimated, particularly when the noise is large, which is to be expected. Indeed, as the noise becomes larger and larger, it becomes difficult to distinguish the noise from the signal.

We also performed other simulations generating sparser matrices. To this end, we randomly set a proportion $q = 0.01$ of correlations to be non null, and distributed according to $\mathcal{U}(0.5, 1)$ in the matrix R. Then the simulations are conducted as presented in Figure 6.8. The resulting $\tilde{R}^{(i)}$ are not correlation matrices, because they are not positive definite, but the model does not require positive definite matrices.

Results of the simulations for the sparse matrix are in Figure 6.10 and Table 6.4, where we see that the estimators of p and σ_l are very accurate, except for p when the noise is very large, as expected. The parameter τ is difficult to estimate when the noise is small. Concerning the values of the Tables 6.3 and 6.4, we see that the true zero rate (TZR) is almost perfect while

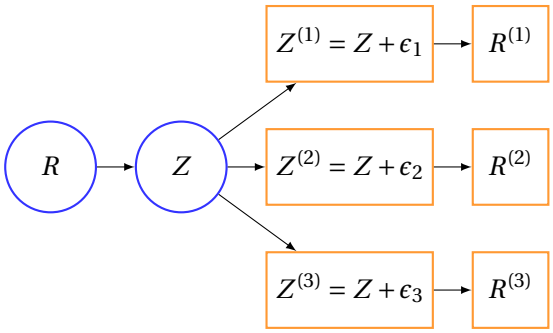


Figure 6.8 – Simulation design used to assess the model. R is the true block diagonal correlation matrix and Z is the Fisher transformed matrix, $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Genes in each block in R have common correlation ρ.

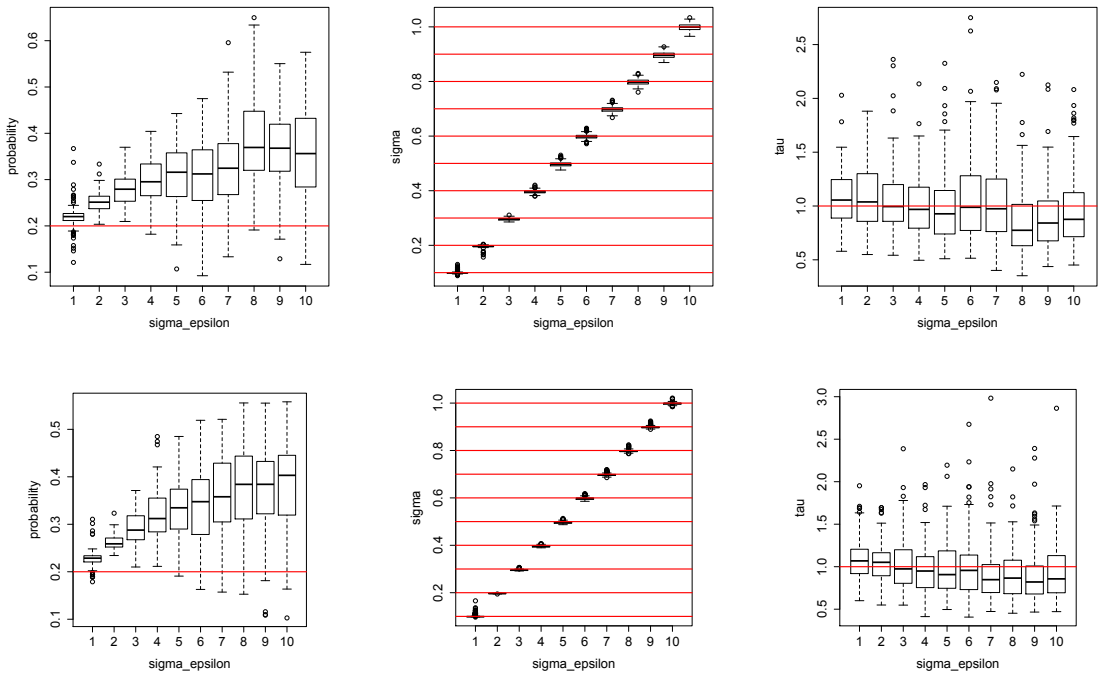


Figure 6.9 – Boxplots of the estimated parameters for non-sparse simulations with $G = 100$ (top) and $G = 1000$ (bottom). Red lines are the true values of the parameter for different values of the noise $\sigma_\epsilon = 0.1, \ldots, 1$ ($\times 10$ of the x-axis the figures).

Table 6.3 – Simulations for $G = 100$, over $R_{\text{sim}} = 100$ simulations, for block diagonal matrices with $B = 5$ blocks. The different quantities examined are the Frobenius, L_1 and operator norms of the difference between the estimated matrix and the truth; the true zero rate (TZR), false zero rate (FZR), true positive rate (TPR) and false positive rate (FPR) in %; the average of the differences of the true non zero entries; the average of all the differences; the proportion (%) of the differences that are smaller than 5% and the estimates of the model parameters.

$G = 100$	$\sigma_\epsilon = 0.1$	$\sigma_\epsilon = 0.2$	$\sigma_\epsilon = 0.3$	$\sigma_\epsilon = 0.4$	$\sigma_\epsilon = 0.5$
Frobenius norm	1.4(0.03)	3.34(0.06)	6.85(0.13)	10.81(0.16)	14.4(0.18)
L_1 norm	0.89(0.01)	2.13(0.04)	4.66(0.11)	7.46(0.11)	9.68(0.09)
operator norm	0.34(0.004)	0.83(0.01)	1.99(0.08)	3.77(0.1)	5.30(0.1)
TZR (%)	99.0(0.2)	99.0(0.5)	98.0(1.0)	97.0(1.0)	96.0(2.0)
FZR (%)	0.0(0.0)	0.07(0.1)	3.0(3.0)	9.0(6.0)	18.0(10.0)
TPR (%)	1.0(0.0)	99.0(0.1)	97.0(3.0)	91.0(6.0)	82.0(10.0)
FPR (%)	30.0(20.0)	1.0(0.5)	2.0(1.0)	3.0(1.0)	4.0(2.0)
Mean $\neq 0$	0.02(0.004)	0.04(0.009)	0.07(0.02)	0.13(0.04)	0.19(0.05)
Mean of differences	0.004(0.0009)	0.009(0.002)	0.01(0.005)	0.03(0.008)	0.05(0.01)
prop (%) $< 5\%$	98.0	94.0	90.0	87.0	84.0
$\widehat{p}$ ($p = 0.2$)	0.23(0.04)	0.25(0.04)	0.28(0.05)	0.30(0.05)	0.31(0.07)
$\widehat{\sigma}$ ($\sigma = \sigma_\epsilon$)	0.11(0.02)	0.19(0.003)	0.30(0.0004)	0.40(0.005)	0.50(0.008)
$\widehat{\tau}$ ($\tau = 1$)	1.43(0.62)	1.08(0.32)	1.06(0.35)	1.00(0.31)	0.99(0.35)
	$\sigma_\epsilon = 0.6$	$\sigma_\epsilon = 0.7$	$\sigma_\epsilon = 0.8$	$\sigma_\epsilon = 0.9$	$\sigma_\epsilon = 1$
Frobenius norm	17.3(0.20)	19.7(0.17)	22.2(0.14)	23.8(0.15)	25.3(0.13)
L_1 norm	11.34(0.12)	12.8(0.09)	13.77(0.08)	14.5(0.07)	15.4(0.09)
operator norm	6.67(0.1)	7.76(0.08)	8.53(0.07)	9.37(0.07)	10.23(0.08)
TZR (%)	96.0(2.0)	95.0(3.0)	93.0(4.0)	94.0(3.0)	94.0(4.0)
FZR (%)	26.0(10.0)	32.0(12.0)	39.0(12.0)	47.0(12.0)	52.0(12.0)
TPR (%)	74.0(10.0)	68.0(12.0)	61.0(12.0)	53.0(12.0)	48.0(12.0)
FPR (%)	4.0(2.0)	5.0(3.0)	7.0(4.0)	6.0(3.0)	6.0(4.0)
Mean $\neq 0$	0.25(0.06)	0.30(0.07)	0.36(0.06)	0.41(0.08)	0.45(0.06)
Mean of differences	0.06(0.01)	0.07(0.01)	0.09(0.01)	0.09(0.01)	0.10(0.01)
prop(%) $< 5\%$	84.0	82.0	79.0	79.0	79.0
$\widehat{p}$ ($p = 0.2$)	0.31(0.08)	0.33(0.09)	0.37(0.09)	0.36(0.09)	0.35(0.1)
$\widehat{\sigma}$ ($\sigma = \sigma_\epsilon$)	0.60(0.009)	0.70(0.008)	0.80(0.009)	0.90(0.012)	1(0.013)
$\widehat{\tau}$ ($\tau = 1$)	1.08(0.42)	1.03(0.38)	0.87(0.33)	0.91(0.33)	0.96(0.35)

Table 6.4 – Simulations for $G = 100$, over $R_{\text{sim}} = 100$ simulations, for sparse matrices ($q = 0.01$). The different quantities examined are the Frobenius, L_1 and operator norms of the difference between the estimated matrix and the truth; The true zero rate (TZR), false zero rate (FZR), true positive rate (TPR) and false positive rate (FPR); the average of the differences of the true non zero entries; the average of all the differences; the proportion of the differences that are smaller than 0.05 and the estimates of the model parameters.

$G = 100$	$\sigma_\epsilon = 0.1$	$\sigma_\epsilon = 0.2$	$\sigma_\epsilon = 0.3$	$\sigma_\epsilon = 0.4$	$\sigma_\epsilon = 0.5$
Frobenius norm	0.29(0.009)	1.03(0.04)	2.47(0.04)	3.75(0.04)	4.79(0.03)
L_1 norm	0.14(0.006)	0.55(0.02)	1.14(0.03)	1.75(0.04)	2.10(0.05)
operator norm	0.11(0.007)	0.47(0.02)	0.81(0.01)	1.12(0.01)	1.29(0.01)
TZR (%)	99.0(0.001)	99.0(0.01)	99.0(0.4)	99.0(0.01)	99.0(0.01)
FZR (%)	0.041(3.0)	2.0(4.0)	14.0(6.0)	30.0(6.0)	46.0(8.0)
TPR (%)	99.0(3.0)	98.0(4.0)	86.0(6.0)	70.0(6.0)	54.0(8.0)
FPR (%)	0.001(0.001)	0.01(0.01)	0.01(0.4)	0.01(0.01)	0.01(0.01)
Mean $\neq 0$	0.02(0.003)	0.05(0.02)	0.13(0.03)	0.23(0.04)	0.34(0.06)
Mean of differences	$2.10^{-4}(10^{-5})$	$6.10^{-4}(10^{-4})$	$2.10^{-3}(10^{-4})$	$2.10^{-3}(10^{-4})$	$4.10^{-3}(10^{-4})$
prop (%) <5%	99.0	99.0	99.0	99.0	99.0
$\widehat{p}$ ($p = 0.01$)	0.04(0.04)	0.01(0.002)	0.018(0.03)	0.016(0.003)	0.016(0.005)
$\widehat{\sigma}$ ($\sigma = \sigma_\epsilon$)	0.13(0.03)	0.20(0.02)	0.30(0.0003)	0.40(0.004)	0.50(0.005)
$\widehat{\tau}$ ($\tau = 1$)	2.87(0.91)	2.45(2.0)	1.44(1.26)	0.99(0.14)	1.02(0.16)

$G = 100$	$\sigma_\epsilon = 0.6$	$\sigma_\epsilon = 0.7$	$\sigma_\epsilon = 0.8$	$\sigma_\epsilon = 0.9$	$\sigma_\epsilon = 1$
Frobenius norm	5.66(0.04)	6.16(0.04)	6.61(0.03)	6.88(0.03)	7.17(0.03)
L_1 norm	2.42(0.05)	2.65(0.05)	2.83(0.05)	2.97(0.05)	2.99(0.04)
operator norm	1.48(0.01)	1.57(0.01)	1.66(0.01)	1.72(0.001)	1.77(0.01)
TZR (%)	99.0(0.01)	99.0(0.01)	99.0(0.01)	99.0(0.1)	99.0(1.0)
FZR (%)	62.0(9.0)	70.0(8.0)	78.0(8.0)	83.0(7.0)	88.0(9.0)
TPR (%)	38.0(9.0)	30.0(8.0)	22.0(8.0)	17.0(7.0)	12.0(9.0)
FPR (%)	0.01(0.01)	0.01(0.01)	0.01(0.01)	0.01(0.1)	0.01(1.0)
Mean $\neq 0$	0.44(0.05)	0.51(0.06)	0.57(0.05)	0.61(0.05)	0.67(0.05)
Mean of differences	$0.005(10^{-4})$	$0.005(10^{-4})$	$0.006(10^{-4})$	$0.007(10^{-4})$	$0.007(10^{-4})$
prop (%) < 5%	99.0	99.0	99.0	99.0	99.0
$\widehat{p}$ ($p = 0.01$)	0.015(0.006)	0.016(0.008)	0.017(0.01)	0.04(0.06)	0.11(0.16)
$\widehat{\sigma}$ ($\sigma = \sigma_\epsilon$)	0.60(0.0006)	0.70(0.0007)	0.80(0.0008)	0.90(0.01)	1(0.01)
$\widehat{\tau}$ ($\tau = 1$)	1.16(0.64)	1.19(0.82)	1.27(0.90)	1.01(0.33)	0.92(0.66)

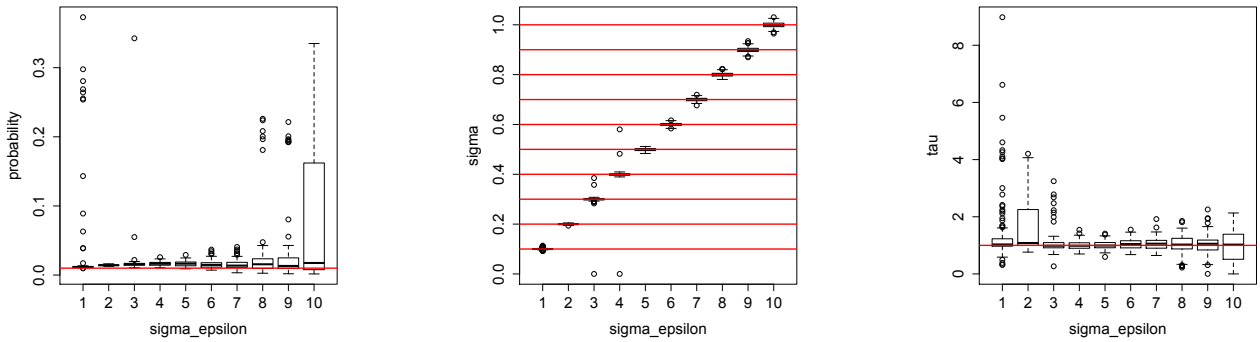


Figure 6.10 – Boxplots of estimated parameters for sparse simulations with $G = 100$. Red lines are the true values of the parameter for different values of the noise $\sigma_\epsilon = 0.1, \ldots, 1$ ($\times 10$ on the x-axis of the figures).

Table 6.5 – Average computation time for the empirical Bayes model for different numbers of genes G, with $L = 3$ studies.

G	100	200	500	1000	2000	5000	10000
time	1.91	5.3	10.4	12.9	18.4	70.5	394

the false zero rate (FZR) tends to be quite high. This means that many values that should not be zero are fixed to zero anyway. However, the mean of the differences of the entries is very small, which also indicates that the matrix is well estimated. The high rate of false zeros may be due to very small entries in the true correlation matrix, which are estimated as being null. However, in simulations performed in Section 7.2.1 we will see this is not a problem for module detection.

6.5 Conclusion

We developed an empirical Bayes procedure for the estimation of a large and sparse correlation matrix common to several studies. The estimated matrix is not a correlation matrix, as it is not positive definite. However, we do not need our matrix to be positive definite, as we will only use it to define modules in the next chapter. On top of being very simple, the method is also fast, as shown in Table 6.5, and can therefore handle very large sets of genes. Computational times may be larger than those presented in Table 6.5, depending on the maximum likelihood estimation, which may require more iterations from time to time. This is due to the fact that we do not use all observations to estimate the parameters, but only a subset, for two reasons: first, it is faster to use fewer observations, and second, as the number of correlations is equal to $G(G-1)/2$, which is very large for large G, taking a subset (usually 10^5, randomly selected), will still give precise estimation.

Simulations from the model and from independent designs show that the model works quite well in estimating a common correlation matrix. In the next chapter, we focus on the detection of gene modules based on this common correlation matrix. The fact that the matrix is sparse aids the detection of modules. We also do not have to deal with one correlation matrix per study in order to identify the common modules. In our case, we only have one matrix $\tilde{R}$ to deal with, which simplifies module detection.

7 Modules

7.1 Clustering

7.1.1 Overview

When one wants to partition a set of data into groups that are not known in advance, clustering methods are very useful. Clustering, which is an unsupervised technique, must be distinguished from classification, a supervised method where one wants to assign objects to known classes. For a full review on clustering techniques, we refer to Gan *et al.* (2007) and Hastie *et al.* (2009, Chapter 14). In the context of genomics data, Thalamuthu *et al.* (2006) provide a good review of existing methods.

The first step for most clustering methods is to define a similarity or dissimilarity measure, such that the more two data points are alike, the larger the similarity. This can be the Euclidean distance, the correlation or any other distance measure. Hastie *et al.* (2009, Chapter 14) distinguish three categories of clustering algorithms; combinatorial, which work directly on the observed data, mode seeking, which are nonparametric approaches, and mixture modeling, where the data are assumed to be independent and identically distributed samples from a population described by a parametrized model.

Combinatorial algorithms are usually not feasible in high dimensions as they consist in partitioning the data in all possible manners and identifying the one which seems to be the most appropriate regarding the optimization of a given criterion.

Hierarchical algorithms divide the dataset into a sequence of nested clusters (Gan *et al.*, 2007, p. 109), and can be separated into two types: agglomerative and divisive. Agglomerative algorithms start with each object clustered alone and then merge the clusters that are the closest according to the similarity measure. Divisive algorithms start with all objects belonging to the same big cluster, and recursively split the clusters into smaller ones, based on the largest between-group dissimilarity. One of the main disadvantages of such methods is that once a point has been attributed to a cluster, it cannot be changed, i.e., there is no correcting step if

a point has been incorrectly assigned to a wrong cluster. Hierarchical clustering algorithms also impose a hierarchical structure on the data, even if such a structure is inappropriate. Hierarchical clustering does not require the number of clusters to be fixed or need a starting configuration before running the algorithm. Agglomerative hierarchical methods can be separated in two groups: those based on graph methods, which use links such as single linkage, complete linkage, average linkage, weighted average linkage, and geometric methods using links such as Ward's, centroids and the median (Gan *et al.*, 2007, p. 117). Geometric distances require one to measure the center of two agglomerated clusters and obtain a measure of dissimilarity between two centers. Divisive clustering is more appropriate if one wishes to obtain a small number of clusters. It can be applied in different ways: either recursively applying the k-means algorithm (to be presented later in this section) with $k = 2$, or splitting the cluster with the largest diameter, or that with the largest average dissimilarity. Once the tree is built, one needs to cut it at the appropriate height to obtain the desired number of clusters. There exist several methods to select the optimal number of clusters, which are described in more detail in Section 7.1.4.

A dendrogram is a useful graphical representation of the tree obtained from a hierarchical clustering algorithm. It consists of a tree in which each node is associated with a height, such that

$$h(A) \leq h(B) \Rightarrow A \subseteq B,\ A \cap B \neq \emptyset$$

for all subsets of data points A and B, and for each set of points (x_i, x_j), h_{ij} is the height of the node specifying the smallest cluster to which both x_i and x_j belong (Gan *et al.*, 2007, p.111). The height is proportional to dissimilarity and is therefore highly interpretable.

Center-based clustering algorithms are usually very efficient for clustering large and high-dimensional databases, and have their own objective functions. The k-means algorithm is one of the most popular non-hierarchical clustering algorithm (Hastie *et al.*, 2009, p. 509). It allocates the remaining data to the nearest of the k clusters, for some fixed k, and then changes the membership of the clusters according to some error loss function until there is no improvement in the error loss function for any change in membership. Gan *et al.* (2007, p. 161) define the error loss function as $E = \sum_{i=1}^{k} \sum_{x \in \mathscr{C}_i} d(x, \mu(\mathscr{C}_i))$, where $\mu(\mathscr{C}_i)$ is the centroid of $\mathscr{C}_i$, and d is a distance measure. The algorithm proceeds from an initial partition of the objects into k clusters, which may be determined randomly or by stepwise assignment (Hastie *et al.*, 2009, p. 518). It then computes the distance of each point to each cluster center and reallocates the point to the closest cluster. Cluster centers are computed again for all modified clusters until there is no further change. This algorithm is efficient in clustering large datasets, but it may end at a local minimum of the loss function. Moreover, its performance is highly dependent on the initialization of the centers, and it requires fixing the number of clusters in advance. The k-means algorithm can be run for different values of k, and the best k is chosen using methods described in Section 7.1.4. Self organizing maps (SOM) can be viewed as a constrained version of the k-means algorithm. The k centers are constrained to lie in a one-

or two-dimensional manifold in the feature (i.e., gene) space. The SOM procedure tries to bend the plane so that the centers approximate the data points as well as possible (Hastie *et al.*, 2009, p. 528). The observations are processed one at a time and each neighboring center is updated through a function which depends on a tuning parameter α. Neighbors are defined as centers having distance smaller than some threshold r with the current center, where r is also to be determined. The performance of the algorithm depends on the choice of the parameters α and r. For more details about this algorithm, we refer to Hastie *et al.* (2009, p. 528-534)

The last clustering method we briefly present is model-based clustering, in which the data are viewed as coming from a mixture of probability distributions, with one distribution per cluster. Two approaches are considered in this case: classification likelihood or mixture likelihood. For more details about this method, we refer to Gan *et al.* (2007, Chapter 14). Model-based algorithms define the number of clusters by comparing different models, using BIC selection for example.

In the next two sections, we present WGCNA (Langfelder and Horvath, 2007) and tight clustering (Tseng and Wong, 2005), which are competitors of our method for module detection, which will use 1−the correlation matrix estimated as described in Section 6.4 as the distance matrix in a hierarchical clustering algorithm (see Section 7.2 for more details). Both WGCNA and tight clustering define modules common to several gene expression studies. In Section 7.2, we will compare our empirical Bayes method with these two other methods, based on simulations.

7.1.2 Weighted correlation network analysis (WGCNA)

The first method is implemented in the R package WGCNA (Langfelder and Horvath, 2008). The method, which is described in Langfelder and Horvath (2007), constructs co-expression modules in each study under analysis and identifies the common modules. Consensus modules are defined as co-expression modules that are shared by at least two networks or studies. Then the authors represent the gene expression profiles of each module by an eigengene, and study differential expression of the eigengenes across studies. Langfelder and Horvath (2007) start by constructing an adjacency matrix and attribute to each gene a scaled connectivity measure. They transform the adjacency matrix using functions that are meant to make the detection of modules easier, by performing soft thresholding, for example. To define the consensus network they state that two nodes should be connected in a consensus network if all, or only a proportion, of the input networks agree on this connection. The dissimilarity matrix, defined as 1− the adjacency matrix, is then used as input for the clustering algorithm, which identifies modules. Langfelder and Horvath (2007) use hierarchical clustering with average linkage to construct the tree and cut it at either a fixed height or using their dynamic tree cut approach (Langfelder *et al.*, 2008), which, according to their simulations, is more powerful in detecting modules. The eigengene for each module is defined as the first singular vector, with the sign fixed to have positive correlation with the mean gene of the module.

7.1.3 Tight clustering

Tight clustering is another method to detect modules common to several studies. It was developed by Tseng and Wong (2005), and produces tight and stable clusters, which can be thought of as being gene modules. Tight clustering has been developed to separate potentially interesting genes, which can be clustered into modules, with what the authors call "scattered genes", which do not belong to any particular cluster. Scattered genes must be treated separately, as, if clustered with other genes, they may make the determination of clusters and the number of clusters difficult, by artificially distorting the clusters.

The method developed by Tseng and Wong (2005) uses a modified k-means algorithm, which circumvents the local minimum problem by specifying starting values obtained from hierarchical clustering. The algorithm proceeds in two parts. In the first part, called Algorithm A, it selects the candidate genes, or, put differently, removes the scattered genes. It selects a random subset of the data, usually 70%, on which k-means algorithm is applied. The resulting cluster centers are used as classifiers for the rest of the data. The subsampling and classification process is repeated several times, and the set of candidate genes is based on a measure of co-membership. The second part of the algorithm defines the clusters. Starting from an initial number of clusters k_0, Algorithm A is applied to the candidate gene set and the top q genes are clustered together. The cluster is formed by agglomerating genes until a threshold on the distance with the previous included genes is reached. k_0 is then decreased by 1 and the process is repeated until the desired number of clusters is found. The algorithm requires two tuning parameters, which control tightness and stability of the selected clusters. The procedure is coded into the tightClust R package.

7.1.4 Choosing the number of modules

In clustering, where the goal is to find unknown groups of objects, the number of clusters is unknown and needs to be determined, either as an input of the algorithm, like k-means, or as the cutting height of the tree in hierarchical clustering. Dudoit and Fridlyand (2002) present several methods for this task. The optimal number of clusters is usually not assessed directly, results from a clustering algorithm for different numbers of clusters k are instead compared, and the configuration optimizing some quantity is selected, thus defining the optimal number of clusters. Methods to determine the number of clusters are either based on internal or external indices. The former use statistics that are functions of between- or within-cluster sums of squares, and that exploit the observations used to produce the clustering. The silhouette plot and the GAP statistic (Tibshirani *et al.*, 2001, to be described later in this section) are examples of these internal indices. External indices are based on measures of agreements between two clustering results. The Rand index (Rand, 1971), to be defined later in this section, is one example of such an external index. Dudoit and Fridlyand (2002) developed a method called Clest, which is a prediction-based resampling method, which uses ideas of both internal and external indices. However, Clest seems more appropriate for selecting small

numbers of clusters, as in cancer subtypes discoveries, rather than the case we are interested in, namely gene clustering.

The GAP statistic

In many clustering methods the number of clusters has to be chosen a priori. However, in real data applications, where we usually have no clue concerning the number of clusters, it can be useful to have a criterion to help select this. Several methods have been developed to tackle this problem, among which is the GAP statistic, developed by Tibshirani *et al.* (2001), which uses resampling to define a reference distribution. We denote by $\mathscr{C}$ a partition which attributes each gene to a cluster or a module, C_i, such that clusters are elements of the partition, $C_i \in \mathscr{C}$. For a predefined number of clusters k from 1 to $K_{\max}$, we define the partition $\mathscr{C}_k = (C_1, \ldots, C_k)$, where $|C_l| = n_l$. Each cluster is summarized by the sum of the pairwise distances of its elements,

$$D_l = \sum_{i,j \in C_l} d_{ij}.$$

Each partition of k clusters is then characterized by the average pairwise distances of its clusters,

$$W_k = \sum_{l=1}^{k} \frac{1}{2n_l} D_l.$$

The idea of the GAP statistic is to compare $\log(W_k)$ with the expected value computed under a reference distribution obtained from the data by resampling. The optimal number of clusters $\widehat{k}$ is obtained as the value maximizing the GAP statistic, defined as

$$\mathrm{GAP}_B(k) = \mathbb{E}_B(\log W_k) - \log W_k, \tag{7.1}$$

where B indicates the number of times data were resampled to estimate the reference distribution. Tibshirani *et al.* (2001) explain how to generate the reference distribution. We present their method and a slightly different version which is more appropriate for our problem. Indeed, in the original paper, the authors use the data matrix to obtain the reference distribution, whereas we want to compute the clusters based on the common correlation matrix. Following Tibshirani *et al.* (2001), we can use two methods to obtain the null reference distribution:

- the first consists in generating uniformly distributed gene expression values for the null distribution over the range of the observed values for that gene;
- the second, a bit more complicated, takes the shape of the distribution into account by generating uniformly distributed gene expression values over a box aligned with the first principal component of the data.

For both methods, the process is repeated B times, and $W_k^{(b)}$ is calculated for each resample, for $b = 1,\dots,B$ and $k = 1,\dots,K_{\max}$, from which an estimate $\widehat{\mu}$ of $\mathbb{E}_B(\log W_k)$ with corresponding standard deviation sd_k is obtained as,

$$\widehat{\mu} = \frac{1}{B}\sum_{b=1}^{B}\log(W_k^{(b)}), \quad \mathrm{sd}_k = \sqrt{\frac{1}{B}\sum_{b=1}^{B}\left(\log W_k^{(b)} - \widehat{\mu}\right)^2}.$$

The optimal number of clusters is chosen to minimize the GAP statistic (7.1), but to take the variance into account, we should rather choose the smallest k satisfying

$$\mathrm{GAP}(k) \geq \mathrm{GAP}(k+1) - \mathrm{sd}_{k+1}\sqrt{1+\frac{1}{B}},$$

as suggested by Tibshirani *et al.* (2001)

In our case, we only have the common correlation matrix at hand, not the gene expression matrix, so, the resampling method used in the GAP procedure is not directly applicable. The idea in the construction of the reference distribution is to break the structure of the correlation matrix, as, under the null hypothesis, the correlation matrix should not have any structure ($k = 1$). We insert the elements of the upper triangular part of the common correlation matrix $\tilde{R}$ in a vector and randomly permute it. This permuted vector corresponds to the upper triangular part of the resampled matrix $\tilde{R}^{(b)}$, which we fill in by imposing 1 on the diagonal and completing the lower triangular part so that the resulting matrix is symmetric. We then apply the procedure explained previously to estimate the number of clusters.

In the original method, the resampling part is done for each value of k, which is very computationally intensive if we want to test many different values for k. Instead, if using a hierarchical clustering algorithm, we construct a hierarchical tree for each resampled dataset, and cut it in order to obtain k clusters, with $k = 1,\dots,K_{\max}$. We also add the possibility of having a lower bound , $K_{\min}$, for k, and therefore can have $k = K_{\min},\dots,K_{\max}$. These two modifications makes the algorithm less computationally intensive, which is particularly interesting in high dimensions.

Consensus clustering

Consensus clustering, developed by Monti *et al.* (2003), is an algorithm which aims to find a consensus across multiple runs of a clustering algorithm. To assess stability of the discovered clusters, and to determine the optimal number of clusters, they use gene resampling. Suppose we want to cluster G genes $\{g_1,\dots,g_G\}$ into k non-overlapping clusters $\{C_1,\dots,C_k\}$. The consensus matrix $\mathscr{M}$, which stores for each pair of items the proportion of clustering runs in which two items are clustered together, is defined through the connectivity matrix $M^{(b)}$ and the indicator matrix $I^{(b)}$, for each perturbed dataset $D^{(b)}$, obtained by resampling the original

dataset D,

$$M^{(b)}(i,j) = \begin{cases} 1, & \text{if items } i \text{ and } j \text{ belong to the same cluster,} \\ 0, & \text{otherwise,} \end{cases}$$

$$I^{(b)}(i,j) = \begin{cases} 1, & \text{if } i \text{ and } j \text{ belong to dataset } D^{(b)}, \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathscr{M}(i,j) = \frac{\sum_{b=1}^{B} M^{(b)}(i,j)}{\sum_{b=1}^{B} I^{(b)}(i,j)}.$$

The consensus matrix with all entries set to either 0 or 1 gives the best clustering, with perfect consensus. Therefore, to find the optimal number of clusters, one needs to identify the consensus matrix having entries mostly 0 and 1. The idea is to construct a consensus matrix $\mathscr{M}^{(k)}$ for each partition containing k clusters, $k = 2,\ldots,K_{\max}$, and to choose the number of clusters that corresponds to the best consensus matrix, i.e. the matrix having entries mostly 0 and 1. To make the task easier, the authors define the consensus distribution and its corresponding area as

$$\mathrm{CDF}(c) = \frac{\sum_{i<j} I_{\{\mathscr{M}(i,j)\leq c\}}}{G(G-1)/2}, \quad A(k) = \sum_{i=2}^{m} (x_i - x_{i-1})\mathrm{CDF}(x_i),$$

where G is the number of rows (the number of genes) in $\mathscr{M}$, and $(x_1,\ldots,x_m)$ denotes the sorted entries of the consensus matrix $\mathscr{M}^{(k)}$, $m = (G-1)G/2$. The optimal number of clusters is chosen as $\widehat{k}_{\mathrm{opt}} = \arg\max_k \Delta(k)$, where

$$\Delta(k) = \begin{cases} A(k), & k = 2, \\ \frac{A(k+1)-A(k)}{A(k)} & k > 2, \end{cases}$$

Although several methods exist to identify the optimal number of clusters, they are usually very difficult to apply in practice due to their large computational requirements, or are not efficient on real data. This fact was noticed by Thalamuthu *et al.* (2006), whose attempts to estimate the number of clusters in real datasets were usually not successful, especially with many scattered genes and large perturbations. We will also encounter this problem in our real data application, and will therefore arbitrarily choose a number of clusters which seems reasonable, i.e. which gives modules of sizes between 5 and approximately 100 genes.

7.1.5 Assessing a partition using the Rand index

The Rand index, first described by Rand (1971), is an objective criterion for comparing two partitions of n objects based on their number of agreements in clustering a set of points. For two partitions $\mathscr{C} = \{C_1,\ldots,C_K\}$ and $\mathscr{P} = \{P_1,\ldots,P_L\}$, we define their $K \times L$ agreement matrix A, with $(A)_{ij} = a_{ij}$ representing the number of elements that are found in both C_i and P_j. In this report, we use the adjusted Rand index, which corrects for classification by chance. We define

the following quantities:

$$a_{\cdot l} = \sum_{k=1}^{K} a_{kl}, \quad a_{k\cdot} = \sum_{l=1}^{L} a_{kl}.$$

The adjusted Rand index is then defined as

$$\text{ARI} = \frac{\left\{\sum_{k,l}\binom{a_{kl}}{2}\right\} - \left\{\sum_{k}^{K}\binom{a_{k\cdot}}{2}\sum_{l}^{L}\binom{a_{\cdot l}}{2}/\binom{N}{2}\right\}}{\frac{1}{2}\left\{\sum_{k}\binom{a_{k\cdot}}{2} + \sum_{l}\binom{a_{\cdot l}}{2}\right\} + \left\{\sum_{k}\binom{a_{k\cdot}}{2}\sum_{l}^{L}\binom{a_{\cdot l}}{2}/\binom{N}{2}\right\}}.$$

A value of ARI close to 1 indicates good agreement between the two partitions. Other measures for evaluating partitions obtained from different clustering algorithms are described by Gan *et al.* (2007, Chapter 17).

7.2 Simulations

In Section 6.4, we estimated a common correlation matrix using an empirical Bayes procedure. We now want to define clusters or modules. To this end, we use some of the clustering algorithms described in Section 7.1. We compare the GAP statistic and consensus clustering (Section 7.1.4) combined with different clustering methods, and also compare our procedure's performance with other methods, WGCNA and tight clustering, described in Sections 7.1.2 and 7.1.3. To assess the performance of the different methods, we either use the Rand index (Section 7.1.5), or

$$\text{sensitivity} = \frac{\text{NTP}}{\text{NTP} + \text{NFN}}, \quad \text{specificity} = \frac{\text{NTN}}{\text{NTN} + \text{NFP}},$$

where

- NTP, the number of true positives, counts the number of genes that are correctly attributed to a module;
- NTN, the number of true negatives, those genes that are not in the module and that should not be;
- NFN is the number of false negatives, genes that were not in the module but should have been;
- NFP is the number of false positives, genes that are in the module but should not have been.

The simulation design used to assess the performance of the different methods is either that from Figure 6.8, or the following: from the gene expression matrix, generated according

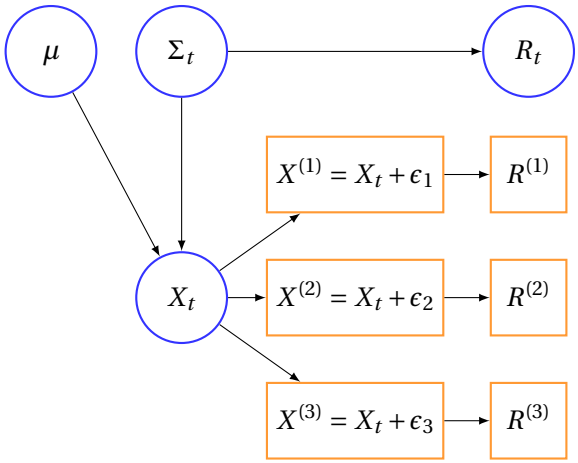


Figure 7.1 – Simulation design used to assess the model. X_t is the true gene expression matrix to which we add noise $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ to obtain noisy realisations. R_t is the true correlation matrix corresponding to X_t, while $R^{(i)}$ are the correlation matrices of the noisy gene expression matrices.

to the procedure defined in Section 4.1.1, we extract the covariance matrix Σ_t, which we transform to a correlation matrix R_t. We then add Gaussian noise to the gene expression matrix, producing three noisy gene expression matrices. For each we compute the matrix of pairwise correlations, which become the inputs of our model. The procedure is illustrated in Figure 7.1, and will be particularly useful for comparing our method with WGCNA and tight clustering, as these two last methods require gene expression matrices as input. Unless otherwise specified, parameters for the simulation are taken as follows: the number of genes is $G = 100$, the number of studies $L = 3$, the number of modules $B = 5$, and the number of simulations $R_{\text{sim}} = 100$. The distance matrix used for a clustering algorithm is $1_G 1_G^T - \tilde{R}$, where $\tilde{R}$ is the matrix estimated from our procedure described in Section 6.4. Several clustering algorithms will be compared in Section 7.3.

7.2.1 Simulations for module detection

We first want to assess the performance of our model combined with different clustering techniques on module detection. Using the simulation design presented in Figure 6.8, which adds noise to the correlation matrix generated according to equation (6.7), we use block correlations $\rho_b \sim \mathcal{U}(0.5, 1)$, producing easy-to-detect modules, and $\rho_b \sim \mathcal{U}(0, 1)$, making modules sightly more difficult to detect, for each block A_b of R, $b = 1, \ldots, 5$. We also use ρ as a tuning parameter, taking values between 0.1 and 0.9, by imposing all blocks to have the same correlation, and using Gaussian noise with fixed variance $\sigma_\epsilon = 0.5$. Results presented in Tables 7.1 and 7.2 correspond to a hierarchical clustering algorithm with Ward's link and the number of clusters, $B = 5$, provided to the algorithm. A summary of the parameters used for these simulations along with their values is presented in Table A.7 of the Appendix.

Table 7.1 – Module detection by our method for the simulation design of Figure 6.8 adding noise, $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$, to the correlation matrix generated according to (6.7). The table presents the mean and standard deviation (%) for the Rand index, sensitivity, and specificity over 100 simulated data.

	$\rho \sim \mathcal{U}(0.5, 1)$			$\rho \sim \mathcal{U}(0, 1)$		
σ_ϵ	Rand index(%)	Sensitivity(%)	Specificity(%)	Rand index(%)	Sensitivity(%)	Specificity(%)
0.1	100 (0.0)	100 (0.0)	100 (0.0)	95.7 (10.4)	96.0 (9.0)	99.5 (1.3)
0.2	100 (0.0)	100 (0.0)	100 (0.0)	91.0 (12.6)	90.5 (12.4)	98.9 (1.7)
0.3	100 (0.0)	100 (0.0)	100 (0.0)	88.0 (18.2)	88.6 (15.1)	98.5 (2.5)
0.4	100 (0.4)	99.9 (1.3)	100 (0.1)	78.1 (23.1)	82.2 (17.6)	97.4 (2.9)
0.5	98.8 (3.8)	98.1 (6.2)	99.9 (0.2)	66.2 (29.0)	73.4 (20.3)	95.9 (3.9)
0.6	96.6 (6.8)	95.4 (9.7)	99.8 (0.4)	64.1 (25.8)	71.1 (17.6)	95.7 (3.4)
0.7	94.9 (07.7)	94.0 (10.3)	99.5 (0.7)	52.2 (27.1)	64.4 (17.6)	94.2 (3.5)
0.8	91.2 (10.9)	91.7 (11.7)	99.1 (1.2)	48.0 (29.0)	60.0 (19.0)	93.5 (4.1)
0.9	89.2 (10.8)	90.5 (11.6)	98.9 (1.1)	45.8 (27.6)	59.3 (18.0)	93.3 (4.0)
1	80.5 (17.5)	83.0 (15.4)	98.0 (2.0)	37.5 (24.6)	54.6 (15.8)	92.1 (3.8)

Table 7.2 – Module detection using ρ, the intra-module correlation, as a tuning parameter. The table presents the mean and standard deviation (%) of the Rand index, sensitivity and specificity over 100 simulated datasets, with fixed noise variance $\sigma_\epsilon = 0.5$.

ρ	Rand index(%)	Sensitivity(%)	Specificity(%)
0.1	0.1(1.1)	28.2 (5.3)	85.2 (1.0)
0.2	6.1(11.2)	33.1 (9.8)	86.8 (2.5)
0.3	43.7 (20.3)	59.2 (14.9)	93.4 (2.7)
0.4	82.0 (8.6)	85.7 (11.5)	98.2 (0.9)
0.5	93.9 (6.8)	93.7 (10.3)	99.5 (0.4)
0.6	98.6 (3.5)	98.3 (5.6)	99.9 (0.2)
0.7	99.6 (2.1)	99.2 (4.1)	100 (0.1)
0.8	100 (0.0)	100 (0.0)	100 (0.0)
0.9	100 (0.0)	100 (0.0)	100 (0.0)

In Table 7.1, the performance of our method in terms of detection of modules is almost perfect if the correlation is large, i.e., $\rho \sim \mathcal{U}(0.5,1)$ for every noise level. When the correlations are smaller, the performance decreases, especially when the noise level is big, as expected.

In Table 7.2, we see that our method detects the modules almost perfectly, as soon as the correlation is large enough, $\rho \geq 0.4$ inside a module, exactly as observed in Table 7.1. For smaller correlations, the performance is less good but still reasonable for $\rho \geq 0.3$. In these simulations, the modules all have the same sizes and in Table 7.2 they all have the same correlation ρ. We also performed simulations with varying module sizes and with varying module correlations, $\rho_b \sim \mathcal{N}(\rho, \sigma_\rho)$ for module b, with ρ and σ_ρ fixed, but the results were very similar to those presented in Tables 7.1 and 7.2.

7.2.2 Comparison of clustering methods

We saw in the previous section that our empirical Bayes model associated with hierarchical clustering using the Ward's link seems to perform quite well concerning module detection when the number of modules is provided to the algorithm. However, we only used hierarchical clustering and the number of modules was assumed to be known. In this section, we compare hierarchical clustering with average, complete, single and Ward's links, and k-means. We also compare two methods for selecting the optimal number of clusters: the GAP statistic and consensus clustering (CC). We use the simulation design of Figure 6.8, with a correlation ρ inside each module varying from 0.1 to 0.9, a fixed noise level $\sigma_\epsilon = 0.5$, and $B = 5$ clusters. Table 7.3 compares these clustering techniques based on the average number of clusters detected by the GAP statistic and the average Rand index over 100 simulations. The GAP statistic only starts to be powerful when modules have large correlations. For values of ρ below 0.5, it systematically finds only one module, i.e., one cluster, as being the optimal number. When the correlation is large, the GAP statistic can correctly identify that there are five modules. It seems from this table that using a hierarchical clustering method with Ward's link and the GAP statistic is the best amongst all methods compared.

Figure 7.2 compares the Rand index obtained from several clustering methods using either the GAP statistic or consensus clustering to determine the optimal number of clusters. From the figure, it becomes clear that the GAP statistic with any method outperforms consensus clustering. Once again, hierarchical clustering with Ward's link and the GAP statistic to select the number of modules seems to be the best choice in our simulations.

7.2.3 Comparison with WGCNA

In this section, we compare our method's performance with that of WGCNA, described in Section 7.1.2. Langfelder and Horvath (2007) provide in supplementary material a simulation design that is supposed to mimic real gene expression data, and is coded in their package. We first used their simulation design with the following settings: 3 studies, 100 simulations,

Table 7.3 – Comparison of different clustering methods. The first line for each method gives the average number of clusters selected using the GAP statistic and the second line gives the mean and standard deviation of the Rand index of the final partition compared to the truth. The simulation design produces clusters of varying sizes, each having the same correlation ρ.

	ρ	0.1	0.2	0.3	0.4	0.5
HC Ward	clusters	1 (0)	1 (0)	1 (0)	1 (0)	1.64 (0.7)
	Rand index	0 (0)	0 (0)	0 (0)	0 (0)	0.3 (0.3)
HC complete	clusters	1 (0)	1(0)	1 (0)	1.08 (0.3)	1.81 (0.9)
	Rand index	0 (0)	0 (0)	0 (0)	0.04 (0.16)	0.32 (0.35)
HC average	clusters	1 (0)	1 (0)	1 (0)	1 (0)	1.27 (0.6)
	Rand index	0 (0)	0 (0)	0 (0)	0 (0)	0.13 (0.27)
HC single	clusters	1 (0)	1 (0)	1.02 (0.14)	1.05 (0.22)	1.10 (0.3)
	Rand index	0 (0)	0 (0)	$10^{-5}(10^{-3})$	0.001 (0.004)	0.004 (0.02)
k-means	clusters	1 (0)	1 (0)	1 (0)	1 (0)	1.15 (0.39)
	Rand index	0 (0)	0 (0)	0 (0)	0 (0)	0.09 (0.23)
	ρ	0.6	0.7	0.8	0.9	
HC Ward	clusters	4.07 (0.6)	4.78 (0.4)	4.99 (0.1)	5 (0)	
	Rand index	0.95 (0.06)	0.99 (0.009)	0.99 (0.001)	1 (0)	
HC complete	clusters	3.48 (1.16)	4.43 (1.07)	4.85 (0.62)	5.06 (0.42)	
	Rand index	0.73 (0.33)	0.85 (0.24)	0.95 (0.12)	0.99 (0.04)	
HC average	clusters	2.93 (1.38)	4.42 (1.09)	4.79 (0.84)	4.96 (0.4)	
	Rand index	0.62 (0.40)	0.91 (0.25)	0.95 (0.20)	0.99 (0.10)	
HC single	clusters	1.66 (0.88)	3.62 (1.55)	4.92 (0.94)	5.05 (0.54)	
	Rand index	0.11 (0.20)	0.59 (0.38)	0.93 (0.23)	0.98 (0.11)	
k-means	clusters	2.56 (0.97)	3.26 (0.85)	3.17 (1.03)	3.29 (1.03)	
	Rand index	0.49 (0.32)	0.62 (0.22)	0.58 (0.23)	0.60 (0.24)	

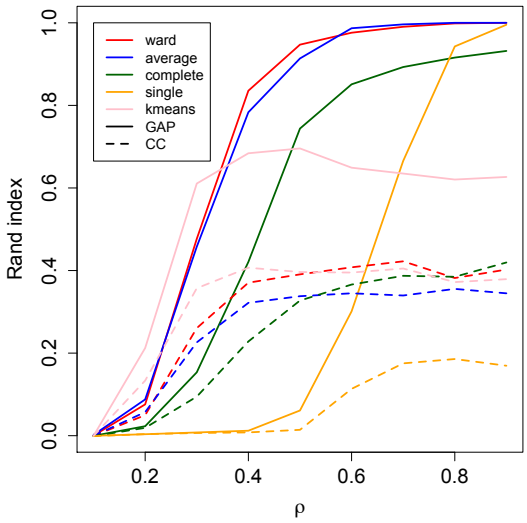


Figure 7.2 – Comparison of different clustering methods: hierarchical clustering with Ward's, average, complete or single linkage and k-means, and techniques to determine the optimal number of clusters, GAP statistic and consensus clustering (CC).

Nmodules = 5, Npatients = 50, nGenes = 100, the eigengenes were generated according to a standard normal distribution, with 20% of the genes in each module, and no grey genes (i.e. scattered genes, see Section 7.2.4), the minimum correlation was taken to be 0.1 or 0.5, and the maximum correlation was bounded by 1, background noise ranged between 0 and 0.3 with six levels as stated in Langfelder and Horvath (2007), and no negative correlation was used.

In Table 7.4, we show the Rand index, sensitivity and specificity of module detection for two versions of the design of Langfelder and Horvath (2007), and for our simulation design (Figure 7.1), with varying ρ and constant noise variance $\sigma_\epsilon = 0.5$. Our approach performs better in detecting modules, even when applied to the design developed by Langfelder and Horvath (2007). The difference between the methods is more striking for small correlations (mincor= 0.1). However, the different levels of noise added to the design of Langfelder and Horvath (2007) do not seem to make big differences in the results. For our simulation design, where we add noise to the gene expression matrix (design illustrated in Figure 7.1), we also find that our approach performs better in detecting modules. This simulation design leads to better performance for our method than that where the noise is added directly to the correlation matrix, as seen in Table 7.2.

To better see the difference in performance between the two methods in terms of the Rand index, with respect to the intra-module correlation ρ, we present the results in Figure 7.3. It is again clear that our empirical Bayes model performs better in terms of the Rand index. Note that in these simulations, we use the fixed cut tree procedure, with the number of clusters

Table 7.4 – Comparison between our empirical Bayes method (EB) and WGCNA, based on the design of Langfelder and Horvath (2007) (WGCNA) and the design presented in Figure 7.1. We present the mean and standard deviations (%) of the Rand index, sensitivity and specificity.

mincor=0.1, WGCNA design						
	EB			WGCNA		
Noise	Rand index(%)	Sensitivity(%)	Specificity(%)	Rand index(%)	Sensitivity(%)	Specificity(%)
0	36.2 (5.5)	73.0 (2.9)	93.7 (0.6)	1.8 (7.2)	25.5 (5.8)	84.3 (1.1)
0.06	35.4 (4.9)	72.6 (2.6)	93.6 (0.6)	6.3 (13.9)	29.1 (10.7)	85.0 (2.0)
0.12	35.5 (4.7)	72.6 (2.6)	93.6 (0.6)	4.8 (11.4)	28.3 (9.5)	84.8 (1.8)
0.18	34.4 (5.1)	72.0 (2.8)	93.5 (0.6)	2.8 (8.4)	26.5 (6.7)	84.5 (1.2)
0.24	34.4 (5.2)	72.0 (2.9)	93.5 (0.6)	2.3 (8.1)	26.0 (6.3)	84.4 (1.2)
0.3	33.5 (4.6)	71.5 (2.6)	93.4 (0.6)	4.1 (9.1)	27.9 (7.8)	84.7 (1.4)
mincor=0.5, WGCNA design						
	EB			WGCNA		
Noise	Rand index(%)	Sensitivity(%)	Specificity(%)	Rand index(%)	Sensitivity(%)	Specificity(%)
0	99.9 (0.5)	100 (0.2)	100 (0.0)	99.7 (2.4)	99.8 (2.0)	99.9 (0.5)
0.06	99.8 (0.8)	99.9 (0.3)	100 (0.1)	99.8 (2.4)	99.8 (2.0)	100 (0.5)
0.12	99.9 (0.4)	100 (0.2)	100 (0.0)	99.8 (2.4)	99.8 (2.0)	100 (0.5)
0.18	99.7 (0.9)	99.9 (0.3)	100 (0.1)	99.7 (2.4)	99.8 (2.0)	99.9 (0.5)
0.24	99.8 (0.8)	99.9 (0.3)	100 (0.1)	100 (0.0)	100 (0.0)	100 (0.0)
0.3	99.9 (0.6)	100 (0.2)	100 (0.1)	99.1 (4.5)	99.2 (3.7)	99.8 (0.9)
Simulation design of Figure 7.1						
	EB			WGCNA		
ρ	Rand index(%)	Sensitivity(%)	Specificity(%)	Rand index(%)	Sensitivity(%)	Specificity(%)
0.1	19.3 (10.0)	43.7 (8.3)	89.5 (1.7)	4.8 (5.9)	31.8 (6.2)	86.3 (1.4)
0.2	62.1 (13.1)	75.1 (12.7)	95.8 (1.5)	29.7 (18.7)	51.5 (13.8)	90.2 (2.9)
0.3	88.8 (7.4)	92.2 (9.2)	98.8 (0.7)	68.3 (19.7)	76.7 (15.3)	95.4 (3.1)
0.4	97.6 (3.3)	98.0 (4.9)	99.8 (0.3)	92.1 (11.3)	94.6 (8.4)	98.8 (1.8)
0.5	99.8 (0.7)	99.9 (0.5)	100 (0.1)	99.5 (2.2)	99.1 (4.0)	99.9 (0.3)
0.6	100 (0.2)	100 (0.1)	100 (0.0)	100 (0.3)	100 (0.1)	100 (0.0)
0.7	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)
0.8	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)
0.9	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)

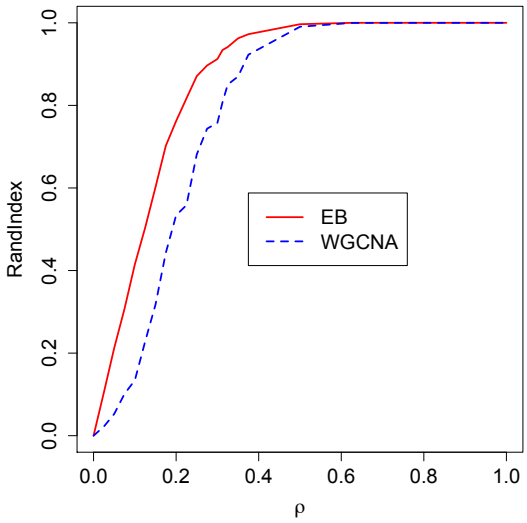


Figure 7.3 – Rand index of module detection as a function of ρ for our procedure (EB) and for WGCNA.

given to both methods, to define the clusters. Langfelder and Horvath (2007) suggest using their dynamic approach, which may give different results, and which we will use in the next section when introducing scattered genes.

7.2.4 Simulations with scattered genes

Scattered genes are genes that do not belong to any module and therefore are assumed to have no or low correlation with other genes. This type of genes is also studied by Tseng and Wong (2005) and Langfelder and Horvath (2007), who both developed methods to identify them.

Thalamuthu *et al.* (2006) compares six clustering methods based on their ability to detect groups of genes in simulated microarray gene expression data. They conclude that in the presence of scattered genes, tight clustering and model based clustering are the best methods overall, while all methods except SOM recover the clusters accurately when no scattered genes are present. Our approach follows this recommendation in the sense that, if we can identify the scattered genes prior to clustering, and remove them from the correlation matrix, most clustering methods will be appropriate in detecting modules. According to Figure 7.2, we choose a hierarchical clustering approach with Ward's link, which we apply after having removed scattered genes.

In order to detect scattered genes, we tried several statistics, including the posterior probability of being scattered, T_q, the sum and variance of absolute correlations, $T_{\rho,\text{sum}}$ and $T_{\rho,\text{var}}$, and the sum and variance of absolute Fisher transformed matrix, $T_{\theta,\text{sum}}$ $T_{\theta,\text{var}}$. The posterior probabil-

ity of being scattered is based on the Fisher transformed matrix, with entries corresponding to the parameter θ, as defined in Section 6.4. Then for each gene g, the posterior probability of being scattered corresponds to the probability that each entry of the gth row is null, except for the diagonal,

$$\begin{aligned} T_{q,g} &= \mathrm{P}\left(\theta_{g,1},\cdots,\theta_{g,G}=0 \mid Z_{g,1}^{(1)},\cdots,Z_{g,1}^{(L_1)},\cdots,Z_{g,G}^{(1)},\cdots,Z_{g,G}^{(L_1)}\right) \\ &= \frac{\prod_{i\neq g} f\left(Z_{gi}^{(1)},\cdots,Z_{gi}^{(L_1)} \mid \theta_{gi}=0\right)\mathrm{P}\left(\theta_{gi}=0\right)}{\prod_{i\neq g}\int_{\theta_{gi}} f\left(Z_{gi}^{(1)},\cdots,Z_{gi}^{(L_1)} \mid \theta_{gi}\right)\pi(\theta_{gi})\mathrm{d}\theta_{gi}} \\ &= \frac{(1-p)^{G-1}\prod_{i\neq g} a_{gi}}{\prod_{i\neq g}\left[(1-p)a_{gi}+\frac{pa_{gi}}{\sqrt{br^2}}\exp\left\{\left(\sum_{l=1}^{L_1} Z_{gi}^{(l)}/\sigma_l^2\right)^2(2b)^{-1}\right\}\right]}, \end{aligned}$$

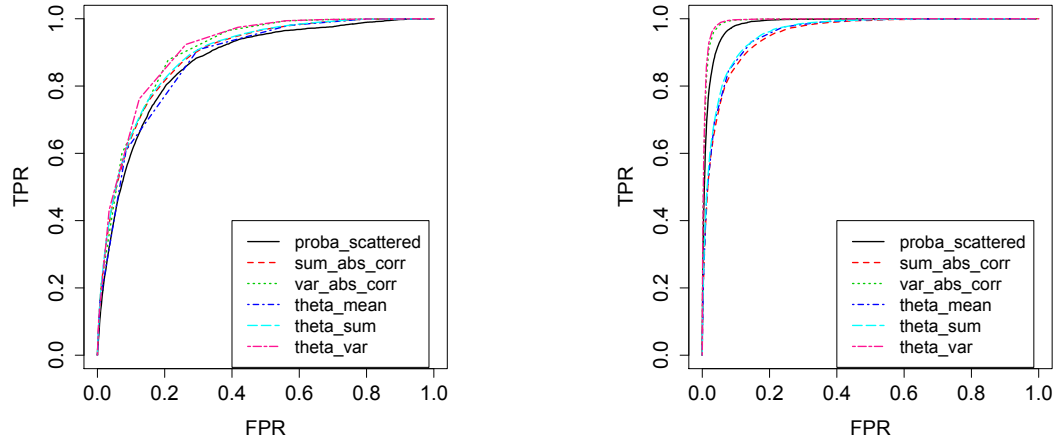
where $a_{gi}=\prod_{l=1}^{L_1}\sigma_l^{-1}\varphi(Z_{gi}^{(l)}/\sigma_l)$, and φ is the standard normal density. For numerical reasons, we prefer to use the log posterior probability of being scattered. The sum and variance of the Fisher transformed matrix are simply obtained as

$$T_{g,\theta,\mathrm{sum}}=\sum_{i\neq g}|\tilde{\theta}_{gi}|, \quad T_{g,\theta,\mathrm{var}}=\mathrm{var}\left(|\tilde{\theta}_{gi}|\right), \quad i\neq g, \tag{7.2}$$

where $\tilde{\theta}$ is the posterior median of θ, as defined in Section 6.4. The sum and variance of the absolute correlations are obtained as the sum and variance for each row of the estimated correlation matrix. For these last four quantities, we also work with $-\log(T)$, to have a wider scale, and scattered genes are identified as the genes with the largest values of $-\log(T)$. We study the power of each statistic in terms of detection of scattered genes, for different values of ρ in Figure 7.4, where we simulated 100 datasets with $G=1000$ genes of which 5% are scattered, and $\rho_b=0.25$, 0.5, 0.75 and 0.9, for all $b=1,\ldots,10$, in (6.7). All statistics seem to have high power of detection of scattered genes, with a slight advantage for the statistics based on the variance of either the absolute correlations or the absolute Fisher values, $T_{\theta,\mathrm{var}}$ and $T_{\rho,\mathrm{var}}$. Looking at one particular simulation in Figure 7.5, we see that $T_{\theta,\mathrm{var}}$ seems to discriminate scattered genes the best. However, even if these statistics seem to have high power, meaning that taking the genes with largest values most probably leads to a list of scattered genes, it is hard to define a meaningful threshold, mainly because we do not know the distribution of these statistics. We tried resampling to obtain a null distribution with which to compare the observed value, but this is computationally intensive, especially since we work with high-dimensional data. We are looking for a threshold in the tail of the distribution of our statistic, from which we could decide whether a gene is scattered or not.

As we are working with the tail of a distribution, extreme value theory can be useful in helping to determine a threshold. The mean residual life plot, defined as

$$\mathbb{E}(X-u \mid X>u)=\frac{\sigma+\xi u}{1-\xi},$$



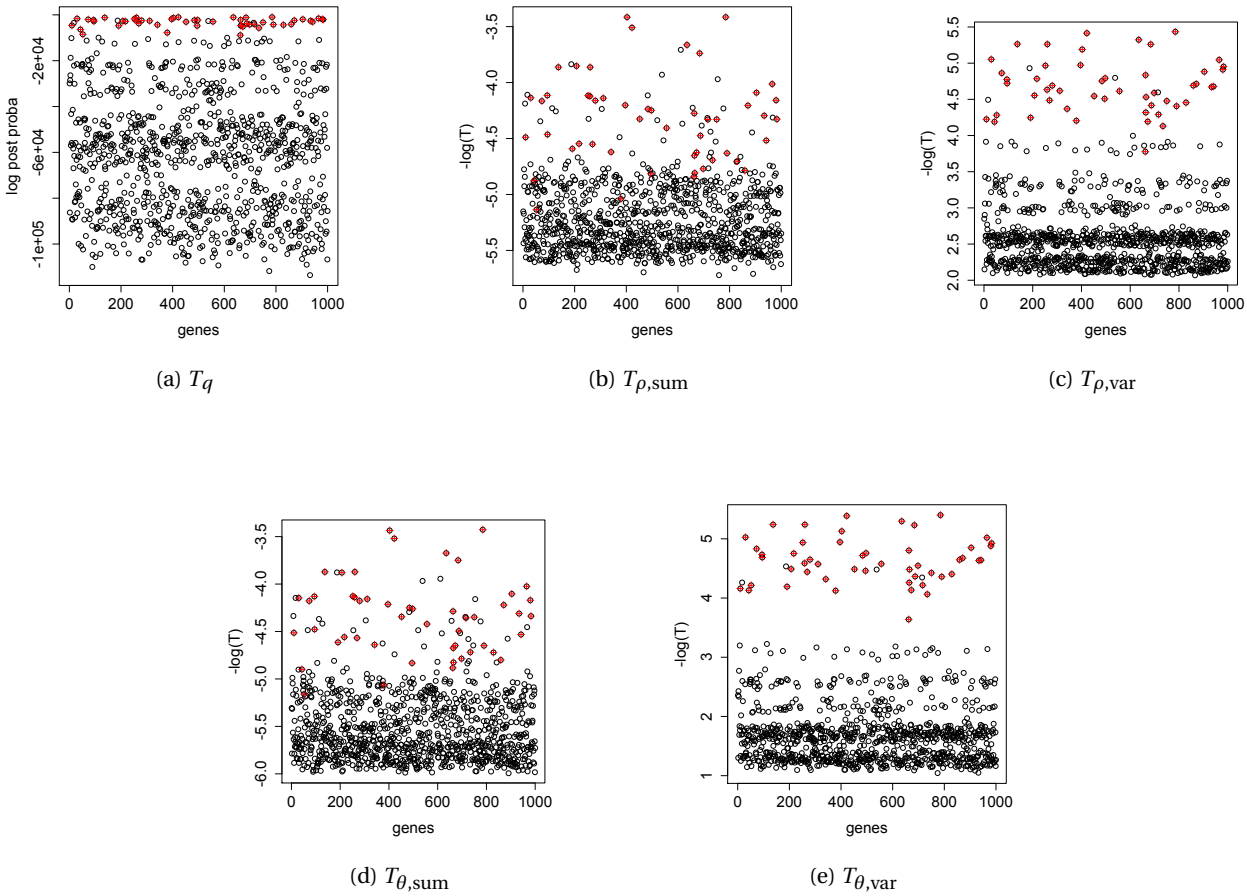
(a) $\rho = 0.25$

(b) $\rho = 0.5$

(c) $\rho = 0.75$

(d) $\rho = 0.9$

Figure 7.4 – Power of different statistics to detect scattered genes for some values of the intra-module correlation, $\rho_b = (0.25, 0.5, 0.75, 0.9)$, $b = 1, \ldots, 10$ in equation 6.7.



(a) T_q (b) $T_{\rho,\text{sum}}$ (c) $T_{\rho,\text{var}}$

(d) $T_{\theta,\text{sum}}$ (e) $T_{\theta,\text{var}}$

Figure 7.5 – Detection of scattered genes (plain red points) for the different statistics, and for $\rho = 0.9$ in (6.7) in one simulation. $T_{\theta,\text{var}}$ better discriminates between scattered and non-scattered genes.

where σ and ξ are the parameters of the generalized Pareto distribution, is a useful tool to help determine the threshold, as it should be linear in u for all values above the threshold u. In Figure 7.6, where we present the mean residual life plot for $-\log(T_{\theta,\text{var}})$, we can clearly identify linearity after $u = 3$, which seems to correspond to a reasonable value to discriminate scattered genes from other genes, in the left panel of the same figure.

Threshold selection remains an open question in statistics of extreme values, although several methods have been developed, e.g., by Wadsworth and Tawn (2012), who also provide a good review of the topic. In our context, we need to find a method to identify scattered genes. We know that the tail of the distribution of our statistic mostly contains scattered genes but can have other genes mixed in too. However, we also know that very far in the tail, or put differently, the most extreme values, are likely to be scattered genes. We decide to model the tail of the distribution as a mixture of two generalized Pareto distributions (GPD), one for the scattered

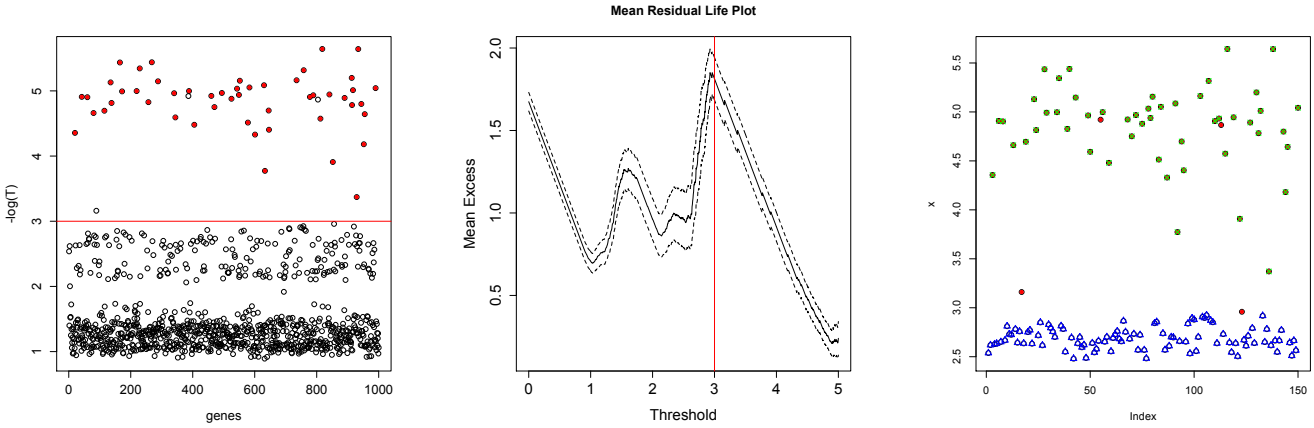


Figure 7.6 – Identification of scattered genes using $-\log(T_{\theta,\text{var}})$ as the statistic, for one simulated dataset and $\rho = 0.9$. *Left*: values of $-\log(T_{\theta,\text{var}})$ with true scattered genes as red plain dots; *center*: mean residual life plot; *right*: values of $-\log(T_{\theta,\text{var}})$ above the 90% quantile, with scattered genes identified by our procedure (red plain dots) corresponding to all values of $-\log(T_{\theta,\text{var}}) > 2.9$, true scattered genes (green crosses) and other genes (blue triangles). All true scattered genes are correctly identified.

genes with parameters σ_1 and ξ_1 and the other with parameters σ_2 and ξ_2. To obtain more accurate estimates, we assume that $\xi_1 = \xi_2 = \xi$, as suggested by Rootzén and Zholud (2014). For an observed value of our statistic t above a threshold u, we have the following density,

$$f(t) = p f_1(t;\sigma_1,\xi) + (1-p) f_2(t;\sigma_2,\xi), \quad t > u, \quad f_i(t;\sigma_i,\xi) = \frac{1}{\sigma_i}\left(1+\xi\frac{x}{\sigma_i}\right)^{-1/\xi-1}, \quad i = 1,2. \tag{7.3}$$

In order to ensure that the observed values are a mixture of scattered and non-scattered genes in the tail, we take u to be a large enough quantile of our observed statistic. The likelihood can be easily obtained, and parameters are estimated by maximum likelihood. Identification of scattered genes is done by computing the posterior probability of belonging to the first mixture distribution for each gene g, based on the value of the corresponding statistic t_g defined in equation (7.2),

$$p_{g,\text{scattered}} = \frac{\widehat{p} f_1(t_g;\widehat{\sigma}_1,\widehat{\xi})}{\widehat{p} f_1(t_g;\widehat{\sigma}_1,\widehat{\xi}) + (1-\widehat{p}) f_2(t_g;\widehat{\sigma}_2,\widehat{\xi})}.$$

We fitted this model to the simulated dataset of Figure 7.6. Estimates of the parameters with corresponding standard errors are presented in Table 7.5. Using the estimated posterior probability of belonging to the first distribution, detects all scattered genes properly and gives a few false positives (50 true positives and 4 false positives, for this particular simulation), as illustrated in the right pannel of Figure 7.6.

Table 7.5 – Estimates of the parameters of the mixture of GPDs, with corresponding standard errors, for one simulation and for $\rho_b = 0.9$, $b = 1, \ldots, B = 10$, in (6.7).

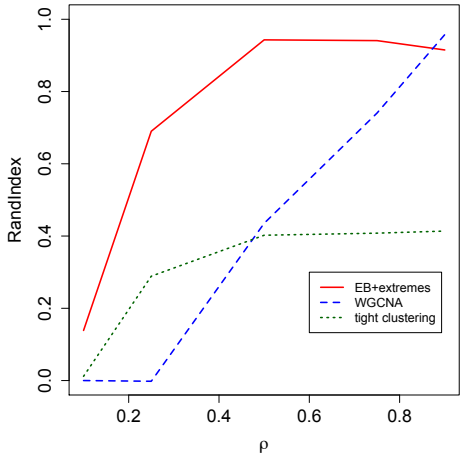
$\widehat{\sigma}_1$	$\widehat{\sigma}_2$	$\widehat{\xi}$	logit($\widehat{p}$)
3.46 (1.33)	0.50 (0.16)	−1.08(0.37)	logit(0.43)=−0.28 (0.21)

We now compare our empirical Bayes approach using the GPD mixture to detect scattered genes, with WGCNA and tight clustering, in terms of detection of scattered genes and accuracy of the gene modules. To assess the performance of each method, we record the number of true and false scattered genes, sensitivity and specificity regarding the detection of scattered genes and the Rand index of the final partition compared to the truth. In all three methods the detection of scattered genes is completely automatic, but the number of modules is provided to the algorithms. For our method, scattered genes are detected by the extreme value method, removed, and the resulting correlation matrix is used as input to the hierarchical clustering algorithm with Ward's linkage. Tight clustering and WGCNA have their own method to detect scattered genes. We compute the Rand index on the entire set of genes by considering the scattered genes to be a module.

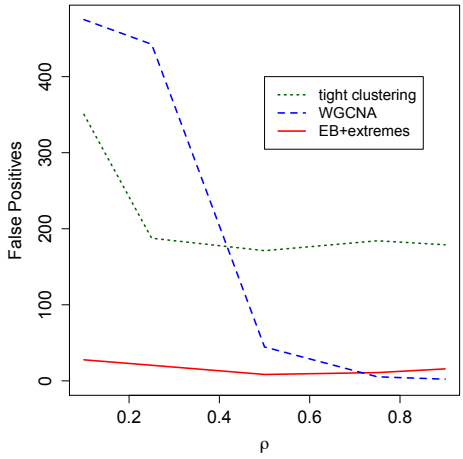
In these simulations, we used $G = 500$ genes, of which 25 are scattered, $B = 10$ modules, and performed 100 simulations for each value of the intra-module correlation $\rho_b = 0.1, \ldots, 0.9$, for all $b = 1, \ldots, 10$, in (6.7). Results of these simulations are presented in Table 7.6 and Figure 7.7. In both, the Rand index for our method is equivalent or better than for other methods, and the difference is more striking for smaller correlations. We see that the scattered gene detection using the GPD mixture tends to be quite conservative, with very few false positives. On the other hand, the number of true positives is a little lower than with the other methods. The plots of true positives and false positives, as well as specificity and sensitivity plots in Figure 7.7, cannot be looked at separately. Indeed, one seeks to obtain many true positives associated with very few false positives. Therefore, even if WGCNA seems to perform very well in detecting scattered genes according to the plots of true positives, when looking at the false positives plot, we see it actually selects all genes to be scattered, and therefore has many false positives too. Therefore, looking at all plots of Figure 7.7 and at Table 7.6, we conclude that our approach is a good compromise, as it has high Rand index, and tends to select scattered genes properly. We would rather have low false positive rate for our method, eventually associated with low true positive rate, as scattered genes are removed from the clustering process, and therefore excluded from modules. So we prefer to have some scattered genes inside modules, rather than correlated genes outside modules.

Table 7.6 – Comparison of our empirical Bayes method with GPD mixture to detect scattered genes, WGCNA and tight clustering. The Rand index is based on all clusters, with scattered genes included as a cluster. Numbers of true positives (TP), which should equal 25, false positives (FP), which is smaller than 475 and should equal 0 ideally, sensitivity and specificity concern the detection of scattered genes only. The Rand index, sensitivity, specificity and their corresponding standard deviations are in percent, whereas TP, FP and corresponding standard deviations are in number of genes.

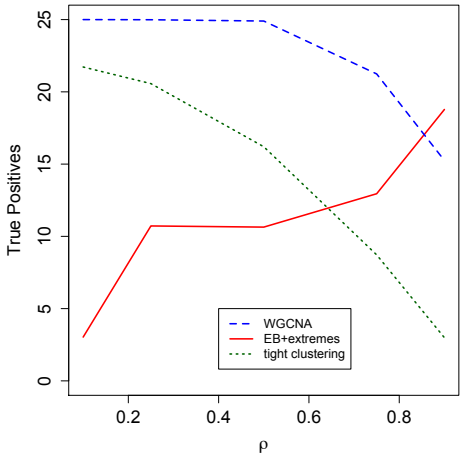
	ρ	0.1	0.25	0.5	0.75	0.9
Rand index	EB	13.9 (4.3)	69.0 (6.7)	94.3 (6.1)	94.1 (6.9)	91.5 (7.8)
(%)	WGCNA	0.0 (0.0)	0.0 (0.7)	43.5 (9.0)	74.1 (8.2)	95.8 (1.9)
	tight	1.2 (1.9)	28.9 (5.5)	40.2 (7.7)	40.8 (7.8)	41.4 (7.6)
TP	EB	3.0 (2.5)	10.7 (6.3)	10.6 (9.9)	12.9 (11.3)	18.8 (10.1)
	WGCNA	25.0 (0.0)	24.9 (0.1)	24.9 (0.3)	21.2 (2.0)	15.3 (2.5)
	tight	21.7 (2.0)	20.6 (2.9)	16.2 (9.7)	8.7 (10.2)	3.0 (6.0)
FP	EB	27.7 (19.5)	20.5 (15.5)	8.4 (11.8)	10.9 (12.1)	15.8 (11.5)
	WGCNA	475.0 (0.0)	442.6 (19.7)	44.6 (21.5)	5.3 (3.8)	2.2 (2.4)
	tight	350.3 (25.4)	187.6 (19.2)	171.2 (17.6)	184.2 (17.9)	178.9 (17.4)
sensitivity	EB	12.1 (10.1)	42.9 (25.1)	42.6 (39.5)	51.8 (45.1)	75.1 (40.3)
(%)	WGCNA	100.0 (0.0)	99.9 (0.4)	99.6 (1.2)	84.9 (8.0)	61.0 (10.0)
	tight	86.9 (7.8)	82.3 (11.6)	64.8 (38.7)	34.9 (40.8)	11.9 (24.1)
specificity	EB	94.2 (4.1)	95.7 (3.3)	98.2 (2.5)	97.7 (2.5)	96.7 (2.4)
(%)	WGCNA	0.0 (0.0)	6.8 (4.2)	90.6 (4.5)	98.9 (0.8)	99.5 (0.5)
	tight	26.3 (5.3)	60.5 (4.0)	63.9 (3.7)	61.2 (3.8)	62.3 (3.7)



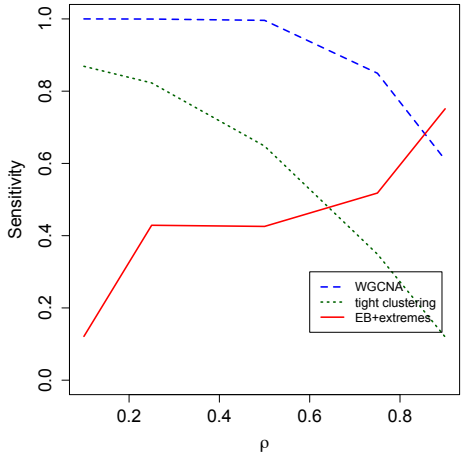
(a) Rand index



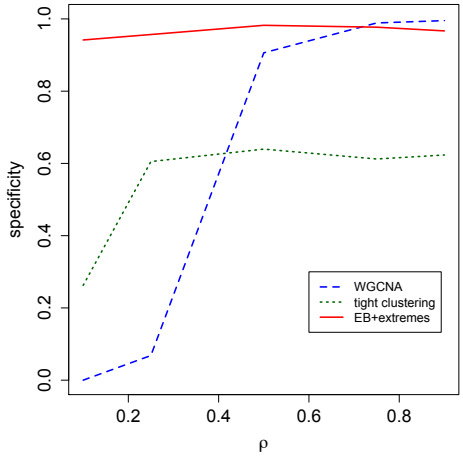
(b) False positives



(c) True positives



(d) Sensitivity



(e) Specificity

Figure 7.7 – Rand index, number of false positives, number of true positives, sensitivity and specificity for the simulations with scattered genes for the comparison of our method using the GPD mixture to select scattered genes, WGCNA and tight clustering.

7.3 Modules in real data

Modules provide a way to reduce the dimension of our problem. We now want to apply the model developed in Chapter 3 to modules defined by our empirical Bayes model and clustering method. Model (3.2) only needs slight modifications to take into account module sizes. We detail the modifications for each data type separately. Suppose we found m modules $\mathcal{M}_1,\dots,\mathcal{M}_m$, of sizes $M_1,\dots,M_m$. We now want to summarize the G genes into m module representatives.

For Type 1 data, on which the modules were defined, we need to obtain an $m \times N$ matrix of module representatives. Langfelder and Horvath (2007) suggest using the module eigengene, which is the first right singular vector of the data matrix restricted to the set of genes of each module. This choice is motivated by the fact that the first singular vector, which also corresponds to the first principal component, explains the most variability of the genes in this module. Denoting by $Y^{(\mathcal{M}_i)}$ the data matrix restricted to genes of module $\mathcal{M}_i$, the eigengene e_i is defined using the singular value decomposition of $Y^{(\mathcal{M}_i)}$,

$$Y^{(\mathcal{M}_i)} = U^{(\mathcal{M}_i)} D^{(\mathcal{M}_i)} V^{(\mathcal{M}_i)T}, \quad e_i = V_1^{(\mathcal{M}_i)},$$

where e_i is the first column of the matrix of right singular vectors $V^{(\mathcal{M}_i)}$. However, the distribution of the first singular vector is difficult to obtain (Johnstone, 2001), and cannot be modeled by a normal distribution, as specified in our model (3.2). For this reason, we choose to summarize the information inside each module by the average gene expression. Therefore,

$$e_i = \frac{1}{M_i} \sum_{g \in \mathcal{M}_i} Y_g,$$

where Y_g denotes the gth row of the data matrix, containing the gene expression for all N individuals. Recalling that $Y_{gj} \mid \mu_g, \beta_g^{(1)}, \sigma_g^2 \sim \mathcal{N}\left(\mu_g + \beta_g^{(1)} I_{j \le n_1}, \sigma_g^2\right)$, and noticing that genes inside a module are not independent, we get

$$\mathbb{E}(e_i) = \frac{1}{M_i} \sum_{g \in \mathcal{M}_i} \mu_g + \beta_g^{(1)} I_{\{j \le n_1\}}, \quad \operatorname{var}(e_i) = \frac{1}{M_i^2} \sum_{g \in \mathcal{M}_i} \left[\sigma_g^2 + \sum_{k \ne g} \rho_{gk} \sigma_g \sigma_k \right],$$

where $\rho_{gk} = \operatorname{corr}(Y_g, Y_k)$. We therefore choose to model the module expression $\tilde{Y}_{ij}$ as

$$\tilde{Y}_{ij} \mid \beta_i^{(1)}, \mu_i, \sigma_i \sim \mathcal{N}\left(\mu_i + \beta_i^{(1)} I_{\{j \le n_1\}}, \frac{\sigma_i^2}{M_i}\right), \quad i = 1,\dots,m, \quad j = 1,\dots,N.$$

$$\mu_i = \sum_{j=n_1+1}^{N} \tilde{Y}_{ij}$$

We use the same symbols for the parameters as in model (3.2), whereas the parameters represent different quantities. However we believe using the same symbols is more helpful than confusing.

For Type 2 data, we need to reduce the vector Z of length G to an m-dimensional vector containing the module scores. Here again, recall that gene score was modeled as

$$Z_g^{(2)} \mid \beta_g^{(2)}, \delta_g^2 \sim \mathcal{N}\left(\beta_g^{(2)} \sqrt{\nu \delta_g^2}, 1\right), \quad \nu = (N - n_1) n_1 / N.$$

To summarize gene scores into module scores, we use the maxmean statistic defined by Efron and Tibshirani (2007), which has power against both shift and scale differences. We use their definition, except that we choose to keep the sign of the shift,

$$e_i = \left(I_{\left\{\overline{s}^{(+)} > \overline{s}^{(-)}\right\}} - I_{\left\{\overline{s}^{(+)} < \overline{s}^{(-)}\right\}}\right) \max\left\{|\overline{s}^{(+)}|, |\overline{s}^{(-)}|\right\}, \quad \begin{cases} \overline{s}^{(+)} & = \sum_{g \in \mathcal{M}_i} z_g I_{\{z_g > 0\}}, \\ \overline{s}^{(-)} & = \sum_{g \in \mathcal{M}_i} (-z_g) I_{\{z_g < 0\}}. \end{cases}$$

Writting $n_+ = \#\{g : z_g > 0\}$ and $n_- = \#\{g : z_g < 0\}$, and noting that genes are not independent, we obtain

$$\begin{aligned} \mathbb{E}(e_i) &= \frac{1}{M_i} \sum_{g \in \mathcal{M}_i} \beta_g \sqrt{\nu \delta_g^2} I_{\{z_g \in \mathcal{S}\}}, \quad \mathcal{S} = \{+, -\}, \\ \operatorname{var}(e_i) &= \frac{1}{M_i^2} \left(n_+ + \sum_{g \in \mathcal{M}_i} \sum_{k \neq g} \rho_{gk} I_{\{z_g, z_k \in \mathcal{S}\}}\right). \end{aligned}$$

We therefore choose to model the module score $\tilde{Z}_i$ as

$$\tilde{Z}_i \mid \beta_i^{(2)}, \delta_i \sim \mathcal{N}\left(\beta_i^{(2)} \sqrt{\frac{\nu \delta_i^2}{v_i}}, 1\right), \quad v_i = \frac{n^{(\mathcal{S})} + \sum_{g \in \mathcal{M}_i} \sum_{k \neq g} \widehat{\rho}_{gk} I_{\{z_g, z_k \in \mathcal{S}\}}}{M_i^2}, \quad \mathcal{S} = \{+, -\},$$

where $\widehat{\rho}_{gk}$ is the (g, k) entry of the estimated common correlation matrix $\tilde{R}$ defined in Section 6.4.

For the Type 3 data, we can encounter three situations:

- all genes in the module are observed — in this case, the module is summarized exactly as the Type 2 data presented previously;
- all genes in the module are missing — in this case the entire module is missing and we impute, similarly as in Chapter 3, writting $\tilde{z}_{\text{ref}}$ for the smallest module score observed,

$$\tilde{Z}_i \sim \mathcal{N}_{-|\tilde{z}_{\text{ref}}|}^{|\tilde{z}_{\text{ref}}|}\left(\beta_i^{(3)} \sqrt{\frac{\nu \delta_i^2}{v_i}}, 1\right); \text{ or}$$

- some genes are missing and some are observed in the module, in which case we summarize exactly as for Type 2, dividing by M_i. This implies that missing genes are arbitrarily set to 0, which makes sense as missing genes are those with small z-scores, not reported as being differentially expressed in the study.

Type 4 data, i.e., ranks, need also to take into account that modules may only be partially observed. For this reason, we first transform the observed ranks to the normal distribution scale, prior to combination:

$$\tilde{R}_g = \Phi^{-1}\left(\frac{R_g}{G+1}\right), \quad g = 1,\ldots,G,$$

The module score is then computed by averaging the observed values inside the module, and modules are attributed a rank based on this score. Here we do not use the maxmean approach previously described, as we do not observe the sign of the underlying statistics. The module rank is modeled using a latent variable $\tilde{u}$ as follows:

$$\tilde{u}_i \sim \mathcal{N}\left(\beta_i^{(4)}\sqrt{\nu}, \frac{\sigma_{u,i}^2}{M_i}\right).$$

To summarize, the complete hierarchical Bayesian model for module i ($i = 1,\ldots,m$) is

$$\begin{aligned}
Y_{ij} \mid \beta_i^{(1)}, \sigma_i^2, \mu_i &\sim \mathcal{N}\left(\mu_i + \beta_i^{(1)} I_{\{j\leq n_1\}}, \frac{\sigma_i^2}{M_i}\right), j = 1,\ldots,N, \quad \sigma_i^{-2} \sim \text{Gamma}(b_1, b_2),\\
Z_i^{(l)} \mid \beta_i^{(l)}, \delta_i &\sim \mathcal{N}\left(\beta_i^{(l)}\sqrt{\frac{\nu\delta_i^2}{v_i}}, 1\right), \quad v_i = \frac{n^{(\mathscr{S})} + \sum_{g\in\mathscr{M}_i}\sum_{k\neq g}\widehat{\rho}_{gk} I_{\{z_g, z_k \in \mathscr{S}\}}}{M_i^2}, \quad \mathscr{S} = \{+,-\}, \quad l = 2,3,\\
u_i \mid \beta_i^{(4)}, \sigma_{u,i}^2 &\sim \mathcal{N}\left(\beta_i^{(4)}\sqrt{\nu}, \frac{\sigma_{u,i}^2}{M_i}\right), \quad \sigma_{u,i}^{-2} \sim \text{Gamma}(d_1, d_2),\\
\beta_i^{(l)} \mid \gamma_i, \sigma_\beta^2 &\sim \mathcal{N}\left(\gamma_i, \frac{\sigma_\beta^2}{M_i}\right), \quad \sigma_\beta^{-2} \sim \text{Gamma}(e_1, e_2).
\end{aligned} \tag{7.4}$$

We use the same symbols as in (3.2), whereas the quantities they represent are different. However it simplifies and help identifying the similarities between the two versions of the model.

7.3.1 Differentially expressed modules

The 11 studies presented in Chapter 5 were considered for analysis of module differential expression. As motivated in Chapter 6, summarizing genes into modules prior to fitting our model could be very interesting both statistically, by fulfilling the independence assumption and reducing the dimension, and biologically.

Modules were identified based on the 9884 genes common to all four Type 1 datasets. For each study, the sample correlation matrices were obtained and the empirical model was applied. It took 445 seconds to get the resulting common correlation matrix. From the transformed matrix on the Fisher scale, we calculated $-\log(T_{\theta,\text{var}})$, defined in equation (7.2), and applied

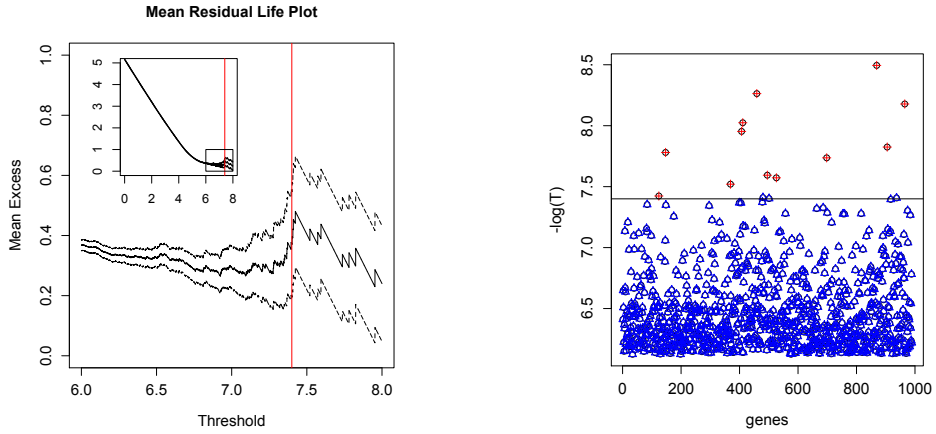


Figure 7.8 – Detection of scattered genes for real data. *Left*: mean residual life plot for real data, *right*: identification of scattered genes (red crosses) and non scattered genes (blue triangles)

Table 7.7 – Estimates and standard errors of the parameters of the mixture of GPD distributions for the detection of scattered genes in real data.

$\widehat{\sigma}_1$	$\widehat{\sigma}_2$	$\widehat{\xi}$	logit($\widehat{p}$)
0.71 (0.14)	0.38 (0.03)	−0.19 (0.05)	logit(0.12)= −1.99(0.86)

our mixture of GPDs, which identified only 11 scattered genes. Figure 7.8 shows the mean residual life plot for our real data, which suggests taking a threshold larger than 6.5, and a plot of the identified scattered genes, where our procedure places the threshold at 7.4. The number of identified scattered genes is very low, so there may be scattered genes in the modules defined hereafter. Hierarchical clustering with Ward's link was performed on the estimated correlation matrix of the remaining genes. This part of the procedure is very fast, as the tree could be obtained within 18 seconds. The GAP procedure being very computationally intensive and not successful in detecting an appropriate number of clusters, we chose to cut the tree to obtain 500 modules, a number large enough to obtain clusters of reasonable sizes, as shown in Figure 7.9.

With the modules thus defined, we added all genes present in the union of all the other types of data, but not present in the modules, as scattered genes, and summarized the module information as described in Section 7.3. This left us with 1962 elements, 500 modules and 1462 scattered genes. We applied the hierarchical model (7.4) with the spike and slab prior to our summarized data. We performed 65000 iterations, using a warmup of 5000 iterations and thinning of 10, which took less than 5 hours, and resulted in 6000 roughly independent samples from the posterior distributions.

With the dimension reduced, we did not even need to run the rank part of the MCMC algorithm more than the other steps. Based on graphical exploration, the convergence of the chain

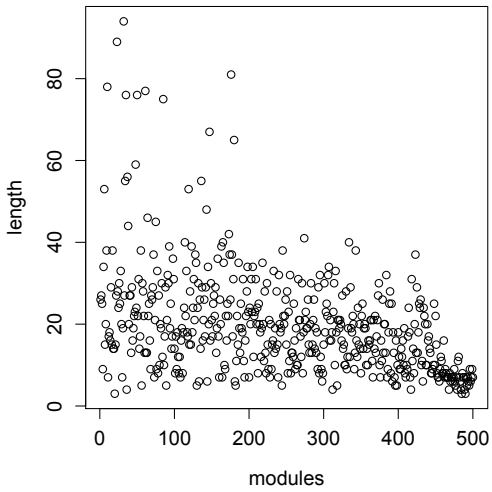


Figure 7.9 – Module size

and independence of the realizations seem to be attained, as presented for one module in Figure 7.10, for the parameter γ. More formal tests were applied and gave no evidence against convergence.

To discriminate between differentially and non-differentially expressed modules, we use again the quantity w defined in Section 3.4. The posterior mean of w identifies 93 elements, 34 of which are single genes, whereas the posterior median of w better separates differentially and non-differentially expressed modules and gives 86 differentially expressed elements. Values of the posterior mean and median of w are presented in Figure 7.11. This figure also has several jumps in the values of the posterior quantities, which are explained by the number of studies in which each module is found, as presented in the right panel of Figure 7.11. The results

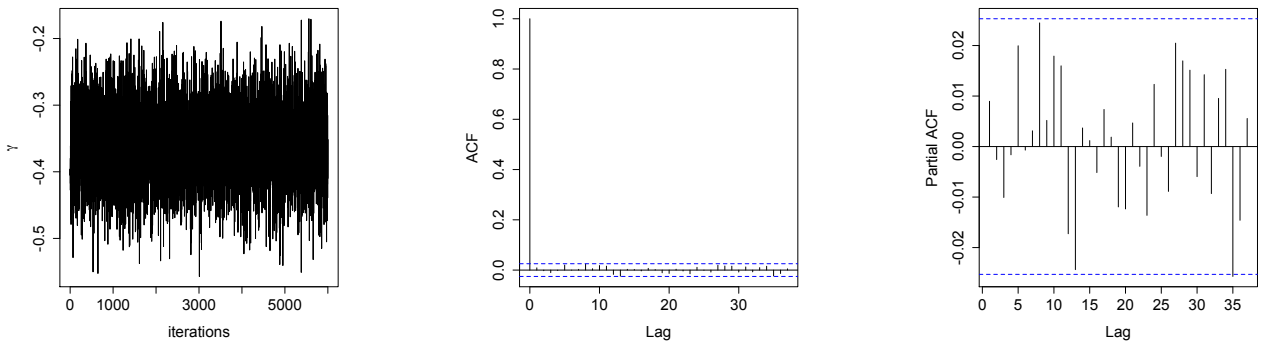
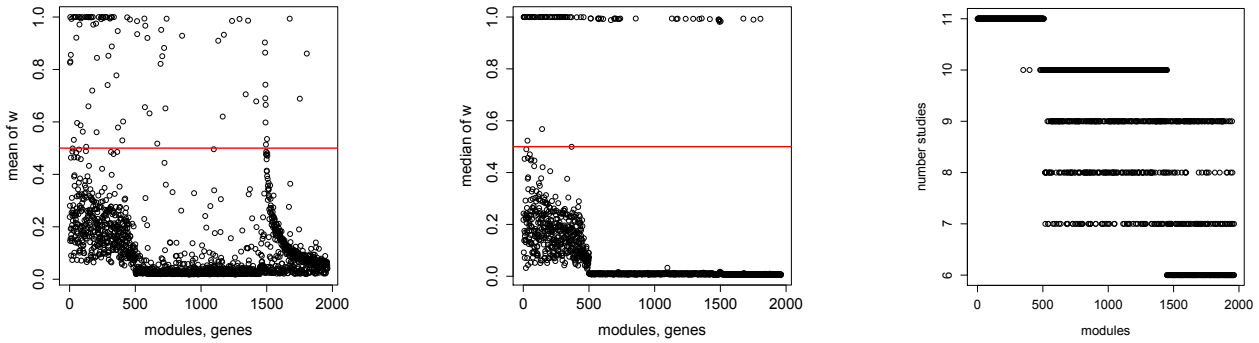


Figure 7.10 – Graphical diagnostics for the convergence of γ.



(a) Posterior mean of w (b) Posterior median of w (c) Number of studies per module

Figure 7.11 – Results for the real data analysis. *Left:* estimate of w based on the posterior mean, with 0.5 threshold to discriminate between differentially and non-differentially expressed genes; *center:* estimate of w based on the posterior median with threshold 0.5; *right:* number of studies to which each module belongs.

for the posterior mean, median and standard deviation of w, along with posterior mean and standard deviation of γ are presented in Table 7.8 for the differentially expressed modules.

We also present the probability that a module present in the top 50 appears in the top r, with $r = 1$, 10, 50, 100 and 500, in Table 7.9. It appears that modules found in the top 50 by our analysis, have a high probability of being in the top 100, whereas posterior probabilities for modules ranked 1000–1050 are not significant, which suggests that our results are quite stable.

7.3.2 Enrichment analysis of modules

The analysis conducted in the previous section identifies several modules as being differentially expressed. We now want to assess the enrichment of these modules, in order to highlight their biological functions, if any. To this end, we used the Molecular Signature database (Liberzon *et al.*, 2011, MSig DB), which provides pathways from several repositories. For our analysis, we kept the pathways from KEGG, Biocarta, Reactome and GO, available to download from MSig under the c2 and c5 collections.

After downloading all the pathways, we removed those too large and too small, keeping only pathways containing between 5 and 200 genes, as suggested in Tseng *et al.* (2012). We also only selected those having common genes with our module set. The enrichment analysis included 1200 genes and 724 pathways. Enrichment of each module in each pathway was tested using Fisher's exact test, and the test with the smallest p-value, or the most enriched term, was recorded for each module. Results are presented in Table 7.10, for the differentially expressed modules identified by our analysis. Differentially expressed scattered genes are included as a module called scattered in Table 7.10. Almost all modules are enriched in one

Table 7.8 – Differentially expressed modules ordered according to the posterior mean of w, $\widehat{w}$. Presented values are the posterior median and mean of w, $\widehat{w}$ and $\bar{w}$, with corresponding standard errors, and the posterior mean of γ with corresponding standard error. The three first columns are in percent.

Module ID	$\tilde{w}$ (%)	$\widehat{w}$ (%)	$sd(w)$(%)	$\widehat{\gamma}$	$sd(\gamma)$
176	100.0	100.0	0.0	1.6	0.1
48	100.0	100.0	0.0	−0.6	0.1
274	100.0	100.0	0.0	0.4	0.1
129	100.0	100.0	0.0	−0.5	0.1
5	100.0	100.0	0.0	−0.4	0.1
255	100.0	100.0	0.0	−0.4	0.1
92	100.0	100.0	0.0	−0.7	0.1
137	100.0	100.0	0.0	−0.7	0.1
221	100.0	100.0	0.0	−0.5	0.1
40	100.0	100.0	0.0	−0.9	0.1
161	100.0	100.0	0.0	−0.6	0.1
279	100.0	100.0	0.0	−0.4	0.1
27	100.0	100.0	0.0	0.5	0.1
322	100.0	100.0	0.0	−1.5	0.1
74	100.0	100.0	0.0	−0.4	0.1
39	100.0	100.0	0.0	−0.6	0.1
241	100.0	100.0	0.0	−0.7	0.1
337	100.0	100.0	0.0	−0.6	0.1
273	100.0	100.0	0.0	−0.5	0.1
242	100.0	100.0	0.0	−0.7	0.1
117	100.0	100.0	0.0	−0.6	0.1
145	100.0	100.0	0.8	−0.4	0.1
67	100.0	100.0	0.0	−0.5	0.1
138	100.0	100.0	0.0	−0.6	0.1
146	100.0	99.9	2.1	0.4	0.1
317	100.0	99.9	2.6	−0.5	0.1
438	99.9	99.9	2.5	−0.5	0.1
165	100.0	99.9	2.4	−0.3	0.1
303	100.0	99.8	3.7	−0.4	0.1
257	100.0	99.6	5.2	0.5	0.1
86	99.9	99.5	6.4	−0.6	0.1
DLGAP5	99.5	99.4	0.2	1.5	0.3
307	100.0	99.4	6.2	0.4	0.1
ITLN1	99.4	99.4	0.3	−2.0	0.4
EVI1	99.4	99.4	0.3	1.8	0.4
14	100.0	99.3	7.3	0.5	0.1
CTRB1	99.3	99.3	0.3	1.8	0.4
240	100.0	99.2	6.9	0.3	0.1
459	99.9	99.1	8.8	0.5	0.1
PEG3	99.5	98.6	8.8	−1.5	0.4
HOXC4 /// HOXC6	99.3	98.5	8.6	−1.8	0.4
CD24	99.5	98.4	9.8	1.5	0.4
201	100.0	97.5	13.4	−0.4	0.1
75	100.0	97.1	12.3	−0.2	0.1
180	100.0	97.1	11.7	−0.3	0.1
GINS1	99.4	96.7	16.1	1.3	0.4
ABCB5	99.4	95.1	20.1	1.2	0.4

Module ID	$\tilde{w}$ (%)	$\widehat{w}$ (%)	$sd(w)$ (%)	$\widehat{\gamma}$	$sd(\gamma)$
366	99.9	94.6	21.3	−0.5	0.2
EPCAM	99.4	93.5	23.4	1.2	0.4
HBA1 /// HBA2	99.4	93.2	23.7	−1.4	0.5
IFNA4	99.4	92.9	24.5	1.2	0.4
51	100.0	92.1	23.1	−0.3	0.1
PTH2R	99.4	92.0	25.9	1.2	0.5
SCN1B	99.4	91.0	27.6	1.2	0.5
NY−REN−7	99.0	90.3	28.0	1.6	0.7
321	100.0	88.8	27.7	−0.3	0.1
CNGA2	99.3	88.2	31.1	1.3	0.6
LOC221061	99.0	86.4	32.9	−1.5	0.8
SPA17	99.3	86.1	33.6	−1.1	0.5
9	100.0	85.6	26.3	0.2	0.1
295	100.0	85.2	28.4	−0.2	0.1
ADH4	99.4	85.1	34.7	1.0	0.5
208	100.0	84.4	30.5	−0.2	0.1
6	100.0	83.0	25.2	−0.2	0.1
3	100.0	82.6	31.5	0.2	0.1
MT4	99.4	82.1	37.4	1.0	0.5
356	100.0	77.7	35.8	−0.2	0.1
ARHI	98.9	74.2	42.7	1.2	0.9
289	100.0	74.1	37.3	−0.2	0.1
172	100.0	72.0	37.1	−0.2	0.1
CFH /// CFHR1	99.2	70.5	44.8	−0.9	0.6
KIAA0186	98.8	69.0	45.3	−1.2	0.9
LOC152573	99.0	68.8	45.4	1.0	0.8
PRSS7	99.2	67.8	45.9	0.9	0.7
KIAA1536	98.8	66.5	46.3	1.1	0.9
143	56.8	65.9	30.5	−0.1	0.1
CDC2	99.2	65.6	46.7	0.8	0.6
HTR6	99.2	65.2	46.8	0.7	0.6
GPR21	99.2	63.3	47.4	0.7	0.6
KCNJ2	99.2	62.0	47.7	−0.7	0.6
407	99.9	60.1	44.2	0.2	0.2
C4A	98.7	59.8	48.2	1.0	0.9
57	44.7	59.6	35.2	−0.1	0.1
83	42.5	58.7	34.9	0.1	0.1
368	49.9	57.9	42.3	0.2	0.2
100	32.8	56.3	42.3	−0.2	0.2
205	40.5	56.1	33.2	−0.1	0.1
LOC114977	98.3	53.5	49.0	−0.8	0.8
32	52.3	53.2	11.8	−0.0	0.0
401	23.6	52.9	43.2	−0.2	0.2
STIL	98.8	51.7	49.1	0.5	0.5
FLJ20059	98.1	51.4	49.2	−0.8	0.9
126	34.9	50.5	32.7	−0.1	0.1

Table 7.9 – Probability (%) of being in the top r for the 50 first differentially expressed modules according to the posterior mean of w, and for modules ranks 1000 to 1049.

		r							r				
Rank	Module ID	1	10	50	100	500	Rank	Module ID	1	10	50	100	500
1	176	32.1	94.2	100	100	100	1000	AFAP1	0	0	0	1	1.7
2	48	11.8	76.6	100	100	100	1001	ACR	0	0	0	0.8	1.1
3	274	3.3	44.4	100	100	100	1002	RTCD1	0	0	0	0.8	1.3
4	129	2.1	31.4	99.8	100	100	1003	KRT76	0	0	0	0.8	1.2
5	5	1.5	28.9	99.8	100	100	1004	CYCS	0	0	0	0.6	1.1
6	255	1.2	25	99.6	100	100	1005	TWF1	0	0	0	1.6	2.3
7	92	0.9	21.3	99	100	100	1006	POLA1	0	0	0	0.7	1
8	137	0.9	18.9	98.7	100	100	1007	PRKCB	0	0	0	1	1.5
9	221	0.6	16.9	98.8	100	100	1008	C19orf57	0	0	0	0.6	1.1
10	40	0.5	16.1	98.4	100	100	1009	DIS3	0	0	0	1	1.4
11	161	0.7	14.2	97.8	100	100	1010	C12orf24	0	0	0	0.8	1.3
12	279	0.4	13.8	97.5	100	100	1011	RASIP1	0	0	0	0.6	1.1
13	27	0.4	11.6	96.8	100	100	1012	EXOC5	0	0	0	0.6	0.9
14	322	0.4	6.9	93	100	100	1013	SEPT2	0	0	0	0.5	1
15	74	0.2	6.1	92.3	100	100	1014	RPS17	0	0	0	0.6	1.1
16	39	0.2	5.8	90.8	100	100	1015	CLCNKB	0	0	0	0.8	1.3
17	241	0.1	5	90.7	100	100	1016	SLC2A2	0	0	0	0.9	1.3
18	337	0.2	4.8	88.2	100	100	1017	ALX1	0	0	0	0.6	0.8
19	273	0.1	4	85.8	100	100	1018	ATF6B	0	0	0	0.6	0.9
20	242	0.1	3.5	86	100	100	1019	PFKFB2	0	0	0	0.5	0.9
21	117	0.1	4.1	84.8	100	100	1020	TCF20	0	0	0	0.6	1.1
22	145	1.7	28.9	99.7	100	100	1021	RAB7A	0	0	0	0.8	1.2
23	67	0	2.4	77.5	100	100	1022	SCGB1A1	0	0	0	0.4	0.9
24	138	0.1	2.4	77.3	100	100	1023	KIF2A	0	0	0	0.7	1.2
25	146	0.1	5.9	90	99.9	100	1024	RPL6	0	0	0	1	1.8
26	317	0.1	3.7	84.7	99.9	100	1025	AGAP2	0	0	0	1	1.5
27	438	0	0.4	39.2	99.9	100	1026	ZBTB48	0	0	0	0.9	1.3
28	165	3.1	42.5	99.7	99.8	100	1027	TLK2	0	0	0	0.7	1.2
29	303	0.3	8.8	94.8	99.7	100	1028	SLC39A7	0	0	0	1.2	2.2
30	257	0	1	61.9	99.6	100	1029	EFR3A	0	0	0	0.7	1
31	86	0	0.4	38.1	99.5	99.8	1030	FKBP1B	0	0	0	1	1.5
32	DLGAP5	0	0	0	64.9	100	1031	PON1	0	0	0	1	1.6
33	307	1.6	28.4	98.7	99.1	100	1032	VHL	0	0	0	0.9	1.5
34	ITLN1	0	0	0	58.5	100	1033	CHP	0	0	0	1	1.9
35	EVI1	0	0	0	57.6	100	1034	CYP51A1	0	0	0	0.4	1
36	14	0.1	2.5	77.7	99.1	100	1035	SRRD	0	0	0	0.6	1
37	CTRB1	0	0	0	46.6	100	1036	RCAN1	0	0	0	0.7	1.2
38	240	1.5	24.9	98.1	98.6	100	1037	SNRNP35	0	0	0	0.8	1.1
39	459	0	0.1	23.7	99	99.6	1038	RPS7	0	0	0	0.7	1.1
40	PEG3	0	0	0	64.8	99.2	1039	HOXA9	0	0	0	0.6	1
41	HOXC4 /// HOXC6	0	0	0	45.8	99.2	1040	MUC5B	0	0	0	0.8	1.3
42	CD24	0	0	0	64.5	99	1041	KIAA0146	0	0	0	0.6	1.1
43	201	0.5	10.6	93	96.6	100	1042	EEF1G	0	0	0	0.7	1.1
44	75	4	50.2	94.3	94.3	100	1043	POLD3	0	0	0	0.7	1.2
45	180	13.7	78.1	93.5	93.5	100	1044	THRSP	0	0	0	1	1.5
46	GINS1	0	0	0	62.7	97.2	1045	AKAP9	0	0	0	0.8	1.2
47	ABCB5	0	0	0	61.5	95.6	1046	HTT	0	0	0	1.1	1.7
48	366	0	0	15.6	94.2	96.6	1047	SFRS11	0	0	0	0.8	1.5
49	EPCAM	0	0	0	60	94	1048	FTSJD2	0	0	0	1	1.4
50	HBA1 /// HBA2	0	0	0	52.4	93.8	1049	PDCD11	0	0	0	0.8	1.1

of the pathways included, while some are strongly enriched with a very small p-value. This strongly suggests that the modules formed by our empirical Bayes method are biologically relevant.

Finally we also compared our list of differentially expressed genes found in the analysis of Section 5.4, with genes included in differentially expressed modules, which revealed that most of the differentially expressed genes are included in differentially expressed modules. However, differentially expressed genes tend to be distributed among the modules rather than all grouped in one module.

7.4 Conclusion

In this chapter we developed methods for the automatic selection of gene modules. From the common correlation matrix estimated in Section 6.4, we identified scattered genes by fitting a mixture of generalized Pareto distributions to the tail of the distribution of $-\log(T_{\theta,\text{var}})$, a method which seems to work well in our simulations, but does not detect many genes in the real data application. Using standard hierarchical clustering methods, we could easily identify modules, and simulations showed that our method was equivalent to or better than competitors. Application to our set of real data also showed promising results. First, the computational time is drastically reduced, which makes the Bayesian meta-analysis easier and more practical to use. The assumptions underlying the model are satisfied and convergence of the MCMC algorithm occurs more rapidly. We found that many differentially expressed genes found in Section 5.4 were retrieved in the module analysis, and most of the modules detected as being differentially expressed are enriched in some biological pathway. The only part of the method which is not automatic yet is the choice of the number of clusters or modules. In simulations where the number of modules was usually small, we could use the GAP statistic, but this method failed on real data and was too computationally intensive.

Table 7.10 – Enrichment analysis of the differentially expressed modules and scattered genes, in terms of the GO, KEGG, Reactome and Biocarta pathways from the MSig database. The p-values are obtained from Fisher's exact test and only the most enriched pathway is presented for each module.

Rank	Module ID	p-value	pathway
1	176	1.03e-21	cell cycle process
2	48	1.4e-03	KEGG melanoma
3	274	9.59e-15	cell cycle process
4	129	3.84e-03	sexual reproduction
5	5	4.36e-04	substrate specific channel activity
6	255	1.3e-05	extracellular matrix part
7	92	3.39e-06	actin binding
8	137	6.35e-05	KEGG tgf beta signaling pathway
9	221	3.25e-04	protein kinase regulator activity
10	40	1.96e-04	Reactome ncam signaling for neurite out growth
11	161	1.07e-03	cell projection
12	279	1.61e-03	Reactome peptide ligand binding receptors
13	27	2.07e-07	Reactome cell cell junction organization
14	322	2.16e-03	inorganic cation transmembrane transporter activity
15	74	6.18e-05	organ morphogenesis
16	39	3.22e-06	KEGG complement and coagulation cascades
17	241	1.33e-04	KEGG vascular smooth muscle contraction
18	337	1.77e-03	chromatin assembly
19	273	2.2e-03	regulation of g protein coupled receptor protein signaling pathway
20	242	3.73e-03	lipid raft
21	117	1.5e-02	KEGG small cell lung cancer
22	145	5.66e-03	activation of mapk activity
23	67	1.73e-03	Reactome g alpha z signalling events
24	138	2.3e-06	Reactome biological oxidations
25	146	1.96e-03	Reactome adherens junctions interactions
26	317	2.4e-02	KEGG cell adhesion molecules cams
27	438	3.52e-02	Reactome adherens junctions interactions
28	165	7.79e-03	Biocarta par1 pathway
29	303	1.78e-04	Reactome sphingolipid metabolism
30	257	2.76e-04	calcium ion binding
31	86	3.52e-02	mitochondrial part
33	307	2.44e-25	Reactome interferon signaling
36	14	5.3e-03	peptide binding
38	240	1.02e-11	Reactome DNA replication
39	459	5.47e-04	Reactome striated muscle contraction
43	201	2.36e-04	RNA polymerase ii transcription factor activity
44	75	3.69e-04	response to other organism
45	180	6.69e-05	Reactome toll receptor cascades
48	366	2.47e-02	sarcomere
52	51	6.45e-04	microtubule associated complex
56	321	2.54e-08	Reactome smooth muscle contraction
60	9	2.6e-03	lipid biosynthetic process
61	295	9.47e-05	KEGG gap junction
63	208	2.91e-03	Reactome signaling by ils
64	6	5.32e-03	protein amino acid dephosphorylation
65	3	5.62e-03	lyase activity
67	356	3.25e-03	Reactome pi3k cascade
69	289	1.77e-03	KEGG amino sugar and nucleotide sugar metabolism
70	172	9.96e-03	KEGG pyrimidine metabolism
76	143	3.72e-08	positive regulation of signal transduction
81	407	3.52e-02	Reactome nrage signals death through jnk
83	57	9.76e-06	RNA processing
84	83	1.71e-04	icosanoid metabolic process
85	368	2.51e-05	magnesium ion binding
86	100	3.39e-03	transcription corepressor activity
87	205	1.47e-03	g protein signaling coupled to cyclic nucleotide second messenger
89	32	1.26e-04	amino acid catabolic process
90	401	6.89e-04	positive regulation of cell differentiation
93	126	4.71e-03	damaged DNA binding
-	scattered	2.36e-03	response to other organism

8 Conclusion and Discussion

This thesis highlighted the need for a model to combine heterogeneous types of microarray data. When only a few studies about the question of interest are published, and even fewer make raw data accessible, it is useful to be able to use all the information available. Our hierarchical Bayesian model, either applied to genes or modules, allows integration of various types of study results, avoiding loss of information, which is common when performing meta-analysis or list aggregation, while maintaining or even increasing the power of detection of differentially expressed genes. Our model is adaptive, but is not heavily influenced by the choice of the hyperparameters. Choosing priors that shrink the parameters of interest close to zero for uninteresting genes aids the detection of the differentially expressed genes, contributing to the high power of our approach. Comparisons of our procedure with existing methods show a clear increase in the number of correctly identified genes and strong control of false positives. The gain of power from including all possible studies is highly significant. Moreover, the simulation design used in this thesis is intended to be realistic: it was constructed to mimic real microarray data; this reinforces the strength of our results, and suggests that our model should be efficient for real data. We emphasize that our simulations were performed from an independent design and not from our model, which is not favored in the simulations.

Application to meta-analysis of 11 real datasets shows promising results by retrieving many genes known to be involved in ovarian cancer and showing significant enrichment, while being robust to false positive housekeeping genes. The criterion for detection of differential expression is a simple thresholding rule which is easy to apply using the posterior output. An important advantage is the absence of multiplicity correction, unlike that needed in the frequentist approach.

One drawback of our model is its large computational requirements. It is infeasible to fit the model to the union of all genes from all studies, so gene selection was essential in the application of Chapter 5. Even if our gene selection is not based on differential expression, we might discard important genes. Including only the best candidate genes is unrealistic and is not appropriate for the chosen priors, which require the inclusion of a large proportion of uninteresting genes. The real data example of Chapter 5 performs gene selection independent

of the model, as we only select genes corresponding to unions or intersections of groups of genes.

In this work, all studies contribute equally to the model and to the detection of differentially expressed genes. However, one might have more confidence in full studies, for example, where the analysis and preprocessing steps are entirely reproducible, rather than in lists of interesting genes that are not reproducible. We could easily associate weights to each of the studies to be combined, reflecting prior confidence in each. Another way of improving the computations would be to use an empirical Bayes approach rather than a purely Bayesian one. Obtaining the posterior densities of each data types given the parameter γ is only possible in closed form for Types 1, 2 and 3 (full studies and z-scores), the posterior densities for the ranks given γ involving an intractable integral. However, one could imagine estimating the hyperparameters from the three first types of studies, using maximum likelihood estimation, and estimating the integral in the posterior density of the ranks by Laplace approximation. Because of the large amount of data available, we believe that the estimation of the hyperparameters would be quite accurate. The entire process should not be too computationally intensive and would drastically reduce the computational time required by our MCMC version. Another necessary assumption that we made concerned the missing values in the Type 1 data, which we suppose to be missing at random. Missing values being usually due to technical problems, the assumption seems reasonable. Moreover, it would make the modeling of these data more complicated, or even impossible, if it was not the case.

In the first part of this thesis, we also assumed that genes are independent, which is not true in practice, though it is a common assumption in the literature, even for single study analysis. Correlations between the genes are included in the simulation design, and do not seem to affect the efficiency of the model. However we believe our procedure would perform even better if the elements on which it is applied are independent. We thus define modules, sets of genes that are correlated or related in some sense, by noticing that genes belonging to the same modules have correlated expression. Using an empirical Bayes model, we estimate a correlation matrix common to all studies for which gene expression matrix is available. Even if this matrix is close to the truth, it is not a correlation matrix per se, as it is not positive definite. We did not need positive definiteness in this work, as the matrix was to be used as input to a hierarchical clustering method. However, it may be possible to impose conditions on the estimates to obtain this property.

With a common correlation matrix thus obtained, we could apply standard hierarchical clustering methods to define modules. The procedure that we defined is particularly attractive due to its simplicity. The empirical Bayes model has only a few parameters, and an explicit expression for the likelihood allows the estimation of the hyperparameters. The posterior median can be obtained explicitly, without resorting to MCMC methods, which resulted in a sparse matrix by performing soft thresholding, thus helping module definition. We did not look at the theoretical properties of our estimator, in terms of risk for example. However, the posterior median was shown to have good properties by Johnstone and Silverman (2005) in

the context of curve estimation using wavelets. Similar theoretical results could be obtained for our method.

The entire process for defining modules is very fast, and could therefore easily be applied as a preprocessing step in every microarray data analysis. In this work, we did not discuss negative correlation. Langfelder *et al.* (2013) consider signed networks, where negatively correlated variables are unconnected, and unsigned networks where the strongly negatively correlated variables are highly related. We assume that negatively correlated genes should belong to different clusters. Indeed, if two genes are negatively correlated, it means they have opposite effect on the response of interest, and therefore it does not make any sense to cluster them together when studying differential expression of modules. However, as a result, we obtain modules that are not uncorrelated, by possibly having a negative correlation.

We also developed a simple method to identify scattered genes, using extreme-value theory. This method was shown to be very robust to false positives, even if the number of true positives was a bit low. Since we would rather not discard genes that might prove to be important, it is better for our method to have high sensitivity rather than high specificity; accidently adding scattered genes to modules is less problematic than incorrectly declaring genes to be scattered. Even if the procedure seems to perform well in simulations, application to real data detects only very few scattered genes, which indicates that we probably miss many scattered genes. We also tried another version of the method, where we fitted a mixture of GPD with different shape parameters, but this led to instability of the estimates and of identified scattered genes, and gave results that did not seem to be accurate. Using only one shape parameter common to the two distributions helps the estimation of the parameters, but seems too stringent. We illustrate by simulations that even if it is very simple, our entire method, comprising the empirical Bayes model, the identification of scattered genes, and hierarchical clustering to define modules, can equal or surpass competing methods by retrieving modules more accurately. This performance was measured in terms of the Rand index in several simulation studies.

Our method is therefore almost entirely automatic, except for the choice of the optimal number of clusters. Finding the optimal number of clusters has been considered and some solutions exist (Dudoit and Fridlyand, 2002; Tibshirani *et al.*, 2001; Monti *et al.*, 2003), usually based on resampling. These methods work well when one wants to cluster a few objects, such as cancer subtypes in microarray gene expression data. However, when it comes to gene clustering, the resampling part becomes computationally expensive and the methods tend to have very low power, as noticed by Thalamuthu *et al.* (2006). Indeed, to be able to identify the optimal number of clusters of G genes, with G of the order of thousands, one needs to perform hierarchical clustering B times for a number of clusters k going from 1 to G. Moreover, B must be large to obtain p-values with sufficient precision, hence its intractability. In this work we choose the number of modules arbitrarily to obtain modules of reasonable size, but being able to select the optimal number of clusters automatically is certainly a very interesting topic for future research.

Automatic module detection on real data was very fast. The hierarchical Bayes model developed in Chapter 3 was then applied on the module set plus some scattered genes. All genes coming from Types 2, 3 and 4 studies that were not present in modules had to be included as scattered genes, as we have no way of knowing whether they could be correlated with other genes based on the information that we have, i.e. z-scores or ranks. This is another argument in favor of making raw data available. Enrichment analysis of the differentially expressed modules showed statistical significance, indicating that the modules automatically defined were biologically meaningful. Moreover, fitting the model to the module set is much faster than on the gene set and does not require gene selection. We also noticed that with a smaller set of objects, latent variables for modeling the ranks tend to be less correlated and therefore we did not need to run that part of the algorithm more than the rest to obtain acceptable correlation between the draws, which further reduced the computational time.

Other ideas for further development include a more powerful method to test for the equality of covariance matrices. Indeed, we saw in Chapter 6 that existing methods have very low power when the sample size is small, even when the number of variables is smaller than the number of samples.

Concerning applications to real data, and more particularly to ovarian cancer data, there are some more interesting questions we could look at. In our real data application, we compared serous ovarian cancer patients with healthy subjects. We did not separate the patients having different characteristics, like different cancer grades. An interesting application could be to compare grade 1 and grade 3 patients. In this case, according to recent knowledge, the type of healthy tissue matters in the comparison. Therefore, it would be more informative to compare grade 1 patients with healthy tissues from the ovary and grade 3 patients with healthy tissues from tube. The control group was also selected as a whole. It could be interesting to make a comparison of two groups of healthy subjects, to see whether there really is no difference between them. This question is especially important in ovarian cancer studies. Indeed the procedure to get healthy samples being so invasive, one can wonder why the patient undertook such surgery in the first place, meaning she certainly has no ovarian cancer but probably has another disease, which is not recorded, but may change gene expression of the tissue collected. The genes or modules found in our analyses could be used to build a prediction model which would help to classify individuals into the cancer or the control groups. This model would be very useful to make a first, less invasive, selection of potential women at risk of ovarian cancer, before undergoing surgery.

Finally, we make some comments concerning development of tools to facilitate microarray gene expression data analysis. When performing gene set enrichment analysis, we noticed that there exist a lot of different methods and databases. A guide to enrichment analysis similar to the one published by Irizarry *et al.* (2009), but for all types of results that one can obtain through an analysis of gene differential expression, would be very useful. Consensus for publication concerning enrichment analysis would also be useful. Development of packages such as the curatedOvarianData R package (Ganzfried *et al.*, 2013), regularly updated, with

preprocessed and normalized datasets, for every type of cancer, would greatly simplify the search for studies to include in meta-analyses. With results from this thesis in mind, such packages could also include published lists of differentially expressed genes and not just raw data or gene expression data.

A Appendix

A.1 Computation of the posterior densities

In this section, we present the details of the calculations needed to obtain the posterior densities of the model parameters. We suppose we have L_i studies of type i. For simplicity, superscripts indicating the study or its type will be omitted in the calculations when not necessary. In what follows, $\pi(\cdot)$ denotes prior densities, whereas, $\pi(\cdot \mid \cdot)$ denotes the posterior.

A.1.1 Realisations of $\pi\left(\beta_g^{(1)} \mid \textbf{rest}\right)$

$$
\begin{aligned}
\pi\left(\beta_g^{(1)} \mid \text{rest}\right) &\propto \exp\left\{\frac{-(\beta_g-\gamma_g)^2}{2\sigma_\beta^2}\right\} \prod_{j=1}^{n_1} \exp\left\{\frac{-(y_{gj}-(\beta_g+\mu_g))^2}{2\sigma_g^2}\right\} \prod_{j=n_1+1}^{N} \exp\left\{\frac{-(y_{gj}-\mu_g)^2}{2\sigma_g^2}\right\} \\
&\propto \exp\left\{\frac{-\beta_g^2\sigma_g^2+2\beta_g\gamma_g\sigma_g^2-\gamma_g^2\sigma_g^2-\sum_{j=1}^{n_1} y_{gj}^2\sigma_\beta^2+2(\beta_g+\mu_g)\sigma_\beta^2\sum_{i=1}^{n_1} y_{gj}}{2\sigma_\beta^2\sigma_g^2}\right. \\
&\qquad \left. -\frac{n_1(\beta_g+\mu_g)^2\sigma_\beta^2}{2\sigma_\beta^2\sigma_g^2}\right\} \\
&\propto \exp\left\{\frac{-\beta_g^2[\sigma_g^2+n_1\sigma_\beta^2]+2\beta_g[\gamma_g\sigma_g^2+\sigma_\beta^2\sum_{j=1}^{n_1} y_{gj}-n_1\sigma_\beta^2\mu_g]}{2\sigma_g^2\sigma_\beta^2}\right\} \\
&\sim \mathcal{N}\left(\frac{\gamma_g\sigma_g^2+\sigma_\beta^2\sum_{j=1}^{n_1} y_{gj}-n_1\sigma_\beta^2\mu_g}{\sigma_g^2+n_1\sigma_\beta^2}, \frac{\sigma_\beta^2\sigma_g^2}{\sigma_g^2+n_1\sigma_\beta^2}\right)
\end{aligned}
$$

A.1.2 Realisations of $\pi\left(\sigma_g^{-2} \mid \textbf{rest}\right)$

$$\begin{aligned}
\pi(\sigma^{-2} \mid \text{rest}) &\propto \prod_{j=1}^{N} \pi(Y_{gj} \mid \mu_g, \beta_g, \sigma_g^{-2}) \pi(\sigma^{-2} \mid b, h) \\
&\propto (\sigma_g^{-2})^{b_1 - 1} \exp\left\{-\sigma_g^{-2} b_2\right\} \exp\left\{\frac{-\sum_{j=1}^{n_1} (Y_{gj} - \mu_g - \beta_g)^2 \sigma_g^{-2}}{2}\right\} \exp\left\{\frac{-\sum_{g=n_1+1}^{n} (Y_{gj} - \mu_g)^2 \sigma_g^{-2}}{2}\right\} \left(\sigma^{-2}\right)^{\frac{N}{2}} \\
&\sim \text{Gamma}\left(b_1 + \frac{N}{2}, b_2 + \frac{1}{2} \sum_{j=1}^{n_1} (Y_{gj} - \mu_g - \beta_g)^2 + \frac{1}{2} \sum_{j=n_1+1}^{n} (Y_{gj} - \mu_g)^2\right)
\end{aligned}$$

A.1.3 Realisations of $\pi\left(\beta_g^{(2)} \mid \textbf{rest}\right)$

$$\begin{aligned}
\pi\left(\beta_g^{(2)} \mid \text{rest}\right) &\propto \pi(\beta_g^{(2)} \mid \gamma_g, \sigma_\beta^2) \pi(z_g^{(2)} \mid \beta_g^{(2)}, \delta_g^2) \\
&\propto \exp\left\{\frac{-(\beta_g - \gamma_g)^2}{2\sigma_\beta^2}\right\} \exp\left\{\frac{-(z_g - \beta_g \sqrt{\delta_g^2 \nu})^2}{2}\right\} \\
&\propto \exp\left\{\frac{-\beta_g^2 + 2\beta_g\gamma_g - \gamma_g^2 - z_g^2\sigma_\beta^2 + 2z_g\beta_g\sigma_\beta^2\sqrt{\delta_g^2\nu} - \beta_g^2\sigma_\beta^2\delta_g^2\nu}{2\sigma_\beta^2}\right\} \\
&\propto \exp\left\{\frac{-\beta_g^2(1 + \delta_g^2\nu\sigma_\beta^2) + 2\beta_g(\gamma_g + z_g\sigma_\beta^2\sqrt{\delta_g^2\nu})}{2\sigma_\beta^2}\right\} \\
&\sim \mathcal{N}\left(\frac{\gamma_g + z_g\sigma_\beta^2\sqrt{\delta_g^2\nu}}{1 + \delta_g^2\nu\sigma_\beta^2}, \frac{\sigma_\beta^2}{1 + \delta_g^2\nu\sigma_\beta^2}\right)
\end{aligned}$$

The calculations are the same for $\beta_g^{(3)}$.

A.1.4 Realisations of $\pi\left(\beta_g^{(4)} \mid \textbf{rest}\right)$

$$\begin{aligned}
\pi(\beta_g^{(4)} \mid \text{rest}) &\propto \pi(\beta_g \mid \gamma_g, \sigma_\beta^2) \pi(u_g \mid \beta_g, \sigma_{u,g}^2) \\
&\propto \exp\left\{\frac{-(\beta_g - \gamma_g)^2}{2\sigma_\beta^2} - \frac{(u_g - \beta_g\sqrt{\nu})^2}{2\sigma_{u,g}^2}\right\} \\
&\propto \exp\left\{\frac{-\beta_g^2(\sigma_{u,g}^2 + \sigma_\beta^2\nu) + 2\beta_g(\gamma_g\sigma_{u,g}^2 + u_g\sigma_\beta^2\sqrt{\nu})}{2\sigma_\beta^2\sigma_{u,g}^2}\right\} \\
&\sim \mathcal{N}\left(\frac{\gamma_g\sigma_{u,g}^2 + u_g\sigma_\beta^2\sqrt{\nu}}{\sigma_{u,g}^2 + \nu\sigma_\beta^2}, \frac{\sigma_\beta^2\sigma_{u,g}^2}{\sigma_{u,g}^2 + \sigma_\beta^2\nu}\right)
\end{aligned}$$

A.1.5 Realisations of $\pi\left(u_g \mid \textbf{rest}\right)$

We have

$$\begin{aligned}\pi(u_g \mid \text{rest}) &\propto \pi(R_g \mid u_{g-1}, u_g, u_{g+1})\pi(u_g \mid \beta_g^{(4)}, \sigma_{u,g}^2) \\ &\propto I_{\{|u_{g+1}|<|u_g|<|u_{g-1}|\}} \exp\left\{\frac{-(u_g - \beta_g\sqrt{\nu})^2}{2\sigma_{u,g}^2}\right\} \\ &\sim \mathcal{N}_{|u_{g+1}|}^{|u_{g-1}|}\left(\beta_g\sqrt{\nu}, \sigma_{u,g}^2\right),\end{aligned}$$

where $\mathcal{N}_b^a$ corresponds to a truncated normal between a and b. To generate from the truncated normal, we first center and scale the random variable u_g, and obtain

$$Z_g = \frac{u_g - \beta_g^{(4)}\sqrt{\nu}}{\sigma_{u,g}}, \quad z^+ = \frac{u_{g-1} - \beta_g^{(4)}\sqrt{\nu}}{\sigma_{u,g}}, \quad z^- = \frac{u_{g+1} - \beta_g^{(4)}\sqrt{\nu}}{\sigma_{u,g}}.$$

Then

$$\begin{aligned}\alpha &= P(Z \leq z \mid z^- < z < z^+) \\ &= \frac{P(Z \leq z, z^- < z < z^+)}{P(z^- < z < z^+)} \\ &= \frac{P(z^- < Z < z)}{\Phi(z^+) - \Phi(z^-)} \\ &= \frac{\Phi(z) - \Phi(z^-)}{\Phi(z^+) - \Phi(z^-)}.\end{aligned}$$

Therefore, $z = \Phi^{-1}\left[\alpha\left\{\Phi(z^+) - \Phi(z^-)\right\} + \Phi(z^-)\right]$, and thus,

$$u_g = \Phi^{-1}\left[\alpha\left\{\Phi(z^+) - \Phi(z^-)\right\} + \Phi(z^-)\right]\sigma_u + \beta_g^{(4)}\sqrt{\nu},$$

where $\alpha \sim \mathcal{U}(0,1)$. The calculations here are for a positive u_g, the negative version being similar. As the sign of u is not observed and interest only lies in the expected value of this variable, we fix the sign of u_g to be the same that of the corresponding $\beta_g^{(4)}$.

A.1.6 Realisations of $\pi\left(\sigma_{u,g}^{-2} \mid \textbf{rest}\right)$

$$\begin{aligned}\pi(\sigma_{u,g}^{-2} \mid \text{rest}) &\propto \pi(\sigma_{u,g}^{-2} \mid d_1, d_2)\pi(u_g \mid \beta_g, \sigma_{u,g}^{-2}) \\ &\sim \text{Gamma}\left(d_1 + \frac{1}{2}, d_2 + \frac{(u_g - \beta_g\sqrt{\nu})^2}{2}\right)\end{aligned}$$

A.1.7 Realisations of $\pi\left(\sigma_\beta^{-2} \mid \textbf{rest}\right)$

$$\begin{aligned}
\pi(\sigma_\beta^{-2} \mid \text{rest}) &\propto \pi(\sigma_\beta^{-2} \mid e_1, e_2) \prod_{g=1}^{p} \prod_{l=1}^{L} \pi(\beta_g^{(l)} \mid \gamma_g, \sigma_\beta^{-2}) \\
&\propto (\sigma_\beta^{-2})^{e_1-1} e^{-\sigma_\beta^{-2} e_2} \prod_{g=1}^{p} \prod_{l=1}^{L} \exp\left\{ \frac{-(\beta_g^{(l)} - \gamma_g)^2 \sigma_\beta^{-2}}{2} \right\} \left(\sqrt{\sigma_\beta^{-2}}\right)^{pL} \\
&\sim \text{Gamma}\left(e_1 + \frac{pL}{2}, e_2 + \sum_{g=1}^{p} \sum_{l=1}^{L} \frac{(\beta_g^{(l)} - \gamma_g)^2}{2} \right)
\end{aligned}$$

A.1.8 Posterior densities for the spike and slab prior

Realisations of $\pi(\alpha \mid \textbf{rest})$

$$\begin{aligned}
\pi(\alpha \mid \text{rest}) &\propto \pi(\alpha) \prod_{g=1}^{p} \pi(c_g \mid c^\star, \alpha) \\
&\propto \prod_{g=1}^{p} (1-\alpha)\delta_{c^\star} + \alpha\delta_1 \\
&\propto (1-\alpha)^{\#\{g:c_g = c^\star\}} \alpha^{\#\{g:c_g=1\}} \\
&\sim \text{Beta}\left(1 + \#\{g : c_g = 1\}, 1 + \#\{g : c_g = c^\star\}\right)
\end{aligned}$$

Realisations of $\pi(c_g \mid \textbf{rest})$

We have

$$\begin{aligned}
\pi(c_g \mid \text{rest}) &\propto \pi(c_g \mid c^\star, \alpha) \pi(\gamma_g \mid c_g) \\
&\propto \{(1-\alpha)\delta_{c^\star} + \alpha\delta_1\} \frac{1}{\sqrt{c_g}} \exp\left(\frac{-\gamma_g^2}{2 c_g \tau_g^2} \right).
\end{aligned}$$

We define the following quantities:

$$w_{1,g} = \frac{1-\alpha}{\sqrt{c^\star}} \exp\left(\frac{-\gamma_g^2}{2c^\star \tau_g^2} \right), \quad w_{2,g} = \alpha \exp\left(\frac{-\gamma_g^2}{2\tau_g^2} \right),$$

then

$$c_g \mid \text{rest} \sim \frac{w_{1,g}}{w_{g,1} + w_{g,2}} \delta_{c^\star} + \frac{w_{2,g}}{w_{1,g} + w_{2,g}} \delta_1.$$

Realisations of $\pi(\tau_g^{-2} \mid \textbf{rest})$

$$\begin{aligned}\pi(\tau_g^{-2} \mid \text{rest}) &\propto \pi(\tau_g^{-2} \mid a_1, a_2)\pi(\gamma_g \mid c_g, \tau_g^{-2}) \\ &\propto (\tau_g^{-2})^{a_1-1} e^{-\tau_g^{-2} a_2} \sqrt{\tau_g^{-2}} e^{-\frac{\gamma_g^2 \tau_g^{-2}}{2c_g}} \\ &\sim \text{Gamma}\left(a_1 + 1/2, a_2 + \frac{\gamma_g^2}{2c_g}\right)\end{aligned}$$

Realisations of $\pi(\gamma_g \mid \textbf{rest})$

$$\begin{aligned}\pi\left(\gamma_g \mid \text{rest}\right) &\propto \pi(\gamma_g \mid c_g, \tau_g^2) \prod_{l=1}^{L} \pi\left(\beta_g^{(l)} \mid \gamma_g, \sigma_\beta^2\right) \\ &\propto \exp\left\{\frac{-\gamma_g^2}{2c_g\tau_g^2}\right\} \exp\left\{\frac{-\sum_{l=1}^{L}\left(\beta_g^{(l)} - \gamma_g\right)^2}{2\sigma_\beta^2}\right\} \\ &\propto \exp\left\{\frac{-\gamma_g^2\left(Lc_g\tau_g^2 + \sigma_\beta^2\right) + 2\gamma_g\left(\sum_{l=1}^{L}\beta_g^{(l)} c_g\tau_g^2\right)}{2\sigma_\beta^2 c_g\tau_g^2}\right\} \\ &\sim \mathcal{N}\left(\frac{\sum_{l=1}^{L}\beta_g^{(l)} c_g\tau_g^2}{Lc_g\tau_g^2 + \sigma_\beta^2}, \frac{\sigma_\beta^2 c_g\tau_g^2}{Lc_g\tau_g^2 + \sigma_\beta^2}\right)\end{aligned}$$

A.1.9 Posterior densities for the horseshoe prior

Realisations of $\pi(\gamma_g \mid \textbf{rest})$

$$\begin{aligned}\pi(\gamma_g \mid \text{rest}) &\propto \prod_{l=1}^{L} \pi(\beta_g^{(l)} \mid \gamma_g, \sigma_\beta^2)\pi(\gamma_g \mid \lambda_g^2, \tau^2) \\ &\propto \prod_{l=1}^{L} \exp\left\{\frac{-(\gamma_g - \beta_g^{(l)})^2}{2\sigma_\beta^2}\right\} \exp\left\{\frac{-\gamma_g^2}{2\lambda_g^2\tau^2}\right\} \\ &\propto \exp\left\{\frac{-\gamma_g^2(L\lambda_g^2\tau^2 + \sigma_\beta^2) + 2\gamma_g\sum_{l=1}^{L}\beta_g^{(l)}\lambda_g^2\tau^2}{2\sigma_\beta^2\lambda_g^2\tau^2}\right\} \\ &\sim \mathcal{N}\left(\frac{\sum_{l=1}^{L}\beta_g^{(l)}\lambda_g^2\tau^2}{L\lambda_g^2\tau^2 + \sigma_\beta^2}, \frac{\sigma_\beta^2\lambda_g^2\tau^2}{\sigma_\beta^2 + L\lambda_g^2\tau^2}\right)\end{aligned}$$

Realisations of $\pi(\lambda_g \mid \textbf{rest})$

$$\pi(\lambda_g \mid \text{rest}) \propto \pi(\gamma_g \mid \lambda_g, \tau)\pi(\lambda_g) \propto \frac{1}{\lambda_g\tau} \exp\left\{\frac{-\gamma_g^2}{2\lambda_g^2\tau^2}\right\} \frac{1}{1+\lambda_g^2}$$

It seems difficult to sample from this posterior distribution. Therefore, we follow the algorithm described in Scott (2011), which was adapted from the method described in Damien (1999). We first define

$$\eta_g = \frac{1}{\lambda_g^2}, \quad \varphi_g = \frac{\gamma_g}{\tau}.$$

We can now compute the posterior distribution of η_g,

$$\pi(\eta_g \mid \varphi_g) \propto \exp\left(\frac{-\varphi_g^2 \eta_g}{2}\right) \frac{1}{1+\eta_g}.$$

Then the algorithm is:

- sample $u_g \sim \mathcal{U}(0, 1/(1+\eta_g))$;
- generate $\eta_g \sim E(\varphi_g^2/2)$, truncated to have 0 probability outside the interval $\left[0, \frac{1-u_g}{u_g}\right]$;
- transform back to the λ-scale to obtain a sample from the desired posterior distribution: $\lambda_g = \sqrt{1/\eta_g}$.

In order to sample from a truncated exponential distribution, we first denote $u^+ = (1-u_g)/u_g$ and omitting the subscript g,

$$\alpha = P(\eta < x \mid \eta < u^+) = \frac{P(\eta < x)}{P(\eta < u^+)} = \frac{1-\exp(-\varphi^2 x/2)}{1-\exp(-u^+\varphi^2/2)},$$
$$x = \frac{-2}{\varphi^2} \log\left[1 - \alpha\left\{1 - \exp\left(\frac{-\varphi^2 u^+}{2}\right)\right\}\right],$$

where $\alpha \sim \mathcal{U}(0,1)$.

Realisations of $\pi(\tau \mid \textbf{rest})$

In order to generate from the posterior distribution of τ, we will use a similar algorithm to that used in Section A.1.9,

$$\begin{aligned}\pi(\tau \mid \lambda, \gamma) &\propto \pi(\tau) \prod_{g=1}^{p} \pi(\gamma_g \mid \lambda_g, \tau) \\ &\propto \frac{1}{1+\tau^2} \prod_{g=1}^{p} \frac{1}{\lambda_g \tau} \exp\left(\frac{-\gamma_g^2}{2\lambda_g^2 \tau^2}\right) \\ &\propto \frac{1}{(1+\tau^2)\tau^p} \exp\left(\frac{-1}{2\tau^2} \sum_{g=1}^{p} \frac{\gamma_g^2}{\lambda_g^2}\right).\end{aligned}$$

We now define the following quantities, inspired by the algorithm proposed by Scott (2011),

$$\eta = \frac{1}{\tau^2}, \quad \frac{d\tau}{d\eta} = \frac{\sqrt{\eta}}{\eta^2} \quad \varphi = \sum_{g=1}^{p} \frac{\gamma_g^2}{\lambda_g^2}.$$

We therefore obtain

$$\pi(\eta \mid \varphi) \propto \frac{1}{1+\eta} \eta^{\frac{p-1}{2}} \exp(-\eta\varphi/2).$$

The algorithm is then

- sample $u \sim \mathcal{U}(0, 1/(1+\eta))$;
- generate $\eta \sim \text{Gamma}\big((p+1)/2, \varphi/2\big)$, truncated to have probability 0 outside $[0, (1-u)/u]$;
- transform back to the τ-scale, by $\tau = \sqrt{1/\eta}$.

If $F(x; a, b)$ denotes the distribution function of a Gamma distribution evaluated at x, with shape and rate parameters a and b, then sampling from a truncated Gamma is done as follows:

$$x = F^{-1}\left[\alpha F\left(u^+; \frac{p+1}{2}, \frac{\varphi}{2}\right); \frac{p+1}{2}, \frac{\varphi}{2}\right], \quad \alpha \sim \mathcal{U}(0,1), \quad u^+ = \frac{1-u}{u}.$$

A.1.10 Posterior densities for the normal-gamma prior

Realisations of $\pi\left(\gamma_g \mid \textbf{rest}\right)$

$$\begin{aligned}
\pi(\gamma_g \mid \text{rest}) &\propto \prod_{l=1}^{L} \pi(\beta_g^{(l)} \mid \gamma_g, \sigma_\beta^2) \pi(\gamma_g \mid \psi_g) \\
&\propto \exp\left\{ \frac{-\sum_{l=1}^{L} (\gamma_g - \beta_g^{(l)})^2}{2\sigma_\beta^2} - \frac{\gamma_g^2}{2\psi_g} \right\} \\
&\sim \mathcal{N}\left(\frac{\sum_{l=1}^{L} \beta_g^{(l)} \psi_g}{L\psi_g + \sigma_\beta^2}, \frac{\sigma_\beta^2 \psi_g}{L\psi_g + \sigma_\beta^2} \right)
\end{aligned}$$

Realisations of $\pi\left(\psi_g \mid \textbf{rest}\right)$

We have

$$\begin{aligned}\pi(\psi_g \mid \text{rest}) &\propto \pi(\psi_g \mid \lambda,\tau)\pi(\gamma_g \mid \psi_g)\\ &\propto \frac{1}{\sqrt{2\pi\psi_g}}\exp\left(\frac{-\gamma_g^2}{2\psi_g}\right)\psi_g^{\lambda-1}\exp\left(\frac{-\psi_g}{\tau}\right)\\ &\sim GIG\left(\lambda-\frac{1}{2},\frac{1}{\tau^2},\gamma_g^2\right),\end{aligned}$$

where

$$GIG(\lambda,\psi,\chi)=\frac{(\psi/\chi)^{\lambda/2}}{2K_\lambda(\sqrt{\psi\chi})}x^{\lambda-1}\exp\left\{-\frac{1}{2}\left(\psi x+\frac{\chi}{x}\right)\right\},$$

and K_λ is the modified Bessel function of the third kind and the parameters satisfy one of the conditions

$$\begin{aligned}&\lambda>0,\ \psi>0,\ \chi\geq 0,\\ &\lambda=0,\ \psi>0,\ \chi>0,\\ &\lambda<0,\ \psi\geq 0,\ \chi>0.\end{aligned}$$

If parameters are close to zero, limiting distributions can be used (Eberlein and Hammerstein, 2004):

- if $\lambda>0$, $\psi>0$ but $\chi=0$ or is close to zero, then a gamma distribution, Gamma$(\lambda,\psi/2)$, can be used instead of the generalized inverse Gaussian;

- if $\lambda<0$, $\chi>0$ but $\psi=0$ or is close to zero, then the inverse gamma distribution, IG$(-\lambda,\chi/2)$, can be used instead of the generalized inverse Gaussian.

So, using our parameters, we obtain the following cases:

- if $\lambda>1/2$, $1/\tau^2>0$, $\beta_g^2=0$, $\psi_g \mid \text{rest} \sim \text{Gamma}\left(\lambda-\frac{1}{2},\frac{1}{2\tau^2}\right)$;

- if $\lambda<0$, $\beta_g^2>0$ and $1/\tau^2=0$, then $\psi_g \mid \text{rest} \sim \text{IG}\left(-\lambda+\frac{1}{2},\frac{\beta_g^2}{2}\right)$.

Realisation of $\pi(\lambda \mid \textbf{rest})$

Now

$$\begin{aligned}\pi(\lambda \mid \text{rest}) &\propto \pi(\lambda) \prod_{g=1}^{p} \pi(\psi_g \mid \lambda) \\ &\propto e^{-\lambda} \prod_{g=1}^{p} \left(\frac{\psi_g^{\lambda-1}}{(2\tau^2)^{\lambda}\Gamma(\lambda)} \right) \\ &\propto e^{-\lambda} \left(\prod_{g=1}^{p} \psi_g \right)^{\lambda-1} \left[(2\tau^2)^{\lambda}\Gamma(\lambda) \right]^{-p}.\end{aligned}$$

As this distribution does not have a recognizable form, we use Metropolis–Hastings step to generate from it. We put a random walk update on $\log\lambda$ leading to the proposal

$$\lambda' = \exp\{\sigma_\lambda^2 z\}\lambda,$$

where z is generated from a standard normal distribution, and σ_λ is chosen so that the overall acceptance probability is about 30% in average. The corresponding τ' can be taken to be $\tau' = \sqrt{2\lambda\tau^2/(2\lambda')}$. Then the acceptance probability is given by

$$\alpha = \min\left[1, e^{-\lambda'+\lambda} \left(\prod_{g=1}^{p} \psi \right)^{\lambda'-\lambda} \left(\frac{\Gamma(\lambda)}{\Gamma(\lambda')} \right)^{p} \frac{(2\tau^2)^{p\lambda}}{(2\tau'^2)^{p\lambda'}} \frac{\lambda'}{\lambda} \exp\left\{ \sum_{g=1}^{p} \psi_g \left(\frac{1}{2\tau^2} - \frac{1}{2\tau^2} \right) \right\} \right],$$

and

$$\begin{aligned}\log(\alpha) = (\lambda' - \lambda) \left[-1 + \sum_{g=1}^{p} \log\psi_g \right] + (\log\lambda' - \log\lambda) + p(\log\Gamma(\lambda) - \log\Gamma(\lambda')) \\ + p\lambda\log(2\tau^2) - \lambda' p\log(2\tau'^2) + \sum_{g=1}^{p} \psi_g \left(\frac{1}{2\tau^2} - \frac{1}{2\tau'^2} \right).\end{aligned}$$

Realisation of $\pi(\tau^2 \mid \textbf{rest})$

$$\begin{aligned}
\pi(\tau^{-2} \mid \text{rest}) &\propto \prod_{g=1}^{p} \pi(\psi_g \mid \lambda, \tau^{-2}) \pi(\tau^{-2}) \\
&\propto \tau^{-2} \exp\left\{-\frac{\tau^{-2} M}{2\lambda}\right\} \prod_{g=1}^{p} \left(\frac{\tau^{-2}}{2}\right)^{\lambda} \exp\left\{-\frac{\psi_g \tau^{-2}}{2}\right\} \\
&\propto \left(\tau^{-2}\right)^{2+\lambda p-1} \exp\left\{-\tau^{-2}\left(\frac{M}{2\lambda} + \sum_{g=1}^{p} \frac{\psi_g}{2}\right)\right\} \\
&\sim \text{Gamma}\left(2 + \lambda p, \frac{M}{2\lambda} + \sum_{g=1}^{p} \frac{\psi_g}{2}\right)
\end{aligned}$$

A.2 Complete results for the real data analysis

In Section 5.4, we only provide the list of the top 100 differentially expressed genes identified by our analysis. In this section, we give the complete list of all 296 differentially expressed genes, sorted according to the parameter $\widehat{w}$, the posterior mean of w. Tables A.2 and A.3 give the mean and standard deviation of the posterior mean of the parameters w, indicating differential expression, and γ, giving information about the direction of the differentials, for each gene.

Table A.1 – List of the top 100 differentially expressed genes. Genes are ordered according to the value of $\widehat{w}$, from the most to the least differentially expressed genes. The estimates $\widehat{w}$ are obtained from the fit of our model to the 11 studies selected for the analysis, $\widehat{\gamma}$ indicates the magnitude and the direction of the differential expression. Genes in bold are known to be involved in ovarian cancer.

Rank	Genes	$\widehat{w}$	sd($\widehat{w}$)	$\widehat{\gamma}$	sd($\widehat{\gamma}$)
1	CP	0.99	$< 10^{-2}$	3.07	0.38
2	**TOP2A**	0.99	$< 10^{-2}$	2.60	0.38
3	BCHE	0.99	0.01	-2.49	0.41
4	NEK2	0.99	0.01	2.48	0.38
5	TTK	0.99	0.01	2.48	0.38
6	CENPA	0.99	0.02	2.45	0.38
7	SPP1	0.99	0.02	2.31	0.39
8	**MELK**	0.99	0.02	2.35	0.39
9	PRAME	0.99	0.02	2.41	0.39
10	ADH1B	0.99	0.02	-2.33	0.41
11	KIAA0101	0.99	0.02	2.41	0.40
12	NMU	0.99	0.02	2.30	0.39
13	IGFBP6	0.99	0.03	-2.22	0.39
14	**KLK6**	0.99	0.02	2.24	0.39
15	EVI1	0.99	0.01	2.67	0.45
16	**CLDN3**	0.99	0.02	2.38	0.41
17	SST	0.99	0.02	2.27	0.41
18	FOLR1	0.99	0.03	2.22	0.39
19	**WFDC2**	0.99	0.03	2.30	0.40
20	UBE2C	0.99	0.03	2.22	0.38
21	CD24	0.99	0.03	2.47	0.45
22	HMGA2	0.99	0.04	2.21	0.43
23	SPARCL1	0.99	0.04	-2.00	0.38
24	KIF2C	0.99	0.04	2.08	0.40
25	ELF3	0.99	0.05	2.11	0.41
26	**MAL**	0.99	0.05	2.07	0.42
27	CDKN2A	0.98	0.05	2.09	0.41
28	**PAX8**	0.98	0.05	2.05	0.41
29	FOXM1	0.98	0.05	2.05	0.40
30	CENPF	0.98	0.06	2.02	0.40
31	TNNT1	0.98	0.06	2.01	0.42
32	**CLDN4**	0.98	0.06	2.05	0.42
33	HMMR	0.98	0.07	1.99	0.42
34	LCT	0.98	0.07	1.94	0.41
35	KIF11	0.98	0.07	1.92	0.41
36	TRIM31	0.98	0.08	1.92	0.43
37	CDC20	0.98	0.08	1.88	0.42
38	**TACSTD1**	0.98	0.08	2.61	0.63
39	**CCNB1**	0.97	0.09	1.87	0.44
40	PTTG1	0.97	0.10	1.93	0.48
41	PRSS8	0.97	0.10	1.87	0.46
42	CCNE1	0.97	0.11	1.80	0.45
43	RRM2	0.96	0.11	1.84	0.47
44	EHF	0.96	0.11	1.90	0.48
45	**S100A1**	0.96	0.11	1.82	0.47
46	ATP6V1B1	0.95	0.13	1.75	0.50
47	KLK7	0.95	0.13	1.72	0.48
48	ABCA8	0.95	0.13	-1.77	0.49
49	ALDH1A1	0.95	0.13	-1.70	0.47
50	SCNN1A	0.94	0.14	1.73	0.51

Rank	Genes	$\widehat{w}$	sd($\widehat{w}$)	$\widehat{\gamma}$	sd($\widehat{\gamma}$)
51	C7	0.94	0.14	-1.69	0.52
52	PDHA2	0.94	0.15	1.70	0.52
53	ANXA8	0.94	0.17	2.22	0.74
54	GLDC	0.94	0.15	1.66	0.49
55	LAMP3	0.94	0.16	1.72	0.54
56	KRT7	0.93	0.16	1.57	0.50
57	CKS2	0.93	0.16	1.61	0.50
58	AOX1	0.93	0.16	-1.63	0.53
59	EZH2	0.93	0.16	1.60	0.49
60	MYH11	0.93	0.16	-1.63	0.51
61	NY-REN-7	0.92	0.20	2.41	0.94
62	CCNA2	0.91	0.18	1.51	0.53
63	TK1	0.91	0.18	1.50	0.52
64	MAD2L1	0.91	0.18	1.51	0.54
65	ACTG2	0.91	0.18	-1.53	0.56
66	SCGB2A1	0.91	0.19	1.65	0.64
67	**MUC1**	0.90	0.19	1.52	0.57
68	SLC2A1	0.90	0.19	1.46	0.53
69	SULT1C2	0.89	0.20	1.47	0.57
70	MGP	0.87	0.21	-1.37	0.55
71	THBD	0.87	0.21	-1.38	0.57
72	IGF2BP3	0.86	0.22	1.36	0.58
73	SPINT2	0.86	0.22	1.36	0.58
74	ZIC1	0.86	0.22	1.38	0.60
75	CGN	0.85	0.24	1.44	0.66
76	RNASE4	0.85	0.22	-1.31	0.57
77	EFEMP1	0.85	0.22	-1.33	0.60
78	APOA1	0.85	0.23	1.31	0.59
79	DXYS155E	0.84	0.28	2.22	1.21
80	TFAP2A	0.83	0.23	1.25	0.57
81	NDP52	0.83	0.29	2.21	1.27
82	FRY	0.82	0.24	-1.25	0.62
83	DEFB1	0.79	0.25	1.12	0.59
84	TYMS	0.79	0.25	1.12	0.56
85	GRPR	0.79	0.25	1.15	0.60
86	MYBL2	0.79	0.25	1.13	0.58
87	MAOB	0.79	0.25	-1.11	0.56
88	CNN1	0.77	0.25	-1.08	0.57
89	CLDN7	0.77	0.26	1.14	0.64
90	BLM	0.77	0.25	1.09	0.59
91	CXCL10	0.76	0.25	1.07	0.61
92	KIF23	0.76	0.25	1.04	0.56
93	KLF4	0.76	0.25	-1.03	0.56
94	CDKN3	0.75	0.26	1.03	0.57
95	HTR3A	0.75	0.25	1.02	0.57
96	EPCAM	0.75	0.27	1.09	0.67
97	GNG11	0.75	0.26	-1.01	0.55
98	KIF14	0.74	0.25	1.00	0.55
99	NDN	0.74	0.25	-0.99	0.54
100	RAD54L	0.74	0.25	0.98	0.55

Table A.2 – List of top 101-200 differentially expressed genes. Genes are ordered according to the value of $\widehat{w}$, from the most to the least differentially expressed genes. The estimates $\widehat{w}$ are obtained from the fit of our model to the 11 studies selected for the analysis.

Rank	Genes	$\widehat{w}$	sd($\widehat{w}$)	$\widehat{\gamma}$	sd($\widehat{\gamma}$)
101	CBS	0.74	0.26	0.99	0.57
102	TROAP	0.74	0.25	0.96	0.55
103	FXYD3	0.74	0.26	1.01	0.60
104	DKFZP586A0522	0.74	0.33	1.60	1.14
105	KIAA1536	0.72	0.33	1.52	1.14
106	GPR19	0.72	0.26	0.93	0.55
107	SE20-4	0.72	0.33	1.65	1.28
108	SFN	0.71	0.26	0.92	0.55
109	ISG15	0.70	0.26	0.91	0.58
110	SLC6A6	0.70	0.26	0.91	0.57
111	MAS1	0.70	0.26	0.90	0.56
112	KRT2	0.70	0.26	0.89	0.55
113	ESPL1	0.70	0.26	0.89	0.53
114	CDH6	0.69	0.26	0.89	0.58
115	SLPI	0.69	0.26	0.86	0.57
116	TRIP13	0.68	0.25	0.83	0.52
117	DCN	0.67	0.27	-0.87	0.57
118	RXRG	0.66	0.25	0.78	0.51
119	GRM8	0.66	0.25	0.79	0.50
120	MTHFD2	0.66	0.25	0.78	0.51
121	FLJ13236	0.66	0.32	1.09	0.86
122	SPINT1	0.65	0.27	0.83	0.58
123	UCP2	0.65	0.25	0.76	0.52
124	TACSTD2	0.65	0.25	0.76	0.55
125	DES	0.65	0.25	-0.74	0.51
126	MAGEA1	0.64	0.25	0.75	0.50
127	MCM4	0.64	0.25	0.75	0.49
128	LMNB1	0.64	0.25	0.76	0.52
129	GIF	0.64	0.25	0.75	0.50
130	CPT1B	0.64	0.30	0.93	0.76
131	TACR3	0.64	0.25	0.73	0.48
132	LAD1	0.64	0.25	0.74	0.50
133	LCN2	0.63	0.25	0.71	0.49
134	ETV4	0.62	0.25	0.69	0.48
135	DLC1	0.62	0.26	-0.73	0.55
136	MAOA	0.62	0.24	-0.68	0.48
137	AQP5	0.62	0.24	0.68	0.48
138	S100A8	0.61	0.24	-0.67	0.50
139	CAV1	0.61	0.24	-0.68	0.46
140	GCG	0.61	0.24	0.66	0.43
141	HOXB1	0.61	0.24	0.66	0.48
142	CDC25C	0.60	0.24	0.66	0.47
143	HNF4A	0.60	0.24	0.65	0.45
144	MMP13	0.60	0.24	0.64	0.46
145	PLK1	0.60	0.24	0.65	0.44
146	PTGIS	0.60	0.25	-0.66	0.50
147	DHCR24	0.60	0.24	0.65	0.44
148	KIFC1	0.60	0.24	0.64	0.44
149	SOX17	0.60	0.26	0.69	0.55
150	C4A	0.60	0.34	1.06	1.01

Rank	Genes	$\widehat{w}$	sd($\widehat{w}$)	$\widehat{\gamma}$	sd($\widehat{\gamma}$)
151	DUSP1	0.60	0.24	-0.63	0.49
152	EPHA1	0.60	0.24	0.63	0.44
153	SPRR1B	0.60	0.24	0.64	0.44
154	CFD	0.60	0.24	-0.64	0.43
155	ID3	0.60	0.24	-0.63	0.49
156	CDH1	0.59	0.24	0.63	0.45
157	FHL1	0.59	0.23	-0.61	0.43
158	PHOX2B	0.59	0.23	0.61	0.43
159	BIK	0.59	0.23	0.61	0.44
160	GLP1R	0.59	0.23	0.63	0.45
161	GABRR1	0.59	0.23	0.61	0.43
162	MMP7	0.59	0.24	0.62	0.47
163	WNT7A	0.59	0.23	0.60	0.44
164	HIST1H2BG	0.58	0.28	0.69	0.59
165	LMOD1	0.58	0.23	-0.59	0.43
166	MYCL1	0.58	0.23	0.59	0.43
167	PNOC	0.58	0.23	0.59	0.45
168	SLC15A1	0.58	0.23	0.59	0.43
169	SCGB1D2	0.57	0.26	0.61	0.57
170	DSC2	0.57	0.23	0.58	0.41
171	PLAG1	0.57	0.23	0.58	0.43
172	CR1	0.57	0.22	0.58	0.41
173	ALDH1A3	0.57	0.23	-0.57	0.46
174	SLC10A1	0.57	0.23	0.58	0.41
175	NAP1L3	0.57	0.22	-0.58	0.42
176	MEP1B	0.57	0.22	0.57	0.40
177	IL9	0.57	0.22	0.57	0.42
178	CDC25A	0.56	0.22	0.57	0.42
179	SPOCK1	0.56	0.22	-0.55	0.41
180	CYP2A6	0.56	0.22	0.56	0.41
181	FLJ14957	0.56	0.34	0.98	1.05
182	MCM2	0.56	0.22	0.55	0.40
183	PROCR	0.56	0.22	-0.55	0.41
184	COL19A1	0.56	0.22	0.54	0.41
185	ALDH3B2	0.56	0.22	0.55	0.39
186	HTR1D	0.56	0.21	0.54	0.39
187	RUNX2	0.55	0.22	0.54	0.40
188	TRIM29	0.55	0.22	0.53	0.42
189	PLS1	0.55	0.22	0.53	0.40
190	RTN1	0.55	0.21	-0.53	0.39
191	MCF2	0.55	0.22	0.52	0.39
192	NR2F6	0.55	0.22	0.52	0.38
193	CCL20	0.55	0.21	0.52	0.40
194	CACNB4	0.55	0.21	0.51	0.38
195	NPY1R	0.55	0.21	-0.53	0.40
196	LAMA2	0.55	0.21	-0.52	0.38
197	PLN	0.55	0.21	-0.52	0.38
198	IDH2	0.55	0.22	0.51	0.40
199	MSLN	0.55	0.21	0.52	0.39
200	RGS1	0.55	0.22	0.52	0.45

Table A.3 – List of top 201-296 differentially expressed genes. Genes are ordered according to the value of $\widehat{w}$, from the most to the least differentially expressed genes. The estimates $\widehat{w}$ are obtained from the fit of our model to the 11 studies selected for the analysis.

Rank	Genes	$\widehat{w}$	sd($\widehat{w}$)	$\widehat{\gamma}$	sd($\widehat{\gamma}$)
201	KIF1A	0.54	0.21	0.51	0.40
202	PCDH11X	0.54	0.21	0.52	0.38
203	GPM6A	0.54	0.21	-0.51	0.37
204	SIM2	0.54	0.21	0.50	0.40
205	TNNI3	0.54	0.21	0.51	0.38
206	CDC7	0.54	0.21	0.52	0.39
207	ERBB3	0.54	0.21	0.51	0.38
208	IFI27	0.54	0.21	0.51	0.40
209	MMP1	0.54	0.21	0.50	0.39
210	SDC2	0.54	0.21	-0.50	0.37
211	ELAVL2	0.54	0.21	0.50	0.38
212	EFG1	0.54	0.33	0.94	1.07
213	MPZ	0.54	0.20	0.50	0.38
214	CXCR4	0.54	0.21	0.48	0.41
215	DEFA4	0.54	0.21	0.50	0.37
216	RFC4	0.53	0.21	0.49	0.38
217	BMP2	0.53	0.20	-0.47	0.36
218	KLF2	0.53	0.22	-0.53	0.39
219	ZNF165	0.53	0.20	0.48	0.36
220	IFNA5	0.53	0.20	0.48	0.36
221	LOC286286	0.53	0.33	0.85	0.98
222	PLK4	0.53	0.20	0.47	0.35
223	ST14	0.53	0.20	0.46	0.35
224	CASR	0.53	0.20	0.46	0.37
225	HK2	0.52	0.20	0.46	0.37
226	EPS8	0.52	0.20	-0.47	0.36
227	E2F3	0.52	0.20	0.46	0.36
228	DDR1	0.52	0.20	0.46	0.36
229	SLC5A1	0.52	0.20	0.46	0.35
230	PGR	0.52	0.19	-0.46	0.35
231	CXCL11	0.52	0.21	0.46	0.37
232	PTH	0.52	0.20	0.45	0.35
233	AP1M2	0.52	0.22	0.49	0.39
234	GGH	0.52	0.20	0.46	0.35
235	NPPB	0.52	0.21	0.45	0.40
236	BAMBI	0.52	0.20	-0.46	0.36
237	SMPDL3B	0.52	0.20	0.47	0.35
238	TYRP1	0.52	0.20	0.45	0.34
239	EMP3	0.52	0.19	-0.46	0.32
240	DFNA5	0.52	0.22	-0.49	0.41
241	BCAT1	0.52	0.20	0.45	0.36
242	XDH	0.52	0.19	0.44	0.34
243	HIST1H1C	0.52	0.19	0.44	0.34
244	GNAT1	0.52	0.19	0.45	0.33
245	KIAA0222	0.52	0.32	0.82	0.95
246	APOD	0.52	0.20	-0.47	0.34
247	MAL2	0.52	0.24	0.54	0.46
248	FRMPD4	0.51	0.19	0.44	0.33
249	CRABP1	0.51	0.19	0.42	0.35
250	CHI3L1	0.51	0.22	0.48	0.45
251	COL4A6	0.51	0.19	-0.44	0.34
252	RAD51	0.51	0.19	0.46	0.35
253	SOX9	0.51	0.19	0.43	0.33
254	KRT23	0.51	0.22	0.48	0.41
255	TDO2	0.51	0.19	0.44	0.34
256	MCM7	0.51	0.19	0.43	0.32
257	TACR1	0.51	0.19	0.44	0.33
258	GRB7	0.51	0.19	0.42	0.32
259	HTR2C	0.51	0.19	0.43	0.32
260	AVPR1B	0.51	0.19	0.44	0.32
261	CD302	0.51	0.19	-0.43	0.32
262	GEM	0.51	0.19	-0.43	0.34
263	ANKRD1	0.51	0.18	0.42	0.32
264	MEST	0.51	0.19	0.42	0.33
265	MFAP4	0.51	0.19	-0.43	0.33
266	UGT8	0.51	0.19	0.42	0.35
267	CHRNB4	0.51	0.18	0.43	0.31
268	KRT19	0.51	0.23	0.52	0.44
269	CDKL5	0.51	0.19	0.43	0.32
270	AOC3	0.51	0.19	-0.42	0.34
271	NBEA	0.51	0.21	-0.47	0.36
272	KLK10	0.51	0.21	0.44	0.39
273	CSN2	0.51	0.18	0.42	0.31
274	APOF	0.51	0.19	0.42	0.34
275	PRKX	0.51	0.18	0.42	0.30
276	CHRM5	0.51	0.19	0.42	0.32
277	ABO	0.50	0.18	0.42	0.33
278	PAX3	0.50	0.18	0.41	0.33
279	PAFAH1B3	0.50	0.19	0.42	0.32
280	ARHI	0.50	0.32	0.78	0.95
281	JAK3	0.50	0.18	0.42	0.31
282	EPHA5	0.50	0.18	0.43	0.31
283	SIX1	0.50	0.19	0.39	0.33
284	CXADR	0.50	0.18	0.40	0.33
285	ITGB8	0.50	0.18	0.41	0.32
286	NPPA	0.50	0.18	0.39	0.33
287	MC4R	0.50	0.19	0.40	0.32
288	PRSS1	0.50	0.18	0.41	0.32
289	SCGB2A2	0.50	0.19	0.42	0.33
290	GNAO1	0.50	0.18	0.41	0.30
291	NT5E	0.50	0.18	-0.40	0.32
292	PRKCI	0.50	0.18	0.41	0.31
293	EDN2	0.50	0.18	0.41	0.31
294	E2F1	0.50	0.18	0.41	0.32
295	RAG1	0.50	0.19	0.41	0.33
296	CKM	0.50	0.18	0.41	0.32

A.3 Computations of the posterior distributions and likelihood of the empirical Bayes model

We start by computing the posterior density of θ_g. As the calculation is the same for all g, we omit the subscript in what follows,

$$\begin{aligned}\pi\left(\theta \mid Z^{(1)},\ldots,Z^{(L_1)}\right) &= \frac{\pi(\theta) f\left(Z^{(1)},\ldots,Z^{(L_1)} \mid \theta\right)}{f\left(Z^{(1)},\ldots,Z^{(L_1)}\right)} \\ &= \frac{\pi(\theta) f\left(Z^{(1)},\ldots,Z^{(L_1)} \mid \theta\right)}{\int_\theta f\left(Z^{(1)},\ldots,Z^{(L_1)} \mid \theta\right)\pi(\theta)\mathrm{d}\theta}.\end{aligned} \tag{A.1}$$

We compute the numerator:

$$\begin{aligned}\pi(\theta) f(Z^{(1)},\ldots,Z^{(L_1)}) &= \left[(1-p)\delta_0 + \frac{p}{\sqrt{2\pi}\tau}\exp\left\{\frac{-\theta^2}{2\tau^2}\right\}\right]\prod_{l=1}^{L_1}\frac{1}{\sqrt{2\pi}\sigma_l}\exp\left\{\frac{-1}{2\sigma_l^2}(Z^{(l)}-\theta)^2\right\} \\ &= \underbrace{(1-p)\prod_{l=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{Z^{(l)}}{\sigma_l}\right)}_{=w2}\delta_0 + \frac{p}{\sqrt{2\pi}^{L_1+1}\tau\prod_{l=1}^{L_1}\sigma_l}\exp\left\{\frac{-1}{2}\sum_{l=1}^{L_1}\frac{Z^{(l)2}-2\theta Z^{(l)}+\theta^2}{\sigma_l^2} - \frac{\theta^2}{2\tau^2}\right\} \\ &= w_2\delta_0 + \frac{p}{\sqrt{2\pi}^{L_1+1}\tau\prod_{l=1}^{L_1}\sigma_l}\exp\left\{\frac{-1}{2}\sum_{l=1}^{L_1}\frac{Z^{(l)2}}{\sigma_l^2}\right\}\times \\ &\quad \exp\left\{\frac{-1}{2}\left(\frac{1}{\tau^2}+\sum_{l=1}^{L_1}\frac{1}{\sigma_l^2}\right)\left[\theta - \sum_{l=1}^{L_1}\frac{Z^{(l)}}{\sigma_l^2}\left(\frac{1}{\tau^2}+\sum_{l=1}^{L_1}\frac{1}{\sigma_l^2}\right)^{-1}\right]^2\right\}\exp\left\{\frac{1}{2}\left(\sum_{l=1}^{L_1}\frac{Z^{(l)}}{\sigma_l^2}\right)^2\left(\frac{1}{\tau^2}+\sum_{l=1}^{L_1}\frac{1}{\sigma_l^2}\right)\right\} \\ &= w_2\delta_0 + \frac{p}{\sqrt{2\pi}\tau}\prod_{l=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{Z^{(l)}}{\sigma_l}\right)\exp\left\{\frac{1}{2b}\left(\sum_{l=1}^{L_1}\frac{Z^{(l)}}{\sigma_l^2}\right)^2\right\}\exp\left\{\frac{-1}{2}b\left(\theta - \sum_{l=1}^{L_1}\frac{Z^{(l)}}{\sigma_l^2}b^{-1}\right)^2\right\} \\ &= w_2\delta_0 + w_1\sqrt{b}\varphi\left(\frac{\theta - b^{-1}\sum_{l=1}^{L_1}Z^{(l)}/\sigma_l^2}{\sqrt{b}^{-1}}\right)\end{aligned}$$

where

$$\begin{aligned}b &= \frac{1}{\tau^2}+\sum_{l=1}^{L_1}\frac{1}{\sigma_l^2}, \quad w_1 = \frac{p}{\sqrt{b}\tau}\exp\left\{\frac{1}{2b}\left(\sum_{l=1}^{L_1}\frac{Z^{(l)}}{\sigma_l^2}\right)^2\right\}\prod_{l=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{Z^{(l)}}{\sigma_l}\right), \\ w_2 &= (1-p)\prod_{l=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{z^{(l)}}{\sigma_l}\right), \quad p_z = \frac{w_1}{w_1+w_2}.\end{aligned}$$

We now compute the denominator of (A.1):

$$
\begin{aligned}
&\int_\theta f\left(Z^{(1)},\dots,Z^{(L_1)} \mid \theta\right)\pi(\theta)\mathrm{d}\theta = \int_\theta \pi(\theta)\prod_{l=1}^{L_1} f\left(Z^{(l)} \mid \theta\right)\mathrm{d}\theta \\
&= w_2 + \int_\theta \frac{p}{\sqrt{2\pi\tau}}\exp\left\{\frac{-\theta^2}{2\tau^2}\right\}\prod_{l=1}^{L_1}\frac{1}{\sqrt{2\pi}\sigma_l}\exp\left\{\frac{-(Z^{(l)}-\theta)^2}{2\sigma_l^2}\right\}\mathrm{d}\theta \\
&= w_2 + \frac{p}{\sqrt{2\pi\tau}}\prod_{L=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{Z^{(l)}}{\sigma_l}\right)\exp\left\{\left(\sum_{l=1}^{L_1}\frac{Z^{(l)}}{2\sigma_l^2}\right)^2\frac{1}{2b}\right\}\int_\theta \exp\left\{\frac{-b}{2}\left(\theta-\sum_{l=1}^{L_1}\frac{Z^{(l)}}{b\sigma_l^2}\right)^2\right\}\mathrm{d}\theta \\
&= w_2 + \frac{p}{\sqrt{b}\tau}\prod_{l=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{Z^{(l)}}{\sigma_l}\right)\exp\left\{\left(\sum_{l=1}^{L_1}\frac{Z^{(l)}}{2\sigma_l^2}\right)^2\frac{1}{2b}\right\}\underbrace{\int_\theta \frac{\sqrt{b}}{\sqrt{2\pi}}\exp\left\{\frac{-b}{2}\left(\theta-\sum_{l=1}^{L_1}\frac{Z^{(l)}}{\beta\sigma_l^2}\right)^2\right\}\mathrm{d}\theta}_{=1} \\
&= w_2 + w_1.
\end{aligned}
$$

Therefore, with $p_z = \frac{w_1}{w_1+w_2}$, we obtain the posterior density for θ:

$$
\pi(\theta \mid z^{(1)},\dots,z^{(L_1)}) = p_z\sqrt{b}\,\varphi\left(\frac{\theta - b^{-1}\sum_{l=1}^{L_1} z^{(l)}/\sigma_l^2}{\sqrt{b}^{-1}}\right) + (1-p_z)\delta_0.
$$

As it is possible to obtain an explicit expression for the likelihood, we adopt an empirical Bayes approach, and estimate the parameters $\sigma_1,\dots,\sigma_{L_1},p,\tau$ by maximum likelihood. The likelihood is

$$
\begin{aligned}
&\mathscr{L}(p,\sigma_1,\dots,\sigma_{L_1},\tau) = \prod_{g=1}^{G} f\left(Z_g^{(1)},\dots,Z_g^{(L_1)} \mid \sigma_1,\dots,\sigma_{L_1},p,\tau\right) \\
&= \prod_{g=1}^{G}\int_\theta \pi(\theta)\prod_{l=1}^{L_1} f\left(Z_g^{(l)} \mid \theta\right)\mathrm{d}\theta \\
&= \prod_{g=1}^{G}(1-p)\underbrace{\prod_{l=1}^{L_1}\frac{1}{\sigma_l}\varphi\left(\frac{Z_g^{(l)}}{\sigma_l}\right)}_{=\tilde{w}_2} + \frac{p\tilde{w}_2}{\tau\sqrt{b}}\exp\left\{\left(\sum_{l=1}^{L_1}\frac{Z^{(l)}}{2\sigma_l^2}\right)^2\frac{1}{2b}\right\},
\end{aligned}
$$

so the log likelihood is

$$
l(\sigma_1,\dots,\sigma_{L_1},p,\tau) = \sum_{g=1}^{G}\log\left[(1-p)\tilde{w}_2 + \frac{p\tilde{w}_2}{\tau\sqrt{b}}\exp\left\{\frac{1}{2b}\left(\sum_{l=1}^{L_1}\frac{Z_g^{(l)}}{\sigma_l^2}\right)^2\right\}\right].
$$

A.4 List of notations

In this section, we give a list of the notations used in this thesis.

- DE : differentially expressed;
- SOC: serous ovarian cancer;
- SAS: spike and slab prior;
- HS: horseshoe prior;
- NG: normal-gamma prior;
- φ: standard normal density;
- Φ: standard normal distribution;
- z_α: αth quantile of the standard normal distribution;
- $\mathcal{N}_b^a$: truncated normal distribution on the interval $[a;b]$;
- $\xrightarrow{d}$: convergence in distribution;
- $\delta_x(\theta)$: distribution with a spike at $\theta = x$;
- $I_{\{A\}}$ indicator function which is equal to 1 when condition A is satisfied;
- $|S|$: determinant of S, with S a matrix;
- $|\mathscr{S}|$: size of the set $\mathscr{S}$;
- $X_{G\times N}$: matrix X with G rows and N columns;
- $\sim F$: follows distribution F;
- $T \underset{H_0}{\sim} F$: under the null distribution, T follows the distribution F;

A.5 List of model parameters

The following tables give lists of parameters encountered in model (3.2) and model of Section 6.4, as well as the parameters used in simulations of Chapter 4 and Sections 6.4.2 and 7.2.

Table A.4 – Parameters of model (3.2).

Type 1	
$Y_{gj}^{(l)}$	gene expression of gene g for individual j in study l
$\mu_g^{(l)}$	baseline expression mean of gene g in study l
$\beta_g^{(1,l)}$	differential expression parameter for gene g in Type 1 study l
$\sigma_g^{2(l)}$	variance of the expression of gene g among all patients in study l
b_1, b_2	hyperparameters for σ_g
Type 2 (Type 3 is similar)	
$Z_g^{(l)}$	z-score of gene g in study l
$\beta_g^{(2,l)}$	differential expression parameter for gene g in Type 2 study l
Type 4	
$R_g^{(l)}$	Rank of gene g in study l
$u_g^{(l)}$	latent variable for gene g in study l; $R_g = \text{rank}\left(\lvert u_g^{(l)} \rvert\right)$
$\beta_g^{(4,l)}$	differential expression parameter for gene g in Type 4 study l
$\sigma_{u,g}^{2(l)}$	variance of the latent variable u_g in study l
d_1, d_2	hyperparameters for $\sigma_{u,g}^{2(l)}$

Table A.5 – Parameters used in the simulations of Chapter 4.

parameter	meaning	usual value
G	number of genes	200
N	number of samples	50
n_1	number of cancer samples	40
$N - n_1$	number of control samples	10
L	number of studies	5
L_1	number of Type 1 studies	2
L_2, L_3, L_4	number of Types 2, 3 and 4 studies	1
a	differential expression parameter	$0.2, \ldots, 2$
k	number of differentially expressed genes	10
R_{sim}	number of simulations	100 or 500
R	number of iterations in the Gibbs sampling algorithm	31500
B	number of blocks in the simulated covariance matrix	1

Table A.6 – Parameters used in the empirical Bayesian model of Section 6.4 to estimate a common correlation matrix.

parameter	meaning
$R^{(l)}$	$G \times G$ gene-gene correlation matrix of study l
$r^{(l)}$	vectorized upper triangular part of $R^{(l)}$ of length $G(G-1)/2$
$Z^{(l)}$	Fisher transformed $r^{(l)}$
θ_g	mean of $Z^{(l)}$
σ_l	variance of $Z^{(l)}$
p	probability that θ_g is not null
$\tilde{\theta}_g$	posterior median of θ_g
p_z	vector of posterior probabilities that θ is not null
τ^2	variance of θ_g

Table A.7 – Parameters used in the simulations of Sections 6.4.2 and 7.2.

parameter	meaning	usual value
G	number of genes	100 or 500
X_t	true gene expression matrix	size $G \times N$
Σ_t	true block diagonal covariance matrix	size $G \times G$, with B blocks
R_t	true block diagonal correlation matrix	size $G \times G$, with B blocks
B	number of blocks in Σ_t or R_t	5 or 10
ϵ	noise	$\sim \mathcal{N}(0, \sigma_\epsilon^2)$
σ_ϵ^2	variance of the noise	$0.1, \ldots, 1$
ρ_b	correlation of the elements in the bth block	$\sim \mathcal{U}(0.5, 1)$ or $\sim \mathcal{U}(0, 1)$ or fixed between 0.1 and 0.9
R_{sim}	number of simulations	100
L	number of studies	3

Bibliography

Abadir, K. M., Distaso, W. and Zikes, F. (2012) Design-free estimation of large variance matrices. Technical report, Imperial College London.

Albert, J. H. and Chib, S. (1993) Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association* **88**, 669–679.

Bai, J. and Shi, S. (2011) Estimating high dimensional covariance matrices and its applications. *Annals of Economics and Finance* **12**, 199–215.

Bandos, A., Rockette, H. and Gur, D. (2004) A permutation test sensitive to differences in areas for comparing ROC curves from a paired design. *Statistics in Medicine* **24**, 2873–2893.

Banerjee, S. and Ghosal, S. (2013) Posterior convergence rates for estimating large precision matrices using graphical models. `arXiv`:1302.2677v1.

Barnard, J., McCulloch, R. and Meng, X.-L. (2000) Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica* **10**, 1281–1312.

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M. *et al.* (2013) NCBI GEO: archive for functional genomics data sets–update. *Nucleic Acids Research* **41**, D991–D995.

Bartlett, M. S. (1937) Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences* **160**, 268–282.

Bengtsson, H., Simpson, K., Bullard, J. and Hansen, K. (2008) `aroma.affymetrix`: A generic framework in R for analyzing small to very large Affymetrix data sets in bounded memory. Technical report, University of California, Berkley.

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* **57**, 289–300.

Bickel, P. J. and Levina, E. (2008) Covariance regularization by thresholding. *The Annals of Statistics* **36**, 2577–2604.

Bignotti, E., Tassi, R. A., Calza, S., Ravaggi, A., Romani, C., Rossi, E., Falchetti, M., Odicino, F. E., Pecorelli, S. and Santin, A. D. (2006) Differential gene expression profiles between tumor biopsies and short-term primary cultures of ovarian serous carcinomas: identification of novel molecular biomarkers for early diagnosis and therapy. *Gynecologic Oncology* **103**, 405–416.

de Borda, J. C. (1781) Mémoire sur les élections au scrutin. *Histoire de l'Académie des Sciences* pp. 657–664.

Bradley, R. A. and Terry, M. E. (1952) Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* **39**, 324–345.

Braun, T. and Alonzo, T. (2008) A modified sign test for comparing paired ROC curves. *Biostatistics* **9**, 364–372.

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., Hostege, 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 experiement (MIAME)-toward standards for microarray data. *Nature Genetics* **29**, 365–371.

Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P. and Sansone, S.-A. (2003) ArrayExpress: a public repository for microarray gene expression data at the EBI. *Nucleic Acids Research* **31**, 68–71.

Brooks, S. P., Gelman, A., Jones, G. L. and Meng, X. L. (eds) (2011) *Handbook of Markov Chain Monte Carlo*. London: Chapman & Hall/CRC Press.

Cai, T. and Liu, W. (2011) Adaptive thresholding of sparse covariance matrix estimation. *Journal of the American Statistical Association* **106**, 672–684.

Cancer Genome Atlas Research Network (2011) Integrated genomic analyses of ovarian carcinoma. *Nature* **474**, 609–615.

Carlin, B. and Louis, T. (2000) *Bayes and Empirical Bayes Methods for Data Analysis*. Second edition. New York: Chapman & Hall.

Carvalho, C. M., Chang, J., Lucas, J. E., Nevins, J. R., Wang, Q. and West, M. (2008) High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association* **103**, 1438–1456.

Carvalho, C. M., Massam, H. and West, M. (2007) Simulation of hyper-inverse Wishart distributions in graphical models. *Biometrika* **94**, 647–659.

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2009) Handling sparsity via the horseshoe. *Journal of Machine Learning Research Workshop and Conference Proceedings* **5**, 73–80.

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2010) The horseshoe estimator for sparse signals. *Biometrika* **97**, 465–480.

Chaipitak, S. and Chongcharoen, S. (2013) A test for testing the equality of two covariance matrices for high-dimensional data. *Journal of Applied Sciences* **13**, 270–277.

Chang, L.-C., Lin, H.-M., Sibille, E. and Tseng, G. C. (2013) Meta-analysis methods for combining multiple expression profiles: comparisons, statistical characterization and an application guideline. *BMC Bioinformatics* **14**, 368–383.

Chen, M., Zhang, M., Wang, X. and Xiao, G. (2013) A powerful Bayesian meta-analysis method to integrate multiple gene set enrichment studies. *Bioinformatics* **29**, 862–869.

Choi, J. K., Yu, U., Kim, S. and Yoo, O. J. (2003) Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics* **19**, i84–i90.

Chon, H. S. and Lancaster, J. M. (2011) Microarray-based gene expression studies in ovarian cancer. *Cancer Control* **18**, 8–15.

Cochran, W. G. (1950) The comparison of percentages in matched samples. *Biometrika* **37**, 256–266.

Cochrane (2010) The Cochrane Library. http://www.cochrane.org/.

Congdon, P. D. (2010) *Applied Bayesian Hierarchical Methods*. London: Chapman & Hall/CRC Press.

Conlon, E. M., Postier, B. L., Methé, B. A., Nevin, K. P. and Lovley, D. R. (2012) A Bayesian model for pooling gene expression studies that incorporates co-regulation information. *PloS One* **7**, e52137.

Conlon, E. M., Song, J. J. and Liu, A. (2007) Bayesian meta-analysis models for microarray data: a comparative study. *BMC Bioinformatics* **8**, 80–101.

Corander, J., Koski, T., Pavlenko, T. and Tillander, A. (2013) Bayesian block-diagonal predictive classifier for Gaussian data. In *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, eds R. Kruse, M. R. Berthold, C. Moewes, M. A. Gil, P. Grzegorzewski and O. Hryniewicz, pp. 543–551. Berlin: Springer.

Damien, P. (1999) Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society, Series B* **61**, 331–344.

Danaher, P., Wang, P. and Witten, D. M. (2014) The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B* **76**, 373–397.

Datta, J. and Ghosh, K. (2012) Asymptotic properties of Bayes risk for the horseshoe prior. *Bayesian Analysis* **7**, 771–792.

Davison, A. C. (2003) *Statistical Models*. Cambridge: Cambridge University Press.

Davison, A. C. (2008) Some challenges for statistics. *Statistical Methods and Applications* **17**, 167–181.

Davison, A. C., Fraser, D. A. S., Reid, N. and Sartori, N. (2014) Accurate directional inference for vector parameters in linear exponential families. *Journal of the American Statistical Association* **109**, 302–314.

DeConde, R. P., Hawley, S., Falcon, S., Clegg, N., Knudsen, B. and Etzioni, R. (2006) Combining results of microarray experiments: a rank aggregation approach. *Statistical Applications in Genetics and Molecular Biology* **5**, 1–24.

DeLong, E. R., DeLong, D. M. and Clarke-Pearson, D. L. (1988) Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics* **44**, 837–845.

Donninger, H., Bonome, T., Radonovich, M., Pise-Masison, C. A., Brady, J., Shih, J. H., Barrett, J. C. and Birrer, M. J. (2004) Whole genome expression profiling of advance stage papillary serous ovarian cancer reveals activated pathways. *Oncogene* **23**, 8065–8077.

Draghici, S. (2003) *Data Analysis Tools for DNA Microarrays*. Boca Raton: Chapman & Hall / CRC.

Dudoit, S. and Fridlyand, J. (2002) A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology* **3**, 1–21.

Dwork, C., Kumar, R., Naor, M. and Sivakumar, D. (2001) Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web*, pp. 613–622.

Eberlein, E. and Hammerstein, E. A. V. (2004) Generalized hyperbolic and inverse Gaussian distributions: limiting cases and approximation of processes. In *Seminar on Stochastic Analysis, Random Fields and Applications IV*, eds R. Dalang, M. Dozzi and F. Russo, volume 58 of *Progress in Probability*, pp. 221–264.

Edgar, R., Domrachev, M. and Lash, A. E. (2002) Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research* **30**, 207–210.

Efron, B. and Tibshirani, R. J. (2007) On testing the significance of sets of genes. *The Annals of Applied Statistics* **1**, 107–129.

Ein-Dor, L., Kela, I., Getz, G., Givol, D. and Domany, E. (2005) Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics* **21**, 171–178.

Ein-Dor, L., Zuk, O. and Domany, E. (2006) Thousand of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proceedings of the National Academy of Sciences* **103**, 5923–5928.

Eisenberg, E. and Levanon, E. Y. (2013) Human housekeeping genes, revisited. *Trends in Genetics* **29**, 569–574.

Fackler, P. L. (2005) Notes on matrix calculus.

Falcon, S. and Gentleman, R. (2007) Using GOstats to test gene lists for GO term association. *Bioinformatics* **23**, 257–258.

Fan, J., Fan, Y. and Lv, J. (2008) High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics* **147**, 186–197.

Fan, J., Liao, Y. and Mincheva, M. (2011) High dimensional covariance matrix estimation in approximate factor models. *Annals of Statistics* **39**, 3320–3356.

Fan, J., Liao, Y. and Mincheva, M. (2013) Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B* **75**(4), 603–680.

Fisher, R. A. (1932) *Statistical Methods for Research Workers*. Fourth edition. London: Oliver and Boyd.

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

Gamerman, D. (1997) *Markov Chain Monte Carlo*. New York: Chapman & Hall.

Gan, G., Ma, C. and Wu, J. (2007) *Data clustering: Theory, Algorithms, and Applications*. Philadelphia: SIAM.

Ganzfried, B. F., Riester, M., Haib-Kains, B., Risch, T., Tyekucheva, S., Jazic, I., Wang, X. V., Ahmadifar, M., Birrer, M. J., Parmigiani, G., Huttenhower, C. and Warldron, L. (2013) `curatedOvarianData`: clinically annotated data for the ovarian cancer transcriptome. *Database* **2013**, bat013.

Gaskins, J. T. and Daniels, M. J. (2013) A nonparametric prior for simultaneous covariance estimation. *Biometrika* **100**, 125–138.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013) *Bayesian Data Analysis*. Second edition. New York: Chapman & Hall/CRC Press.

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721–741.

George, E. and McCulloch, R. (1993) Variable selection via Gibbs sampling. *Journal of the American Statistical Association* **88**, 881–889.

George, E. O. (1977) *Combining independent one-sided and two-sided statistical tests – Some theory and applications*. Ph.D. thesis, University of Rochester.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (eds) (1996) *Markov Chain Monte Carlo in Practice*. New York: Chapman & Hall.

Goeman, J. J. and Bühlmann, P. (2007) Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics* **23**, 980–987.

Goldstein, D. R., Delorenzi, M., Luthi-Carter, R. and Sengstag, T. (2009) Comparison of meta-analysis to combined analysis of a replicated microarray study. In *Meta-analysis and Combining Information in Genetics and Genomics*, eds R. Guerra, B. Allison and D. R. Goldstein. Boca Raton: Chapman & Hall/CRC.

Green, P. J. (2001) A Primer on Markov Chain Monte Carlo. In *Complex Stochastic Systems*, eds O. E. Barndorff-Nielsen, D. R. Cox and C. Klüppelberg, pp. 1–62. New York: Chapman & Hall/CRC.

Griffin, J. E. and Brown, P. J. (2010) Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis* **5**, 171–188.

Guerra, R. and Goldstein, D. R. (eds) (2009) *Meta-analysis and Combining Information in Genetics and Genomics*. Boca Raton: Chapman & Hall/CRC.

Guo, J., Levina, E., Michailidis, G. and Zhu, J. (2011) Joint estimation of multiple graphical models. *Biometrika* **98**, 1–15.

Hafdahl, A. R. (2007) Combining correlation matrices: Simulation analysis of improved fixed-effects methods. *Journal of Educational and Behavioral Statistics* **32**, 180–205.

Hastie, T. J., Friedman, J. and Tibshirani, R. J. (2009) *The Elements of Statistical Learning*. Second edition. Springer.

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109.

Huang, D. W., Sherman, B. T. and Lempicki, R. A. (2008) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols* **4**, 44–57.

Hung, J.-H., Yang, T.-H., Hu, Z., Weng, Z. and DeLisi, C. (2012) Gene set enrichment analysis: performance evaluation and usage guidelines. *Briefings in Bioinformatics* **13**, 281–291.

Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B. and Speed, T. P. (2003) Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Research* **31**, e15.

Irizarry, R. A., Wang, C., Zhou, Y. and Speed, T. P. (2009) Gene set enrichment analysis made simple. *Statistical Methods in Medical Research* **18**, 565–575.

Ishwaran, H. and Rao, J. S. (2003) Detecting differentially expressed genes in microarrays using Bayesian model selection. *Journal of the American Statistical Association* **98**, 438–455.

Ishwaran, H. and Rao, J. S. (2005) Spike and slab gene selection for multigroup microarray data. *Journal of the American Statistical Association* **100**, 764–780.

Jacob, F., Goldstein, D. R., Fink, D. and Heinzelmann-Schwarz, V. (2009) Proteogenomic studies in epithelial ovarian cancer: established knowledge and future needs. *Biomarkers in Medicine* **3**, 743–756.

Johnstone, I. M. (2001) On the distribution of the largest eigenvalue in principal components analysis. *Annals of Statistics* **29**, 295–327.

Johnstone, I. M. and Silverman, B. W. (2005) Empirical Bayes selection of wavelet thresholds. *Annals of Statistics* **33**, 1700–1752.

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

Kooperberg, C., Aragaki, A., Strand, A. D. and Olson, J. (2005) Significance testing for small microarray experiments. *Statistics in Medicine* **24**, 2281–2298.

Kulesh, D. A., Clive, D. R., Zarlenga, D. S. and Greene, J. J. (1987) Identification of interferon-modulated proliferation-related cDNA sequences. *Proceedings of the Nationaly Academy of Sciences of the United States of America* **84**, 8453–8457.

Kulinskaya, E., Morgenthaler, S. and Staudte, R. G. (2008) *Meta Analysis: A Guide to Calibrating and Combining Statistical Evidence*. Chichester: John Wiley.

Langfelder, P. and Horvath, S. (2007) Eigengene networks for studying the relationships between co-expression modules. *BMC Systems Biology* **1**, 1–54.

Langfelder, P. and Horvath, S. (2008) WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* **9**, 559–572.

Langfelder, P., Mischel, P. and Horvath, S. (2013) When is hub gene selection better than standard meta-analysis? *PloS one* **8**, e61505.

Langfelder, P., Zhang, B. and Horvath, S. (2008) Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. *Bioinformatics* **24**, 719–720.

Larsson, O. and Sandberg, R. (2006) Lack of correct data format and comparability limits future integrative microarray research. *Nature Biotechnology* **24**, 1322–1323.

Lashkari, D. A., DeRisi, J. L., McCusker, J. H., Namath, A. F., Gentile, C., Hwang, S. Y., Brown, P. O. and Davis, R. W. (1997) Yeast microarrays for genome wide parallel genetic and gene expression analysis. *Proceedings of the National Academy of Sciences of the United States of America* **94**, 13057–13062.

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

Ledoit, O. and Wolf, M. (2012) Nonlinear shrinkage estimation of large-dimensional covariance matrices. *Annals of Statistics* **40**, 1024–1060.

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdóttir, H., Tamayo, P. and Mesirov, J. P. (2011) Molecular signatures database (MSigDB) 3.0. *Bioinformatics* **27**, 1739–1740.

Lili, L. N., Matyunina, L. V., Walker, L., Benigno, B. B. and McDonald, J. F. (2013) Molecular profiling predicts the existence of two functionally distinct classes of ovarian cancer stroma. *BioMed Research International* **2013**, 1–9.

Lin, S. (2010) Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**, 555–570.

Lin, S. and Ding, J. (2009) Integration of ranked lists via cross entropy Monte Carlo with applications to mRNA and microRNA studies. *Biometrics* **65**, 9–18.

Martoglio, A.-M., Tom, B. D. M., Starkey, M., Corps, A. N., Charnock-Jones, D. S. and Smith, S. K. (2000) Changes in tumorigenesis and angiogenesis related gene transcript abundance profiles in ovarian cancer detected by tailored high density cDNA arrays. *Molecular Medicine* **6**, 750–765.

Meinhold-Heerlein, I., Bauerschlag, D., Zhou, Y., Sapinoso, L. M., Ching, K., H. Frierson, J., Bräutigam, K., Sehouli, J., Stickeler, E., Könsgen, D., Hilpert, F., von Daisenberg, C. S., Pfisterer, J., Bauknecht, T., Jonat, W., Arnold, N. and Hampton, G. M. (2007) An integrated clinical-genomics approach identifies a candidate multi-analyte blood test for serous ovarian carcinoma. *Clinical Cancer Research* **13**, 458–466.

Merck (2014) The Merck manuals, online medical library. http://www.merckmanuals.com.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* **21**, 1087–1092.

MGED (2010) Minimum information about a microarray experiment. http://www.mged.org/Workgroups/MIAME/miame.html.

Mitchell, T. J. and Beauchamp, J. J. (1988) Bayesian variable selection in linear regression. *Journal of the American Statistical Association* **83**, 1023–1032.

Mok, S. C., Bonome, T., Vathipadiekal, V., Bell, A., Johnson, M. E., Wong, K.-K., Park, D.-C., Hao, K., Yip, D. K. P., Donninger, H., Ozbon, L., Samini, G., Brady, J., Randonivich, M., Pise-Masison, C. A., Barett, J. C., Wong, W. H., Welch, W. R., Berkowitz, R. S. and Birrer, M. J. (2009) A gene signature predictive for outcome in advanced ovarian cancer identifies a survival factor: Microfibril-associated glycoprotein 2. *Cancer Cell* **16**, 521–532.

Monti, S., Tamayo, P., Mesirov, J. and Golub, T. (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning* **52**, 91–118.

Mootha, V. K., Lindgren, C. M., Eriksson, K.-F., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E. *et al.* (2003) PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics* **34**, 267–273.

NCBI (2010) pubmed. http://www.ncbi.nlm.nih.gov/pubmed/.

Pang, X. and Gill, J. (2011) Spike and slab prior distributions for simultaneous Bayesian hypothesis testing model selection and prediction or nonlinear outcomes. *Unpublished Manuscript* pp. 1–51.

Parmigiani, G., Garett, E. S., Irizarry, R. A. and Zeger, S. L. (2003) *The Analysis of Gene Expression Data: Methods and Software*. New York: Springer.

Pepe, M., Longton, G. and Janes, H. (2009) Estimation and comparison of receiver operating characteristic curves. *The Stata Journal* **9**, 1–14.

Pihur, V., Datta, S. and Datta, S. (2009) RankAggreg, an R package for weighted rank aggregation. *BMC Bioinformatics* **10**, 62–73.

Polson, N. G. and Scott, J. G. (2010) Shrink globally, act locally: sparse Bayesian regularization and prediction. *Bayesian Statistics* **9**, 501–538.

Pourahmadi, M. (2011) Covariance estimation: The GLM and regularization perspectives. *Statistical Science* **26**, 369–387.

Ramasamy, A., Mondry, A., Holmes, C. C. and Altman, D. G. (2008) Key issues in conducting a meta-analysis of gene expression microarray datasets. *PLoS Medicine* **5**, e184.

Rand, W. M. (1971) Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* **66**, 846–850.

Rhodes, D. R., Barrette, T. R., Rubin, M. A., Ghosh, D. and Chinnaiyan, A. M. (2002) Meta-analysis of microarrays: interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer. *Cancer Research* **62**, 4427–4433.

Rhodes, D. R., Yu, J., Deshpande, N., Varambally, R., Ghosh, D., Barrette, T., Pandey, A. and Chinaiyan, A. M. (2004) Oncomine: A cancer microarray database and integrated data-mining platform. *Neoplasia* **6**, 1–6.

Robbins, H. (1985) *The Empirical Bayes Approach to Statistical Decision Problems*. New York: Springer.

Robert, C. P. and Casella, G. (2005) *Monte Carlo Statistical Methods*. Second edition. New York: Springer-Verlag.

Roberts, G. O., Gelman, A. and Gilks, W. R. (1997) Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability* **7**, 110–120.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C. and Müller, M. (2011) pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics* **12**, 77–85.

Rootzén, H. and Zholud, D. (2014) Efficient estimation of the number of false positives in high-throughput screening. *submitted to Biometrika* pp. 1–11.

Rothman, A. J. (2012) Positive definite estimators of large covariance matrices. *Biometrika* **99**, 733–740.

Rothman, A. J., Bickel, P. J., Levina, E. and Zhu, J. (2008) Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics* **2**, 494–515.

Runcie, D. E. and Mukherjee, S. (2013) Dissecting high-dimensional phenotypes with Bayesian sparse factor analysis of genetic covariance matrices. *Genetics* **194**, 753–767.

Schott, J. R. (2007) A test for the equality of covariance matrices when the dimension is large relative to the sample sizes. *Computational Statistics & Data Analysis* **51**, 6535–6542.

Scott, J. G. (2009) *Bayesian Adjustment for Multiplicity*. Ph.D. thesis, Department of Statistical Science, Duke University.

Scott, J. G. (2011) Bayesian estimation of intensity surfaces on the sphere via needlet shrinkage and selection. *Bayesian Analysis* **6**, 307–327.

Shen, R., Ghosh, D. and Chinnaiyan, A. M. (2004) Prognostic meta-signature of breast cancer developed by two-stage mixture modeling of microarray data. *BMC Genomics* **5**, 94–116.

Smyth, G. K. (2005) limma: Linear models for microarray data. In *Bioinformatics and Computational Biology solutions using R and Bioconductor*, eds R. Gentleman, V. Carey, S. Dudoit, R. Irizarry and W. Huber, pp. 397–420. New York: Springer.

Srivastava, M. S., Kollo, T. and von Rosen, D. (2011) Some tests for the covariance matrix with fewer observations than the dimension under non-normality. *Journal of Multivariate Analysis* **102**, 1090–1103.

Srivastava, M. S. and Yanagihara, H. (2010) Testing the equality of several covariance matrices with fewer observations than the dimension. *Journal of Multivariate Analysis* **101**, 1319–1329.

Stanford (2010) Stanford microarray database. http://smd.stanford.edu/.

Stein, C. M. (1975) Estimation of a covariance matrix. In *Rietz Lecture*. Atlanta, Georgia: 39th annual meeting IMS.

Stevens, J. R. and Doerge, R. W. (2005) Combining affymetrix microarray results. *BMC Bioinformatics* **6**, 57–76.

Stouffer, S. A., Suchman, E. A., DeVinney, L. C., Star, S. A. and Williams, R. M. J. (1949) *The American Soldier, Vol.1 : Adjustment during Army Life.* Princeton University Press.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S. *et al.* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* **102**, 15545–15550.

Sutton, A. J., Abrams, K., Jones, D. R., Sheldon, T. A. and Song, F. (2000) *Methods for Meta-Analysis in Medical Research.* New York: John Wiley.

Thalamuthu, A., Mukhopadhyay, I., Zheng, X. and Tseng, G. C. (2006) Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics* **22**, 2405–2412.

The Gene Ontology Consortium (2008) The Gene Ontology project in 2008. *Nucleic Acids Research* **36**, D440–D444.

Tibshirani, R. J., Walther, G. and Hastie, T. J. (2001) Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: series B* **63**, 411–423.

Tseng, G., Ghosh, D. and Feingold, E. (2012) Comprehensive literature review and statistical considerations for microarray meta-analysis. *Nucleic Acids Research* **40**, 3785–3799.

Tseng, G. C. and Wong, W. H. (2005) Tight clustering: A resampling-based approach for identifying stable and tight patterns in data. *Biometrics* **61**, 10–16.

Venkatraman, E. (2000) A permutation test to compare receiver operating characteristic curves. *Biometrics* **56**, 1134–1138.

Wadsworth, J. and Tawn, J. (2012) Likelihood-based procedures for threshold diagnostics and uncertainty in extreme value modelling. *Journal of the Royal Statistical Society: Series B* **74**, 543–567.

Wang, X., Kang, D. D., Shen, K., Song, C., Lu, S., Chang, L.-C., Liao, S. G., Huo, Z., Tang, S., Ding, Y. *et al.* (2012) An R package suite for microarray meta-analysis in quality control, differentially expressed gene analysis and pathway enrichment detection. *Bioinformatics* **28**, 2534–2536.

Warrenfeltz, S., Pavlik, S., Datta, S., Kraemer, E. T., Benigno, B. and McDonald, J. F. (2004) Gene expression profiling of epithelial ovarian tumours correlated with malignant potential. *Molecular Cancer* **3**, 27–43.

Welsh, J. B., Warrinkar, P. P., Sapinoso, L. M., Kern, S. G., Behling, C. A., Monk, B. J., Lockhart, D. J., Burger, R. A. and Hampton, G. M. (2001) Analysis of gene expression profiles in normal and neoplastic ovarian tissue samples identifies candidate molecular markers of epithelial ovarian cancer. *Proceedings of the National Academy of Sciences* **98**, 1176–1181.

Whitehead, J. (1997) *The Design and Analysis of Sequential Clinical Trials*. Second edition. Chichester: John Wiley.

Wirapati, P., Sotiriou, C., Kunkel, S., Farmer, P., Pradervand, S., Haibe-Kains, B., Desmedt, C., Ignatiadis, M., Sengstag, T., Schütz, F., Goldstein, D. R., Piccart, M. and Delorenzi, M. (2008) Meta-analysis of gene expression profiles in breast cancer: toward a unified understanding of breast cancer subtyping and prognosis signatures. *Breast Cancer Research* **10**, R65.

Yoshihara, K., Tajima, A., Komata, D., Yamamoto, T., Kodama, S., Fujiwara, H., Suzuki, M., Onishi, Y., Hatae, M., Sueyoshi, K., Fujiwara, H., Kudo, Y., Inoue, I. and Tanaka, K. (2009) Gene expression profiling of advanced-stage serous ovarian cancers distinguishes novel subclasses and implicates ZEB2 in tumor progression and prognosis. *Cancer Science* **100**, 1421–1428.

Zhang, X., Feng, J., Cheng, Y., Yao, Y., Ye, X., Fu, T. and Cheng, H. (2005) Characterization of differentially expressed genes in ovarian cancer by cDNA microarrays. *International Journal of Gynecological Cancer* **15**, 50–57.

Zhao, Y., Chen, M.-H., Pei, B., Rowe, D., Shin, D.-G., Xie, W., Yu, F. and Kuo, L. (2012) A Bayesian approach to pathway analysis by integrating gene–gene functional directions and microarray data. *Statistics in Biosciences* **4**, 105–131.

LEBOUCQ Alix
Chemin Vermont 4
1006 Lausanne
078/860 45 53

alix.leboucq@gmail.com
26 years old
Swiss/French
Single

Strengths Quick learner, adaptable, highly organized, effective

Education

2010 - 2014 (forecast) PhD in statistics, EPFL

2005 - 2010 Master in mathematics, statistics and finance, EPFL
- -Award for the best poster
- -Academic exchange of 6 months at Newcastle University (UK)

2005 High school diploma, passed with merit, French school Valmont, Lausanne

Experience

2010 - 2014 Collaborations with biologists during PhD thesis

2012 Referee for Biometrika

2008 - present Teaching assistant EPFL: supervision of semester projects, replace the professor for several lectures, preparation and correction of exercises and exams

Publications

2014 PhD Thesis in biostatistics and genomics, "*Meta-analysis of incomplete microarray studies*", under the supervision of Prof. A. C. Davison and Dr. D. R. Goldstein, key words: clustering, hierarchical Bayesian model, large covariance matrix estimation, MCMC, meta-analysis, microarray gene expression data

2014 A. Leboucq, A. C. Davison, D. R. Goldstein (2014), "Meta-Analysis of Incomplete Microarray Studies", *submitted to Biostatistics*

2013 M. Buehler, B. Tse, A. Leboucq, et al. (2013), "Meta-Analysis of Microarray Data Identifies GAS6 Expression as an Independent Predictor of Poor Survival in Ovarian Cancer," *BioMed Research International,* vol. 2013, Article ID 238284

2008 - 2010 Several semester projects in data analysis and biostatistics, key words: additive regression models, power of a test, survival analysis

Languages

French Native language

English Fluent (C2)

2009 Academic exchange of 6 months at Newcastle University, UK

German Intermediate level (B1)

Computing Latex, R, C, C++, Matlab, Microsoft Word, Microsoft Excel

Associations

2011 - 2014 PhD representative at the CUSO (Conférence Universitaire de Suisse Occidentale) organisation of two editions of the Young Researchers Conference

2012 - 2014 Assistant representative at the EDMA (doctoral school of mathematics) Organisation of the career day (2012), and other activities for PhD students

2007-2009 President and vice-president of the association Coaching, EPFL, organisation of the welcome day for all first year EPFL for over than 800 participants.

Hobbies Ballet, ski, snowshoes, cycling