université
PARIS-SACLAY

Integration of complex omics data through Multiscale Gaussian Graphical Models

Intégration de données omiques complexes par inférence de modèle graphique gaussien multi-échelles

Thèse de doctorat de l'université Paris-Saclay

École Doctorale n°574 : Ecole Doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat: Mathématiques Appliquées
Graduate School : Mathématiques, Référent : Université d'Évry Val d'Essonne

Thèse préparée dans le Laboratoire de Mathématiques et Modélisation d'Evry (Université Paris-Saclay, CNRS, Univ Evry), sous la direction de Christophe AMBROISE, professeur, le co-encadrement de Geneviève ROBIN, chargée de recherche

Thèse soutenue à Paris-Saclay, le 08 Septembre 2023, par

Edmond SANOU

Composition du jury
Membres du jury avec voix délibérative

Edouard Duchesnay Directeur de recherche, CEA, Université Paris Saclay	Président
Stéphane Canu Professeur, INSA Rouen, Normandie Université	Rapporteur & Examinateur
Catherine Matias Directrice de recherche, CNRS, Université Paris Cité	Rapporteur & Examinatrice
Mélina Gallopin Maîtresse de conférence, Université Paris Saclay	Examinatrice
Vincent Segura Chargé de recherche, INRAE, Université de Montpellier	Examinateur

THESE DE DOCTORAT

NNT : 2023UPASM020

université PARIS-SACLAY | ÉCOLE DOCTORALE de mathématiques Hadamard (EDMH)

Titre: Intégration de données omiques complexes par inférence de modèle graphique gaussien multi-échelles

Mots clés: Sélection de voisinage – Classification hiérarchique convexe – Modèles graphiques gaussiens – Données omiques – Optimisation non lisse

Résumé: Cette thèse se concentre sur l'inférence de modèles graphiques gaussiens multi-échelles appliqués à des données omiques. Les nombreuses méthodes statistiques existantes pour l'inférence de réseaux supposent généralement que le réseau est parcimonieux (peu d'interactions réelles parmi les interactions possibles) et font parfois l'hypothèse de l'existence d'une structure sous-jacente, qu'elle soit connue ou non. Ces a priori permettent d'obtenir un résumé synthétique des interactions présentes entre les variables d'un ensemble de données.

Dans un premier temps, nous avons développé une nouvelle approche d'inférence de graphes permettant d'estimer des graphes à plusieurs niveaux de granularité tout en recouvrant une structure de classification hiérarchique sur les variables. Pour cela, nous nous sommes basés sur les techniques de sélection de voisinage et de classification hiérarchique convexe. La fonction de pseudo-vraisemblance dérivée a été optimisée grâce à une méthode de continuation utilisant le lissage de Nesterov.

Dans un second temps, nous avons effectué des analyses de données omiques provenant de populations naturelles de peupliers. Ces analyses ont consisté à étudier conjointement des données omiques de différentes natures afin de mettre en lumière en particulier les mécanismes de régulation entre données épigénétiques et données génétiques. Nous avons également pris en compte le problème de la nature hétérogène des sources de données grâce à des transformations de variables permettant de revenir au cadre gaussien.

Title: Integration of complex omics data through Multiscale Gaussian Graphical Models

Keywords: Neighborhood selection – Convex hierarchical clustering – Gaussian graphical models – Omic data - Nonsmooth optimization

Abstract: This thesis addresses the inference of multiscale Gaussian graphical models with applications to omic data. Most existing statistical methods for inferring networks assume sparsity (few fundamental interactions among possible interactions) and sometimes assume a known or unknown underlying structure. These priors enable summarizing the interactions among variables in a data set.

To estimate graphs at several levels of granularity and uncover a hierarchical clustering structure on variables, we developed a novel graphical inference approach. This approach relies on neighborhood selection and convex hierarchical clustering techniques. The resulting pseudo-likelihood function is optimized via a continuation method using Nesterov smoothing.

We also analyzed omics data from natural populations of poplars, jointly studying different types of omics data to highlight the mechanisms of regulation between epigenetic and genetic data. To address the heterogeneity of data sources, we transformed variables to return to the Gaussian framework.


université
evry
val-d'essonne
université
PARIS-SACLAY

LaMME

EPITREE

INRAE

cnrs

Acknowledgments

Mes premiers remerciements vont à mes directeurs de thèse Christophe Ambroise et Geneviève Robin. Merci à vous de m'avoir guidé, soutenu et épaulé avec votre culture et rigueur scientifiques, votre compréhension et votre dévouement. C'était un privilège de travailler avec vous, j'ai beaucoup appris.

Merci à Stéphane Canu et Catherine Matias d'avoir accepté de rapporter la thèse ainsi qu'à Edouard Duchesnay, Mélina Gallopin, Vincent Segura d'avoir accepté de faire partie du jury.

Je tiens à remercier le laboratoire de Mathématique qui m'a accueilli pendant toute la durée de la thèse. Merci à Ludivine, Odélia, Salim, Ayoub, Perrine, Kyllian, Claire, Arnaud, Liudmila, Elisabetta, Antoine, Kamari, Chiara et tous les autres doctorants du laboratoire. Merci à Valérie, Dominique, Eugénie et Maurice pour leur aide. Merci à Vincent R, Guillem, Cyril, Marie S, Franck, Margot et tous les autres membres permanenents du laboratoire.

Je tiens à remercier Stéphane Maury et tous les autres partenaires du projet EPITREE qui a co-financé la thèse. Merci à Mamadou, Jérome, Emile, Vincent S, Odile et tous les autres collègues du projet pour ces échanges autour des problématiques biologiques du projet.

Merci à la communauté Groupe Jeunes SFdS pour leur accueil et les activités scientifiques que nous avons pu organiser ensemble. Merci à Geneviève, Marie C, Arthur, Imke, Iqraa, Margaux et tous les autres membres de bureau que j'ai eu l'honneur de rencontrer ou de voir en visio.

Je tiens à remercier François Coquet, Guillaume Desachy, Boureima Sangaré, Serge Somda et les tous les autres enseignant.e.s que j'ai pu avoir au cours de ma scolarité qui m'ont transmis le goût de l'apprentissage et des statistiques.

Merci à Amine de m'avoir accompagné dans mon projet de thèse, pour les discussions enrichissantes pendant la thèse et tous les autres rencontres de l'IRSN. Merci notamment à Mariem, Mariam, Frédéric, Miray, Lydia, Mani, Clément, Ségolène, Anaig et Fazia.

Merci à Elisabeth, Guillemette, Théophile, Camélia, Svetlana, Jean Paul, Edith et Max d'avoir contribué à rendre ces années de thèse agréable.

Enfin, merci à ma famille et à mes amis pour leur soutien et encouragement.

Contents

1 - Introduction

Inferring networks from biological data is important to gain a more complete understanding of the biological mechanisms underlying a particular phenomenon. Networks are a natural way to integrate and describe how biological variables interact, allowing the identification of complex interactions. With the emergence of high-throughput sequencing techniques, it is possible to generate large amounts of omics data related to the genome, transcriptome, genetic variations and metabolome. The highly dimensional nature of these data is the main difficulty from a statistical and interpretation point of view. The objective of the research is to propose and study a network inference method that takes into account a group structure or hierarchy between variables. Identifying groups of densely connected nodes in the network may correspond to biological variables with related functions and offers the possibility of constructing multiscale structures to synthesize the information retrieved by the groups and improve interpretability. This clustering task can be considered before or in conjunction with the network inference task. Probabilistic graphical models represent a well-suited model for inferring relationships between variables. The research will focus on a general subclass of graphical models where the conditional distribution at the nodes is Gaussian and examine methods for estimating clusters with convex criteria.

The research is motivated by the EPITREE (Evolutionary and functional impact of epigenetic variations in forest trees) project, which seeks to understand how epigenetics, in this case DNA methylation, gene expression and allelic variation, influence mechanisms of adaptation and phenotypic plasticity in forest trees. Epigenetics is the study of heritable changes that affect gene expression without altering the DNA, while phenotypic plasticity is the ability of an individual genotype to express different values of a given phenotypic trait under different environmental conditions (Rey et al., 2016). Trees are remarkable organisms that are long-lived, have complex life cycles, and produce wood, while providing a wide range of ecosystem services. In recent decades, widespread forest dieback due to drought and heat stress has been observed worldwide (Anderegg et al., 2016). These events highlight the vulnerability of forest ecosystems to environmental change and the urgent need to understand how trees respond to environmental stress. Indeed, climate change is the factor that will have the greatest impact on biodiversity by 2100, after land use (Chapin Iii et al., 2000). It is known that forest trees have complex mechanisms that allow them to adapt to environmental stressors (Bruce et al., 2007). Understanding the molecular mechanisms underlying their adaptation is therefore essential for developing conservation and management strategies for forest ecosystems. The EPITREE project studies the molecular mechanisms underlying tree adaptation, focusing on two tree models, poplar and oak. These species were chosen for their genetic diversity and their potential to adapt to changing

environments.

The project consists of several work packages, which include screening of candidate regions for epigenomic analysis and genome sequencing using modern omics technologies. These technologies have generated data on methylated single polymorphisms, differentially methylated regions, gene expression and single nucleotide polymorphisms. The PhD research is primarily motivated by the fourth work package, which aims to perform an integrative analysis to model the multi-scale relationships between quantitative traits and their molecular determinants. Specifically, this work package aims to quantify the contribution of genetic and epigenetic diversity to phenotypic variation, to study the impact of oak and poplar evolution on epigenomic plasticity, and to determine whether differentially methylated regions at the gene level are conserved between the two species. In addition, this project aims to improve models for predicting quantitative trait variation by combining genetic and epigenetic information. Graphical models are a useful tool for inferring interactions between genetic information and methylation patterns, and may provide answers to some of the questions raised in the project.

Probabilistic graphical models (PGMs, Lauritzen (1996); Koller and Friedman (2009)) have become a popular tool for analyzing high-dimensional data and capturing the interactions between variables. They are widely used in various applications, such as genomics and image analysis, to reduce the number of parameters by selecting the most relevant interactions between variables. One class of PGMs that is particularly useful in Gaussian settings is the undirected Gaussian graphical models (GGMs). In high-dimensional statistics, Gaussian graphical models are often assumed to be sparse, meaning that only a small number of variables interact compared to the total number of possible interactions. This sparsity assumption offers both statistical and computational advantages by simplifying the dependence structure between variables (Dempster, 1972) and enabling the development of efficient algorithms. To support this approach, many researchers have developed methods to estimate sparse GGMs from data. These methods include neighborhood selection and penalized maximum likelihood estimation.

In undirected Gaussian graphical models, inferring the conditional independence graph (CIG) involves identifying the support of the precision matrix $\mathbf{\Omega}$ (the inverse of the variance-covariance matrix). To learn the CIG of GGMs, several ℓ_1-penalized methods have been proposed in the literature. One popular method is the neighborhood selection (MB, Meinshausen and Bühlmann (2006)) approach based on nodewise regression using the least absolute shrinkage and selection operator (LASSO). This method focuses on learning only the structure of the network. The MB method has spawned a long line of work in nodewise regression methods, including extensions with various forms of sparsity-inducing penalties such as the Dantzig selector (Yuan, 2010) and the Clime estimator (Cai et al., 2011). Another family of sparse CIG inference methods directly estimates $\mathbf{\Omega}$ via the minimization of the ℓ_1-penalized negative log-likelihood (Banerjee et al., 2008). This

method, called the graphical LASSO (GLASSO, Friedman et al. (2008)), benefits from many optimization algorithms (Yuan and Lin, 2007; Rothman et al., 2008a; Banerjee et al., 2008; Hsieh et al., 2014).

LASSO-type regularization methods are widely used for conditional independence graph estimation and have been shown to be robust to high-dimensional problems. However, they have limitations in the presence of strongly correlated variables, which are well-known and have been discussed in the literature (Bühlmann et al., 2013; Park et al., 2006). To overcome these limitations and improve the estimation procedure, previous works have attempted to integrate clustering structures among the variables. Several studies have proposed different methods for incorporating clustering structures, including Honorio et al. (2009), Ambroise et al. (2009), Mazumder and Hastie (2012a), Tan et al. (2015), Devijver and Gallopin (2018), and Yao and Allen (2019).

The methods discussed earlier utilize the group structure to simplify the graph inference problem and infer the conditional independence graph between single variables. However, the inference of the CIG between groups of variables or representative variables of the groups has received less attention. Although some works have addressed this problem, they have mostly focused on two-level estimations, i.e., at the level of single variables and provided known groups (see, e.g., Cheng et al. (2017)). The research problem addressed in this work aims to define an inference method that allows for more than two levels of granularity estimations with unknown groups. This problem is mainly motivated by applications in biological data analysis where data from multiple sources, typically multi-omic data, need to be analyzed. In such cases, it might be necessary to group variables sharing the same characteristics and simultaneously take that into account in the network inference procedure, using a unique cost function instead of alternating clustering and network inference tasks.

Our research aims to address the inference of hierarchical clustering structures that are more intuitive for interpretation and focus on learning the inferred network structure rather than estimating the coefficients of the precision matrix. Although the applications are mainly motivated by biological questions from the EPITREE project, answering these questions may require using other dedicated machine learning tools. From a mathematical background, our research is located at the crossroads of probabilistic graphical inference, clustering, and convex optimization. From a statistical biological background, we address various biological questions, primarily concerning the impact of epigenetics on poplar local adaptation, which requires tools such as differential gene analysis, enrichment analysis of gene sets, transformation of count data, and gene selection methods. Omics applications include applications on transcriptomic (gene expression), epigenetic (especially DNA methylation), genetic (especially SNPs), and an illustration on metagenomic data (microbial abundance).

The proposed methodology for graphical inference is called Multiscale Graphi-

cal LASSO (MGLASSO), which is a pseudo-likelihood-based method for estimating hierarchical clustering structures and graphical models that depict the conditional independence structure between clusters of variables at each level of the hierarchy. MGLASSO combines neighborhood selection with a fused-LASSO type penalty for clustering (Pelckmans et al., 2005; Hocking et al., 2011; Lindsten et al., 2011). While the use of fusion penalties in Gaussian graphical model inference has been widely studied, previous works have mainly focused on penalized likelihood and investigated fusion penalties for enforcing local constancy in the nodes of the inferred network (Honorio et al., 2009; Yao and Allen, 2019; Lin et al., 2020). In contrast, MGLASSO employs a pseudo-likelihood criterion that is more computationally efficient and establishes a link with multiscale graphical models. Although the criterion used is similar to that used in supervised convex clustering (Hallac et al., 2015; Chu et al., 2021), MGLASSO behaves more like a multitask learning problem (Chiquet et al., 2011) due to its coupling with Gaussian graphical inference. In biological applications, MGLASSO relies on various data transformations adapted to the nature of the data, including the center-log ratio (Aitchison, 1982) for compositional data.

MGLASSO, like Yao and Allen (2019), is a method that combines Gaussian graphical models and convex clustering. However, unlike their work, we focused on the neighborhood selection framework and proposed adding a sparsity-inducing penalty (LASSO) to produce sparser results. We have also made available a beta version R package on the CRAN that implements the approach (Sanou, 2022). The algorithm uses a basic path algorithm to highlight the estimated multiscale structures. Our approach can also be seen as an extension of the SpiecEasi (Kurtz et al., 2015) method to multiscale networks when applied to compositional data with the centered log ratio transformation. Our biological applications, within the framework of the EPITREE project, demonstrate that the DNA methylation epigenetic mark for poplars can be used as markers of the genetic structure of the studied populations (Sow et al., 2023).

The remaining of the manuscript is structured into three chapters. Chapter 2 provides a foundational understanding of graphical modeling, convex clustering, and convex optimization, which are necessary for comprehending the rest of the manuscript. In Chapter 3, the Multiscale Graphical Lasso (MGLASSO) is introduced as a novel approach to Gaussian graphical model inference that combines LASSO and fuse-group-LASSO penalties. Finally, in Chapter 4, the results of applying MGLASSO and other data analysis tools in the context of the EPITREE project are presented.

Publications

- E. Sanou, C. Ambroise, G. Robin, Inference of Multiscale Gaussian Graphical Model, Computo, 2023.

- M. Sow et al., Epigenetic Variation in Tree Evolution: a case study in black poplar (Populus nigra) Mamadou, bioRxiv 2023.07.16.549253.
- E. Sanou, mglasso: Multiscale Graphical Lasso, *CRAN package, 2022*. https://CRAN.R-project.org/package=mglasso

Notations

In this section we describe the notations used in the remaining of the manuscript.

Specific sets

$\mathcal{X}$	discrete or continuous space
$\mathbb{S}^p_{\succ 0}$	set of real symmetric $p \times p$ positive definite matrices

Vectors and matrices

$\mathbf{X} = (X^1, \ldots, X^p)$	p-dimensional random vector or data matrix if $\in \mathbb{R}^{n \times p}$
$\mathbf{X}^A$	subset of $\mathbf{X}$ with variables indices taken in A
$Val(\mathbf{X})$	set of values that $\mathbf{X}$ can take
$\mathbf{X}^k$	k-th column in $\mathbf{X}$
$\mathbf{X}_i$	i-th row in $\mathbf{X}$
$\mathbf{X}^{\setminus(i,j)}$	$\mathbf{X}$ deprived of columns i-th and j-th columns
$\boldsymbol{\beta}^i \in \mathbb{R}^{p-1}$	regression vector
β^i_k	multiple regression coefficient of i-th variable on k-th variable
$Vec(.)$	convert a matrix into a column vector.

Probability

$f(.)$	probability density function.
$\phi(.)$	potential functions.
$\perp$	independence.
$\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \vert \mathbf{Z}$	$\mathbf{X}$ is conditionally independent of $\mathbf{Y}$ given $\mathbf{Z}$.

Gaussian distribution

$\boldsymbol{\Sigma}$	covariance matrix
$\mathbf{S}$	empirical covariance matrix
$\boldsymbol{\Omega}$	$= \boldsymbol{\Sigma}^{-1}$, precision matrix

2 - Mathematical background

Contents

This chapter – which can be read independently – establishes the necessary mathematical background for the rest of the manuscript and reviews existing research of the field.

The manuscript focuses on a subclass of graphical models with no directional arrows between the edges, as described in Section 2.1. The foundations of the Gaussian graphical model, a particular case of the undirected graph model, are introduced. The model and the central notion of conditional independence are defined. Section 2.1.3 presents an overview of state-of-the-art inference approaches for Gaussian graphical models.

Section 2.2 presents convex clustering, a method used to group data points based on a convex criterion and will be used in conjunction with the Gaussian graphical model inference problem later in the manuscript. Some essential properties of the convex approach and the links between convex clustering and clustering approaches, such as hierarchical agglomerative clustering and the k-means method, are discussed.

In Section 2.3, non-smooth convex optimization techniques are reviewed, including subgradient methods, proximal methods such as the alternating direction multiplier method, and smoothing methods, with emphasis on the Nesterov smoothing technique.

2.1 Undirected Graphical Models

Probabilistic graphical models (Lauritzen, 1996; Koller and Friedman, 2009), often seen as a marriage between probability and graph theory (Barber, 2012), are widely used in high-dimensional data analysis to synthesize the interactions between variables. In many applications, such as statistical physics, genomics, image analysis, or social network analysis, graphical models can reduce the number of parameters by selecting the most relevant interactions through parsimony. A graph consists of a set of nodes and edges between the nodes. There are two prominent traditional families of graphical models depending on the nature of the edge that connects the nodes:

- *Markov random fields* (MRF), also called undirected graphical models: These models were first applied in 1902 by Gibbs (Bryan, 1902) to describe the behavior of a system of interacting particles. MRFs are based on undirected graphs with only undirected edges, hence useful to describe soft constraints between variables. Figure 2.1 illustrates a 3-states Potts model. This model (Potts, 1952) is a generalization of the Ising model which arose in statistical physics to model interactions between spins of atoms. In the K-states Potts model, each node takes values in a discrete space $= \{0, 1, \ldots, K-1\}$ where $K > 2$ is an integer. Nodes can only interact with their nearest neighbors, and the model encourages neighboring nodes on the square lattice to be in the same state.

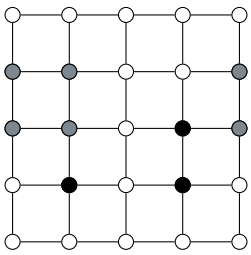


Figure 2.1: Potts model

- *Bayesian networks* also *called directed graphical models*: Directed graphical models' idea can be traced back to 1921. They were used by Wright (1934) to model genetic inheritance in human family trees. They are expressed via *directed acyclic graphs* (DAG) and help provide causal interpretations. DAGs are graphs with only directed edges, i.e., arrows and no directed cycle. Figure 2.2 shows a DAG that models the transmission of blood type from parents F, M to child C according to their respective genotypes. Human genetic material is encoded in DNA strings stored in 23 pairs of chromosomes. For each pair, one chromosome comes from the father and the other from the mother. Chromosomes can be divided into regions called loci,

which are responsible for observable traits such as eye color or blood type. The genotype corresponding to the blood type is the pair (X^F, X^M) which takes values in $\{A, B, O\}^2$. These types of models can be used for genetic counseling, for example, to predict the genotypes of expected children.

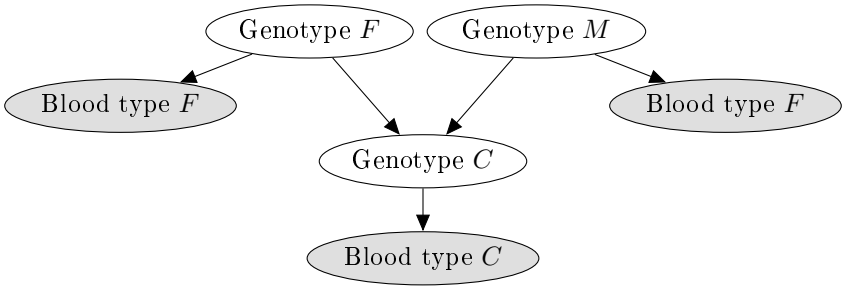


Figure 2.2: Directed acyclic graph

Another family of graphs, less used, derived from the previous ones is the mixed graphs (Sadeghi and Lauritzen, 2014). They contain more than one type of edge: directed, undirected and bidirected. Regardless of the type of graph, graphical models gravitate around the central concept of conditional independence. It occurs when two sets of variables are independent given an additional set.

The following sections focus on the undirected Gaussian graphical model and its inference problem. The three Markov properties are recalled as well as their links with conditional independence. Then, an overview of existing sparse inference approaches is given. Finally, inference methods that assume the existence of an underlying clustering structure on the variables are presented.

2.1.1 Markov properties and factorization

This section contains some theoretical results related to undirected graphical models.

2.1.1.1 Markov properties for Undirected Graphical Models

As mentioned, a graph is undirected if all edges have no directional arrows. We usually denote undirected graphs as G. Let $V = \{1, \dots, p\}$ and let $\mathcal{P}_2(V)$ be the subsets of V of size 2.

Definition 2.1. *For $E \subseteq \mathcal{P}_2(V)$, an undirected graph is a pair $G = (V, E)$ where V is the set of vertices and E the set of edges. The graph is complete if $E = \mathcal{P}_2(V)$. For a subset $\mathcal{C} \subset V$, $G_{\mathcal{C}} = (\mathcal{C}, E_{\mathcal{C}})$ is the subgraph of G induced by $\mathcal{C}$, with $E_{\mathcal{C}}$ the set of edges in $\mathcal{C} \times \mathcal{C}$. Whenever $G_{\mathcal{C}}$ is complete, $\mathcal{C}$ is said to be a clique of G.*

For example, the graph in Figure 2.3 consists of four nodes forming a complete graph with six undirected edges. The subgraphs induced by the sets of nodes

$\{X^1, X^2, X^3\}$, $\{X^2, X^3, X^4\}$ or $\{X^1, X^2, X^4\}$ are cliques of size 3 of the graph G.

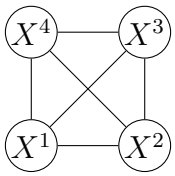


Figure 2.3: The set $\mathcal{C}_1 = \{X^1, X^2, X^3\}$ is an example of clique. The set $\mathcal{C}_2 = \{X^1, X^2, X^3, X^4\}$ is a complete graph.

Undirected Graphical Models can be translated to conditional independence constraints through a set of rules called *Markov properties*.

Definition 2.2 (Conditional independence)**.** *Let $\mathbf{X}^A, \mathbf{X}^B, \mathbf{X}^C$ be sets of random variables. $\mathbf{X}^A$ is said to be independent of $\mathbf{X}^B$ given $\mathbf{X}^C$ in a joint probability distribution P and writes $\mathbf{X}^A \perp\!\!\!\perp \mathbf{X}^B | \mathbf{X}^C$ iff.*

$$P(\mathbf{X}^A = \mathbf{x}^A, \mathbf{X}^B = \mathbf{x}^B | \mathbf{X}^C = \mathbf{x}^C) = P(\mathbf{X}^A = \mathbf{x}^A | \mathbf{X}^C = \mathbf{x}^C) P(\mathbf{X}^B = \mathbf{x}^B | \mathbf{X}^C = \mathbf{x}^C)$$

for all values $\mathbf{x} = (\mathbf{x}^A, \mathbf{x}^B, \mathbf{x}^C) \in Val(\mathbf{X})$, where $Val(\mathbf{X})$ denote the set of values that $\mathbf{X}$ can take.

The concept of conditional independence provides a precise and formal meaning to the idea of information irrelevance. The notation $\mathbf{X}^A \perp\!\!\!\perp \mathbf{X}^B | \mathbf{X}^C$ can be interpreted as follows: given the information contained in $\mathbf{X}^C$, knowledge of $\mathbf{X}^B$ is irrelevant for understanding $\mathbf{X}^A$.

Let us consider conditional independence in the context where $X = (X^v)$, where $v \in V$, is a p-dimensional random vector taking values in some space $\mathcal{X}^p = \otimes_{s=1}^p \mathcal{X}_s$, with a joint probability distribution P. Depending on the application, the space $\mathcal{X}^p$ can be continuous or discrete. When the collection of random variables is associated with an undirected graph G, three Markov properties can be established.

Let us first introduce the notion of separation, which is based on the idea that specific subsets of nodes in a graph can block the flow of information between other subsets.

Definition 2.3 (Separation)**.** *Let A, B, S be subsets of V. The set S is said to separate set A from set B if any path from any element of A to any element of B passes through S.*

Definition 2.4 (Global Markov property)**.** *Let $A, B, S \subset V$ be three disjoint subsets such that S separates A from B in the graph G. X satisfies the global Markov property with respect to the graph G iff.*

$$X^A \perp\!\!\!\perp X^B | X^S. \tag{2.1}$$

where $X^A = (X^k) \backslash k \in A$.

The global Markov property is closely related to the other two Markov properties: the local and pairwise Markov properties.

Denote $\mathrm{ne}(i) = \{u \in V : \{u, i\} \in E\}$ the neighbourhood or Markov blanket of node i and $\mathrm{cl}(i) = \mathrm{ne}(i) \cup \{i\}$, the closure of node i. The local Markov property states that each node is conditionally independent of its non-neighbors, given its neighbors.

Definition 2.5 (Local Markov property)**.** *X satisfies the local Markov Property with respect to the graph G iff.*

$$X^i \perp\!\!\!\perp X^{V \setminus \mathrm{cl}(i)} | X^{\mathrm{ne}(i)} \tag{2.2}$$

for any node $i \in V$.

In contrast, the pairwise Markov property states that two non-adjacent nodes are conditionally independent given their common neighbors.

Definition 2.6 (Pairwise Markov property)**.** *X satisfies the pairwise Markov property with respect to the graph G iff.*

$$X^i \perp\!\!\!\perp X^j | X^{V \setminus \{i,j\}} \tag{2.3}$$

whenever there is no edge between nodes i and j ie $(i, j) \notin E$.

The global Markov property (2.1) implies the local property (2.2), which in turn implies the pairwise property (2.3) (Lauritzen, 1996). When X has a continuous and strictly positive joint density with respect to the Lebesgue measure, the three properties are equivalent.

Thanks to the Markov properties, particularly the global Markov property, the graph in Figure 2.4 can be interpreted as follows: $X^3 \perp\!\!\!\perp X^4 | X^2$. This is because variable X^2 separates the variables X^1 and X^4.

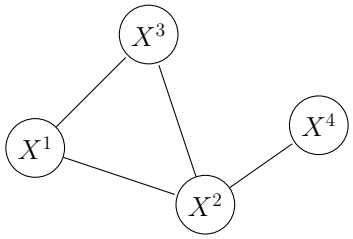


Figure 2.4: An undirected graph

2.1.1.2 Factorization

Another alternative to connect the graphical and probabilistic structure is through factorization, which is used as a basis for many inference algorithms.

Definition 2.7. *The density function f of probability distribution P with respect to product measure ν is said to factorize with respect to the graph G if it can be represented as follows*

$$f(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \phi_C(x_C), \tag{2.4}$$

where $\mathcal{C}$ denotes the set of maximal cliques, $\phi = (\phi_C, c \in \mathcal{C})$ is a collection of positive potential functions and Z is a normalization constant.

The factorization properties of graphical models make it possible to perform tractable computations with multivariate distributions. By representing a joint probability distribution as a product of factors, where each factor depends only on a subset of variables corresponding to a clique in the graph, computations can be performed locally on the cliques and then combined using factorization. This enables the development of efficient inference and learning algorithms for large and complex models.

It is also worth noting that factorization implies the global Markov property for any probability distribution P (Lauritzen, 1996). In the special case of strictly positive distributions, there is an equivalence between factorization and the three Markov properties, through the following theorem:

Theorem 2.1 (Hammersley-Clifford)**.** *A strictly positive and continuous probability distribution P with respect to a product measure ν factorizes with respect to a graph $G = (V, E)$ if and only if P satisfies the pairwise Markov property with respect to G.*

In other words, a probability distribution thus satisfies the Markov properties with respect to an undirected graph if and only if it can be expressed as a positive product of potential functions, where each potential function depends only on a subset of variables corresponding to a clique in the graph. An undirected graphical model can then be considererd as a pair (G, P) where G is a graph with undirected edges and P is a distribution that factorizes with respect to G.

The choice of the potential functions ϕ_C determines the structure of the dependencies between variables in the graphical model.

2.1.1.3 Pairwise Markov networks

In a pairwise Markov network, the joint distribution over a set of random variables is represented as a product of potential functions, each of which involves at most two variables. Log-linear models are a way of parameterizing the potential functions in a pairwise Markov network. Instead of directly specifying the potential functions, they model the logarithm of these functions as linear combinations of features. Each feature represents a particular configuration of the variables, and its weight determines the contribution of that configuration to the overall potential

function. The potential functions are defined as:

$$\phi_C(x_C) = \exp\left(-\epsilon_C(x_C)\right),$$

where $\epsilon_C(x_C) = -\log \phi_C(x_C)$ is called an energy function (Koller and Friedman, 2009). Generally, ϵ_C is chosen such as

$$\epsilon_C(x_C) = -w_C f_C(x_C)$$

where w_C is the clique weight and f_C a feature function over the clique.

Definition 2.8. *Pairwise Markov Networks can be defined as a subclass of undirected graphical models for which the factorizing density function can be written as*

$$f(x) = \frac{1}{Z} \exp\left(\sum_{i=1}^{p} w_i f_i(x_i) + \sum_{i<j} w_{ij} f_{ij}(x_i, x_j)\right) \tag{2.5}$$

where w_i, $w_{ij}, \forall i \neq j$ are weights, f_i, f_{ij} feature functions and Z is a normalization constant.

This family of Markov networks is well-known for its simplicity in parameterization. Pairwise Markov networks restrict the potential functions to be over single or pairs of variables, which significantly reduces the computational burden and improves the interpretability of the model during the learning step. Some examples of pairwise Markov networks for which conditional probability distributions of nodes belong to the exponential family include the Poisson model for count data, Ising or Potts models for categorical data, and the Gaussian model for continuous variables. Gaussian graphical models, which use Markov random fields, are a popular type of pairwise Markov network.

2.1.2 Gaussian graphical models

Gaussian graphical models (GGMs) or covariance selection models (Lauritzen, 1996) are a special class of undirected graphical models used in gaussian settings. Let $\mathbf{X} = (X^1, \ldots, X^p)^T \in \mathbb{R}^p$ be a p-dimensional Gaussian random vector, with zero mean and covariance matrix $\mathbf{\Sigma} \in \mathbb{S}^p_{\succ 0}$, where $\mathbb{S}^p_{\succ 0}$ denote the set of real symmetric $p \times p$ positive definite matrices. Some properties of graphical models specific to Gaussian distributions are given below.

Proposition 2.1. *The conditional independence structure of $\mathbf{X} \sim \mathcal{N}_p(0, \mathbf{\Sigma})$ is characterized by the graph G which is uniquely determined by the support of the precision or concentration matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$.*

Proof. The probability density function f of the multivariate normal distribution is defined as

$$f(\mathbf{X}) = \frac{(\det(\mathbf{\Omega}))^{1/2}}{(2\pi)^{p/2}} \exp\left(-\frac{1}{2}\mathbf{X}^T \mathbf{\Omega} \mathbf{X}\right) \tag{2.6}$$

The quadratic term can be rewritten as

$$\exp\left(-\frac{1}{2}X^T\mathbf{\Omega}X\right) = \exp\left(-\frac{1}{2}\sum_{(i,j)\in E}\Omega_{ij}X^iX^j\right) = \prod_{(i,j)\in E}\phi_{(i,j)\in E}(X^i, X^j)$$

with $\phi_{(i,j)\in E}(X^i, X^j) = \exp\left(-\frac{1}{2}\mathbf{\Omega}_{ij}X^iX^j\right)$. Hence, the distribution can be factorized in terms of potentials over cliques composed of at most two nodes. The Hammersley-Clifford theorem (2.1) thus ensures that, for any two vertices $i, j \notin V$, $\Omega_{ij} = 0$ if and only if the i-th and j-th variables are conditionally independent given the others i.e. $X^i \perp\!\!\!\perp X^j|\mathbf{X}^{\setminus(i,j)}$. □

Proposition 2.2. *The entries of the precision matrix are proportional to partial correlation coefficients.*

Indeed, the partial correlation between X^i and X^j given $\mathbf{X}\setminus\{X^i, X^j\}$ is equal to $\frac{-\Omega_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}}$. The reader may refer to Lauritzen (1996) for proof.

The following corollary can be derived from partial correlations coefficients being directly related to regression coefficients.

Proposition 2.3. *Given the regression problem $X^i = \mathbf{X}^{\setminus i}\beta^i + \epsilon^i$, where ϵ^i is the normal residuals vector, the regression coefficient is given by*

$$\beta^i_k = -\Omega_{ik}/\Omega_{ii}.$$

Proposition 6.1 suggests that the GGMs can be estimated by a series of regressions as outlined by Meinshausen and Bühlmann (2006). The next section will introduce some GGMs estimation methods.

2.1.3 Inference of Gaussian Graphical Models

Let P be an unknown probability distribution that factorizes over a graph G. Given a set of independent and identically distributed (iid) samples from P, the task of learning a Gaussian graphical model is to estimate the potential functions that best fit the distribution (Maathuis et al., 2018). In other words, the goal is to infer the edges of the graph and the parameters of the distribution. However, we also include inference methods in this review that consistently recover the graph structure without necessarily providing consistent parameter estimation. We differentiate learning approaches into three main classes: those without sparsity constraints, those which include sparsity-based approaches, and those with additional constraints on the node structure.

2.1.3.1 Maximum Likelihood Estimation

A natural way to estimate a zero-mean GGM is by using the *maximum likelihood estimator (MLE)*. The goal is to maximize the following strictly concave (Hastie et al., 2015) log-likelihood function given x a realization of X:

$$l(\mathbf{\Omega}) = \sum_{i=1}^{n}\log f(x_i|\mathbf{\Omega}) \propto \log\det(\mathbf{\Omega}) - \mathrm{tr}(\mathbf{\Omega}\mathbf{S}) \tag{2.7}$$

where $\boldsymbol{S} = \frac{1}{n} X^T X$ is the empirical covariance matrix and $\log\det$ the logarithm of the determinant of the matrix. The solution to this problem is unique and given by

$$\hat{\boldsymbol{\Omega}}_{MLE} = \boldsymbol{S}^{-1} \tag{2.8}$$

whenever $\boldsymbol{S}$ is non singular (Maathuis et al., 2018).

However, MLE is not computable when n is lower than p. Even if calculable, it often performs poorly (Mazumder and Hastie, 2012a). Indeed the estimator can result in complete graphs. In the context of high-dimensional statistics, GGMs are generally assumed to be sparse, meaning that a small number of variables interact compared to the total number of possible interactions. Dempster (1972) introduced the idea of estimating the network structure by setting some elements of $\boldsymbol{\Omega}$ to zero. This assumption has been shown to simplify the structure of dependencies between variables.

2.1.3.2 Edge recovery with sparsity constraints

Following the idea of Dempster (1972), several authors proposed approaches to recover the precision matrix support using sparsity.

Lasso penalized pseudo-likelihood estimation

Meinshausen and Bühlmann (2006) originally introduced a *nodewise regression* approach for *neighbourhood selection* based on the least absolute shrinkage and selection operator (Lasso, Tibshirani (1996)). They regress each variable $X^i, i \in V$ on the predictors $X^{\setminus\{i\}} := \{X_k | k \in V \setminus \{i\}\}$, taking advantage of the link between regression coefficients and precision matrix entries (see proposition 6.1).

Denote $\hat{\boldsymbol{\beta}}^i \in \mathbb{R}^{p-1}$ the regression vector when X^i is considered as response variable. $\forall i \in [1, p]$, the Lasso-based neighborhood regression is solved by optimizing the following problem:

$$\hat{\boldsymbol{\beta}}^i := \hat{\boldsymbol{\beta}}^i(\lambda) = \underset{\boldsymbol{\beta}^i \in \mathbb{R}^{p-1}}{\operatorname{argmin}} \frac{1}{n} \left\| \mathbf{X}^i - \mathbf{X}^{\setminus i} \boldsymbol{\beta}^i \right\|_2^2 + \lambda \left\| \boldsymbol{\beta}^i \right\|_1 . \tag{2.9}$$

where λ is a non negative regularization parameter and $\mathbf{X}^{\setminus i}$ denotes the matrix $\mathbf{X}$ deprived of column i. The neighbourhood is thus all the vertices corresponding to the non-zero regression coefficients ie $\widehat{\mathrm{ne}(i)} = \{u \in V \setminus \{i\} | \hat{\beta}_j^i \neq 0\}$. The different regressions are then combined applying an 'AND' or 'OR' rules to infer the conditional independence graph. 'AND' add an edge between X^i and X^j if $\beta_j^i \neq 0$ and $\beta_i^j \neq 0$. While in 'OR', an edge is added when $\beta_j^i \neq 0$ or $\beta_i^j \neq 0$.

Later, authors like Rocha et al. (2008), Ambroise et al. (2009) showed that the neighborhood selection could be seen as a pseudo-likelihood (Besag, 1975) approximation of the global likelihood.

Proposition 2.4. *The neighborhood selection with Lasso applied to each node of a gaussian graphical model is equivalent to maximizing a penalized pseudo-likelihood estimation.*

A proof is given in the appendix 7.1. Penalized pseudo-likelihood approximations are less time-consuming to optimize than penalized maximum likelihood methods that will be introduced later. They might not necessarily yield better parameter estimates, but in the context of GGMs edge recovery, they lead to consistent graph structure estimation under 'AND' or 'OR' rules (Meinshausen and Bühlmann, 2006).

Other penalized pseudo-likelihood methods include the work of Rocha et al. (2008) who merged all the linear regressions in Meinshausen and Bühlmann (2006) into a single one and added symmetry constraints. A symmetrized version have also been proposed by Friedman et al. (2010) and Peng et al. (2009).

Lasso penalized global likelihood estimation

Several authors have subsequently taken the idea of Lasso penalization of Meinshausen and Bühlmann (2006) and applied it to the overall likelihood of the model. In contrast to neighborhood selection approaches, which are mainly concerned with estimating the structure of the graph, likelihood-based approaches allow learning the graph structure and consistently estimating its parameters simultaneously. They seek to optimize the following convex function (Boyd et al., 2011)

$$\widehat{\boldsymbol{\Omega}} = \underset{\boldsymbol{\Omega} \in \mathbb{S}^p_{\succ 0}}{\arg\max} \ \log\det(\boldsymbol{\Omega}) - \operatorname{tr}(\boldsymbol{\Omega}\mathbf{S}) - \lambda \left\|\boldsymbol{\Omega}\right\|_1 \tag{2.10}$$

where $\mathbf{S}$ is the empirical covariance matrix, $\operatorname{tr}(A)$ the trace of a matrix A, λ a positive penalty parameter and $\left\|\boldsymbol{\Omega}\right\|_1 = \sum_{i,j} |\Omega|_{ij}$. Computation is one of the main challenges of problem (2.10). A variety of optimization algorithms have been proposed over the last decades. Some of them are highlighted here.

Yuan and Lin (2007) solved the problem (2.10) by applying the interior points methods. They exploited the link between (2.10) and the determinant maximization problem (Vandenberghe et al., 1998). However, interior point algorithms are not efficient for large-scale problems (Hastie et al., 2015). Banerjee et al. (2008) introduced a dual block coordinate descent approach interpretable as recursive ℓ_1-norm penalized regressions. The approach has been refined by Friedman et al. (2008) and called *graphical Lasso*. First order optimization methods include d'Aspremont et al. (2008); Lu (2010). Later other approaches emerged, which allowed improving the estimation process significantly. Mazumder and Hastie (2012b) and Witten et al. (2011) used the block diagonal screening rule to speed up the graphical Lasso algorithm. They showed that the sample covariance matrix's thresholding leads to the conditional independence graph connected components. Indeed, the inverse of a block diagonal matrix has the same blocks as the matrix. The inference problem can then be decomposed into low-dimensional subproblems within each connected component. Hsieh et al. (2013) developed a proximal Newton approach based on the work of QUIC algorithm (Hsieh et al., 2014) that can scale up to problems with a million variables. It is a block coordinate descent algorithm combined with METIS graph clustering. Their algorithm converges faster than the methods mentioned above.

The Graphical Lasso properties have been largely studied in Yuan and Lin (2007); Ravikumar et al. (2011); Rothman et al. (2008b). Hence the inference of GGMs with Lasso penalty has then been extended to different contexts to address specific applications/issues: for multiple sources data integration (Chiquet et al., 2019), gene regulation networks (Charbonnier et al., 2010), clustered samples (Chiquet et al., 2011; Danaher et al., 2014), multiple response variables (Chiquet et al., 2017b) or even accounting missing data (Robin et al., 2019).

Other sparsity penalties

Remark that in the literature of GGMs, sparsity penalties different from Lasso have also been explored. The ℓ_2 penalty has been studied by Kuismin et al. (2017). They called the approach ROPE, i.e., the Ridge type operator for precision matrix estimation. Using the ridge penalty yields a closed form solution to the inference problem contrarily to Graphical Lasso methods. However, the estimated precision matrix might not be sparse.

Nonconcave penalties as SCAD (Smoothly Clipped Absolute Deviation, Fan and Li (2001)) and adaptive Lasso (Zou, 2006) have been used in place of Lasso by Fan et al. (2009) to address the ℓ_1 penalty bias that can occur because of the linear increase of the penalty on regression coefficients. However, adaptive Lasso-like penalties are sensitive to the choice of the initial estimate of the precision matrix, and the nonconvexity of SCAD can make the computation difficult.

Other penalties that belongs to the successive regressions family include the Dantzig selector (Candes and Tao, 2007), the Sqrt-Lasso (Belloni et al., 2011), the scaled-Lasso (Sun and Zhang, 2012), the grouped-Lasso (Yuan and Lin, 2006) and the constrained ℓ_1 minimizaton introduced by Yuan (2010), Liu and Wang (2017), Sun and Zhang (2013), Friedman et al. (2010) and (Cai et al., 2011) respectively.

Kovács et al. (2021) proposed the graphical Elastic Net. The combination of ℓ_1 and ℓ_2 penalties in the inference problem leads to stable estimations when dealing with highly correlated variables. The approach includes in the estimation procedure the addition of a *target matrix* which is prior knowledge that is provided beforehand. Indeed, in strongly correlated variables presence, Lasso performance is known to be impaired (Bühlmann et al., 2013; Park et al., 2006; Vigneau, 2020; Grimonprez et al., 2018). In the following, we focus on approaches that tackle this Lasso bias.

2.1.3.3 Inference while taking into account underlying structure

In a penalized regression problem, the Lasso selects only one feature from a group of correlated features (Bühlmann et al., 2013). A variety of solutions have been proposed using different sparsity penalties. Among them, The Elastic-net (Zou and Hastie, 2005) applies a linear combination of Lasso and ridge penalties which encourages a grouping effect and can select groups of variables. OSCAR (Bondell and Reich, 2008) achieved that by mixing Lasso and ℓ_∞ penalizations. For its part, the clustered Lasso (She, 2008) defined a sort of generalized Fused Lasso (Tibshirani et al., 2005) criterion where there is no order on the variables.

Another family of approaches associates a preliminary clustering step with the model fitting problem. Variables from the same cluster are then averaged to form new representative variables. Among them, we have Bühlmann et al. (2013) for the Cluster Representative Lasso and Park et al. (2006).

In the graph inference problem, to overcome this, in addition to sparsity, several previous works attempt to estimate CIG by integrating clustering structures among variables for the sake of both statistical sanity and interpretability. A non-exhaustive list of works that integrate a clustering structure to speed up or improve the estimation procedure includes Honorio et al. (2009); Ambroise et al. (2009); Mazumder and Hastie (2012a); Tan et al. (2015); Yao and Allen (2019); Devijver and Gallopin (2018).

Duchi et al. (2012) proposed to penalize prior known groups of variables together via block ℓ_1 penalties. In Ambroise et al. (2009), the clustering structure is uncovered via a mixture model. Different penalty levels are then used for each group. The penalization is lowered when the variables belong to the same cluster and increased in the opposite case. Tan et al. (2015) derives from the block diagonal screening rule a link between graphical Lasso and single linkage hierarchical clustering. They then propose to adjust the penalization parameters to each cluster of connected components. A similar approach is presented in Devijver and Gallopin (2018) where clusters are selected in a non-asymptotic fashion. Other two-stage methods include Marlin and Murphy (2009).

Honorio et al. (2009) infer GGMs graph with a prior knowledge of local neighborhood called *local constancy*. They suppose that when X^i is neighbor to X^j, then X^k spatial neighbor of X^i is likely to be neighbor to X^j. The same goes for non-neighbors nodes. They optimize the following cost function:

$$\max_{\mathbf{\Omega}\in\mathbb{S}^p_{\succ 0}} \quad \log(\det(\mathbf{\Omega})) - \operatorname{tr}(\mathbf{S\Omega}) - \lambda_1||\mathbf{\Omega}||_1 - \lambda_2||\mathbf{D}\otimes\mathbf{\Omega}||_1, \tag{2.11}$$

where λ_1 and λ_2 non negative penalty weights, $\mathbf{D}\in\mathbb{R}^{q\times p}$ the matrix of prior neighbourhood relationships. The problem (2.11) is solved using a coordinate descent-like algorithm. They showed that their method outperforms the graphical Lasso of Friedman et al. (2008), covariance selection of Banerjee et al. (2008) and neighborhood selection of Meinshausen and Bühlmann (2006) in easy and hard settings. Ganguly and Polonik (2014) extends the work of Honorio et al. (2009) to the neighborhood selection case. Their *neighborhood-fused Lasso* consists in estimating the regression coefficients through

$$\hat{\beta}_i(\lambda_1,\lambda_2) = \arg\min_{\beta_i} \frac{1}{n}||X^i - \mathbf{X}^{-i}\beta^i||^2 + \lambda_1||\beta_i|| + \lambda_2||D^i\beta^i||_1 \tag{2.12}$$

In a recent work, Lin et al. (2020) proposed to simultaneously estimate a precision matrix and uncover a clustering structure for variables. They maximize

the following function:

$$\max_{\mathbf{\Omega}\in\mathbb{S}^p_{\succ 0}} \left\{ \log\det(\mathbf{\Omega}) - \mathrm{tr}(\mathbf{S\Omega}) - \lambda_1 \sum_{i<j} |\mathbf{\Omega}_{ij}| - \lambda_2 \sum_{i<j}\sum_{s<t} |\mathbf{\Omega}_{ij} - \mathbf{\Omega}_{st}| \right\} \tag{2.13}$$

The term $\sum_{s<t} |\mathbf{\Omega}_{ij} - \mathbf{\Omega}_{st}|$ do the pairwise differences between all the concentration coefficients.

In the next section, we introduce convex clustering. This is a tool that can be coupled with the Gaussian graphical model inference problem in order also to recover clustering and graph structures simultaneously.

2.2 Convex Clustering

As mentioned earlier, LASSO-type regularization methods commonly used for estimating conditional independence graphs are known to struggle in cases with highly correlated variables (Bühlmann et al., 2013; Park et al., 2006). To address this limitation and improve the estimation process, some researchers have proposed incorporating clustering structures into the analysis. Various methods have been proposed for this purpose, including those described by Honorio et al. (2009), Ambroise et al. (2009), Mazumder and Hastie (2012a), Tan et al. (2015), Devijver and Gallopin (2018), and Yao and Allen (2019).

Clustering is an approach that aims to detect group patterns in data, and various algorithms exist for this purpose. A comprehensive review can be found in books such as Hartigan (1975). Here, we will focus on hierarchical agglomerative clustering (HAC), k-means, and their convex relaxation.

2.2.1 Hierarchical Agglomerative Clustering

Hierarchical Agglomerative (or bottom up) Clustering (Johnson, 1967) is a partitioning algorithm of a data set into successively big clusters. Clusters at each level of the hierarchy are obtained by the merging of lower levels clusters. The process starts with n observations or clusters and ends with one cluster. This can be achieved by defining a dissimilarity between clusters also known as linkage function.

Contrarily to k-means or k-medoids algorithms, there is no need to specify first the number of clusters. Indeed, the cluster structure is recovered at multiple levels of granularity.

Some common linkages functions are given below. Let A and B be two clusters. Denote $d(A, B)$ their dissimilarity and $d(x_i, x_j)$ the distance between observations $x_i \in A$ and $x_j \in B$.

In single linkage,

$$d(A, B) = \min\left\{d(x_i, x_j)\right\}, \quad \forall x_i \in A, x_j \in B \tag{2.14}$$

In complete linkage,

$$d(A,B) = \max\{d(x_i,x_j)\}, \quad \forall x_i \in A, x_j \in B \tag{2.15}$$

In average linkage,

$$d(A,B) = \frac{1}{n_A n_B} \sum_{i=1}^{n_A} \sum_{i=1}^{n_B} d(x_i,x_j) \tag{2.16}$$

where n_A, n_B are the number of observations in clusters A and B respectively.

The Ward's distance is given by

$$d(A,B) = \frac{n_A n_B}{n_A + n_B} d(g_A, g_B)^2 \tag{2.17}$$

where g_A and g_B are the barycenters of clusters A and B respectively.

In order to reduce the computational cost of the HAC algorithm, some dissimilarity update formulas between clustering tree levels have been proposed. Among them, the Lance-Williams formula (Lance and Williams, 1967) which allows to compute distance between a newly formed cluster and the other observations as a function of previous level distances.

The result of HAC can be represented in the form of a tree called dendrogram whose leaves are observations and nodes intermediate clusters.

HAC is an attractive algorithm in term of ease of interpretation of the hierarchical clustering tree. However, the clustering errors made at lower levels remain, as one's go up in the hierarchy. Moreover, the algorithm is highly sensitive to perturbations in the input dataset (Chi and Steinerberger, 2019) and is defined in an iterative way without cost function.

2.2.2 K-means clusetring

Let $X = \{x_1, \ldots, x_n\}$ be a dataset of n observations of a random variable x. In k-means (Lloyd, 1982; MacQueen, 1967), the objective is to find the assignment of the unlabeled n observations to the desired number of clusters K, and recover a set of centroids vectors $\{c_k, k = 1, \ldots, K\}$. Let r_{ik} be the binary variable which indicates if point x_i belongs to cluster k. We seek to minimize the following cost function known as the *inertia*, with respect to $r = \{r_{ik}\}$ and $c = \{c_k\}$:

$$J(r,c) = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|x_i - c_k\|_2^2 . \tag{2.18}$$

This criterion ensures that each data point x_i is assigned to its closest centroid c_k. Remark that dissimilarity measures different from the Euclidean distance can be used.

The problem (2.18) can be solved using alternating minimization. First, for each centroid, the set of closest observations is determined. Secondly, the centroid

is updated by the mean of assigned observations as follows

$$c_k = \frac{1}{n_k} \sum_{i \in \mathcal{A}_k} x_i,$$

where $n_k = \sum_{j=1}^{n} r_{jk}$ and $\mathcal{A}_k$ is the set of observations indices composing cluster k. The operation is repeated until the algorithm converges. Note that the previous algorithm is closely related to the Expectation-Maximization algorithm (Dempster et al., 1977) for Gaussian Mixture Models (GMMs). As a reminder, a GMM assign observations to overlapping clusters while defining a probabilistic model and has the following form:

$$p(x|\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K) = \sum_{k=1}^{K} \pi_k p_k(x) \tag{2.19}$$

where the k-th mixture component p_k is the normal distribution with mean μ_k and covariance matrix Σ_k, π_k are weights with $\sum_k \pi_k = 1$. K-means can be seen as a special case of GMM (Murphy, 2022) in which $\Sigma_k = I$ the identity matrix and $\pi_k = 1/K$.

When c_k is defined as the observation in X which average dissimilarity to the cluster's observations is minimal, it's called a medoid. A partitioning around medoids algorithm (Rdusseeun and Kaufman, 1987) also known as k-medoids can thus be derived and is more robust to outliers.

Despite being a popular clustering algorithm, K-means algorithm does not necessarily provide global optima results and is sensible to the algorithm initialization values. Moreover, it might be subject to instabilities due to its non-convex objective functions.

Some theoretical results of the method have been surveyed in Steinley (2006).

2.2.3 Convex relaxation of k-means and HAC

The convex relaxation of k-means and HAC (Pelckmans et al., 2005; Hocking et al., 2011; Lindsten et al., 2011) also known as sum of norms clustering can be formulated as follows. Given $X = \{x_1, \ldots, x_n\} \in \mathbb{R}^{n \times p}$, we look to minimize with respect to the centroids matrix $\alpha \in \mathbb{R}^{n \times p}$ the criterion

$$\frac{1}{2} \sum_{i=1}^{n} \|x_i - \alpha_i\|_2^2 + \lambda \sum_{i<j} w_{ij} \|\alpha_i - \alpha_j\|_q \tag{2.20}$$

where λ is a sparsity penalization parameter, $\{w_{ij}\}$ are symmetric positive weights, $\alpha_i \in \mathbb{R}^p$ is the centroid to which observation x_i is assigned to, and $\|.\|_q$ is the ℓ_q-norm on $\mathbb{R}^p$ with $q \geq 1$. The reader is refered to Lindsten et al. (2011) for the proof of the link between the formulation (6.2) and k-means. The relation between HAC and (6.2) is proved in Hocking et al. (2011). Indeed clustering methods like k-means and HAC as outlined by Lindsten et al. (2011); Chi and Lange (2015);

Radchenko and Mukherjee (2017) and Hastie et al. (2009) have a greedy strategy. A simple approach like convex clustering solves the clustering problem for a grid of penalization parameters λ independently and thus provide a global optimum. The optimization problem being convex, the final solution is independent from the initialization.

Unlike sparsity in GGMs used for features selection, in convex clustering sparsity allows to determine a clustering structure. The data attachment term (first term) allows centroids to be kept close to the observations that make up their cluster. The penalization term $\|\alpha_i - \alpha_j\|_q$ also knowned as *fusion term* is a fused-group Lasso like penalty (Yuan and Lin, 2006; Tibshirani et al., 2005) when $q > 1$ and fused-Lasso term when $q = 1$. It encourages the centroids to be sparse in their differences. The common used norms are the ℓ_1 and the ℓ_2 norms. When $q = 1$, the regularization term tends to produce sparse entries in the vectors α_i. In the case $q = 2$, sparsity is observed in the whole vector instead of its entries.

The penalty λ controls the tradeoff between the model fit and the number of clusters. Two observations x_i and x_j belong to the same cluster when their estimated centroids are identical ie $\hat{\alpha}_i = \hat{\alpha}_j$. Usually, strict equality is not required for the assignement to clusters. A fusion threshold can be defined instead. When λ increases, the fusion strength increases too and centroids tend to fuse together. For a value of λ large enough, all the clusters merge into a single one.

Now let's consider the regularization path of solutions also so called *clusterpath* in Hocking et al. (2011) obtained while convex clustering (6.2). Like for HAC, a dendrogram can be recovered under some conditions. Recovering tree structures is closely related to the type of norm and weights used. Some theoretical properties and algorithms derived for hierarchical convex clustering problems will be addressed in the following sections.

2.2.4 Convex hierarchical clustering

While solving problem (6.2) for a grid of λ values, it may happen that observations that were once in the same clusters for a given grid value, split for others. Some authors have studied conditions under which a tree structure without splits is obtained. These conditions are highly dependent on the weights $\{w_{ij}\}$.

When weights are identity ie $w_{ij} = 1$, Hocking et al. (2011) proved that a tree structure is recovered in the ℓ_1-norm space . Chiquet et al. (2017a) extended the conclusions of Hocking et al. (2011). They showed no split occurs for arbitrary norms ℓ_q, $q = \{1, \ldots, \infty\}$.

Theorem 2.2. *(Chiquet et al., 2017a) The solutions path $S(\lambda, \mathbf{w})$ of the problem (6.2) contains no splits when $q \in \{1, \ldots, \infty\}$ and $w_{ij} = 1$.*

Theoritical guarantees on cluster recovery with identity weigths have been derived by Panahi et al. (2017). Tan and Witten (2015) studied some statistical properties and thus proved a link between single linkage clustering and convex

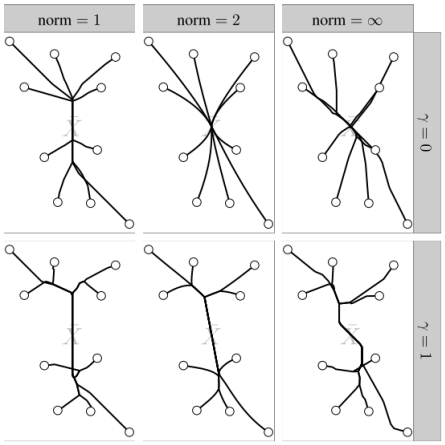


Figure 2.5: Effect of weights w_{ij} choice on the clustering path of 10 data points using distance decreasing weights of the form $\exp(-\gamma \|x_i - x_j\|_2^2)$ (Hocking et al., 2011). Solutions paths while varying λ the fusion penalty, are observed for ℓ_1, ℓ_2 and ℓ_∞ for two different weights choice. For norms different from ℓ_1, fusions are abrupt when $\gamma = 0$ i.e. $w_{ij} = 1$ (first-row graphs). Adding weights improves the tree structure (second-row graphs).

clustering. They also derived upperbounds of penalty parameter value λ for which all the clusters merge.

The use of distance decreasing weights have also been treated in Chiquet et al. (2017a); Chi and Steinerberger (2019). Figure 2.5 from Hocking et al. (2011) illustrates the effect of a distance decreasing weight on the solutions path of a 10 observations convex clustering problem.

Theorem 2.3. *(Chiquet et al., 2017a) The solutions path $S(\lambda, \mathbf{w})$ of the problem (6.2) contains no splits when $q = 1$ and $w_{ij} = f(|x_i - x_j|)$ with f, a decreasing positive function.*

Sun et al. (2021) derived conditions for perfect recovery in the general weighted convex clustering model.

A variety of algorithms can be used to solve the convex clustering problem. Hocking et al. (2011) proposed a homotopy algorithm to recover the solutions path of problem (6.2) for ℓ_1-norm with identity weights. This is inspired from the general fused Lasso path algorithm proposed by Hoefling (2010). A more general subgradient descent algorithm has been proposed by the same authors for ℓ_1, ℓ_2 norms spaces with arbitrary weights. A Franck-Wolf algorithm is used for ℓ_∞-norm space. Chi and Lange (2015) used two methods which are Alternating direction method of multipliers (ADMM) and Alternating minimization algorithm (AMA). In a recent work, Sun et al. (2021) used a semismooth Newton based augmented Lagrangian method. This approach performs better than the previously mentioned methods.

Convex clustering can thus allow, to achieve clustering and estimation simultaneously. This can motivate the association with graphical modelling approaches to recover simultaneously networks and clustering partition.

We thus made a review of the litterature on inference of GGM with structure constraints. In the following of the manuscript, we contribute to bring some light on the field by connecting graphs inference with convex clustering.

2.3 Non-smooth convex optimization

Convex clustering and Gaussian graphical model inference, especially neighborhood selection, rely on optimizing a convex function that involves the sum of a smooth term (e.g., squared error loss) and non-smooth penalty functions (e.g., LASSO, group-fused LASSO, or Elastic-net). The problem of optimizing non-smooth cost functions with structured parsimony has attracted some attention over the last few years. We refer to Bach et al. (2011) for an exhaustive survey on the topic.

Define the following optimization problem

$$\underset{\boldsymbol{x} \in \mathcal{X}}{\text{minimize}} \quad \mathcal{J}(\boldsymbol{x}) \tag{2.21}$$

where $\mathcal{J} : \mathcal{X} \to \mathbb{R}$ is a cost function with $\mathcal{X}$ a generic finite-dimensional continuous parameter space. In non-smooth optimization, the cost function is not continuously differentiable everywhere. Indeed, there exist points where the gradient of $\mathcal{J}$ is not well defined.

Definition 2.9. *The gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $\boldsymbol{x}$ is the vector of its partial derivatives:*

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)^\top. \tag{2.22}$$

We focus on optimization problems where the non-smooth cost function can be partitioned as the sum of smooth terms $\mathcal{J}_s$ (i.e., differentiable) and non-smooth terms $\mathcal{J}_{ns}$:

$$\mathcal{J}(\boldsymbol{x}) = \mathcal{J}_s(\boldsymbol{x}) + \mathcal{J}_{ns}(\boldsymbol{x}). \tag{2.23}$$

Moreover, we require the function $\mathcal{J}$ to be convex.

Definition 2.10. *Let $\mathcal{C}$ be a convex set. For any $\boldsymbol{x}$, $\boldsymbol{x}' \in \mathcal{S}$, we have*

$$\gamma \boldsymbol{x} + (1 - \gamma)\boldsymbol{x}' \in \mathcal{C}, \quad \text{for all } \gamma \in [0, 1]. \tag{2.24}$$

Definition 2.11. *A function $f(x)$ is considered to be convex when it is defined on a convex set and if, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$, and for any $0 \leq \gamma \leq 1$, we have:*

$$f(\gamma \boldsymbol{x} + (1 - \gamma)\boldsymbol{y}) \leq \gamma f(\boldsymbol{x}) + (1 - \gamma) f(\boldsymbol{y}). \tag{2.25}$$

The function f is strictly convex when the inequality (2.25) is strict.

Definition 2.12. *The function f is strongly convex with parameter $c > 0$ if for all $\boldsymbol{x}, \boldsymbol{y}$ in $\mathbb{R}^n$, the following holds:*

$$(\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}))^\top (\boldsymbol{x} - \boldsymbol{y}) \geq c \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 . \tag{2.26}$$

Strong convexity implies strict convexity. In the next sections we review some optimization methods that can tackle the problem defined in equation (2.21).

In the following, we consider three class of approaches that can be used for unconstrained non-smooth optimization problems: subgradient methods, proximal algorithms and smoothing techniques.

2.3.1 Subgradient methods

Historically, the subgradient approaches were the first to be used for non-smooth optimization problems (Nesterov, 2005). The usual gradient based methods for smooth optimization become ineffective.

Definition 2.13. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. The subgradient of f at $\boldsymbol{x}$ is any vector $\boldsymbol{v}$ that satisfies the inequality $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + v^\top(\boldsymbol{y} - \boldsymbol{x})$ for all $\boldsymbol{y}$.*

The function f is said to be subdifferentiable at $\boldsymbol{x}$ if there exist at least one subgradient at $\boldsymbol{x}$. The set of the subgradients is denoted $\partial f(\boldsymbol{x})$. To minimize f, at iteration k, the subgradient method takes a step of size $\eta_k > 0$ in the opposite direction from $\boldsymbol{v}^{(k)}$:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \eta_k \boldsymbol{v}^{(k)}. \tag{2.27}$$

When f is differentiable, $\boldsymbol{v}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$.

Let $f^\star$ denote the optimal value of the minimization problem. Denote $\{\eta_k\}$ the sequence of step sizes. Several methods exist for its choice (Boyd et al., 2003). Some of them are given below with their convergence results.

- Constant step size: $\eta_k = \eta$ with $\eta > 0$. The algorithm is guaranteed to converge within some range of the optimal value. When f is differentiable, the algorithm converges to the optimal value.
- Constant step length: $\eta_k = d\big(\|v^{(k)}\|_2\big)^{-1}$ with $d > 0$. The algorithm is guaranteed to converge within a range of the optimal value.
- Square summable but not summable: $\sum_{k=1}^{\infty} \eta_k^2 < \infty$ and $\sum_{k=1}^{\infty} \eta_k = \infty$. The algorithm is guaranteed to converge to the optimal value.
- Nonsummable diminishing: $\lim_{k\to\infty} \eta_k = 0$ and $\sum_{k=1}^{\infty} \eta_k = \infty$. The algorithm is guaranteed to converge to the optimal value.

2.3.2 Proximal methods

The proximal algorithms are a class of algorithms generally used for non-smooth, constrained, or distributed problems (Parikh et al., 2014), typically when the objective function can be decomposed as a sum of a smooth and a non-smooth term. These approaches solve the convex optimization problem while using proximal operators to approximate the non-differentiable components of the objective function.

Definition 2.14. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a closed proper convex function. The proximal operator* prox *of f is defined as*

$$\text{prox}_{\lambda f}(\boldsymbol{\nu}) = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{1}{2\lambda} \|\boldsymbol{x} - \boldsymbol{\nu}\|_2^2 \tag{2.28}$$

The parameter $\lambda > 0$ controls to which extent the proximal operator maps a given point towards the minimum of f. Proximal operators can be evaluated for various functions for which an analytical solution exists to the convex optimization problem. When a closed-form solution is unavailable for the proximal operator, generic optimization algorithms like the subgradient or the gradient method can be used. In the following, we recall the proximity operators for some common norms functions.

Let f be the ℓ_2 norm in $\mathbb{R}^n$. Its proximal operator is given by the block-wise soft-thresholding:

$$\text{prox}_{\lambda f}(\boldsymbol{v}) = \left(1 - \frac{\lambda}{\|\boldsymbol{\nu}\|_2}\right)_+ \boldsymbol{\nu}.$$

The proximal operator of the ℓ_1 norm writes as follows for all $j = 1, \ldots, n$:

$$(\text{prox}_{\lambda f}(\boldsymbol{v}))_j = \left(1 - \frac{\lambda}{\|\nu_j\|_1}\right)_+ \nu_j.$$

This is also known as element-wise soft thresholding.

The sum of norms function $f = \sum_{g \in \mathcal{G}} \|\boldsymbol{x}_g\|_2$ where $\mathcal{G}$ is a partition of $\{1, \ldots, p\}$ admits the following proximal operator:

$$\text{prox}_{\lambda f}(\boldsymbol{v}) = \left(1 - \frac{\lambda}{\|\boldsymbol{\nu}_g\|_2}\right)_+ \boldsymbol{\nu}_g.$$

Some popular methods based on proximal operators include, among other approaches, the proximal gradient method, the fast iterative shrinkage-thresholding algorithm (FISTA, Beck and Teboulle (2009)), and the alternating direction method of multipliers (Boyd et al., 2011). In the following, we focus on the specific case of the alternating direction method of multipliers (ADMM).

2.3.2.1 The alternating direction method of multipliers

The ADMM is an approach dedicated to composite objective function optimization. It solves problems for which the objective function can be decomposed into multiple smaller subproblems, simpler to solve, and coupled through some constraint. Consider the problem

$$\begin{aligned} &\text{minimize} && f(\boldsymbol{x}) + g(\boldsymbol{y}) \\ &\text{subject to} && \boldsymbol{x} = \boldsymbol{y}. \end{aligned} \tag{2.29}$$

where $f, g : \mathbb{R}^n \to \mathbb{R}$ are non-smooth (and smooth) convex functions. The ADMM combines an alternate minimization of the augmented Lagrangian with respect to variables $\boldsymbol{x}$ and $\boldsymbol{y}$ with an update of the Lagrange multiplier.

The augmented Lagrangian is defined as the Lagrangian of problem (2.29) augmented with a penalty term for the constraint. It takes the form

$$L_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \boldsymbol{u}^\top (x - y) + \rho/2 \left\| x - y \right\|_2^2, \tag{2.30}$$

where $\boldsymbol{u}$ is the Lagrange multiplier, $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^n$ the primal variables.

The updates of the ADMM can be written as follows:

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}, \boldsymbol{y}^{(k)}, \boldsymbol{u}^{(k)}) \\ \boldsymbol{y}^{(k+1)} &= \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}^{(k+1)}, \boldsymbol{y}, \boldsymbol{u}^{(k)}) \\ \boldsymbol{u}^{(k+1)} &= \boldsymbol{y}^{(k)} + \rho(\boldsymbol{x}^{(k+1)} - \boldsymbol{y}^{(k+1)}). \end{aligned} \tag{2.31}$$

Parikh et al. (2014) show the close connexion between the formulation of the ADMM in terms of augmented Lagrangian and the proximal operators. By defining the scaled dual variable $\boldsymbol{z}^{(k)} = (1/\rho)\boldsymbol{u}^{(k)}$, and $\lambda = 1/\rho$ it can be shown that the updates (2.31) are equivalent to:

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \text{prox}_{\lambda f}(\boldsymbol{y}^{(k)} - \boldsymbol{z}^{(k)}) \\ \boldsymbol{y}^{(k+1)} &= \text{prox}_{\lambda g}(\boldsymbol{x}^{(k+1)} - \boldsymbol{z}^{(k)}) \\ \boldsymbol{z}^{(k+1)} &= \boldsymbol{z}^{(k)} + \boldsymbol{x}^{(k+1)} - \boldsymbol{y}^{(k+1)}. \end{aligned} \tag{2.32}$$

In practice, the algorithm can be stopped when the primal and dual residuals are below a fixed tolerance. Following Boyd et al. (2011), the primal residuals $p^{(k)}$ and dual residuals $d^{(k)}$ are defined as:

$$p^{(k+1)} = x^{(k+1)} + y^{(k+1)} \tag{2.33}$$

$$d^{(k+1)} = \rho(y^{(k+1)} - y^{(k)}) \tag{2.34}$$

Other stopping criteria based on combined residuals can also be used. The combined residual measures both the primal and dual residuals simultaneously and have been shown to lead to more robust solutions (see, e.g., Chan et al. (2016)).

Under mild conditions on functions f, g and the unaugmented Lagrangian (see Boyd et al. (2011)), the ADMM algorithm satisfies for each iteration, $\forall \rho > 0$:

- Residual convergence: $p^{(k)} \to 0$ as $k \to \infty$.
- Objective convergence: $f(\boldsymbol{x}^{(k)}) + g(\boldsymbol{y}^{(k)}) \to p^\star$ as $k \to \infty$ where $p^\star$ is the optimal value of the problem (2.29).
- Dual variable convergence: $\boldsymbol{u}^{(k)} \to \boldsymbol{u}^\star$ as $k \to \infty$ with $\boldsymbol{u}^\star$ the dual optimal point.

In some cases, the algorithm can be slow to converge with a fixed parameter ρ. Boyd et al. (2011) proposed the residual balancing technique, which consists of adjusting ρ to keep primal and dual residuals norms within a certain factor of one another. However, proceeding so does not ensure convergence. Other variations of the ADMM have been proposed for accelerated performances (see, e.g., Buccini et al. (2020)) for a general framework of ADMM acceleration with a guarantee of convergence. In the case of using the proximal version of ADMM, the algorithm speed also depends on how efficiently the proximity operators are solved.

2.3.3 Smoothing methods

Smoothing methods are a set of algorithms that transform non-smooth optimization problems into differentiable problems. They create a smooth approximation of the original problem, which can then be solved via traditional optimization methods.

Some popular smoothing techniques include the Moreau-Yosida regularization, Nesterov smoothing, and Tikhonov regularization. In the following, we will focus on Nesterov's technique.

2.3.3.1 Nesterov smoothing

Let f be a non-differentiable convex function $f : \mathbb{R}^n \to \mathbb{R}$ of the form:

$$f(\boldsymbol{x}) = \max_{\boldsymbol{u} \in U} \left\{ \boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{x} - \phi(\boldsymbol{u}) \right\} \tag{2.35}$$

with $\phi : U \to \mathbb{R}$ and U a convex, compact set. Let μ be a positive smoothing parameter, the following smooth approximation is proposed by Nesterov (2005):

$$f_\mu(x) = \max_{\boldsymbol{u} \in U} \left\{ \boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{x} - \phi(\boldsymbol{u}) - \mu d(\boldsymbol{u}) \right\} \tag{2.36}$$

where $\boldsymbol{A}$ is matrix in $\mathbb{R}^{m \times n}$ and $d(\boldsymbol{u})$ is a called a *prox function*. The Nesterov smoothing method consists in adding a regularization term to the objective function f, which is often chosen to be the quadratic function $d(\boldsymbol{u}) = \frac{1}{2} \|\boldsymbol{u}\|_2^2$. Denote $\boldsymbol{u}_\mu^\star(\boldsymbol{x})$ the optimal solution of problem (2.36), the solution is unique since $d(\boldsymbol{u})$ is strongly convex with convexity parameter $\sigma > 0$ (Nesterov, 2005).

Proposition 2.5. *The function $f_\mu(\boldsymbol{x})$ is well defined, continuously differentiable, and convex. The gradient*

$$\nabla f_\mu(\boldsymbol{x}) = \boldsymbol{A}^\top u_\mu^\star(\boldsymbol{x}) \tag{2.37}$$

is Lipschitz continuous with constant

$$L_\mu = \frac{1}{\mu}\sigma \left\|\boldsymbol{A}\right\|_2^2. \tag{2.38}$$

with $\left\|\boldsymbol{A}\right\|_2^2$ *the spectral norm of* $\boldsymbol{A}$.

Proposition 2.6. *Denote* $M = \max_{\boldsymbol{u}} d(\boldsymbol{u})$, *the following inequality holds for any* $\boldsymbol{x} \in \mathbb{R}^n$

$$f_\mu(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq f_\mu(\boldsymbol{x}) + \mu M. \tag{2.39}$$

2.3.3.2 CONESTA

The CONESTA algorithm is dedicated to a general class of composite functions formulated as the sum of a differentiable loss function and two regularization functions. The algorithm is based on a Nesterov smoothing approach and dynamically adapts the smoothing parameter according to a duality gap calculation.

Consider the problem

$$\text{minimize} \quad \mathcal{J}(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x}) + h(\boldsymbol{x}) \tag{2.40}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth convex function and $g, h : \mathbb{R}^n \to \mathbb{R}$ are non-smooth convex functions. An analytical expression of function g proximal operator is assumed to be available. The function h proximal operator is assumed to be expensive to compute or not known and has the form of an $l_{1,2}$ group-norm:

$$h(\boldsymbol{x}) = \sum_{\phi \in \Phi} \left\|\boldsymbol{D}_\Phi \boldsymbol{x}_\Phi\right\|_2$$

with Φ the set of possibly overlapping groups of indices, and $\boldsymbol{D}_\Phi$ a linear operator on the group. Denote $h_\mu(\boldsymbol{x})$ the smooth approximation of function h by the Nesterov approach and $\nabla h_\mu(\boldsymbol{x})$ its gradient. The new smoothed optimization function can be written as:

$$\mathcal{J}_\mu(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x}) + h_\mu(\boldsymbol{x}). \tag{2.41}$$

where $(f + h_\mu)(\boldsymbol{x})$ is the smooth part.

Assuming the smooth function f has the form $f(\boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\right\|_2^2$ where $\boldsymbol{A}^{\in}\mathbb{R}^{m \times n}$ is a matrix of features and $\boldsymbol{y}$ the response variable, it can be rewritten as a function of $\boldsymbol{A}\boldsymbol{x}$ with the constraint $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$. The Lagrangian function of (2.40) is thus:

$$\mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{\nu}) = \mathcal{J}_\mu(\boldsymbol{x}) + \nu^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}) \tag{2.42}$$

where $\boldsymbol{\nu} \in \mathbb{R}^m$ is the dual variable composed of the multipliers of the equality constraint. The Lagrangian dual function of problem (2.40) is obtained by minimizing the Lagrangian function with respect to $\boldsymbol{x}$:

$$\mathcal{H}_\mu(\boldsymbol{\nu}) = \inf_{\boldsymbol{x}} \mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{\nu}). \tag{2.43}$$

For $\boldsymbol{x}$ primal feasible i.e., equality constraint satisfied, and $\boldsymbol{\nu}$ dual feasible i.e. solution to the dual problem, $\delta_\mu(\boldsymbol{x}, \boldsymbol{\nu}) = \mathcal{J}_\mu(\boldsymbol{x}) - \mathcal{H}_\mu(\boldsymbol{\nu})$ is called the duality gap between $\boldsymbol{x}$ and $\boldsymbol{\nu}$.

Definition 2.15. *The Fenchel conjugate of a function $l : \mathbb{R}^n \to R$ is the function $l^\star : \mathbb{R}^n \to R$ defined by*

$$l^\star(\boldsymbol{\omega}) = \sup_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ \boldsymbol{\omega}^\top \boldsymbol{x} - l(\boldsymbol{x}) \right\}. \tag{2.44}$$

Exploiting the link between the Fenchel conjugate function and the Lagrangian dual function (Borwein and Lewis, 2006), the duality gap can be written as:

$$\delta_\mu(\boldsymbol{x}, \boldsymbol{\nu}) = \mathcal{J}_\mu(\boldsymbol{x}) + f^\star(\boldsymbol{\nu}) + g^\star(\boldsymbol{\nu}) + h_\mu^\star(\boldsymbol{\nu}). \tag{2.45}$$

where $f^\star$, $g^\star$ and $h^\star$ are the Fenchel conjugates of f, g and h respectively.

Moreover, the duality gap upperbounds the distance to the minimum. The following inequality holds:

$$\delta_\mu(\boldsymbol{x}, \boldsymbol{\nu}) \geq \mathcal{J}_\mu(\boldsymbol{x}) - \mathcal{J}_\mu(\boldsymbol{x}^\star) \geq 0. \tag{2.46}$$

A zero duality gap is thus equivalent to optimality.

In CONESTA, the duality gap is used as a stopping criterion as it measures the algorithm's progress. The duality gap is calculated at each iteration and compared to a threshold value ϵ. At convergence for a fixed smoothing parameter, the obtained approximation satisfies:

$$\delta_\mu(\boldsymbol{x}, \boldsymbol{\nu}) \geq \epsilon_\mu \tag{2.47}$$

The precision is updated dynamically throughout the algorithm and is chosen to be a linear function of the smoothing parameter. The optimization algorithm can be guided toward a better solution by dynamically updating the tolerance level. The smoothing parameter μ is selected to minimize the number of iterations needed to reach the desired precision.

Proposition 2.7. *For any tolerance level $\epsilon > 0$, the optimal smoothing parameter which minimizes the worst case bound on the number of iterations is given by*

$$\mu_{opt}(\epsilon) = \frac{-M \left\|\boldsymbol{D}\right\|_2^2 + \sqrt{(M \left\|\boldsymbol{D}\right\|_2^2)^2 + ML(\nabla(f)) \left\|\boldsymbol{D}\right\|_2^2)\epsilon}}{ML(\nabla(f))} \tag{2.48}$$

where M is the maximum of the Nesterov smoothing prox function, $L(\nabla(f))$ the Lipschitz constant of the gradient of f and $\boldsymbol{D}$ the vertical concatenation of the matrices $\boldsymbol{D}_\phi$.

The updates of the CONESTA algorithm write:

$$\begin{aligned}
\epsilon_{\mu}^{(k)} &= \epsilon^{(k)} - \mu^{(k)} M \\
\boldsymbol{x}^{(k+1)} &= \arg\min_{\boldsymbol{x}} \mathcal{J}_{\mu^{(k)}}(\boldsymbol{x}) \\
\epsilon^{(k)} &= \delta_{\mu^{(k)}}(\boldsymbol{x}^{(k+1)}, \boldsymbol{\nu}^{(k+1)}) + \mu^{(k)} M \\
\epsilon^{(k+1)} &= \tau \epsilon^{(k)} \\
\mu^{(k+1)} &= \mu_{opt}(\epsilon^{(k+1)})
\end{aligned} \tag{2.49}$$

where $\tau \in]0, 1[$ is the factor by which the distance to the minimum is geometrically decreased. The smoothed minimization problem can be solved using classical proximal algorithms.

3 - Multiscale Graphical LASSO

Contents

In this chapter, we introduce the Multiscale Graphical Least Absolute Shrinkage Operator (MGLASSO), a novel method to estimate simultaneously a hierarchical clustering structure, and graphical models depicting the conditional independence structure between variables at multiple levels of granularity. Some of the chapter's material has been published in Sanou et al..

We focus on the neighborhood selection framework and the convex clustering theory presented in the previous chapter to propose a convex optimization problem with a hybrid penalty term combining graphical LASSO and group-fused LASSO penalties. The method allows for highlighting common patterns in the data through clustering while computing interactions between those clusters. In the estimated graphs, the variables belonging to the same clusters are likely to share the same neighborhood.

Thanks to the continuation of Nesterov smoothing in a shrinkage-thresholding algorithm (Hadj-Selem et al., 2018), the method achieves faster convergence speed than other competing approaches such as alternating direction methods of multipliers (ADMM, Boyd et al. (2011)). A quick empirical convergence analysis is presented for that purpose. We then show how we implemented the approach in practice. The graphs inferred by the MGLASSO are characterized by their multi-scale structure, following a grid of fusion regularization parameters. The proposed model selection method based on the stability approach to regularization selection (StARS, Liu et al. (2010)) focuses on selecting the LASSO regularization parameter. We discuss some other selection methods that have been proposed for the LASSO model selection problem. The performances are evaluated in different simulation settings, from graph models to phylogenetic tree-based models. The application of MGLASSO on real data is available in the next chapter.

In Section 3.1 we present the basic framework of MGLASSO. In Section 3.2, we present how the structure of the graphical model is learned. Section 3.3 describes the model selection procedure. In Section 3.4, MGLASSO is compared to some network inference and clustering approaches on synthetic data.

3.1 Model presentation

3.1.1 Problem formulation

The Multiscale Graphical LASSO method aims at inferring a graphical Gaussian model while hierarchically grouping variables. It infers conditional independence between different groups of variables. The approach is based on neighborhood selection (Meinshausen and Bühlmann, 2006) and considers an additional fused-LASSO type penalty for clustering. In the spirit of hierarchical convex clustering, the hierarchical structure is recovered by spanning the regularization path.

Let $\mathbf{X} = (X^1, \dots, X^p)^\top$ be a p-dimensional Gaussian random vector, with mean vector $\boldsymbol{\mu} \in \mathbb{R}^p$ and positive definite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$. Let $G = (V, E)$ be a graph encoding the conditional independence structure of the normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $V = \{1, \dots p\}$ is the set of vertices and E the set of edges. The graph G is uniquely determined by the support of the precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ (Dempster, 1972). Specifically, for any two vertices $i \neq j \in V$, the edge (i, j) does not belong to the set E if and only if $\Omega_{ij} = 0$. The variables X^i and X^j are said to be independent conditionally to the remaining variables $X^{\setminus(i,j)}$. We note,

$$X^i \perp\!\!\!\perp X^j | X^{\setminus(i,j)} \Leftrightarrow \Omega_{ij} = 0.$$

Let $\boldsymbol{X} = (\boldsymbol{X}_1^\top, \dots, \boldsymbol{X}_n^\top)^\top$ be the $n \times p$-dimensional data matrix composed of n i.i.d samples of the Gaussian random vector $\mathbf{X}$. To perform graphical model inference, Meinshausen and Bühlmann (2006) consider p separate linear regressions

of the form:

$$\hat{\boldsymbol{\beta}}^i(\lambda) = \underset{\boldsymbol{\beta}^i \in \mathbb{R}^{p-1}}{\operatorname{argmin}} \frac{1}{n} \left\| \mathbf{X}^i - \mathbf{X}^{\setminus i} \boldsymbol{\beta}^i \right\|_2^2 + \lambda \left\| \boldsymbol{\beta}^i \right\|_1, \tag{3.1}$$

where λ is a non-negative regularization parameter, $\mathbf{X}^{\setminus i}$ denotes the matrix $\mathbf{X}$ deprived of column i, $\boldsymbol{\beta}^i = (\beta^i_j)_{j \in \{1,\dots,p\} \setminus i}$ is a vector of $p-1$ regression coefficients and $\|.\|_1$ is the ℓ_1-norm. These LASSO regularized problems estimate the neighborhoods, one variable at a time. The final edge set estimates $\hat{E}$ can be deduced from the union of the estimated neighborhoods using an AND or OR rule (Meinshausen and Bühlmann, 2006). The Meinshausen-Bûhlmann (MB) approach is based on the central relationship between simple linear regression and precision matrix coefficients. It can be shown that $\beta^i_j = -\frac{\Omega_{ij}}{\Omega_{ii}}$ (Lauritzen, 1996).

Consider the data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ without any underlying distribution and the clustering analysis of the p points in $\mathbb{R}^n$. The convex clustering problem (Hocking et al., 2011; Lindsten et al., 2011; Pelckmans et al., 2005) is the minimization of the quantity

$$\frac{1}{2} \sum_{i=1}^{p} \left\| \boldsymbol{X}^i - \boldsymbol{\alpha}^i \right\|_2^2 + \lambda \sum_{i<j} w_{ij} \left\| \boldsymbol{\alpha}^i - \boldsymbol{\alpha}^j \right\|_q \tag{3.2}$$

with respect to the matrix $\boldsymbol{\alpha} \in \mathbb{R}^{p \times n}$, where λ is a sparsity penalization parameter, $\{w_{ij}\}$ are symmetric positive weights, $\boldsymbol{\alpha}^i \in \mathbb{R}^n$ is the centroid to which $\boldsymbol{X}^i$ is assigned to, and $\|.\|_q$ is the ℓ_q-norm on $\mathbb{R}^p$ with $q \geq 1$. Points $\boldsymbol{X}^i$ and $\boldsymbol{X}^j$ are assigned to the same cluster if $\hat{\boldsymbol{\alpha}^i} \approx \hat{\boldsymbol{\alpha}^j}$. The regularization path of solutions to problem in (3.2) can be represented as a dendrogram. The path properties have been studied in Chi and Lange (2015) and Chiquet et al. (2017a), among others.

We propose merging the p independent LASSO regressions of the MB approach into a single optimization criterion where a convex clustering fusion penalty in ℓ_2 is applied on the regression vectors considered as cluster centers. Namely, the Multiscale Graphical LASSO (MGLASSO) pseudo-likelihood problem minimizes in a Gaussian framework the following quantity:

$$\mathcal{J}_{\lambda_1,\lambda_2}(\boldsymbol{\beta}; \mathbf{X}) = \frac{1}{2} \sum_{i=1}^{p} \left\| \mathbf{X}^i - \mathbf{X}^{\setminus i} \boldsymbol{\beta}^i \right\|_2^2 + \lambda_1 \sum_{i=1}^{p} \left\| \boldsymbol{\beta}^i \right\|_1 + \lambda_2 \sum_{i<j} \left\| \boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij} \boldsymbol{\beta}^j \right\|_2, \tag{3.3}$$

with respect to $\boldsymbol{\beta} := [\boldsymbol{\beta}^1, \dots, \boldsymbol{\beta}^p] \in \mathbb{R}^{(p-1) \times p}$, where $\mathbf{X}^i \in \mathbb{R}^n$ denotes the i-th column of $\mathbf{X}$, λ_1 and λ_2 are penalization parameters, $\boldsymbol{\tau}_{ij} \in \mathbb{R}^{(p-1) \times (p-1)}$ is a permutation matrix, which permutes the coefficients in the regression vector $\boldsymbol{\beta}^j$ such as

$$\left\| \boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij} \boldsymbol{\beta}^j \right\|_2 = \sqrt{\sum_{k \in \{1,\dots,p\} \setminus \{i,j\}} (\beta^i_k - \beta^j_k)^2 + (\beta^i_j - \beta^j_i)^2},$$

as illustrated in Figure 3.1. The coefficient β^i_k is to be read as the multiple regression coefficients of $\boldsymbol{X}^i$ on $\boldsymbol{X}^k$.

3.1.2 Grouping effect

The MGLASSO criterion can be seen as a multitask regression problem where the set of responses is identical to the set of predictors. The LASSO penalty term encourages sparsity in the estimated coefficients. The group-fused term encourages fusion in the regression coefficients $\boldsymbol{\beta}^i$ and $\boldsymbol{\beta}^j$. An ℓ_1-norm of the differences would affect each variable individually Degras (2021). Moreover, some simulations studies by Tan and Witten (2015) showed that the convex clustering in ℓ_2 outperforms the ℓ_1 case.

The clustering effect in MGLASSO occurs at two scales: first, by using the norm penalty ℓ_2 to select groups of correlated variables, and second by considering the parsimony of the differences of the regression vectors. The natural tendency of this type of penalty would be to produce groups of nearly equal regression vectors (Zeng et al., 2017). Using this form of penalization is helpful when no prior information is available on the groups. Intuitively, one can notice that in a framework where the data X has a block-diagonal correlation structure with similar correlation levels, the model excels when these correlation levels are high enough (Park et al., 2006).

Let us illustrate by an example the effect of the fusion term in the proposed approach. Two variables i and j are in the same group when $\|\boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j\|_2 \approx 0$. Considering a cluster $\mathcal{C}$ of q variables, it is straightforward to show that $\forall (i,j) \in \mathcal{C}^2$, we have $\hat{\beta}^i_j = \beta_{\mathcal{C}}$, where $\beta_{\mathcal{C}}$ is a scalar. Thus the algorithm is likely to produce precision matrices with blocks of constant entries for a given value of λ_2, each block corresponding to a cluster. In the same vein as Park et al. (2006), a cluster composed of variables that share the same coefficients can be summarized by a representative variable. A component-wise difference between two regression vectors without reordering the coefficients would not necesarily cluster variables which share the same neighborhood. The permutation $\boldsymbol{\tau}_{ij}$ reoders coefficients in such a way that differences are taken between symmetric coeffecients and those corresponding to the same set of predictors. The model is thus likely to cluster together variables that share the same neighboring structure and encourages symmetric graph structures.

$$(\boldsymbol{\beta}^i, \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j) = \begin{pmatrix} \beta^i_1 & \beta^i_2 & \dots & \beta^i_j & \dots\dots & \beta^i_k & \dots & \beta^i_p \\ & & & & & & & \\ \beta^j_1 & \beta^j_2 & \dots & \beta^j_k & \dots\dots & \beta^j_i & \dots & \beta^j_p \end{pmatrix} \tag{3.4}$$

Figure 3.1: Illustration of the permutation between regression coefficients in the MGLASSO model.

The greater the regularization weight λ_2 is, the larger groups become. This is the core principle of the convex relaxation of hierarchical clustering introduced by

Hocking et al. (2011). Hence, we can derive a hierarchical clustering structure by spanning the regularization path obtained by varying λ_2 while λ_1 is fixed.

In practice, when external information about the clustering structure is available, the problem can be generalized into:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{p} \frac{1}{2} \left\| \mathbf{X}^i - \mathbf{X}^{\setminus i} \boldsymbol{\beta}^i \right\|_2^2 + \lambda_1 \sum_{i=1}^{p} \left\| \boldsymbol{\beta}^i \right\|_1 + \lambda_2 \sum_{i<j} w_{ij} \left\| \boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij} \boldsymbol{\beta}^j \right\|_2 ,$$

where w_{ij} is a positive weight. In the remainder of the manuscript, we will assume that $w_{ij} = 1$.

3.1.3 Local constancy

We succintly show the link between the MGLasso model and the local constancy notion introduced by Honorio et al. (2009) and derived a sort of local constancy definition in the sense of the MGLasso. Honorio et al. (2009) introcuded a Gaussian graphical model in which the locality information i.e., known interactions in a dataset are taken in account as a prior for learning the graph structure G. Local constancy encourages the search of dependencies between clusters of variables, instead of variables taken individually. Honorio et al. (2009) enforced that by using the following penalty on the precision matrix in addition to the Lasso penalty:

$$P_{\lambda_2}(\boldsymbol{\Omega}) = \lambda_2 \left\| \boldsymbol{D} \oslash \boldsymbol{\Omega} \right\|_1 ,$$

where D is a $m \times p$ matrix, with m the number of local neigbors and $\oslash$ the diagonal excluded matrix product (Ganguly and Polonik, 2014). The k-th row $D_k, . = e_i - e_j$ where (X^i, X^j) are local neighbors and e_t a canonical basis vector in $\mathbb{R}^p$ with the t-th element set to 1.

If the node X^1 is independent (or dependent) of node X^2, a local neighbor $X^{1'}$ of X^1 is more likely to be independent or (dependent) of X^2. Denote G_{local} the domain or prior knowledge graph (Ganguly and Polonik, 2014) and E_{local} the associated set of edges. As a remark, the local edges don't necessarily belong to E, the set of edges of graph G. An illustration of the local constancy is given in Figure 3.2.

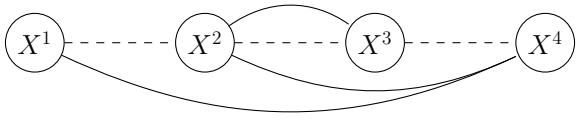


Figure 3.2: Illustration of locally constant interactions. The prior knowledge on the neighborhood structure is represented by dashed lines. X^1 and X^2 are local neighbors and mutually connected to X^4 in the true graph (with solid lines). The interaction between $\{X^1, X^2\}$ and X^4 is locally constant. The edge (X^2, X^3) is not locally constant as the local neighbor X^1 is not connected to X^3.

The difference matrix $\boldsymbol{D}$ according to the penalty of Honorio et al. (2009), corresponding to Figure 3.2 is

$$\boldsymbol{D} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

In the MGLasso model with unitary weights, i.e. all $w_{ij} = 1$, implicitly all nodes are supposed to be local neighbors. Indeed, the fusion penalty includes all the pairwise differences between the p regression vectors. The local graph G_{local} is hence the complete graph where all the nodes are connected. This assumption might be relevant when no locality information is available. For a model with 4 variables, the local graph is given in Figure 3.3.

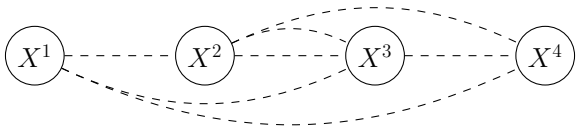


Figure 3.3: Illustration of locality graph in the MGLasso model with 4 variables. All the nodes are expected to be local neighbors. The pairwises differences includes all the existing pairs of variables.

The local constancy is enforced at the scale of the whole variable neighborhood instead of neighbors taken individually in Honorio et al. (2009), by using the group fused penalty term on the regression vectors,

$$P_{\lambda_2}(\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^p) = \lambda_2 \sum_{i<j} \left\| \boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij} \boldsymbol{\beta}^j \right\|_2 .$$

Let $ne(X^1)$ be the markov blanket or neighborhood of node X^1. A local neighbor $X^{1'}$ of X^1 is more likely to have the same markov blanket as node X^1 in the MGLasso.

One might notice that when local neighbors are neighbors in the estimated graph $\hat{G}$, the MGLasso model is likely to produce clustering structures where all the variables belonging to the same cluster are mutually connected in term of conditional dependence. The permutation $\boldsymbol{\tau}_{ij}$ encourages a symmetry structure.

3.2 Model learning

This section introduces a complete numerical scheme to apply MGLASSO in practice using convex optimization algorithms. The objective function in 6 is the sum of three convex components: a smooth function (squared loss) and two non-smooth penalty functions: the LASSO function, which is separable i.e. can be

broken down into independent components involving single entries corresponding to predictors coefficients, and the group-fused LASSO penalty, which is non-separable. Several approaches can be used to tackle its minimization problem. Among them, we have subgradients methods (Shor, 2012), alternating direction methods of multipliers (ADMM, Boyd et al. (2011)) or continuation methods combined with smoothing techniques for the non-smooth parts of the criterion (see, e.g., Hadj-Selem et al. (2018)).

We compare the performances of the subgradient descent algorithm, the ADMM, and the continuation with Nesterov smoothing in a shrinkage-thresholding algorithm (CONESTA, Hadj-Selem et al. (2018)), the optimization algorithm used in practice to solve MGLASSO. Except for the subgradient case, the algorithms are generally applied to reformulated versions of the initial MGLASSO criterion. Section 3.2.1.1 derives the subgradients equations. Section 3.2.1.2 presents the ADMM algorithm. Section 3.2.1.3 reviews the principles of the continuation with Nesterov smoothing in a shrinkage-thresholding algorithm (CONESTA, Hadj-Selem et al. (2018)). The empirical convergence results of the algorithms in various settings can be found in Section 3.2.1.4.

3.2.1 Optimization algorithms

3.2.1.1 Optimization via subgradient descent

The first natural approach that can be used to solve the non-smooth optimization problem 6 is the subgradient descent algorithm (Boyd et al., 2003). The objective in the equation 6 is a non-smooth convex function in $\boldsymbol{\beta}$ and hence admits a non-unique solution for $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$. That solution is characterized by the following Karush-Kuhn-Tucker conditions, which are the MGLASSO equations for the subgradient: for all $i = 1, \ldots, p$:

$$\begin{aligned}\partial \mathcal{J}_{\lambda_1,\lambda_2}(\boldsymbol{\beta}^i) = &-\mathbf{X}^{\setminus i\top}\left(\mathbf{X}^i - \mathbf{X}^{\setminus i}\boldsymbol{\beta}^i\right) + \lambda_1 \operatorname{Sign}(\boldsymbol{\beta}^i) \\ &+ \lambda_2 \sum_{j \neq i}\left(\frac{\boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j}{\|\boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j\|_2}\mathbb{1}_{\boldsymbol{\beta}^i \neq \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j} + \gamma_{ij}\mathbb{1}_{\boldsymbol{\beta}^i = \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j}\right) = \mathbf{0}_p, \quad (3.5)\end{aligned}$$

with $\gamma_{ij} \in \mathbb{R}^p \;/\; \|\gamma_{ij}\|_2 \leq 1$ and $\operatorname{Sign}(\mathbf{x}) = (\operatorname{Sign}(x_i)), (1 \leq i \leq p)$

$$\operatorname{Sign}(x_i) = \begin{cases} 1 & \text{if } x_i > 0, \\ \in [-1, 1] & \text{if } x_i = 0, \\ -1 & \text{if } x_i < 0, \end{cases}$$

for all $\mathbf{x} \in \mathbb{R}^p$. These equations are sufficient and necessary for the solution.

The subgradient method solves problem 6 iteratively. The update at iteration k is given in algorithm 1 where $\nu^{(k)} > 0 \in \partial \mathcal{J}_{\lambda_1,\lambda_2}(\boldsymbol{\beta}^{(k)})$ and $\eta_k = 0.1/\sqrt{k}$ is the non-summable diminishing step size i.e. $\lim_{k\to\infty} \eta_k = 0$ and $\sum_{k=1}^{\infty} \eta_k = \infty$.

Algorithm 1: Subgradient method update

1 $\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} - \eta_k \nu^{(k)}$;

The algorithm is stopped when $\|\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)}\|_2 \leq \epsilon$ with ϵ the stopping tolerance. The results obtained through the subgradient method are not always sparse, and the convergence is slow (Bach et al., 2011).

Proposition 3.1. *For diminishing step sizes η_k, with $\lim_{k\to\infty} \eta_k = 0$ and $\sum_{k=1}^{\infty} \eta_k = \infty$, the subgradient algorithm is guaranteed to converge to an optimal value $J^\star$. Namely,*

$$\lim_{k\to\infty} J_{\lambda_1,\lambda_2}\left(\boldsymbol{\beta}^{(k)}\right) = J^\star.$$

Proof. Following Boyd et al. (2003), the proof is based on the fact that we assume J to be a Lipschitz continuous function and hence have the subgradients bounded by the Lipschitz constant. □

3.2.1.2 Optimization via ADMM

A second strategy for solving the optimization problem (6) for fixed λ_1 and λ_2 is the Alternating Direction Method of Multipliers (ADMM, Boyd et al. (2011)). The approach is adapted to cost functions with a separable structure. We introduce additional variables to account for the difference between two regression vectors: for $(i,j) \in \{1,\ldots,p\}^2$, $i<j$, let $\Delta_{i,j} \in \mathbb{R}^{(p-1)}$. We convert the unconstrained optimization (6) to the following constrained problem:

$$\begin{aligned} \arg\min \quad & \sum_{i=1}^{p} \frac{1}{2}||\mathbf{X}^i - \mathbf{X}^{\setminus i}\boldsymbol{\beta}^i||_2^2 + \lambda_1 \sum_{i=1}^{p} ||\boldsymbol{\beta}^i||_1 + \lambda_2 \sum_{i<j} ||\Delta_{i,j}||_2 \\ \text{such that} \quad & \Delta_{i,j} = \boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j \text{ for all } (i,j) \in \{1,\ldots,p\}^2. \end{aligned} \tag{3.6}$$

The augmented Lagrangian function writes, for $\boldsymbol{\beta} \in \mathbb{R}^{p(p-1)}$, $\Delta = (\Delta_{i,j}) \in \left(\mathbb{R}^{(p-1)}\right)^{p(p-1)/2}$, $\Gamma = (\Gamma_{i,j}) \in \left(\mathbb{R}^{(p-1)}\right)^{p(p-1)/2}$ and $\rho > 0$:

$$\begin{aligned} L_\rho(\boldsymbol{\beta}, \Delta, \Gamma) = & \frac{1}{2} \sum_{i=1}^{p} ||\mathbf{X}^i - \mathbf{X}^{\setminus i}\boldsymbol{\beta}^i||_2^2 + \lambda_1 \sum_{i=1}^{p} ||\boldsymbol{\beta}^i||_1 + \lambda_2 \sum_{i<j} ||\Delta_{i,j}||_2 \\ & + \sum_{i<j} \langle \Gamma_{i,j}, \Delta_{i,j} - \boldsymbol{\beta}^i + \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j \rangle + \sum_{i<j} \rho/2 \|\Delta_{i,j} - \boldsymbol{\beta}^i + \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j\|_2^2, \end{aligned} \tag{3.7}$$

where $\langle u, v \rangle$ denotes the standard scalar product in $\mathbb{R}^{(p-1)}$. ADMM consists of separate iterative updates of the primal variables $\boldsymbol{\beta}$ and Δ, and of the dual variable Γ. At iteration $k+1$, we apply the following update rules described in algorithm 2.

Algorithm 2: ADMM updates

while *not converged* **do**
  (i) $\boldsymbol{\beta}^{(k+1)} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{p(p-1)}} L_\rho(\boldsymbol{\beta}, \Delta^{(k)}, \Gamma^{(k)})$;
  (ii) $\Delta^{(k+1)} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{p(p-1)}} L_\rho(\boldsymbol{\beta}^{(k+1)}, \Delta, \Gamma^{(k)})$;
  (iii) $\Gamma_{i,j}^{(k+1)} = \Gamma_{i,j}^{(k)} + \rho(\Delta_{i,j}^{(k+1)} - \boldsymbol{\beta}^{i(k+1)} + \boldsymbol{\beta}^{j(k+1)})$;
end

Standard optimization procedures can be applied to carry out each one of these updates.

Update (i) boils down to a weighted LASSO problem. Denote $b = \mathrm{Vec}(\boldsymbol{\beta}) \in \mathbb{R}^{p(p-1)}$, obtained by stacking the rows of $\boldsymbol{\beta}$ above each other. Denote also, for $i \in \{1, \dots, p\}$, $\boldsymbol{J}_i \in \mathbb{R}^{(p-1)\times p(p-1)}$, the block matrix containing p sub-matrices of size $(p-1) \times (p-1)$, each null except for the i-th block which is equal to the identity matrix of order $(p-1)$. Up to the terms which do not depend on $\boldsymbol{\beta}$, minimizing the augmented Lagrangian with respect to b is equivalent to

$$b^{(k+1)} \in \underset{b\in\mathbb{R}^{p(p-1)}}{\arg\min} \frac{1}{2} \sum_{i=1}^{p} \left\| \mathbf{X}^i - \mathbf{X}^{\setminus i} \boldsymbol{J}_i b \right\|_2^2 + \sum_{i<j} \frac{\rho}{2} \left\| \Delta_{i,j}^{(k)} + \rho^{-1} \Gamma_{i,j}^{(k)} - (\boldsymbol{J}_i - \boldsymbol{J}_j) b \right\|_2^2 + \lambda_1 \|b\|_1, \quad (3.8)$$

which is a standard ℓ_1 penalized quadratic minimization problem. The main issues here are the problem's dimensionality and the non-separability of the quadratic term with respect to the $\boldsymbol{\beta}^i$, $i \in \{1, \dots, p\}$. On the other hand, update (ii) corresponds to a classical ℓ_2 penalized problem. Denoting, for $(i,j) \in \{1, \dots, p\}^2$, $v_{i,j}^{(k)} = -\rho^{-1}\Gamma_{i,j}^{(k)} + (\boldsymbol{J}_i - \boldsymbol{J}_j) b^{(k+1)}$, we solve

$$\Delta^{(k+1)} \in \underset{\Delta}{\arg\min} \sum_{i<j} \frac{\rho}{2} \left\| \Delta_{i,j} - v_{i,j}^{(k)} \right\|_2^2 + \lambda_2 \sum_{i<j} \|\Delta_{i,j}\|_2 , \quad (3.9)$$

which is separable across (i,j) pairs and solved in closed form. We obtain the following update rule for all $(i,j) \in \{1, \dots, p\}^2$:

$$\Delta_{i,j}^{(k+1)} = \left(1 - \lambda_2 \rho^{-1} \|v_{i,j}^{(k)}\|_2^{-1}\right)_+ v_{i,j}^{(k)}. \quad (3.10)$$

In contrast to the general residual balancing technique used to stop the ADMM algorithm (Boyd et al., 2011), the criterion we apply is inspired by the work of Chan et al. (2016), which sums up the primal and dual residuals. We then stop the algorithm when the *combined residual* $\zeta^{(k+1)} \leq \epsilon$ where $\epsilon > 0$ is the tolerance and

$$\zeta^{(k+1)} = \frac{1}{\sqrt{p(p-1)}} \left\| b^{(k+1)} - b^{(k)} \right\|_2 + \frac{1}{\sqrt{p(p-1)^2/2}} \left(\left\| \Delta^{(k+1)} - \Delta^{(k)} \right\|_2 + \left\| \Gamma^{(k+1)} - \Gamma^{(k)} \right\|_2 \right). \quad (3.11)$$

The factors $\frac{1}{\sqrt{p(p-1)}}$ and $\frac{1}{\sqrt{p(p-1)^2/2}}$ account for the fact that the ℓ_2 norms are in $\mathbb{R}^{p(p-1)}$ and $\mathbb{R}^{p(p-1)^2/2}$ respectively.

The MGLASSO problem is convex. It can be shown that the ADMM algorithm converges, i.e., in primal and dual residuals, when the penalty parameter ρ is held constant (Boyd et al., 2011).

Proposition 3.2. *The iterates $\boldsymbol{\beta}^{(k)}$ in the ADMM algorithm converge to the global minimizer $\boldsymbol{\beta}^\star$ of the multiscale graphical LASSO criterion $\mathcal{J}_{\lambda_1,\lambda_2}(\boldsymbol{\beta})$.*

Proof. The proposition is a direct corollary of the convergence of the ADMM algorithm for convex problems' optimization. □

Although the ADMM algorithm converges for the MGLASSO optimization problem, it results in a slow convergence in its vanilla version presented afore. The following section introduces a faster algorithm for the MGLASSO optimization problem, which is based on a continuation with Nesterov's smoothing (Nesterov, 2005) technique.

3.2.1.3 Optimization via CONESTA

The continuation with Nesterov smoothing in a shrinkage-thresholding algorithm (CONESTA, Hadj-Selem et al. (2018)) is dedicated to high-dimensional regression problems with structured sparsity. It optimizes criteria like (6) by using a smooth approximation of the fused LASSO penalty. We reformulate (6) to comply with the form of loss function required by CONESTA. The objective of MGLASSO writes :

$$f(\widetilde{\boldsymbol{\beta}}) = \frac{1}{2}||\mathbf{Y} - \widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}}||_2^2 + \lambda_1||\widetilde{\boldsymbol{\beta}}||_1 + \lambda_2 \sum_{i<j} ||\boldsymbol{D}_{ij}\widetilde{\boldsymbol{\beta}}||_2, \tag{3.12}$$

where $\mathbf{Y} = \mathrm{Vec}(\mathbf{X}) \in \mathbb{R}^{np}, \widetilde{\boldsymbol{\beta}} = \mathrm{Vec}(\boldsymbol{\beta}) \in \mathbb{R}^{p(p-1)}, \widetilde{\mathbf{X}}$ is a $\mathbb{R}^{[np]\times[p\times(p-1)]}$ block-diagonal matrix with $\mathbf{X}^{\setminus i}$ on the i-th block. The matrix $\boldsymbol{D}_{ij}$ is a $(p-1)\times p(p-1)$ matrix chosen so that $\boldsymbol{D}_{ij}\widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j$.

Notice that the above equation is a multivariate linear regression problem. Denote $s(\widetilde{\boldsymbol{\beta}}) = \sum_{i<j} ||\boldsymbol{D}_{ij}\widetilde{\boldsymbol{\beta}}||_2$ the $\ell_{1,2}$ group-norm. A smooth function can approximate this non-smooth penalty with a known gradient computed using Nesterov's smoothing (Nesterov, 2005). Given a smoothness parameter $\mu > 0$, the smooth approximation is defined as follows:

$$s_\mu(\widetilde{\boldsymbol{\beta}}) = \max_{\boldsymbol{\alpha}\in\mathcal{K}} \left\{ \boldsymbol{\alpha}^\top \mathbf{D}\widetilde{\boldsymbol{\beta}} - \frac{\mu}{2}\|\boldsymbol{\alpha}\|_2^2 \right\}, \tag{3.13}$$

where $\mathcal{K}$ is the cartesian product of ℓ_2-unit balls, $\mathbf{D}$ is the vertical concatenation of the matrices $\boldsymbol{D}_{ij}$ and $\boldsymbol{\alpha}$ is an auxiliary variable resulting from the dual reformulation of $s(\widetilde{\boldsymbol{\beta}})$. Note that $\lim_{\mu\to 0} s_\mu(\widetilde{\boldsymbol{\beta}}) = s(\widetilde{\boldsymbol{\beta}})$.

A Fast Iterative Shrinkage-Thresholding Algorithm (FISTA, Beck and Teboulle (2009)) step can then be applied after computing the gradient of the smooth part. Denote $g(\widetilde{\boldsymbol{\beta}}) = ||\mathbf{Y} - \widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}}||_2^2$, the gradient is:

$$\nabla_{\widetilde{\boldsymbol{\beta}}}(g + \lambda_2 s_\mu) = -\widetilde{\mathbf{X}}^\top(\mathbf{Y} - \widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}}) + \boldsymbol{D}^\top \boldsymbol{\alpha}_\mu^\star(\widetilde{\boldsymbol{\beta}}), \tag{3.14}$$

where $\alpha_\mu^\star(\widetilde{\boldsymbol{\beta}})$ is the solution to problem (3.13).

The main ingredient of CONESTA remains in determining the optimal smoothness parameter using the duality gap, which minimizes the number of FISTA iterations for a given tolerance ϵ. Let the smooth approximation of the function $f(\widetilde{\boldsymbol{\beta}})$ be:

$$f_\mu(\widetilde{\boldsymbol{\beta}}) = l(\widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}}) + Q_\mu(\widetilde{\boldsymbol{\beta}}) \tag{3.15}$$

where $l(\widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}}) = \|\widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}} - \mathbf{Y}\|_2^2$ and $Q_\mu(\widetilde{\boldsymbol{\beta}} = \lambda_1||\widetilde{\boldsymbol{\beta}}||_1 + \alpha_\mu^\star(\widetilde{\boldsymbol{\beta}})^\top \mathbf{D}\widetilde{\boldsymbol{\beta}} - \frac{\mu}{2}\|\alpha_\mu^\star\|_2^2$. The duality gap is given by

$$\mathrm{GAP}_\mu(\widetilde{\boldsymbol{\beta}}^{(k)}) = f_\mu(\widetilde{\boldsymbol{\beta}}^{(k)}) + l^\star\left(\widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}}^{(k)} - \mathbf{Y}\right) + Q_{\mu,k}^\star\left(-\widetilde{\mathbf{X}}^\top\left(\widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}}^{(k)} - \mathbf{Y}\right)\right) \tag{3.16}$$

where $l^\star$ and $Q_{\mu,k}^\star$ are the Fenchel conjugates.

The specification of μ is subject to dynamic update. A sequence of decreasing optimal smoothness parameters is generated to adapt the FISTA algorithm stepsize towards ϵ dynamically. Namely, $\mu^{(k)} = \mu_{opt}(\epsilon^{(k)})$. The smoothness parameter decreases as one gets closer to $\widetilde{\boldsymbol{\beta}}^\star$, a solution of problem (3.12). Since $\widetilde{\boldsymbol{\beta}}^\star$ is unknown, the approximation of the distance to the minimum is achieved via the duality gap. Indeed

$$\mathrm{GAP}(\widetilde{\boldsymbol{\beta}}^{(k)}) \geq f(\widetilde{\boldsymbol{\beta}}^{(k)}) - f(\widetilde{\boldsymbol{\beta}}^\star) \geq 0.$$

The CONESTA routine is spelled out in the algorithm 3 where $L(g + \lambda_2 s_\mu)$ is the Lipschitz constant of $\nabla(g + \lambda_2 s_\mu)$, k is the iteration counter for the inner FISTA updates and i is the iteration counter for CONESTA updates.

Algorithm 3: CONESTA solver

Input : $f(\widetilde{\boldsymbol{\beta}}) = g(\widetilde{\boldsymbol{\beta}}) + h(\widetilde{\boldsymbol{\beta}}) + s(\widetilde{\boldsymbol{\beta}})$, $\epsilon > 0$, $\lambda_1 > 0, \lambda_2 > 0$, $\boldsymbol{\tau} \in (0, 1)$
Output: $\widetilde{\boldsymbol{\beta}}$

Initialize $\widetilde{\boldsymbol{\beta}}^{(0)}, \epsilon^{(0)} = \boldsymbol{\tau}\,\mathrm{GAP}_{\mu=10^{-8}}(\widetilde{\boldsymbol{\beta}}^{(0)}), \mu^{(0)} = \mu_{opt}(\epsilon^{(0)})$;
repeat
  $\epsilon_\mu^{(i)} = \epsilon^{(i)} - \mu^{(i)}\lambda_2\frac{d}{2}$;
  /* FISTA */
  $k = 2$ /* new iterator */
  $\widetilde{\boldsymbol{\beta}}^{(1)}_{\mathrm{FISTA}} = \widetilde{\boldsymbol{\beta}}^{(0)}_{\mathrm{FISTA}} = \widetilde{\boldsymbol{\beta}}^{(k)}$ /* Initial parameters value */
  $t_\mu = \frac{1}{L(g+\lambda_2 s_\mu)}$ /* Compute stepsize with $L(g + \lambda_2 s_\mu)$ the Lipschitz constant of $\nabla(g + \lambda_2 s_\mu)$ */
  repeat
    $\boldsymbol{z} = \widetilde{\boldsymbol{\beta}}^{(k-1)}_{\mathrm{FISTA}} + \frac{k-2}{k+1}(\widetilde{\boldsymbol{\beta}}^{(k-1)}_{\mathrm{FISTA}} - \widetilde{\boldsymbol{\beta}}^{(k-2)}_{\mathrm{FISTA}})$
    $\widetilde{\boldsymbol{\beta}}^{(k)}_{\mathrm{FISTA}} = \mathrm{prox}_{\lambda_1 h}(\boldsymbol{z} - t_\mu \nabla(g + \lambda_2 s_\mu)(\boldsymbol{z}))$
  until $\mathrm{GAP}_\mu(\widetilde{\boldsymbol{\beta}}^{(k)}_{\mathrm{FISTA}}) \leq \epsilon_\mu^{(i)}$;
  $\widetilde{\boldsymbol{\beta}}^{(k+1)} = \boldsymbol{\theta}^k_{\mathrm{FISTA}}$;
  $\epsilon^{(i)} = \mathrm{GAP}_{\mu=\mu_i}\widetilde{\boldsymbol{\beta}}^{(k+1)} + \mu^{(i)}\lambda_2\frac{d}{2}$;
  $\epsilon^{(i+1)} = \boldsymbol{\tau}\epsilon^{(i)}$;
  $\mu^{(i+1)} = \mu_{opt}(\epsilon^{(i+1)})$
until $\epsilon^{(i)} \leq \epsilon$;

Proposition 3.3. *The iterates $\boldsymbol{\beta}^{(k)}$ in the CONESTA algorithm converge to the global minimizer $\boldsymbol{\beta}^\star$ of the multiscale graphical LASSO criterion $\mathcal{J}_{\lambda_1,\lambda_2}(\boldsymbol{\beta})$.*

Proof. The proposition is straightforwardly deduced from the convergence theorem of the CONESTA algorithm in the continuation scheme (see Theorem 3 Hadj-Selem et al. (2018)). □

3.2.1.4 Empirical convergence analysis

In this section, we present the numerical results for convergence analysis. We use the objective function criterion at $\beta^{(k)}$ to assess the convergence of algorithms for different levels of tolerance for simulated data sets. We show the computation times results for different simulation models and the scalability of the algorithms according to the stopping tolerance and the difficulty of the problem (number of variables in the data set).

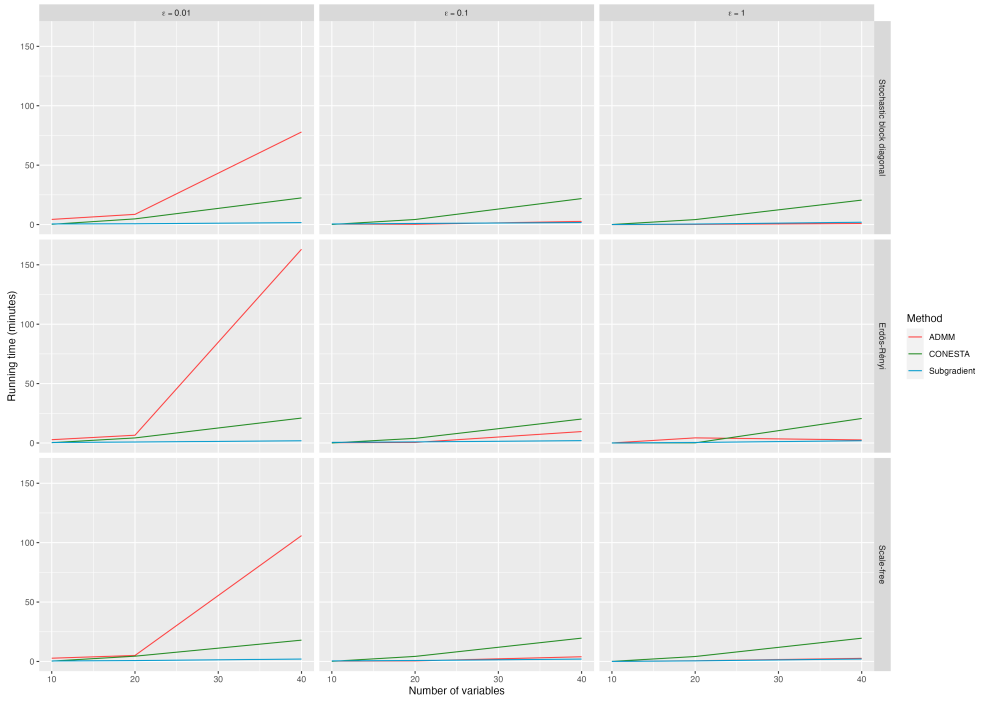


Figure 3.4: Scalability of the algorithms on three types of graph structures. The mean running time in minutes is plotted against the dimension of the data set for different tolerance levels.

Our analysis differs from Hadj-Selem et al. (2018), where the convergence error i.e $f(\beta^{(k)}) - f(\beta^\star)$ is used to assess convergence. Indeed, the authors proposed in Löfstedt et al. (2018) an approach to simulate a response variable Y and predictors X from a regression vector $\beta^\star$ for linear regression with control over the penalization parameters. This simulation framework does not fit the specific case of reformulated Gaussian graphical models in which the equality $Y = \mathrm{vec(X)}$ is required.

Details on the simulation models are provided in Section 3.4.1.1. We used an Erdös-Rényi simulation framework parameterized by $\alpha = 0.1$. The stochastic block model contains 5 blocks of equal size. The scale-free model contains as many expected edges as the number of variables. We set $\lambda_1 = 0.5$ and $\lambda_2 = 10$ to run the three optimization algorithms. The number of observations is fixed at $n = 20$. The number of variables p is chosen in $\{10, 20, 40\}$. The stopping tolerance levels fixed are $10^0, 10^{-1}$ and 10^{-2}. The algorithms have been implemented in Python and launched via an R program. The timing results are shown in minutes. We choose the ADMM penalty parameter ρ fixed and proportional to the parameter λ_2. The algorithms are stopped when the stopping tolerance is not reached after a maximum number of iterations equals 10000.

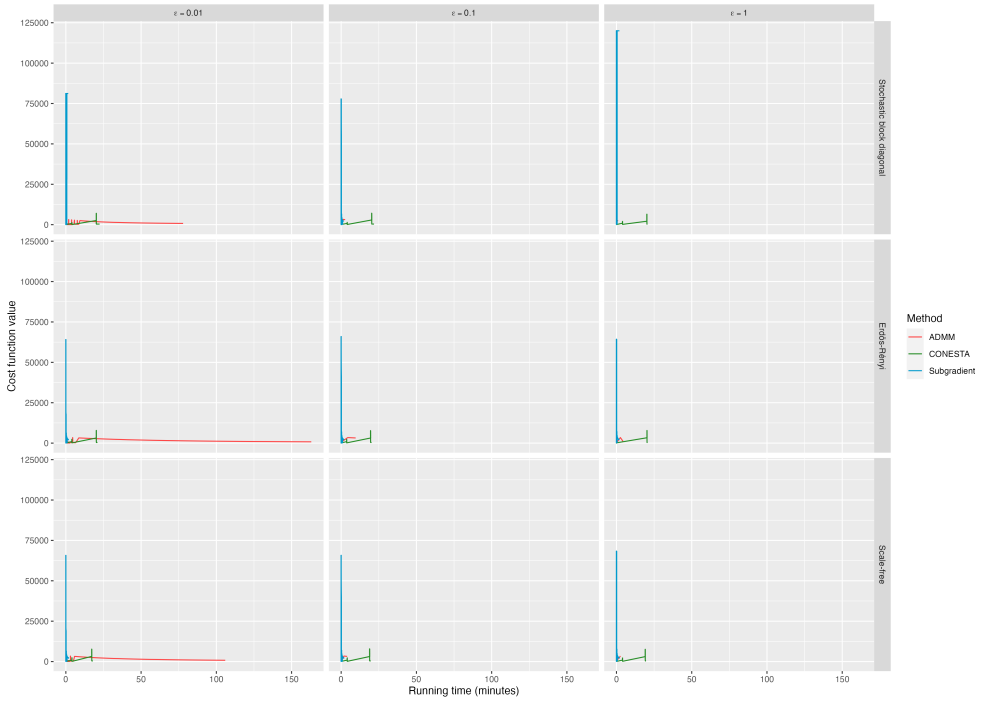


Figure 3.5: Comparison of the effect of different tolerance levels. We plot the convergence as a function of the running time.

Figure 3.5 and Figure 3.4 indicate that the subgradient method is the fastest algorithm for all stopping tolerance values. However, the algorithm does not converge and does not yield sparse solutions. ADMM is faster than CONESTA for higher stopping tolerance ($\epsilon = 1$ and $\epsilon = 0.1$). However, for lower stopping tolerance ($\epsilon = 0.01$) CONESTA is preferable to ADMM.

The results of our convergence analysis corroborate the more expanded analysis done in Hadj-Selem et al. (2018). The CONESTA is a superior approach for lower stopping tolerances and is the one chosen in practice for the MGLASSO problem optimization.

3.2.2 Practical implementation

3.2.2.1 Path algorithm

In practice, the MGLASSO is implemented as a path-algorithm in which the LASSO penalty is fixed. At each evaluation of the criterion 6 for a pair (λ_1, λ_2), the CONESTA solver is applied. In order to determine clustering assignments as the fusion penalty increases, the scheme described in algorithm 4 has been proposed. If $d(i,j) = \left\| \hat{\boldsymbol{\beta}}^i - \boldsymbol{\tau}_{ij} \hat{\boldsymbol{\beta}}^j \right\|_2 \leq \epsilon_{fuse}$ then the elements of the pair (i,j) are assigned to the same cluster.

Algorithm 4: MGLASSO algorithm

Input : $\mathbf{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^p\} \in \mathbb{R}^{n \times p}$, $\lambda_1 \geq 0, \lambda_{2\text{initial}} > 0$, $\eta > 1$, $\epsilon_{fuse} \geq 0$

Output: $\boldsymbol{\beta}(\lambda_1, \lambda_2) \in \mathbb{R}^{p \times (p-1)}$, $C(\lambda_1, \lambda_2)$ for all (λ_1, λ_2)

Initialize $\boldsymbol{\beta}^i = (\mathbf{X}^i)^\dagger \mathbf{X}^i, \quad \forall i = 1, \ldots, p$;
Set $\lambda_2 = 0$;
Compute $\boldsymbol{\beta}$ using CONESTA solver in algorithm 3 ;
Compute clusters C ;
Set $\lambda_2 = \lambda_{2\text{initial}}$;
/* Clustering path */
while $\text{Card}(C) > 1$ **do**
    Compute $\boldsymbol{\beta}$ with CONESTA algorithm 3 ;
    Compute clusters C;
    $\lambda_2 = \lambda_2 \times \nu$;

3.2.2.2 The mglasso package

The algorithm 4 is available as an R (R Core Team, 2022) package **mglasso** (version 0.1.2, Sanou (2022)) available on the comprehensive R archive network (CRAN) and Github (https://github.com/desanou/mglasso). It is based on the Python (vanRossum, 1995) library **pylearn-parsimony** (Hadj-Selem et al., 2018) and the R package **reticulate** (Ushey et al., 2020) which provides an R interface to Python code. The **mglasso** package provides both Python and R functions and can be installed at https://cran.r-project.org/package=mglasso.

```
R> install.packages("mglasso")
R> library(mglasso)
```

In the following, we show how to use in practice the principal functions and highlight a hidden function that builds the linear operator A in the reformulated criterion 3.12.

3.2.2.2.1 Installing Python libraries from Github into R

After installing **mglasso**, the required python libraries are delay loaded. That allows the package to be loaded even when the Python engine is not correctly set up. The next step is then installing the Python dependencies. For libraries not available on Conda distribution or PyPi, **reticulate** does not offer an easy option to install them directly. The function `install_pylearn_parsimony()` in **mglasso** needs to be run in order to install all the libraries not automatically installed by reticulate. The following code installs the libraries in the Conda environment *rmglasso*.

```
R> install_pylearn_parsimony(envname = "rmglasso", method =
    "conda")
R> reticulate::use_condaenv("rmglasso", required = TRUE)
R> reticulate::py_config() ## Initialize the Python engine
```

3.2.2.2.2 The MGLASSO function

The CONESTA solver is loaded from a library that provides optimization methods for regression models with structured and sparse penalties. In order to solve the MGLASSO problem, we construct a linear operator $\mathbf{A} = \{\mathbf{A}_k\}_{k\in\{1,\ldots,p\}}$ which encodes the structure of the fusion penalty. Instead of constructing the matrix $\boldsymbol{D}$ (see the equation (3.13)) with $\frac{p(p-1)}{2}$ components, each corresponding to the pairwise difference structure, the component $\mathbf{A}_k$ in $\mathbf{A}$ encodes the neighborhood structure of the k-th variable. In other words, $\mathbf{A} = \{\mathbf{A}_k\}_{k\in\{1,\ldots,p\}}$ is a reorganization of the elements of $\boldsymbol{D} = \{\boldsymbol{D}_{ij}\}_{1\leq i<j\leq p}$ so that the structure of the differences involving the variable k is stored in the component $\boldsymbol{A}_k$ (see Algorithm 5). This step, not directly accessible, is done internally via the following python routine:

```
Python> A_=linear_operator_from_num_variables(p, type_, W_)
```

Algorithm 5: Component k of linear operator $\boldsymbol{A}$.

Input : $p \in \mathbb{N}$, $k \in \{1, \ldots, p\}$,
Output: $\boldsymbol{A}_k$

$l = 1$
for $i \leftarrow 1$ **to** $p - 1$ **do**
  for $j \leftarrow i + 1$ **to** p **do**
    if $i = k$ **then**
      $\boldsymbol{A}_k[l, (i-1) \times p + k] = 1$;
      $\boldsymbol{A}_k[l, (j-1) \times p + j] = -1$;
    else if $j = k$ **then**
      $\boldsymbol{A}_k[l, (i-1) \times p + j] = 1$;
      $\boldsymbol{A}_k[l, (j-1) \times p + i] = -1$;
    else
      $\boldsymbol{A}_k[l, (i-1) \times p + k] = 1$;
      $\boldsymbol{A}_k[l, (j-1) \times p + k] = -1$;
    end
    l = l+1 ;
  end
end

Let us simulate multivariate random data from a simple block-diagonal model to show how to estimate a Gaussian graphical model through MGLASSO. The data set contains $n = 50$ observations of 9 Gaussian variables with $K = 3$ blocks. To set up the block-diagonal model, we first simulate a correlation matrix (Figure 3.6) for which intra-clusters correlation levels are set arbitrarily to $\rho = 0.85$. The multivariate random data $\mathbf{X}$ is then simulated from that correlation matrix.

```
R> n = 50
R> K = 3
R> p = 9
R> rho = 0.85
R> blocs <- list()
```

```
R> for (j in 1:K) {
R>   bloc <- matrix(rho, nrow = p/K, ncol = p/K)
R>   for(i in 1:(p/K)) { bloc[i,i] <- 1 }
R>   blocs[[j]] <- bloc
R>  }
R> mat.correlation <- Matrix::bdiag(blocs)
R> set.seed(11)
R> mglasso_data <- mvtnorm::rmvnorm(n, mean = rep(0,p), sigma =
    as.matrix(mat.correlation))
R> corrplot::corrplot(as.matrix(mat.correlation), method =
    "color", tl.col="black")
```

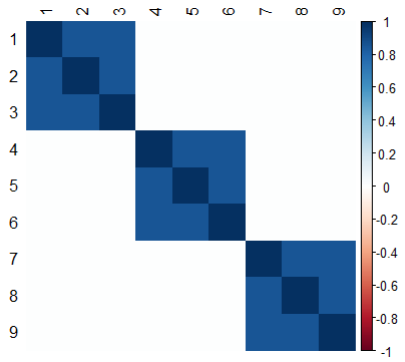


Figure 3.6: Correlation matrix for a 3-block diagonal model with 3 variables per clusters.

The MGLASSO routine is called via the function `mglasso()` to estimate graphs and clusters for a fixed LASSO penalty λ_1 and a grid of group fused LASSO regularization parameters λ_2s.

```
R> mglasso_data <- scale(mglasso_data)
R> fit_mglasso <- mglasso(mglasso_data, lambda1 = 0.2*n,
    lambda2_start = 0.1, fuse_thresh = 1e-3, verbose = FALSE)
```

`mglasso()` returns a list with the following entries:

- `fit_mglasso$out` returns a list for which each element corresponds to a λ_2 value and a clustering level. An element `fit_mglasso$out$levelk` contains the regression vectors' matrix beta `beta` and the computed cluster partition `clusters` with k clusters;
- `fit_mglasso$lambda1` stores the LASSO parameter.

3.2.2.2.3 Clustering path

We explore the clustering path using the function `plot_clusterpath()`. It returns the algorithm regularization path representation when λ_1 is kept fixed, and

λ_2 varies. We plot the estimated $\hat{\mathbf{X}}$ at each level onto the two principal components of the input data in Figure 3.7.

```
R> library(ggplot2)
R> library(ggrepel)
R> mglasso:::plot_clusterpath(as.matrix(mglasso_data),
    fit_mglasso)
```

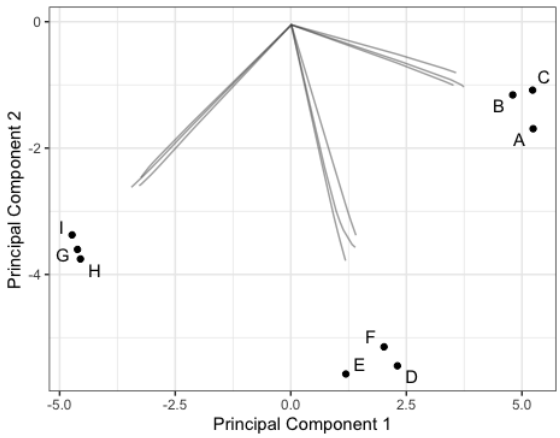


Figure 3.7: Clustering path

3.2.2.2.4 Graphs path

The function `plot_mglasso` provides the adjacency matrices for each clustering level in Figure 3.8.

```
R> plot_mglasso(res)
```

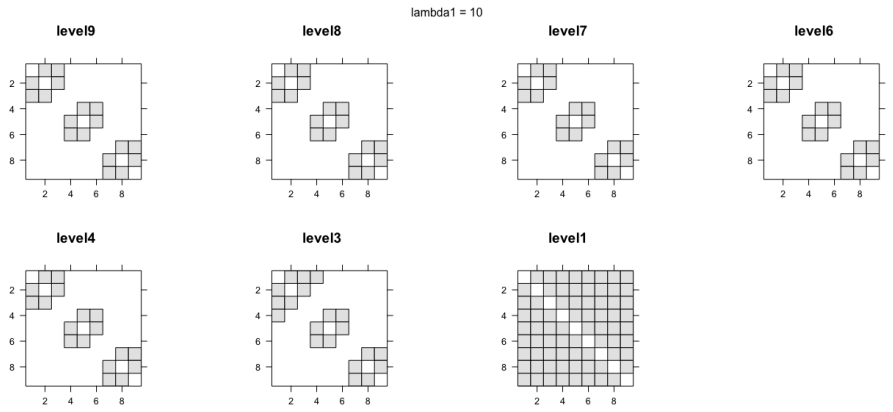


Figure 3.8: Estimated adjacency matrices for different levels of λ_2 values

3.3 Model selection

A crucial question for practical applications consists in defining a rule to select the penalty parameters (λ_1, λ_2). This selection problem operates at two levels: λ_1 controls the sparsity of the graphical model, and λ_2 controls the number of clusters in the optimal clustering partition. First, these two parameters can be dealt with separately: the sparsity parameter λ_1, chosen via model selection, while the clustering parameter λ_2 varies across a grid of values to obtain graphs with different levels of granularity. Secondly, considering λ_1 fixed, a model selection rule can be applied to tune λ_2. Finally, the parameters can be selected simultaneously. The approach privileged in practical applications of MGLASSO is the first scheme that allows highlighting the multiscale structure in a Gaussian data set. Nonetheless, the selection criterion used for that scheme can be easily extended to others.

3.3.1 Selection of LASSO regularizer

The problem of model selection in graphical models is difficult in the high dimensional case where the number of samples is small compared to the number of variables, as classical Akaike information criterion (AIC, Akaike (1998)) and Bayesian information criterion (BIC, Schwarz (1978)) tend to perform poorly (Liu et al., 2010).

In the following, we focus on the StARS stability selection approach proposed by Liu et al. (2010) as suggested by some preliminary tests where we compared the Extended BIC (EBIC, Foygel and Drton (2010)), the BIC calibrated with slope heuristics (Baudry et al., 2012), the Rotation invariant criterion implemented in the Huge package (Zhao et al., 2012), the GGMSelect procedure (Giraud et al., 2012), cross-validation (Bien and Tibshirani, 2011) and StARS. The results of the comparative analysis are shown in Section 3.3.2.

The StARS method uses k subsamples of data to estimate the associated graphs for a given range of λ_1 values. For each value, a global instability of the graph edges is computed. The optimal value of λ_1 is chosen so as to minimize the instability, as follows. Let $\lambda_1^{(1)}, \ldots, \lambda_1^{(K)}$ be a grid of sparsity regularization parameters, and $S_1, \ldots, S_N$ be the N bootstrap samples obtained by sampling the rows of the data set $\mathbf{X}$. For each $k \in \{1, \ldots, K\}$ and for each $j \in \{1, \ldots, N\}$, we denote by $\mathcal{A}^{k,j}(\mathbf{X})$ the adjacency matrix of the estimated graph obtained by applying the inference algorithm to S_n with regularization parameter $\lambda_1^{(k)}$. For each possible edge $(s,t) \in \{1, \ldots, p\}^2$, the probability of edge appearance is estimated empirically by

$$\hat{\theta}_{st}^{(k)} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{A}_{st}^{k,j}.$$

Define

$$\hat{\xi}_{st}(\lambda_1^{(k)}) = 2\hat{\theta}_{st}^{(k)} \left(1 - \hat{\theta}_{st}^{(k)}\right)$$

the empirical instability of edge (s,t) (that is, twice the variance of the Bernoulli

indicator of edge (s,t)). The instability level associated with $\lambda_1^{(k)}$ is given by

$$\hat{D}(\lambda_1^{(k)}) = \frac{\sum_{s<t} \hat{\xi}_{st}(\lambda_1^{(k)})}{\binom{p}{2}}.$$

StARS selects the optimal penalty parameter as follows

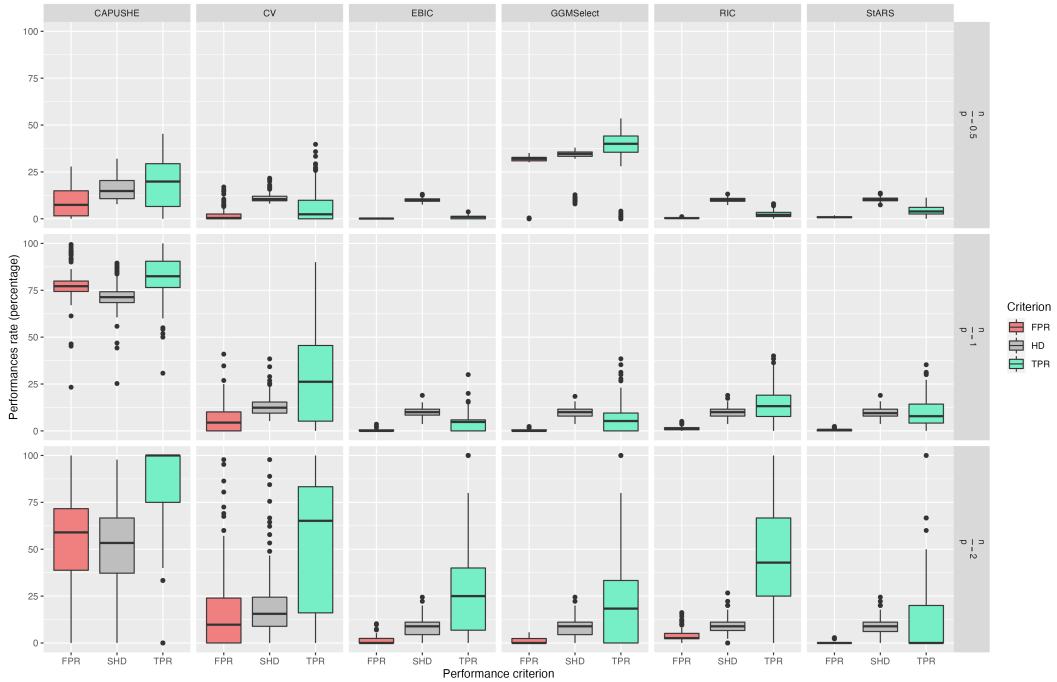
$$\hat{\lambda} = \max_k \left\{ \lambda_1^{(k)} : \hat{D}(\lambda_1^{(k)}) \leq \upsilon, k \in \{1, \ldots, K\} \right\},$$

where υ is the threshold chosen for the instability level.

3.3.2 Other selection approaches

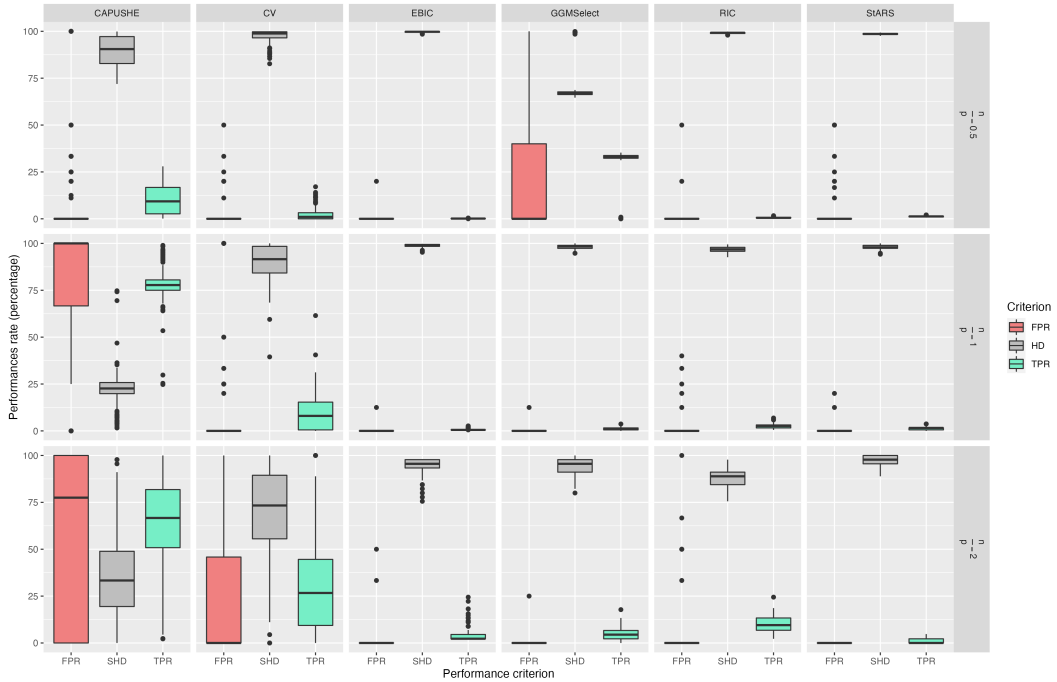
The goal is to outline how the model selection methods compare in different simulation settings. The performances are evaluated in Erdös-Rényi, scale-free, and stochastic block models. The selection methods compared are cross-validation (CV), CAPUSHE for selection based on slope heuristic, extended Bayesian information criterion (EBIC), GGMSelect, rotation invariant criterion (RIC), and the stability-based approach StARS. We used an Erdös-Rényi simulation framework parameterized by $\alpha = 0.1$. The stochastic block model contains 5 blocks with equal sizes. The scale-free model contains as many expected edges as the number of variables. Details on the simulation models are provided in Section 3.4.1.1. The number of observations is fixed to $n = 20$. The number of variables p is chosen in $\{10, 20, 40\}$. For each simulation setting, data are replicated 200 times. Performances are assessed via the false positive rate (FPR), true positive rate (TPR), and structural Hamming distance. The parameter λ_1 takes values in $[0, 1]$. The parameter λ_2 is fixed to 0.

In Figures, 3.9a, 3.9c, 3.9b, each row of graphs represents a fixed value of the ratio n/p. Each of the 6 columns corresponds to a model selection criterion. The graphs give the performance values in percentage on the y-axis and the performance criterion on the x-axis. The boxplots are taken over 200 simulated datasets. On the one hand, low values of structural Hamming distances and false positive rates indicate good performances. On the other hand, high values of true positive rate correspond to good performances.

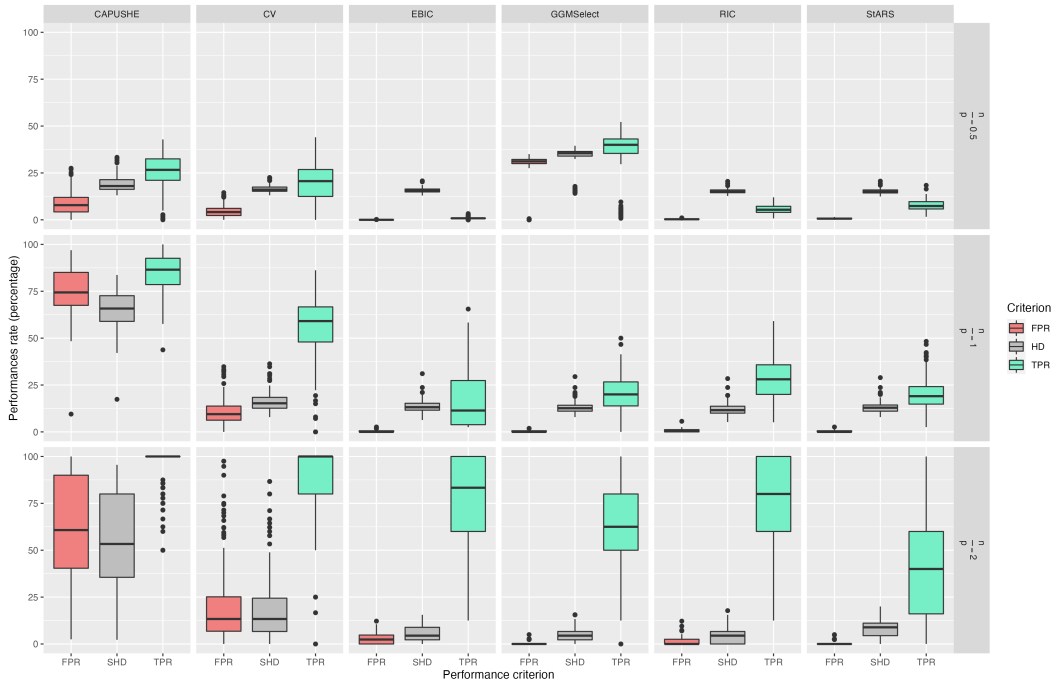


(a) Erdös-Rényi model

In general, no matter the selection criterion used, the performances get better when n/p is higher. The parameterization of the scale-free model does not scale well to the criteria used. If one is interested in minimizing the risk of false edge detections, StARS and RIC methods are preferred. They show consistent behavior over the different simulation settings. If the goal is to encourage edge detection GGMSelect approach is well suited. Note that the analysis is not meant to give an overall best selection criterion. According to the simulation setting, data nature, or the analysis objective, a selection criterion can be preferred to another. We preferred a criterion based on bootstrap that requires mild working conditions.



(b) Scale-free model



(c) Stochastic block model

Figure 3.9: False positive rate, True positive rate, and structural Hamming distance for different model selection criteria in a neighborhood selection problem. Measures are taken on 200 data sets simulated with p chosen in $\{20, 40, 80\}$ and $n = 40$.

3.4 Model performance

In this Section, we conduct a simulation study to evaluate the performance of the MGLASSO method, both in terms of clustering and support recovery. Receiver Operating Characteristic (ROC) curves are used to evaluate the adequacy of the inferred graphs with the ground truth, for the MGLASSO and GLASSO in its neighborhood selection version, in the Erdös-Rényi (Erdős et al., 1960), scale-free (Newman et al., 2001), and Stochastic Block Models (SBM, Fienberg and Wasserman (1981)) frameworks. The Adjusted Rand indices are used to compare the partitions obtained with MGLASSO, hierarchical agglomerative clustering, and k-means clustering in a stochastic block model framework and a hierarchical model.

3.4.1 Synthetic data models

3.4.1.1 Single level random graphs models

We consider three different synthetic network models: the Stochastic Block Model (Fienberg and Wasserman, 1981), the Erdös-Renyi model (Erdős et al., 1960) and the Scale-Free model (Newman et al., 2001). In each case, Gaussian data is generated by drawing n independent realizations of a multivariate Gaussian distribution $\mathcal{N}(0, \mathbf{\Sigma})$ where $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$ and $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$. The support of $\mathbf{\Omega}$, equivalent to the network adjacency matrix, is generated from the three different models. The difficulty level of the problem is controlled by varying the ratio $\frac{n}{p}$ with p fixed at 40: $\frac{n}{p} \in \{0.5, 1, 2\}$.

3.4.1.1.1 Stochastic Block-Model

We construct a block-diagonal precision matrix $\mathbf{\Omega}$ as follows. First, we generate the support of $\mathbf{\Omega}$ as shown in Figure 3.10, denoted by $\boldsymbol{A} \in \{0, 1\}^{p \times p}$. To do this, the variables are first partitioned into $K = 5$ hidden groups, noted $C_1, \ldots, C_K$ described by a latent random variable Z_i, such that $Z_i = k$ if $i = C_k$. Z_i follows a multinomial distribution

$$P(Z_i = k) = \pi_k, \quad \forall k \in \{1, \ldots, K\},$$

where $\pi = (\pi_1, \ldots, \pi_k)$ is the vector of proportions of clusters whose sum is equal to one. The set of latent variables is noted $\mathbf{Z} = \{Z_1, \ldots, Z_K\}$. Conditionally to $\mathbf{Z}$, A_{ij} follows a Bernoulli distribution such that

$$A_{ij} | Z_i = k, Z_j = l \sim \mathcal{B}(\alpha_{kl}), \quad \forall k, l \in \{1, \ldots, K\},$$

where α_{kl} is the probability of inter-cluster connectivity, with $\alpha_{kl} = 0.01$ if $k \neq l$ and $\alpha_{ll} = 0,75$. For $k \in \{1, \ldots, K\}$, we define $p_k = \sum_{i=1}^{p} \mathbf{1}_{\{Z_i = k\}}$. The precision matrix $\mathbf{\Omega}$ of the graph is then calculated as follows. We define $\Omega_{ij} = 0$ if $Z_i \neq Z_j$; otherwise, we define $\Omega_{ij} = A_{ij}\omega_{ij}$ where, for all $i \in \{1, \ldots, p\}$ and for all

$j \in \{1, \ldots, p | Z_j = Z_i\}$, ω_{ij} is given by :

$$\omega_{ii} := \frac{1 + \rho(p_{Z_i} - 2)}{1 + \rho(p_{Z_i} - 2) - \rho^2(p_{Z_i} - 1)};$$
$$\omega_{ij} := \frac{-\rho}{1 + \rho(p_{Z_i} - 2) - \rho^2(p_{Z_i} - 1)}.$$

If α_{ll} were to be equal to one, this construction of $\mathbf{\Omega}$ would make it possible to control the level of correlation between the variables in each block to ρ. Introducing a more realistic scheme with $\alpha_{ll} = 0.75$ allows only to have an approximate control.

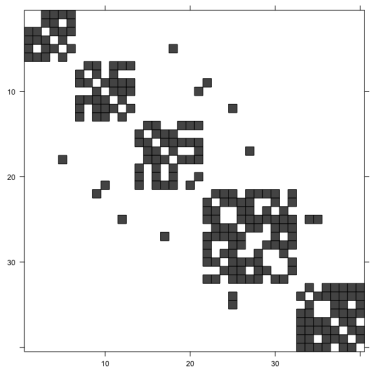


Figure 3.10: Adjacency matrix of a stochastic block model defined by $K = 5$ classes with identical prior probabilities set to $\pi = 1/K$, inter-classes connection probability $\alpha_{kl} = 0.75, k \neq l$, intra-classes connection probability $\alpha_{ll} = 0.01$ and $p = 40$ vertices.

3.4.1.1.2 Erdös-Renyi Model

The Erdös-Renyi model is a special case of the stochastic block model where $\alpha_{kl} = \alpha_{ll} = \alpha$ is constant. We set the density α of the graph to 0.1; see Figure 3.11 for an example of the graph resulting from this model.

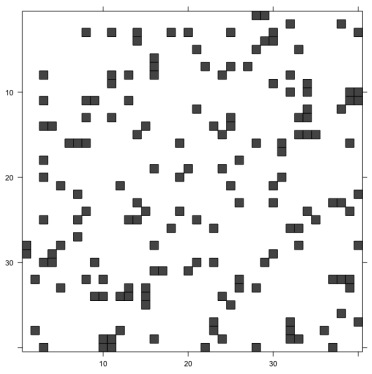


Figure 3.11: Adjacency matrix of an Erdös-Renyi model with probability of connection $\alpha = 0.1$ and $p = 40$ vertices.

3.4.1.1.3 Scale-free Model

The scale-free Model generates networks whose degree distributions follow a power law. The graph starts with an initial chain graph of 2 nodes. Then, new nodes are added to the graph one by one. Each new node is connected to an existing node with a probability proportional to the degree of the existing node. We set the number of edges in the graph to 40. An example of scale-free graph is shown in Figure 3.12.

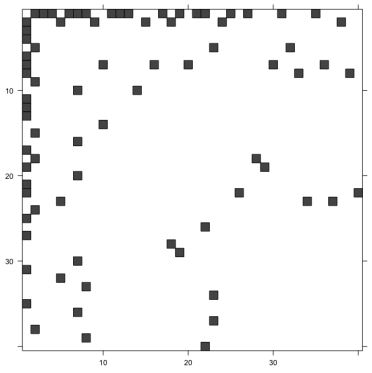


Figure 3.12: Adjacency matrix of a scale-free model with 40 edges and $p = 40$ nodes.

3.4.1.2 Multilevel or hierarchically structured models

The motivation of these models is to simulate hierarchically-structured groups of variables to illustrate how the performances of the MGLASSO compare to other standard clustering methods. To do so, we will construct variance-covariance matrices with underlying hierarchical clustering structure, using stochastic phylogeny trees. We first recall briefly the principles of stochastic phylogeny trees, before

describing the detailed model. Phylogeny trees were initially used to describe relationships between species and their evolution through time. The tree model describes when speciation occurs, and the evolutionary model describes how the species' quantitative traits evolve. The first can be chosen in a family available in the **ape** R package (Paradis et al., 2004), and the second is usually a stochastic process, e.g., the Brownian motion or the Ornstein-Uhlenbeck process. The tree model and the evolutionary model are intimately related. When speciation occurs from a specie, the children carry the parent trait and evolve independently. Hence, a correlation or covariance can be defined between the children's traits, and this quantity is known to be proportional to the shared evolution time (Bastide et al., 2017). Here we are interested in the covariance structure between the leaves, i.e., at the end of the evolution process, to derive the hierarchically structured covariance matrix.

Let us introduce some terminology about trees. Let $G = (V, E)$ be an undirected graph with edge set E and vertices set V. Let $T = (V, F)$ be any rooted ultrametric tree whose leaves correspond with V. Ultrametric trees are a class of phylogenetic trees where leaves are equidistant to the root. Each edge $f \in F$ has an associated branch length l_f. For leaf i, $t_i = l_f = h$ denotes the distance from the root to the leaf, and $pa(i)$ is the leaf's parent. For leaves $i, j \in V$, $\mathrm{mrca}(i, j)$ denotes their most recent common ancestor and $t_{\mathrm{mrca}(i,j)} = t_{i,j}$ is the distance from the root to $\mathrm{mrca}(i, j)$. The phylogenetic distance $d_{i,j}$ between the leaves is $d_{i,j} = t_i + t_j - 2t_{i,j}$. Note that T and G do not share the same edge set.

Denote $X_{i1 \leq i \leq \mathrm{Card}(V)}$ a sequence of continuous random variables describing a given trait at each leaf. Let us assume that the branch l_f has child leaf i and parent node $pa(i)$. A quantitative trait evolves on this branch according to a stochastic process $(W_t^f, 0 \leq t \leq l_f)$ with distribution $\mathcal{P}(\omega_e)$, independently from other species, conditionally on $W_0^e = X_{pa(i)}$. At leaf i, we have $X_i = W_{l_f}^f$.

A possible choice of the tree model is the coalescent model. We refer the reader to Degnan and Salter (2005) for details on coalescent models. We assume the trait of interest evolves along the coalescent tree under the Ornstein-Uhkenbeck (OU) process. A univariate OU process $W_t, 0 \leq t \leq h$ is characterized by an optimal value θ. Its stochastic differential equation can be written as :

$$dW_t = -\alpha(W_t - \theta)dt + d\epsilon_t$$

where α is the selection strength. The Brownian motion $\epsilon \sim \mathcal{N}(0, \sigma^2)$ where σ^2 is the variance term. Note that when $\alpha = 0$, the Ornstein-Uhlenbeck process is equivalent to the Brownian motion.

If W_0 is known and fixed, it can be shown that

$$\mathrm{E}(W_t) = W_0 e^{-\alpha t} + \theta(1 - e^{-\alpha t})$$

and

$$\mathrm{Cov}(W_t, W_s) = \frac{1}{2\alpha}\sigma^2 \left[e^{-\alpha|t-s|} - e^{-\alpha(t+s)}\right].$$

At leaf i in the ultrametric case, we have X_i denote the OU process with constant length branch $l_f = h$. The distribution of X_i conditionally to the parent node $X_{pa(i)}$ can be written as:

$$X_i | X_{pa(i)} \sim \mathcal{N}\left(X_{pa(i)} e^{-\alpha h} + \theta_i (1 - e^{-\alpha h}), \frac{1}{2\alpha} \sigma^2 (1 - e^{-2\alpha h}) \right).$$

By integrating the distance constraint on the tree, the covariance between two leaves can be written as:

$$\Sigma_{i,j} = \frac{\sigma^2}{2\alpha} (e^{-\alpha d_{ij}} - e^{-2\alpha h}).$$

We refer the reader to Bichat et al. (2020) and Bastide et al. (2017) for details and proofs of the derivations.

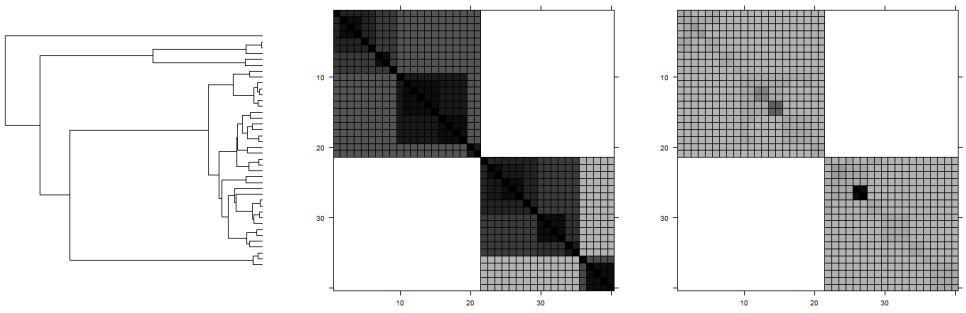


Figure 3.13: Tree, Covariance and precision matrices of the phylogeny based hierarchical model

The simulation procedure follows: we first simulate a coalescent tree. Then with an OU process on its branches, we compute the covariance at leaves. This covariance matrix is finally used to simulate a multivariate Gaussian dataset if its inverse exists. Figure 3.13 shows the covariance and inverse covariance matrices derived with the OU process for a coalescent tree model. The benefit of this simulation scheme for the performance evaluation is that the ground truth tree can be cut into any number of clusters to match the number of estimated clusters by the compared methods.

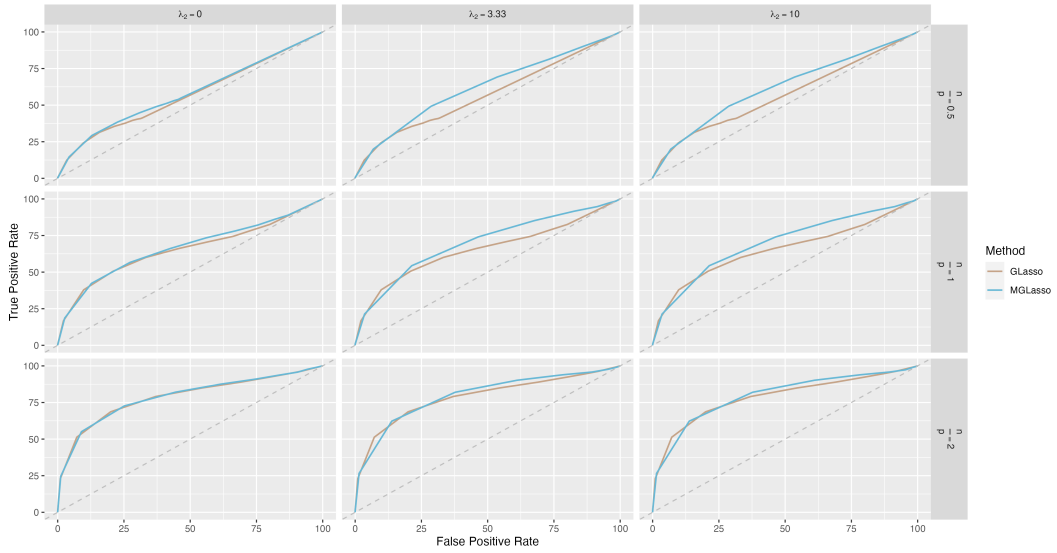
3.4.2 Support recovery

In a small simulation study, we compare our approach to the GLASSO in its neighborhood selection version. The goal is to measure the accuracy of the compared methods for edge detection in sparse graphs using ROC curves. There are 3 scenarios with a number of variables fixed to 40 and a varying ratio of sample size over number of variables $\frac{n}{p} \in \{0.5, 1, 2\}$. The data were generated from Gaussian distributions using the models described in Section 3.4.1.1. The stopping

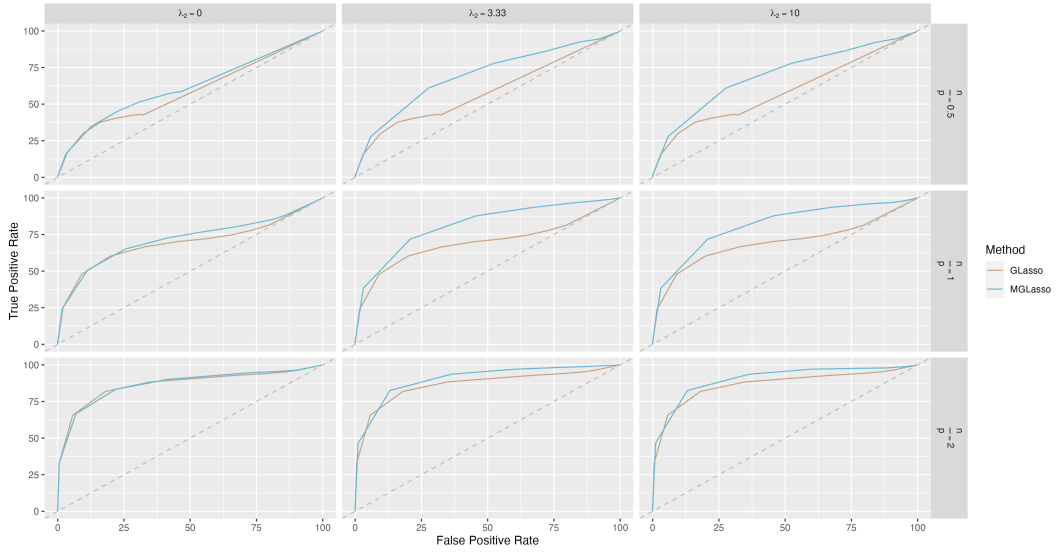
tolerance for the MGLASSO method was chosen to be 0.001. As mentioned earlier in Section 3.2.2, the MGLASSO is coded in both R and Python, and the GLASSO is performed using the **huge** package in R. For the approaches to be comparable, LASSO penalties used for MGLASSO, need to be rescaled by a factor of n in **huge** implementation. Timings results are not computed. However, GLASSO is much faster than MGLASSO in all the described settings.

In GLASSO and MGLASSO, the sparsity is controlled by a regularization parameter λ_1, and they are selected through the StARS approach. Moreover, MGLASSO admits an additional regularization parameter, λ_2, which controls the strength of the convex clustering. To compare the two methods, in each ROC curve, we vary the parameter λ_1 while the parameter λ_2 (for MGLASSO) is kept constant. We have chosen to compare ROC curves for different fusion penalty parameters instead of the results for a particular value of λ_2 to highlight the fusion penalty effect slightly. The GLASSO method does not aim at proposing clusters of variables and focuses on the graph inference task. For this reason, we did not derive clusters from its estimated graphs, as the approach chosen for clustering might be subject to extensive discussion. We computed ROC curves for 3 different penalty levels for the λ_2 parameter. Since GLASSO does not depend on λ_2, the GLASSO ROC curves are replicated.

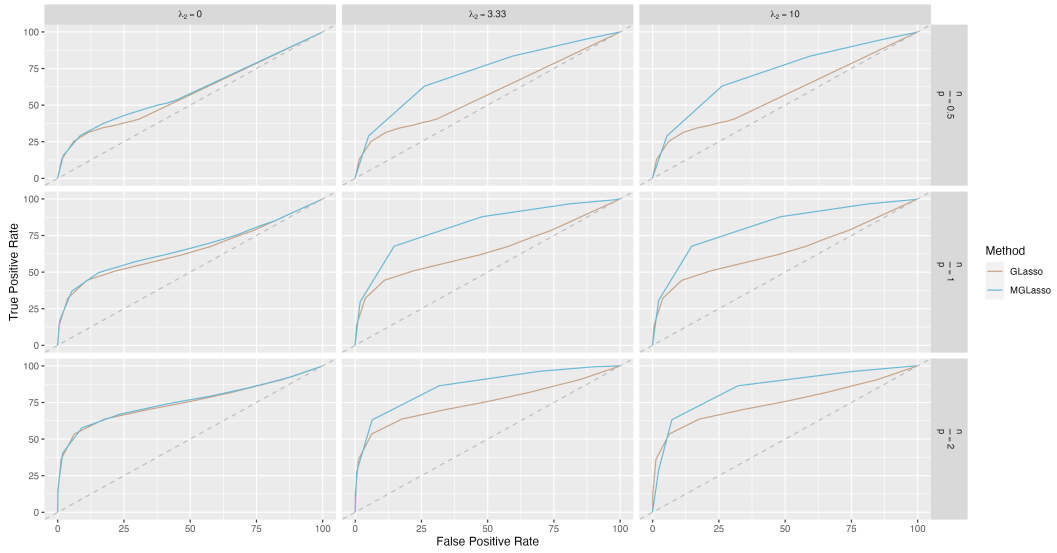
In a decision rule associated with a sparsity penalty level λ_1, we recall the definition of the two following functions. The true positive rate is given by $\frac{TP(\lambda_1)}{TP(\lambda_1)+FN(\lambda_1)}$. The false positive rate is defined as follows $1 - \frac{TN(\lambda_1)}{TN(\lambda_1)+FP(\lambda_1)}$, where TP is the number of true positives, TN the number of true negatives, FN the number of false negatives and FP the number of false positives. The ROC curve represents the true positive rate as a function of the false positive rate. For a given level of true positive rate, the best method minimizes the false positive rate.



(a) Mean ROC curves for MGLASSO and GLASSO graph inference in the Erdos-Rényi model.



(b) Mean ROC curves for MGLASSO and GLASSO graph inference in the scale-free model.



(c) Mean ROC curves for MGLASSO and GLASSO graph inference in the stochastic block model.

Figure 3.14: We varied the fusion penalty parameter of MGLASSO $\lambda_2 \in \{0, 3.33, 10\}$ alongside the ratio $\frac{n}{p} \in \{0.5, 1, 2\}$. Within each panel, the ROC curve shows the True positive rate (y-axis) vs. the False positive rate (x-axis) for both MGLASSO (blue) and GLASSO (brown). Since GLASSO does not have a fusion penalty, its ROC curves were replicated for panels belonging to the same row. We also plot the random classifier (dotted grey line). The results have been averaged over 50 simulated datasets and suggest that MGLASSO performs no worse than GLASSO. For $\lambda_2 = 0$, the MGLASSO approach is equivalent to GLASSO in its neighborhood selection version.

Figures 3.14a, 3.14b, 3.14c show the average values of ROC curves for MGLASSO and GLASSO for different configurations as averaged over 50 simulations. Based on these empirical results, we first observe that, in all the considered simulation models, MGLASSO improves over GLASSO in support recovery in the high-dimensional setting where $p < n$. In addition, in the absence of a fusion penalty, i.e., $\lambda_2 = 0$, MGLASSO performs no worse than GLASSO in each one of the 3 models. This configuration corresponds to the baseline and thus suggests that the CONESTA algorithm achieves similar results as the **huge** algorithm (Zhao et al., 2012) for GLASSO neighborhood selection with no added fusion term. The entire regularization path of the MGLASSO according to the fusion penalty is not presented, and differences between the non-zeros fusion penalty parameter for MGLASSO are minimal. Indeed, in a multiple configuration analysis, the grid of penalty values may not induces the same amount of fusion according to the size of the simulated dataset. A finer grid can be used for each configuration taken independently to handle this problem; however, it would be prohibitively expensive to compute. Note

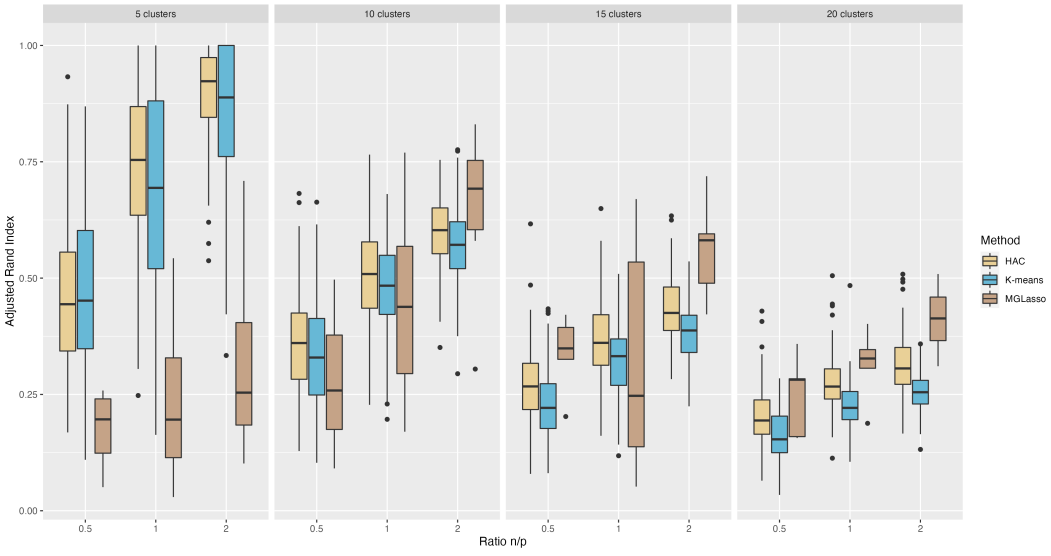
that a shrinkage effect might be observed in the regression vectors' estimates as the λ_2 penalty parameter increases. This shrinkage effect of group-fused penalty terms was also observed in Chu et al. (2021). Moreover, the MGLASSO may fail to recover the edges in a stochastic block model framework where the inter-clusters edge connection probability is high.

3.4.3 Clustering

In this section we evaluate the performance of MGLASSO and compare with existing methods in the stochastic-block model and the hierarchically structured covariance matrix model.

We first compare the partitions estimated by MGLASSO, Hierarchical Agglomerative Clustering (HAC) with Ward's distance and k-means to the true partition in a stochastic block model framework. Euclidean distances between variables are used for HAC and k-means. The criterion used for the comparison is the adjusted Rand index (ARI). A larger value of ARI is an indicator of good clusters recovery.

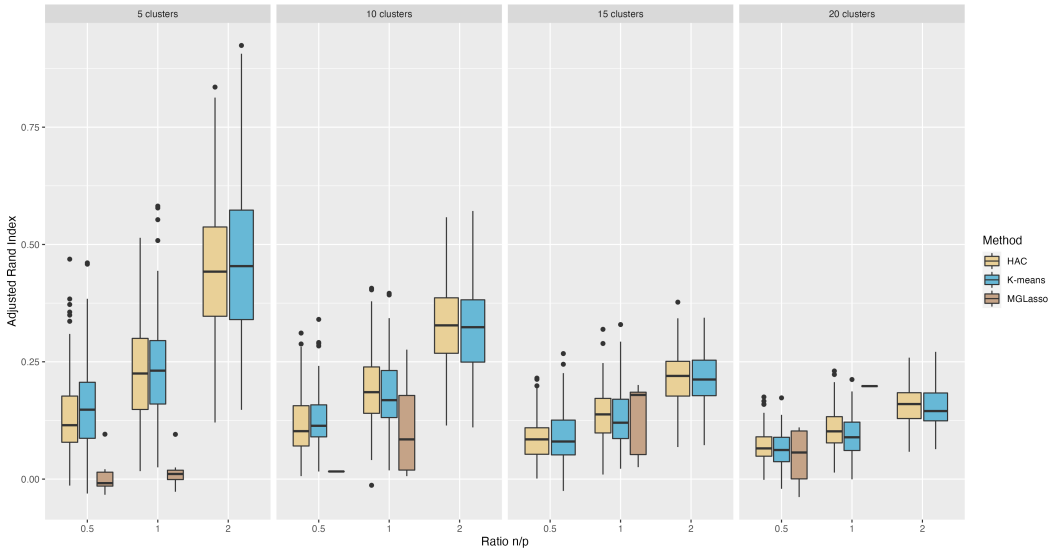
For the HAC and k-means, we vary the number of clusters from 2 to $p = 40$. For the MGLASSO, the fusion penalty controls the number of clusters. The partitions infered by MGLASSO do not necessarily cover the entire range of possible number of clusters as HAC and k-means. According to the strength of the fusion, multiple variables may merge together. The estimated partitions by the 3 methods are then compared to the oracle partition used in the simulation of the stochastic-block model. Note that it remains possible to compare partitions with different number of clusters with the ARI.



(a) Correlation level $\rho = 0.1$.

We study the influence of the correlation level inside clusters on the clustering performances through two different parameters: $\rho \in \{0.1, 0.3\}$; the vector of

cluster proportions is fixed at $\pi = (1/5, \ldots, 1/5)$. Hundred Gaussian data sets were then simulated for each configuration (ρ, n/p fixed).The optimal sparsity penalty for MGLASSO was chosen by the Stability Approach to Regularization Selection (StARS) method (Liu et al., 2010). The parameter λ_2 has been varied.



(b) Correlation level $\rho = 0.3$.

Figure 3.15: Boxplots of Rand index for HAC, k-means and MGLASSO in the stochastic-block model. The number of estimated clusters $\{5, 10, 15, 20\}$ vary alongside the ratio $\frac{n}{p} \in \{0.5, 1, 2\}$. Within each panel, the boxplots of ARI between true partition (with 5 classes) and estimated clustering partitions on 100 simulated datasets for k-means (blue), hierarchical agglomerative clustering (yellow), and MGLASSO (brown) methods are plotted against the ratio $\frac{n}{p}$. The cluster assignments of MGLASSO are computed from a distance between estimated regression vectors for a given value of λ_2. Missing boxplots for MGLASSO thus mean computed partitions in the grid of values of λ_2 do not yield the fixed number of clusters. The higher the ARI values, the better the estimated clustering partition is.

The results shown in Figures 3.15b and 3.15a suggest that, particularly at the lower to medium levels of the hierarchy (between 20 and 10 clusters), the hierarchical clustering structure recovered by MGLASSO is comparable to popular clustering methods used in practice. For the higher levels (5 clusters), the performances of MGLASSO deteriorate. As expected, the three compared methods also underperform as the level of correlation inside clusters decreases. Note that the MGLASSO performances may be sensible to fusion threshold ϵ_{fuse} set to 10^{-6} for this simulation study. Using non-trivial weights could also help improve the overall performance.

The expected empirical evidence according to which the MGLASSO would work fairly well for correlated variables is somehow highlighted with Figures 3.15b - 3.15a. However, the result suggests that controling the correlation between groups of predictors used for each node wise regression in the simulation model, is not an easy task in a multitask learning framework where the set of predictors is the same as the set of responses.

As also noticed in Wang et al. (2018), some preliminary tests suggest that the MGLASSO clusters recovery performance is robust to the Lasso penalty parameter used. Higher values for the penalty contributes, however, to increase the strength of the fusion, i.e., the number of variables merging simultaneously. Nevertheless, the performances of the MGLASSO get significantly improved in the hierarchically-structured model described in the next paragraph.

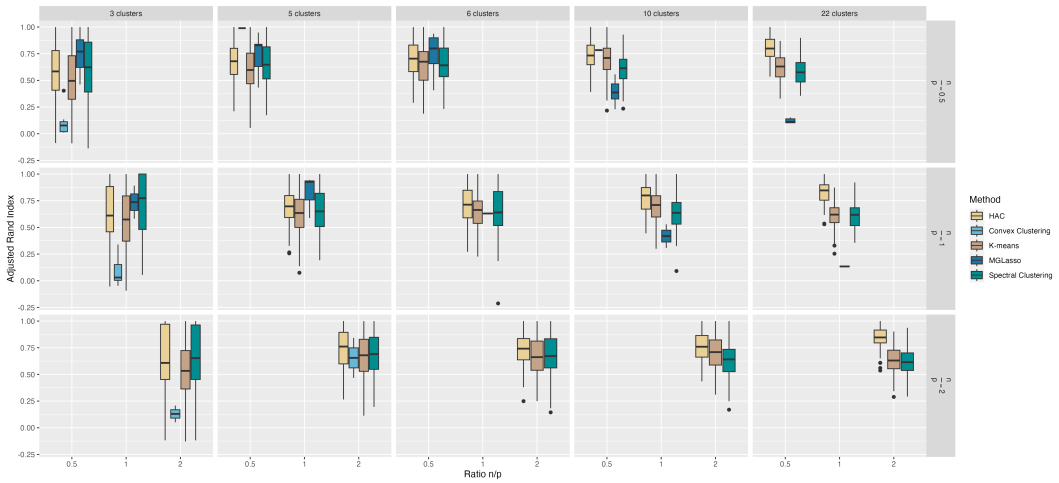


Figure 3.16: Comparisons of adjusted Rand indices for MGLASSO and some clustering methods in the hierarchically structured model. The Rand index is computed between the estimated partition in k clusters and the partition obtained after cutting the tree used for simulation in k clusters. The boxplots of the indices distribution are provided for different ratios n/p. The compared approaches are the HAC, k-means, spectral clustering and vanilla convex clustering. Missing boxplots for MGLASSO or convex clustering mean computed partitions do not yield the fixed number of clusters.

Figure 3.16 compares the performances of MGLASSO, HAC, k-means, spectral clustering, and convex clustering under different n/p ratios for different numbers of clusters using the hierarchically structured covariance matrix simulation framework. The significant difference with the previous simulation framework is that the oracle partition number is changed via cutting the tree used for simulation, according to the number of estimated clusters by the compared methods.

The convex approaches (MGLASSO, classical convex clustering) do not share the same number of boxplot points as HAC, k-means, and spectral clustering. In

a multiple simulation framework, it appears difficult to control or select a desired number of clusters for these approaches. Each boxplot represents the distribution of 130 simulations for HAC, k-means, and spectral clustering, and at least 1 matching simulation for MGLASSO and convex clustering if the results are available. The MGLASSO performance decreases with the number of clusters. So it is easier to find pertinent clusters at higher levels of the tree than at lower levels. The higher-level tree performances for MGLASSO are relatively better than the other clustering approaches results. For the classical convex clustering with unitary weights, clusters tend to fuse abruptly, comparatively to MGLASSO. Intermediate partitions with a number of clusters between 1 and 40 are harder to be found.

Another GGM inference method that is based on the theory of convex clustering is the Clustered Gaussian graphical model developed by Yao and Allen (2019). This approach would be a potential candidate for the evaluation of the performance simultaneously in terms of support recovery and clustering results. However, the approach does not lead to sparse graphs after some preliminary tests. So it would be unfair to compare it to the MGLASSO. An alternative comparison method with a graph and clustering method would be to apply a two-step approach where the graph is first estimated, and community detection or classical clustering algorithms are then used to infer clusters. However, proceeding in such a scheme diverts from the initial objective of the MGLASSO, which is the simultaneous clustering and graph inference. Nonetheless, a not yet published approach that attempts to achieve the simultaneous task of graph inference and cluster estimation appears to be the work of Lin et al. (2020). However, an explicit rule for cluster deduction is not proposed in the paper, nor an explicit link with the theory of convex clustering is established, which leaves it impractical to make an eventual comparison with the MGLASSO.

3.5 Inference of microbial networks via MGLASSO

This section presents a second application of the MGLASSO model on real data, particularly microbial abundance data that was collected as part of the American gut project (McDonald et al., 2018). The use of neighborhood selection in sparse microbial network inference has already been reported in works like Kurtz et al. (2015). We show here how the MGLASSO model, also belonging to neighborhood selection approaches, can be used for the problem while drawing a parallel with the cited work. Human gut microbes play a critical role in homeostasis, disease, and digestion. Knowledge of their interactions can serve as an initial community-level description of the underlying microbial ecosystem (Yoon et al., 2019). We perform multi-scale network clustering and inference analysis in which we focus on deriving representative variables for microbial species, unlike the previous application of MGLASSO, in which networks were drawn across the whole set of variables in the model. The material of the analysis is described in Section 3.5.1. We briefly

reintroduce the inference approach proposed by Kurtz et al. (2015) in Section 3.5.2. MGLASSO learning is presented in Section 3.5.3. We show some results in Section 3.5.4.

3.5.1 Material

We used the microbial abundance dataset called `amgut1.filt` included in the **SpiecEasi** package (Kurtz et al., 2015). It describes the microbial community structure of $n = 289$ samples via an abundance table for $p = 127$ types of microbes (operational taxonomic units). The data has been pre-processed by Kurtz et al. (2015) and is a subset acquired from the first round of the American gut project (AGP, McDonald et al. (2018)). Samples of feces, skin and other body regions were taken from thousands of participants and profiled using 16S rRNA sequencing. The project aimed to study the associations between the human microbiome and factors such as diet.

3.5.2 SpiecEasi method

We briefly recall the SpiecEasi method for the inference of sparse microbial association networks and refer to the seminal article for more details. The technique essentially combines transformation for relative abundance data with a neighborhood selection approach for Gaussian graphical inference (Meinshausen and Bühlmann, 2006) coupled with the StARS scheme for model selection (Liu et al., 2010). The transformation applied to the data is the centered log ratio and is presented below.

For $j = 1, \dots, n$, let $\mathbf{Y}_j = (Y_{j1}, \dots, Y_{jp}) \in \mathbb{N}^p$ be the p-dimensional row-vector of number of OTUs observed from the j-th sample with an associated relative abundance vector $\widetilde{\mathbf{Y}}_j = \frac{\mathbf{Y}_j}{\sum_{i=1}^p Y_{ij}}$. Note that $\{\widetilde{\mathbf{Y}}_j\}_{j=1,\dots,p}$ belongs to the p-dimensional *unit simplex* :

$$\mathbb{S}^p = \left\{ \mathbf{x} = (x_1, \dots, x_p) \in \mathbb{R}^p \;\middle|\; \sum_{i=1}^p x_i = 1, \mathbf{x} \geq 0 \right\}. \tag{3.17}$$

Kurtz et al. (2015) then applied the centered log ratio transformation (clr, Aitchison (1982)) to the relative abundances:

$$\operatorname{clr}\left(\widetilde{\mathbf{Y}}_j\right) = \log\left(\frac{\widetilde{\mathbf{Y}}_j}{\left(\prod_{i=1}^p \widetilde{Y}_{ij}\right)^{1/p}} \right), \text{for all} \quad j = 1, \dots, p. \tag{3.18}$$

where $\left(\prod_{i=1}^p \widetilde{Y}_{ij}\right)^{1/p}$ is the geometric mean of the relative abundance vector. A pseudo-count is added to $\{\mathbf{Y}_j\}_{j=1,\dots,n}$ to avoid numerical problems due to zero counts.

3.5.3 MGLASSO learning

We applied the MGLASSO method introduced in Chapter 3, on the OTUs previously transformed by the clr approach. The MGLASSO with a value of $\lambda_2 = 0$ is therefore equivalent to the SpiecEasi method.

For the model fitting, the LASSO penalty parameter λ_1 is selected via the StARS approach with a variability threshold set at 0.05 in a grid of values defined according to the same rules as in Section 4.4.3. The fusion penalty λ_2 is chosen in the interval $[0, 20]$ with irregular steps. This was done in an attempt to control abrupt merges in the clustering path and to reduce the number of values to be evaluated due to computation time. We chose 20 equidistant values in each of the following intervals: $[0.20]$, $[1.20]$ and $[0.4]$. The CONESTA solver precision is set to $\epsilon = 0.01$.

To represent the graphs estimated on the grid of penalization parameters λ_2, we first selected the representative variables by clusters to contrast with the representation used in the section 4.4 where the matrices of adjacency on the whole set of nodes have been presented. The representative variable is one of the cluster's variables—the fusion principle. Indeed, theoretically, variables that belong to the same cluster share the same neighborhood. To compute clustering partitions, the fusion threshold was set to $\epsilon_{fuse} = 0.001$.

3.5.4 Results

Using the parameters above, we compute a clustering path of MGLASSO solutions and then display the estimated networks for specific values of the λ_2 fusion penalty.

Figure 3.17 shows how the predicted data computed from the estimated regression vectors scales on the λ_2 fusion penalty parameter grid. The variables were not clearly separated on the first two principal components. The results are therefore displayed on the components 3 and 4. The path is not always agglomerative and shows several cases of abrupt merges. The OTUs have been colored according to the biological clusters, here, the taxonomic classifiers corresponding to the phylum level (Rank 2). The taxonomy table is loaded from the `amgut1.filt.phy` dataset available in the **SpiecEasi** package. The 27 OTUs belonging to the phylum Bacteroidetes form a pure cluster that MGLASSO successfully identifies. The 20 phylum Proteobacteria has been divided into three subgroups. The Firmicutes phylum of 76 OTUs is split into two main clusters. The phyla Actinobacteria, Tenericutes, and Verrucomicrobia, which respectively count 2, 1, and 1 OTU(s), are less predominant in the clustering. Figure 3.18 displays networks and clusters for different levels of granularity corresponding to the values of the λ_2 fusion penalty parameter. The networks are built on representative variables of the computed clustering partition. The first graph with $\lambda_2 = 0$ displayed in Figure 3.18a corresponds to the SpiecEasi network of 175 edges. Figures 3.18b, 3.18c and 3.18d show intermediate graphs for the following (number of clusters, number of edges) pairs : (63 clusters, 548 edge), (31 clusters, 240 edges) and (15 clusters, 91 edges) respectively. Most networks are dense. The last network is represented on 2 representative variables for a partition in 2 clusters (Figure 3.18e). The nodes are colored according to their phylum taxonomic classifier. It is important to note that not all clusters are pure. Labeling a representative variable with a given phylum

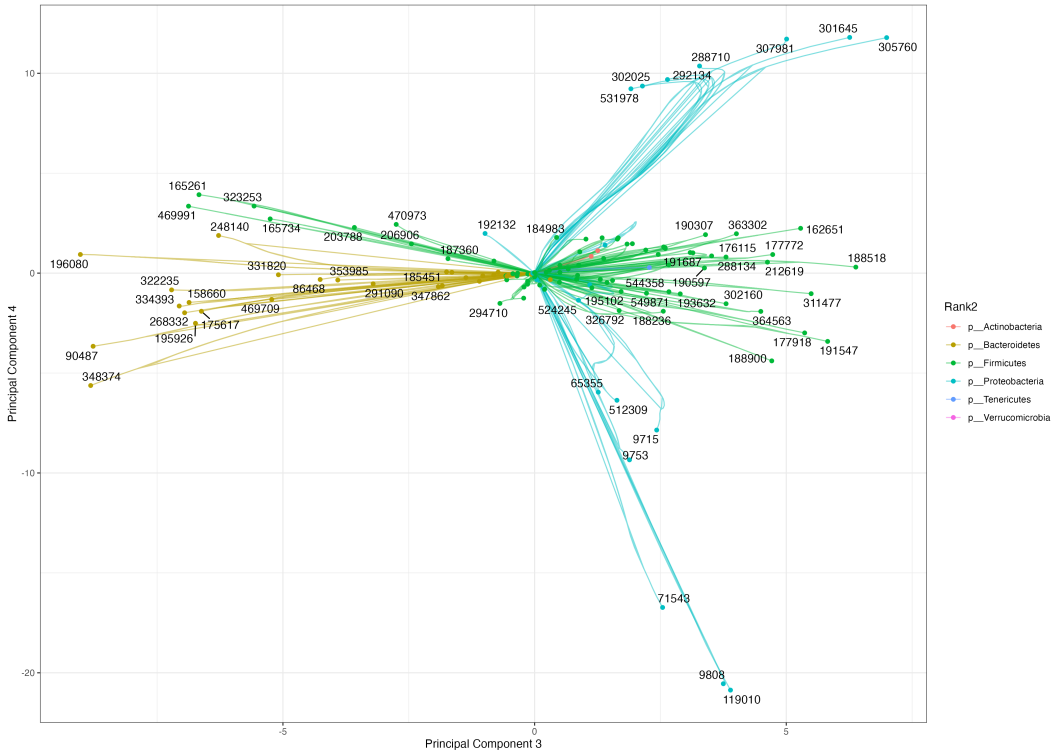
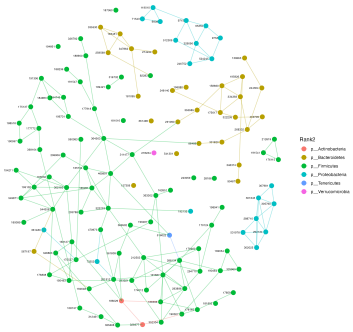
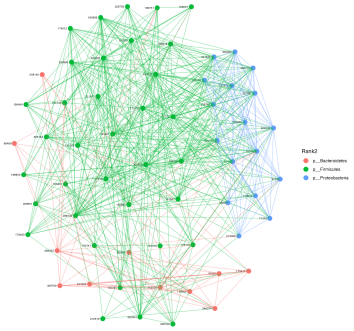


Figure 3.17: Clustering path of MGLasso solutions on human microbiome data composed of 127 operational taxonomic units. OTUs are colored according to their phylum classification. The path displays abrupt merges. The pure cluster on the graph's left side (down) corresponds to the phylum Bacteroidetes.

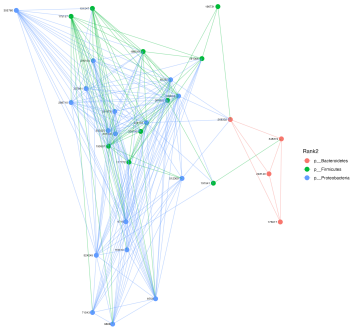
does not necessarily mean that all the OTUs in the cluster belong to the phylum concerned.



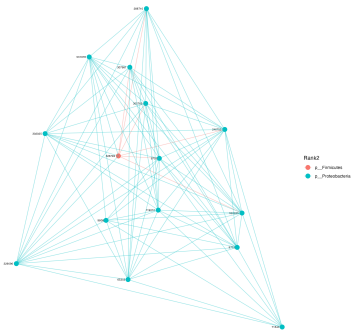
(a) $\lambda_2 = 0$



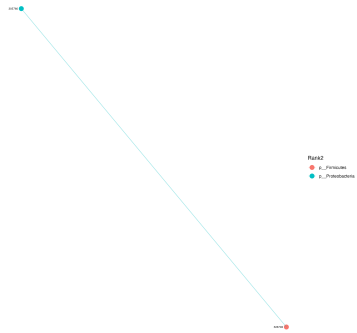
(b) 63 clusters



(c) 31 clusters



(d) 15 clusters



(e) 2 clusters

Figure 3.18: Graphs estimated at multiple levels of granularity for the microbiome data. The first graph, showing a network inferred by the MGLASSO method when $\lambda_2 = 0$, corresponds to the SpiecEasi network. Increasing the fusion penalty makes it possible to uncover graphs built on the representative variable of each cluster. OTUs are colored according to their phylum taxonomic classifier.

3.5.5 Discussion

The analysis presented above was exploratory and aimed not only to illustrate the MGLASSO model but also to show its connection to the SpiecEasi approach when applied in a microbial network inference framework. It has the advantage of proposing multi-scale networks with clustering of OTUs, which makes it possible to find interesting links between OTUs that do not necessarily belong to known taxonomic classifications. The partition into 2 groups found in data from the American gut project suggests a relationship between a group composed exclusively of OTUs belonging to the phylum Proteobacteria and the other phyla represented by the phylum Firmicutes. Setting weights for the fusion penalty can improve MGLASSO's clustering path with less abrupt fusions.

4 - Applications on omics data

Contents

This chapter focuses on applications of the MGLASSO model and other statistical models on omics data. In Section 4.3, various biological questions are answered, in particular within the framework of the EPITREE project, which aims to study the evolutionary and functional impact of epigenetic variation in forest trees. Note that we have carried out other analyses within the project that are not reported in this manuscript. In Section 4.4, MGLASSO is illustrated in an integrative analysis of transcriptomic and DNA methylation data from the EPITREE project. Section 3.5 presents another example of applying MGLASSO to the inference of sparse microbial networks.

4.1 Elements of omics data

Omics is a field of study in biology that involves the comprehensive characterization and quantification of molecules, such as genes, proteins, and metabolites, within a biological system. This can include genomics (study of genes and genomes), proteomics (study of proteins), metabolomics (study of metabolites), and other subfields. The goal of omics is to understand the interactions and relationships between these molecules in order to gain a better understanding of the overall function and regulation of biological systems. The omics data are generated from high-throughput sequencing technologies, also known as next-generation sequencing (NGS). NGS opposed to first-generation technologies, are new sequencing approaches that read genomes at a higher speed and a relatively cheaper cost. In this section, we review some concepts of biology and some omics data type that will help understand the remainder of the chapter.

4.1.1 Genomics

The deoxyribonucleic acid of an organism (DNA) is a helix molecule of two-paired strands that contains information about its functioning, development, reproduction, and growth. It is structured into chromosomes found in the cell's nucleus. The DNA sequence is an oriented sequence made up of four character states, also known as bases: adenine (A), thymine (T), cytosine (C), and guanine (G). In the paired structure, C is linked to G, and A is linked to T to form a base pair (bp). Some DNA regions (coding regions) can be transcribed into messenger ribonucleic acids (mRNA). The mRNAs can, in turn, be translated into proteins (chains of amino acids) which are involved in various cellular processes – the central dogma of molecular biology (see Figure 4.1).

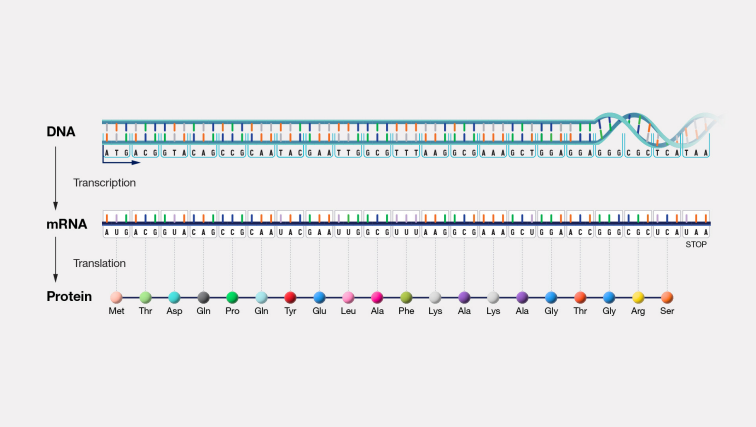


Figure 4.1: Central dogma of molecular biology (source: National Human Genome Research Institute).

The genome is the complete DNA sequence of an organism. Its size differs between species. For example, the poplar (*Populus trichocarpa*) genome has 19

chromosomes for about 0.485 billion base-pairs while the human (*Homo sapiens*) genome has 42 chromosomes for approximately 3 billion base-pairs. The genomic sequence can be segmented into genes and intergenic regions. Genes can, in turn, be segmented into units called exons and introns. The exons correspond to coding regions in the gene, while introns are the non-coding regions. The expression of genes can be measured as the number of RNA transcripts produced by the gene.

4.1.2 Transcriptomics

Transcriptomics studies the ribonucleic acid (RNA) available in a cell or a set of cells. The transcriptome is the complete set of RNA transcripts, some coding (messenger) and others not (e.g., ribosomal, transfer, or small nuclear). De novo RNA-sequencing (RNA-Seq) is a breakthrough technology in biology that easily captures any RNA and, therefore, measures genes' expression. It refers to a method of transcriptome assembly where a reference genome is not used to align the RNA-seq reads. Instead, the reads are assembled into contigs and then into longer transcript sequences (isoforms) without the use of a reference genome. De novo assembly is useful for organisms for which a reference genome is not yet available or for organisms that have a high degree of genetic diversity. It can also be used to identify novel transcripts and isoforms, which can provide important information about the regulation of gene expression. The de novo RNA-seq pipeline includes several steps: RNA isolation, quality control, RNA sequencing, assembly and annotation. This approach is computationally intensive and requires large amounts of computational resources and memory, specially when dealing with large or complex transcriptomes.

In the RNA-Seq experiment, the total numbers of mapped reads, i.e., sequencing depths, are different for samples (see Figure 4.2), and more reads would map to longer genomic regions. In order to have comparable counts between these samples, RNA-Seq data are usually normalized via diverse approaches, which might take into account other within and between samples non-uniformities (Soneson and Delorenzi, 2013).

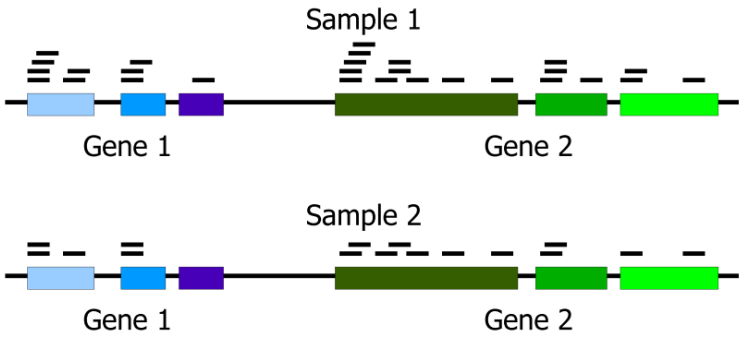


Figure 4.2: Genes in two samples with different numbers of reads (source: (Rau, 2017))

4.1.3 Genetics

Population genetics is the study of the variations in DNA sequence between individuals in a population, while quantitative genetics focus on individuals' traits and their heritability (Harmon, 2019). The genotype of an organism is its heritable genetic identity.

Multiple variants exist in the genomes of a population of a given species and can be regrouped into 2 significant categories, which are the single nucleotides variations and the structural variations which extend to multiple nucleotides (Manzoni et al., 2018). The SNVs are also known as singles nucleotide polymorphisms (SNPs, Figure 4.3) when common in a population (frequency higher than 1%). For most SNPs, two variants are only observed and called alleles (Neuvial et al., 2011). Since organisms inherit one copy of each SNP position from each parent, the organism's genotype at an SNP position is either AA, AB or BB with A and B the two alleles (LaFramboise, 2009). The minor allele frequency (MAF) is the proportion of the less common allele in a population (with a proportion lower than 0.5). Thus, an SNP is characterized by its genomic position, alleles, and MAF.

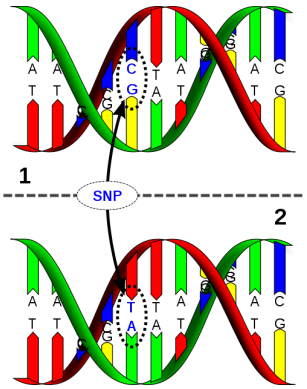


Figure 4.3: Single nucleotide polymorphism (source: David Hall)

The SNPs' genetic variants can be captured by whole genome sequencing based on NGS technologies.

4.1.4 Epigenetics

Epigenetics studies heritable changes that affect gene expression without changing the DNA sequence (Plomion et al., 2016). The epigenome is the pattern of these changes that mark the genome. For an organism, the genome is usually stable across the cells. However, the epigenome is highly dynamic across cell types and in time (Robinson et al., 2014a). Its profiling is thus more complex than profiling the genome.

DNA methylation is a commonly studied epigenetic mark. It consists of the addition of a methyl group to a cytosine position. Methylation occurs predominantly on CpG(CG) dinucleotides. Non-CpG methylation sites include CHG and CHH contexts where H is a DNA base different from the guanine.

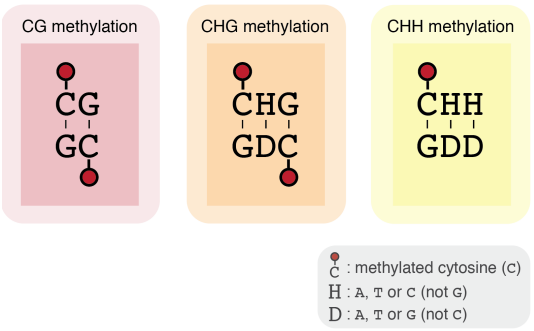


Figure 4.4: Different methylation contexts (image readapted from Colette Picard and Robert Erdmann)

The gold-standard method for methylation analysis is whole genome bisulfite sequencing (WGBS). The WGBS uses the sodium bisulfite treatment to convert unmethylated cytosines to thymine (via uracil) while the methylated cytosines remain untouched (Wreczycka et al., 2017). This allows quantifying the proportion of methylated cytosines over a number of reads (read coverage). When interested in the sequencing of a specific region, approaches such as methyl capture sequencing (MC-Seq, Teh et al. (2016)) can be used. They are cheaper than WGBS and need the prior definition of probes to survey the genomic loci.

4.1.5 Microbiomics

Microbiomics studies the microbiome, which is the collection of microorganisms (such as viruses and bacteria) that live in a particular environment[1]. Therefore, the metagenome is the total genomic DNA of the microorganisms, and the metatranscriptome is their total transcribed RNA (Morgan and Huttenhower, 2012). Microbial communities differ from multicellular organisms such as poplars or humans, which have a unique genomic signature. They do not necessarily carry identical genomes, and it can be impractical to sequence each cell's genome entirely. Some molecular markers (DNA sequences), such as the 16S ribosomal RNA gene, have thus been defined in microbial ecology, which can uniquely identify distinct genomes.

Shotgun metagenomics and 16S metagenomics are two different methods used to study the microbial diversity in a sample. Shotgun metagenomics involves sequencing all the DNA in a sample, regardless of its origin. The DNA is then broken down into smaller fragments, which are sequenced in parallel. The resulting sequences are then assembled into contigs and then into larger sequences called scaffolds. Shotgun metagenomics provides a comprehensive overview of the microbial diversity in a sample, including both known and unknown species. It also allows for the identification of functional genes and metabolic pathways present in the sample. 16S metagenomics, on the other hand, focuses specifically on the

[1] https://www.genome.gov/genetics-glossary/Microbiome

16S rRNA gene, which is a conserved gene found in all bacteria and archaea. The 16S rRNA gene is used as a marker to identify and classify different bacterial and archaeal species. This method is typically used to study the composition and diversity of bacterial and archaeal communities in a sample, but it does not provide information about the functional genes or metabolic pathways present in the sample. In summary, shotgun metagenomics is a more comprehensive approach that allows for the identification of both known and unknown species and functional genes, while 16S metagenomics is more focused and is used specifically to identify and classify different bacterial and archaeal species.

The operational taxonomic unit (OTU) is a cluster of microorganisms with a similar marker gene sequence beyond a fixed threshold. It replaces species because named species genomes are not always available for specific marker genes. A typical microbiome dataset is the count of sequences per OTUs for a given number of samples. In order to have comparable OTUs between samples, relative counts are usually computed, which gives rise to compositional data.

When interested in identifying SNPs in the microbial sequences, whole DNA sequencing, also known as whole metagenome shotgun (WMS), can be conducted. Then, the sequenced genome can be compared to reference genomes.

4.2 The EPITREE Project

The evolutionary and functional impact of epigenetic variation in forest trees (EPITREE, ANR-17-CE32) is a forest research project that focuses on how genetic and epigenetic variations contribute to phenotypic plasticity and adaptation to the local environment. EPITREE was born from a need to understand the mechanisms underlying forest tree adaptation to manage genetic resources better. Indeed, over the last few years, widespread forest die-off has been observed due to drought constraints. These trees play an essential role in providing ecosystem balance on earth. Two tree species are studied in the project, namely the poplar and oak.

The project is one of some, interested in the genetic bases of trees' local adaptation. However, the existing studies usually focus on the contribution of SNPs. Epigenetic mechanisms are not studied in depth. The dynamic nature of the epigenome makes it an interesting subject to study in these long-lived organisms. EPITREE focuses on studying the variations of DNA methylation, gene expression, and genome structural variations for a better understanding of the contribution of epigenetic variations in local adaptation and phenotypic plasticity. The project is subsetted in multiple work packages which center around the identification of epigenomic candidate regions, the characterization of epigenomic variation in natural populations and its functional consequences, the characterization of the epigenomic plasticity and its functional consequences in response to environmental constraints, data generation, and integrative multi-omics analysis.

Understanding the evolutionary pattern of species is essential in biology, espe-

cially for long-lived organisms such as trees. The study of genetic and epigenetic variations of these trees' natural populations can help shed light on their mechanisms of adaptation to their local environment, which is relevant in the current context of climate change. Simultaneous interest in genetic and epigenetic variation is a recently explored area of research focused on annual plants, which are short-lived organisms (Sow et al., 2018). Characterizing these variations for trees remains an open question that can bring added value in forest management (Amaral et al., 2020). One of the objective of the EPITREE project is to provide answers to this concern by producing diversified data on populations of trees located on different geographical sites, focusing on oak and poplar.

In the framework of the PhD research, we contribute to highlight poplar populations' genetic and epigenetic structure through SNP data and DNA methylation data, respectively. We also show how methylation is used as a marker of population differentiation by proposing genes whose methylation pattern closely follows the genetic structure of populations. We briefly study the link between methylation and expression patterns for a specific class of genes whose expression profiles are stable between trees belonging to the same meta-populations. These chosen areas of the analysis result from discussions with experts from the EPITREE project. We also laid the groundwork for further analysis by briefly exploring the structure and interplay between transcriptomic and methylation patterns in different methylation contexts using the MGLASSO model. The study detailed in Section is summarized in the application section on real data of the MGLASSO article (Sanou, 2022).

4.2.1 Data and material

Three datasets of different omic natures were used for the analysis. These are SNP data, DNA methylation data, and gene expression data. We present them in this section and briefly describe some applied preprocessing and transformations. Note that details of plant material, experimental designs, and bioinformatics preprocessing can be found in Sow (2019), Sow et al. (2023), and Chateigner et al. (2020). Twenty genotypes corresponding to ten natural populations (two genotypes per population) of poplars (*Populus nigra*), representative of the diversity of the black poplar collection in Western Europe (see Figure 4.5), were investigated.

4.2.1.1 SNP data

The SNP data contained 9.7 million SNPs across the 20 genotypes and was collected via the whole genome sequencing (WGS) method. An illustration of the dataset is given in Table 4.1. Missing values are ignored and the SNPs are filtered out, so those with a minor allele frequency (MAF) less than 5% are discarded (see Figure 4.6). This step made it possible to select 5.1 million SNPs.

4.2.1.2 Methylation data

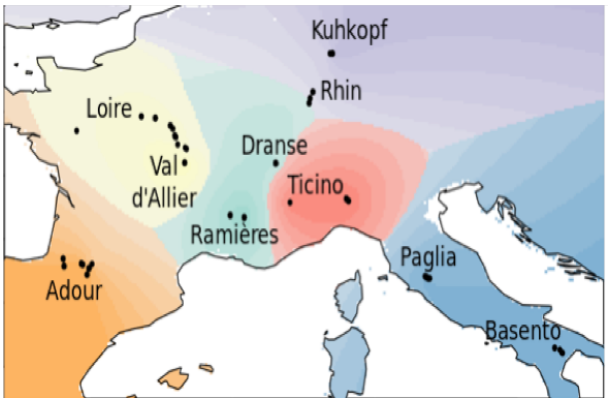


Figure 4.5: Map of Western Europe illustrating natural populations of black poplars. The ten natural populations are partitioned into six ancestral populations whose colors are shown on the map. Black dots indicate sites where populations of *Populus nigra* were present, including sites in France, Germany, and Italy.

	1-A26	1-J31	ALL-014	ALL-019	AST-005	BDX-003
Chr01-314	1	2	1	1	1	1
Chr01-321	1	0	1	1	1	1
Chr01-357	1	0	1	1	1	1
Chr01-363	1	1	1	1	1	1
Chr01-369	1	1	1	1	1	1

Table 4.1: Sample of SNP data: allele count profiles for five genomic positions over six genotypes. The geographical location of the genotypes is as follows: 1-A26, 1-J31: Ramière; ALL-014, ALL-019: Val d'Allier; AST-005, BDX-003: Adour.

DNA methylation levels per cytosine (single methylated polymorphism, SMP) are measured by the whole genome bisulfite sequencing (WGBS) method for all 20 samples. Datasets are available for the three methylation contexts, with 3.53 million SMPs in the CpG context, 10.41 million in the CHG context, and 53.72 million in the CHH context. The DNA methylation data provided are count data, and a sample is shown in Table 4.2.

Methylation proportions per cytosine are computed by taking the ratio between the number of methylated reads and the total number of reads in the sample. Filters based on standard deviation (Akalin, 2020) are applied to select the cytosines that vary the most between the samples. The histogram of the distribution of standard deviations in the CpG context is given in Figure 4.7. We chose an arbitrary threshold of at least 20. After the filtering step and the selection of complete cases, we end up with 260078 SMPs in the CpG context, 92690 SMPs in the CHG context, and 82436 SMPs in the CHH context.

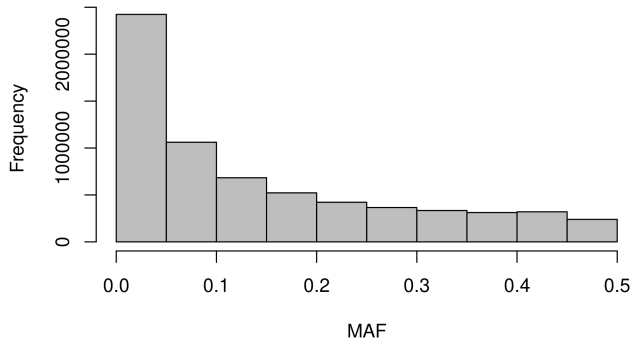


Figure 4.6: Histogram of minor allele frequencies. The SNPs with a minor allele frequency (MAF) less than 5% are discarded from the study.

	chr	start	end	strand	coverage1	numCs1	numTs1
10	Chr01	387	387	+	13	2	11
14	Chr01	416	416	+	21	2	19
18	Chr01	466	466	+	37	6	31
20	Chr01	504	504	+	44	9	35
21	Chr01	506	506	-	25	3	22

Table 4.2: Sample of DNA methylation data in CHG context for five genomic positions: `chr`, `start`, `end` and `strand` locate the genomic position while `coverage1` , `numCs1` and `numTs1` are the number of reads, the number of methylated reads and the number of unmethylated reads, respectively for sample 1 (Ramière).

The proportions of DNA methylation per gene in the gene body and promoter regions are also available. They result from the correspondence between SMPs and genes. A sample of the data is given in Table 4.3. After discarding missing values, 40238, 40318 and 40407 genes methylation levels are available for the CpG, CHG and CHH contexts, respectively.

4.2.1.3 Expression data

The raw genes expression data are count data. They were collected via the RNA sequencing method (RNA-seq) on the genotypes of interest. After being normalized with the trimmed mean of M-values (TMM, Robinson and Oshlack (2010)), the counts per millions (CPM) are computed. A mixed linear model is then fitted on the gene expressions to correct for the effects of cofactors such as date, block, or time of sampling. The best linear unbiased predictors of between and within population random effects are summed to obtain the best linear unbiased prediction for each gene. The process is described in detail in Chateigner et al.

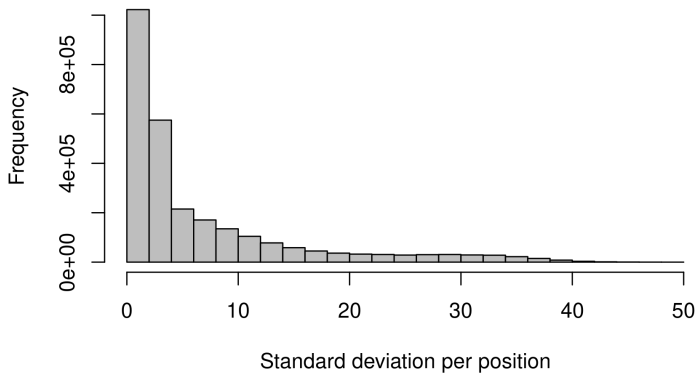


Figure 4.7: Histogram of the distribution of standard deviations of methylation levels per genomic position in the CpG context. Positions with a standard deviation greater than 20 are selected.

	geneid	ratio1.Adour.gbM.CG	ratio2.Adour.gbM.CG
1	Potri.001G000200	0.80	0.74
2	Potri.001G000300	0.70	0.64
3	Potri.001G000400	0.29	0.33
4	Potri.001G000500	0.01	0.01
5	Potri.001G000600	0.01	0.01

Table 4.3: Sample of DNA methylation data in the CpG context for five genes (gene body region). The methylation proportions are given for two genotypes located in the Adour area.

(2020). A sample of the RNA-seq normalized data is given in Table 4.4. The transcriptomic dataset used contains 34229 genes.

	1-A26	1-J31	AST-05	BDX-03
Potri.001G000200.1	-0.01	-0.01	-0.01	-0.02
Potri.001G000400.1	-0.13	0.12	0.07	-0.13
Potri.001G000700.1	-0.08	0.01	-0.02	-0.05
Potri.001G000800.1	0.19	0.46	-0.22	-0.50
Potri.001G000900.1	-0.06	-0.03	0.07	0.06

Table 4.4: Sample of normalized gene expression data for five genes over four genotypes. The data is normalized in two rounds: first by the trimmed mean of M-values and second by a correction for experimental design effects via a mixed linear model (Chateigner et al., 2020). The geographical location of the genotypes is as follows: 1-A26, 1-J31: Ramière; AST-005, BDX-003: Adour.

4.3 Differential analysis and selection of markers

Some material in this section is part of a collaborative article submitted (Sow et al., 2023) to a biology journal. Note that the analysis is narrated here in a language where the emphasis is on the statistical methodologies used, with an attempt to bring together omics data of different natures. We refer the reader to Sow et al. (2023) for a deeper understanding of the biological motivations for this study. We present the data used for the analyses in Section 4.2.1. In Sections 4.3.1, 4.3.2, and 4.3.3, we introduce statistical methods. Section 4.3.4 describes the results.

4.3.1 Differential analysis

4.3.1.1 Binomial regression model for methylation data

The problem addressed here is the definition of a method to identify positions of the genome for which the methylation proportions are significantly different between sample groups. A *treatment* variable or clustering usually determines the groups. The issue of differential analysis for methylation data has been tackled in Robinson et al. (2014b); Wreczycka et al. (2017). Common methods are based on standard statistical tests, regression models, or even hidden Markov Models. Here, the focus is put on a regression-based method. These approaches are the most suited when dealing with multiple groups in the samples and allow adding additional covariates in the model if needed.

Denote $\mathbf{y}_j = (y_{1j}, \ldots, y_{nj})^\top \in]0,1[^n$ the vector encoding the proportion of methylated cytosines at a position j of the DNA, in the n samples. Denote $\mathbf{N}_j \in \mathbb{N}^n$ the vector of reads coverages. The proportion of methylated cytosines equals the ratio between the number of cytosines and the read coverage. Let $X_j \in \{1, \ldots, K\}^n$ be the categorical n-dimensional covariate with $K \geq 2$ categories corresponding to the groups to which the samples belong.

Let's introduce $\tilde{\mathbf{Y}}_j$, the response variable deduced from $\mathbf{y}_j$ corresponding to the number of successes in the N_j trials. We have $\tilde{Y}_{ij}|X_{ij} = x_{ij} \sim \text{Binomial}(N_{ij}, \pi_{ij})$

where $\pi_{ij} = \pi(x_{ij})$. When $N_{ij} = 1$, we return to classical logistic regression for binary variables. We introduce dummy variables to encode the contribution of the categorical variable as follows: $\text{dummy}(x_{ij}) = [\mathbb{1}(x_{ij} = 1), \ldots, \mathbb{1}(x_{ij} = K)]$.

Consider the generalized linear model with the link function $h(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$, with $\pi \in]0, 1[$:

$$\text{logit}(\pi_{ij}) = \log\left\{\frac{\pi_{ij}}{1-\pi_{ij}}\right\} = \sum_{k=1}^{K} \beta_{jk} \mathbb{1}(x_{ij} = k)$$

and $\sum_k \beta_{jk} = 0$.

The log-likelihood is defined as follows:

$$\log L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \{\tilde{y}_{ij} \log(\pi_{ij}) + (N_{ij} - y_{ij}) \log(1 - \pi_{ij})\} \tag{4.1}$$

The model without intercept is fitted via the following negative convex log-likelihood function, except for one constant:

$$\begin{aligned} \underset{\boldsymbol{\beta}_j}{\arg\min} \quad & \sum_{i=1}^{n} \left\{ -\tilde{y}_{ij} \left(\sum_{k=1}^{K} \beta_{jk} \mathbb{1}(x_{ij} = k) \right) + N_{ij} \log\left(1 + \exp\left(\sum_{k=1}^{K} \beta_{jk} \mathbb{1}(x_{ij} = k) \right) \right) \right\} \\ \text{subject to} \quad & \sum_{k} \beta_{jk} = 0. \end{aligned} \tag{4.2}$$

In order to establish whether there is an association between the groups and the response, the deviance of the null model is compared to the deviance of the fitted model $\mathcal{M}$ via:

$$G = 2(\log \hat{L}_{\text{null}} - \log L_{\mathcal{M}}),$$

where $\hat{L}_s$ and $L_{\mathcal{M}}$ are respectively the maximized likelihoods of the null model and the fitted model $\mathcal{M}$.

The alternative hypothesis stating $H_{1j} : \boldsymbol{\beta}_{jk} \neq 0$ for at least one coefficient k was tested against the null hypothesis $H_{0j} : \boldsymbol{\beta}_{jk} = 0$ for all k. The statistic G follows a chi-square probability distribution χ^2 with K degrees of freedom under the null hypothesis. In practice, with a modality of the categorical variable chosen as a reference class, $G \sim \chi^2(K - 1)$. Bonferroni corrections are applied to correct for multiple tests. The differential analysis procedure described is available in the **methylKit** package and named weighted fractional logistic regression.

4.3.1.2 ANOVA model for expression profiles

The problem addressed here is the definition of a method to identify differentially expressed genes from multiple samples. Various tools and methods can be used depending on the nature of the gene expression data (Soneson and Delorenzi, 2013). Regarding counts, methods based on Poisson and Negative Binomial models are the most common. In our case, the data has already been transformed from

counts to continuous data, which eases standard methods based on hypothesis testing.

Differential analysis is conducted for gene expression data by one-way univariate analysis of variance (ANOVA). Let $\mathbf{y}_j = (y_{1j}, \ldots, y_{nj})^\top \in \mathbb{R}^n$ be the vector encoding the normalized gene expression values in the n samples. Let $X_j \in \{1, \ldots, K\}^n$ be the n-dimensional categorical covariate with $K \geq 2$ categories corresponding to the groups to which the samples belong.

Consider the generalized linear model with the identity link function $h(y) = y$, where $y \in \mathbb{R}$:

$$y_{ij} = \sum_{k=1}^{K} \beta_{jk} \mathbb{1}(x_{ij} = k),$$

and $\sum_k \beta_{jk} = 0$ for identifiability constraints.

The least squares estimation procedure can be used to estimate the model coefficients, which are the solution of:

$$\begin{aligned} \underset{\boldsymbol{\beta}_j}{\arg\min} \quad & \sum_{i=1}^{n} \left(y_{ij} - \sum_{k=1}^{K} \beta_{jk} \mathbb{1}(x_{ij} = k) \right)^2 \\ \text{subject to} \quad & \sum_{k} \beta_{jk} = 0. \end{aligned} \tag{4.3}$$

An ANOVA table for hypothesis testing can be constructed by partitioning the sum of squares and using a Fisher statistic. Denote $\hat{y}_{ij}$ the fitted values for gene j and observation i and $\bar{y}_j$ the mean expression in the sample. The regression residuals can be broken down as follows:

$$\sum_{i=1}^{n} (y_{ij} - \hat{y}_{ij})^2 = \sum_{i=1}^{n} (y_{ij} - \bar{y}_j)^2 - \sum_{i=1}^{n} (\hat{y}_{ij} - \bar{y}_j)^2. \tag{4.4}$$

The statistic

$$f = \frac{\sum_{i=1}^{n} (\hat{y}_{ij} - \bar{y}_j)^2 / (K-1)}{\sum_{i=1}^{n} (y_{ij} - \hat{y}_{ij})^2 / (N-K)} \sim F(K-1, N-K), \tag{4.5}$$

where $F(\mathrm{df}_1, \mathrm{df}_2)$ is the Fisher distribution with df_1 and df_2 degrees of freedom. The null hypothesis is formulated as follows: H_{0j} : The gene j has the same expression level in all groups. Bonferroni corrections are applied for the correction of multiple tests. The ANOVA model is fitted using the **stats** package.

4.3.2 Clustering of SNP and methylation data

Identifying the genetic structure of a population of species is common in genetics (Robin and Ambroise, 2019). As part of our analysis, the genetic structure is used to find epigenetic markers that mimic the same population structure. We briefly recall the clustering procedure developed by a co-author of the paper Sow et al. (2023) to find the genetic structure of populations. Hierarchical ascendant

clustering with Ward's method is applied to the VanRaden Genomic Relationship Matrix (GRM, VanRaden (2008)) converted into a dissimilarity matrix. The R package **stats** is used.

The epigenetic structure is also uncovered using hierarchical clustering with Ward's method on the correlation similarity matrix between populations. The approach is implemented in the **methylKit** package.

4.3.3 Gene sets testing

We perform enrichment analysis to provide a biological interpretation of differentially methylated cytosines. We seek significantly enriched gene cluster in gene ontology terms, recovered from clustering on the cytosines. This approach is slightly different from classical gene set enrichment analysis (GSEA), which considers both a set of differentially expressed or methylated genes and the complementary set.

Cytosine groups are found using k-means with euclidean distances on the methylation profiles. The centered log-ratio previously transforms these profiles. Note that the clustering step can make it possible to find co-methylated profiles and help characterize genes whose biological function is not yet known (Rau, 2017). The number of clusters is selected using the slope heuristic (Baudry et al., 2012) with a number of clusters varying between 1 and 30. Clustering is done using the **coseq** package (Rau et al., 2017) initially developed for clustering RNA-Seq data.

In the second step, a map is established between the positions of the cytosines and the known genes of poplar using the genome of Populus trichocarpa v3.1. The Gene Ontology database (GO, Ashburner et al. (2000)) is used for functional gene annotations. We limit ourselves to the GO terms of biological processes. Molecular functions and cellular components are not taken into account.

Finally, the enrichment analysis is performed using the **topGo** package after the clustering step. The tests are based on the number of genes per ontology belonging to the cluster. For cluster k and GO term Y, the following question is answered: Is there an association between the genes of the cluster k and the annotations for the GO term Y? The contingency table 4.5 can therefore be built. The p-values are calculated using Fisher's exact (hypergeometric) test. The null

	Differentially expressed genes in Cluster k	Differentially expressed genes in other clusters	Total
In GO term Y	n_{11}	n_{12}	$n_{1.}$
Not in GO term Y	n_{21}	n_{22}	$n_{2.}$
Total	$n_{.1}$	$n_{.2}$	$n_{..}$

Table 4.5: Contingency table between cluster's genes and ontology term.

hypothesis H_0 states: the genes of the cluster k are not overrepresented in GO Y.

Only ontologies with a p-value less than 0.05 after Benjamini-Hochberg correction are considered significant.

Complementary GO enrichment analysis is also performed using **Metascape** (Zhou et al., 2019) bioinformatics software with default settings. This approach is based on the sets of differentially and non-differentially methylated genes. Additionally, it performs clustering of GO terms and returns the GO with the lowest p-value per cluster.

4.3.4 Results

4.3.4.1 Populations' genetic and epigenetic structure

The recovered groups from SNP data and the natural groups from the geographical structure are concordant in differentiating the 10 natural populations of poplar genotypes (see figures 4.8 and 4.5). Indeed, each pair of poplars per geographical site first merged before joining other sites. One genotype from the Rhine population was discarded due to some filtering constraints on the methylation data and removed from the SNPs and expression datasets. The hierarchical tree in Figure 4.8 summarizes the pattern of evolutionary relatedness among the group of poplars' species. With a partition into 3 clusters, we note that the first group, composed of the populations of Basento and Paglia, corresponds to geographical sites in the south of the map (Figure 4.5). The second group, composed of Dranse, Kuhkopf, Rhin, and Ticino, corresponds to the geographical sites of the center. The last group, composed of populations from Adour, Val d'Allier, Loire, and Ramières, corresponds to the geographical sites of the West.

DNA methylation data's epigenetic analysis must be considered a triple analysis according to the methylation contexts (CpG, CHG, and CHH). To make the reading pleasant and relevant, we focus only on the results obtained in the CpG context. Note that the analysis is readily reproducible in other methylation contexts. The clustering tree obtained on the methylation data in a CpG context is given in Figure 4.9. The structure is different from the genetic structure of the population. However, the two genotypes belonging to the same natural population merged all the same before joining other geographical sites.

The three meta-populations identified earlier are used for the differential analysis step to identify epigenetic markers of the population's genetic structure and, therefore, implicitly of the geographical structure.

4.3.4.2 Epigenetic markers of the genetic population structure

To characterize which cytosines differentiate the three meta-populations based on the genetic structure (Section 4.3.4.1), we performed a differential analysis for DNA methylation data in the CpG context. Figure 4.10 illustrates the histogram of unadjusted p-values for cytosines. After Bonferroni's correction, 69189 cytosine positions out of 260078, or about 26%, were significantly associated with the genetic structure. The overall significance level was set at 0.01.

For the enrichment analysis, we selected the top 1500 of the most differenti-

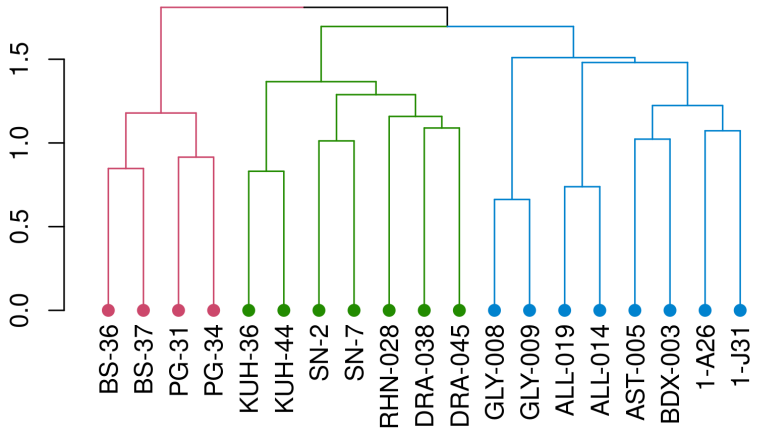


Figure 4.8: Genetic structure of natural poplar populations from SNP data. The clustering dendrogram is obtained by hierarchical ascendant clustering based on the similarity matrix of genomic relationships between genotypes. A 3 cluster partition is indicated by different colors for the branches and leaves of the tree.

ated cytosines, i.e., those with the lowest p-values. Their underlying population structure is compared to the genetic structure in Figure 4.11. The two dendrograms are different. However, the DNA methylation dendrogram shows that the three poplar meta-populations are well discriminated.

We conducted an enrichment analysis on epigenetic markers, identifying 8 clusters obtained using the clustering approach described in Section 4.3.3. The mapping step identified 746 genes. Figure 4.12 shows the enriched GO for a selected cluster. Compared to the other clusters, this displayed cluster is enriched in genes involved in biological regulation and response to stimuli, among others.

A gene set enrichment analysis is also done using the Metascape approach in Figure 4.13 and also highlights relevant biological processes.

4.3.4.3 Methylation and expression profiles

The objective is to verify if there are patterns between the expression profiles and the methylation profiles of the housekeeping and differentially methylated genes within the meta-populations.

Housekeeping genes are, by definition, genes whose expression level is stable between populations. In order to identify them, we performed a differential analysis between genotypes belonging to the same meta-populations for the RNA-Seq data and selected the non-differentially expressed genes. This differential analysis within

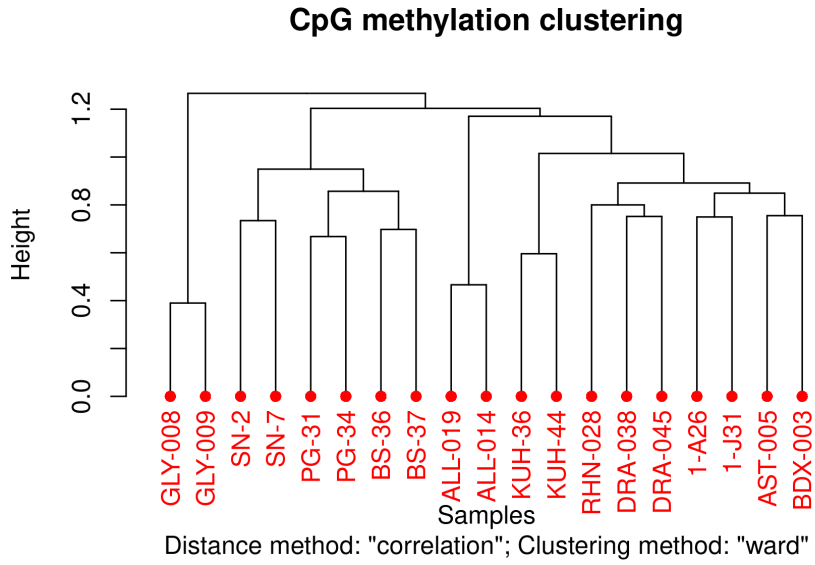


Figure 4.9: Epigenetic structure of natural poplar populations from DNA methylation data in the CpG context. The clustering dendrogram is obtained by hierarchical ascendant clustering based on the similarity matrix of correlation between genotypes.

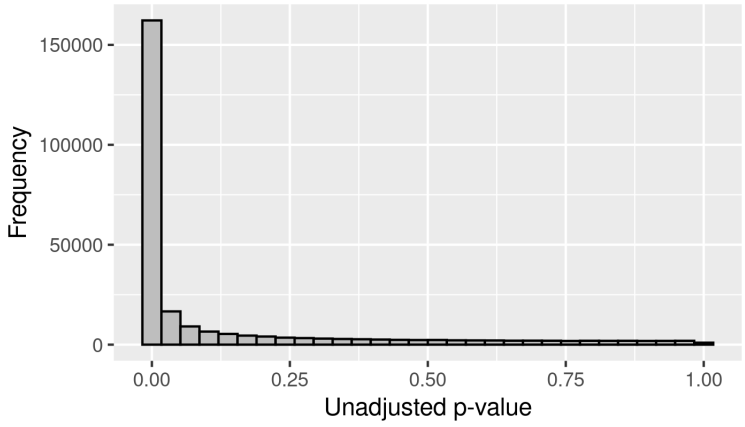


Figure 4.10: Histogram of unadjusted p-values

meta-populations is also performed for DNA methylation data. The results are presented only for the CpG context. We used the methylation dataset provided for genes and not for SMPs.

For expression data (34229 genes), we identified 34227, 33332, and 32660 non-differentially expressed genes in the first meta-population (Paglia, Basento), the second (Dranse, Kuhkopf, Rhin, Ticino) and the third (Adour, Val d'Allier, Loire, Ramières), respectively. For DNA methylation data, we obtained 8558, 6154, and

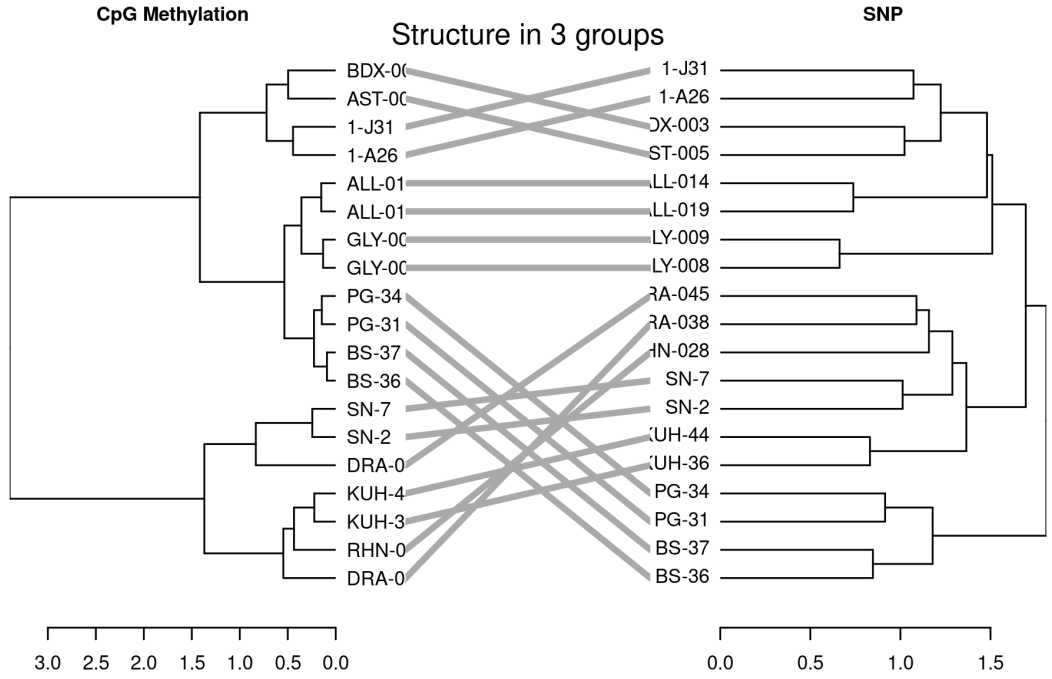


Figure 4.11: Epigenetic structure of natural poplars based on 1500 markers from DNA methylation data in the CpG context. The hierarchical tree on DNA methylation data is plotted against the one computed from the whole SNP data.

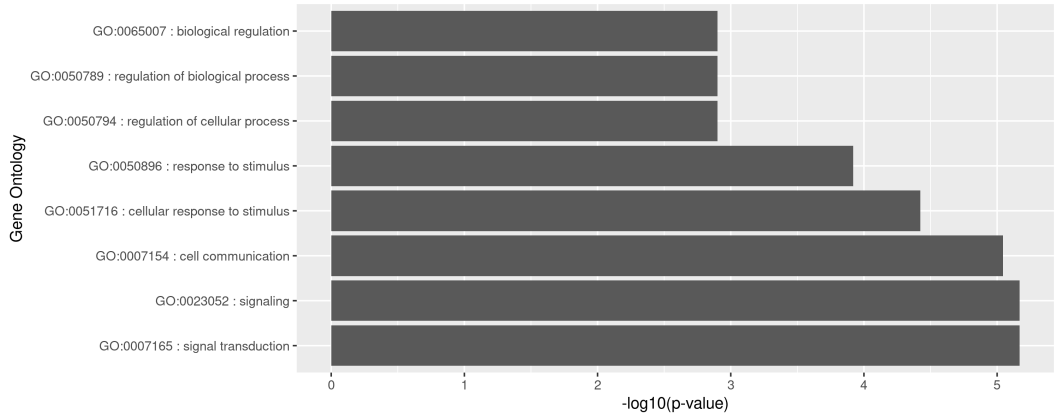


Figure 4.12: Top 8 enriched gene ontology terms for the biological process ontology in one selected cluster. The bar plot displays the $-10\log$ of the adjusted p-value for the Fisher's exact test.

7427 differentially methylated genes in the first, second, and third metapopulations, respectively.

A simple linear regression model is then fitted between the expression and

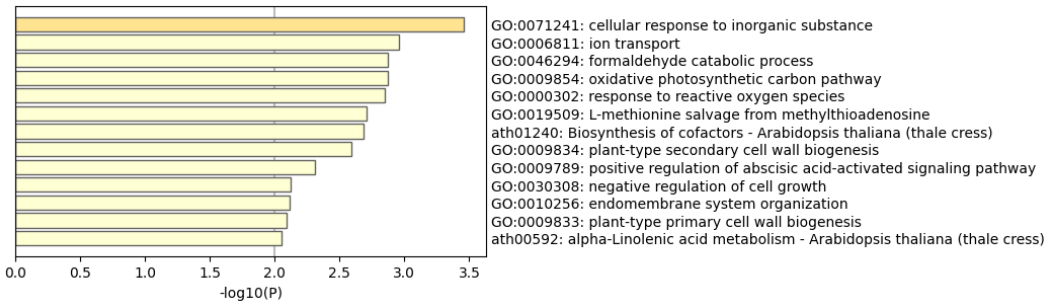


Figure 4.13: Encriched gene ontology with Metascape.

methylation levels onto the intersection of the two subsets of genes, with the expression considered the response. Figure 4.14 shows the match between expression and methylation profiles for the natural population of Paglia. The analysis did not identify a general trend for the different subpopulations nor any significant linear model coefficients.

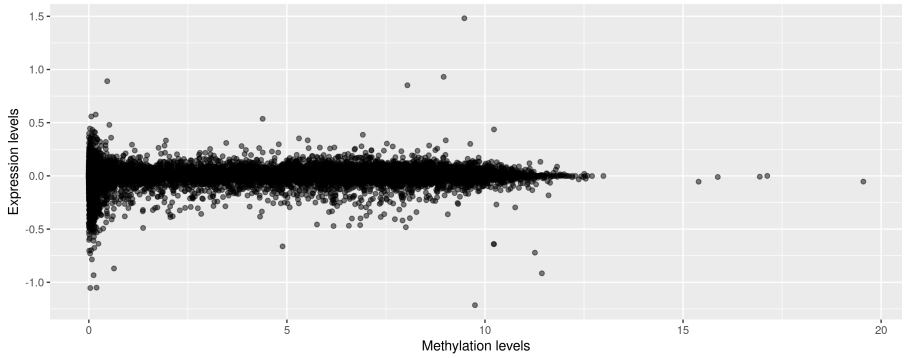


Figure 4.14: Methylation profiles plotted against expression profiles for housekeeping and differentially methylated genes. The samples are the mean expression and methylation levels of the genotypes belonging to the natural population of Paglia in the CpG context. We do not observe a significant linear trend between the two genomic entities.

4.3.5 Discussion

The population structure was investigated for epigenetic data (DNA methylation in a CpG context) and genetic data (SNPs). As mentioned earlier in the introductory part of the review, studies exploring genetic and epigenetic variation in trees are less well explored than in annual plants.

The results obtained on the poplar forest tree suggest that, as well as the SNPs data, there is a relationship between DNA methylation and the local adaptation of trees. This result has also been noted by Lamka et al. (2022). Indeed, the hierarchical clustering trees have shown that it is always possible to merge first the

genotypes from the same geographical sites. Our results also suggest that DNA methylation can be used as a marker of population differentiation according to the genetic structure. To this end, we have proposed a list of differentially methylated genes markers of the genetic structure of the population.

When examining methylation within meta-populations, we did not find a clear pattern between methylation and expression levels for housekeeping genes. The analysis could be renewed by focusing on a smaller sample of genes and by defining, together with the biologists, much more stringent filters. We refer to Sow et al. (2023) for more details on the biological findings of this study.

4.4 Integrative analysis of methylation and transcriptomic data through MGLASSO

This section presents an application of our MGLASSO approach (Chapter 3), to transcriptomic and DNA methylation data. They were collected as part of the EPITREE project (Maury et al., 2019) and have already been presented in the previous section (Section 4.3). We performed an integrative analysis in which we modeled the multi-scale relationships between natural poplar populations for a set of selected genes. Unlike Section 4.3, where clustering is performed on omics data of different natures separately, in what follows, we incorporate several types of omic data to generate a clustering path of the natural populations of poplars and multi-scale interaction graphs. We laid the groundwork for further analysis by briefly exploring the structure and interplay between transcriptomic and methylation patterns in different methylation contexts. The study detailed here is summarized in the application section on real data of the MGLASSO article (Sanou, 2022). We present the data in Section 4.4.1. Section 4.4.2 introduces the approach used for gene selection. Section 4.4.3 gives details on how the model is learned. We present some results in Section 4.4.4.

4.4.1 Data and pretreatments

As mentioned earlier, poplar is often used as a model tree in studying molecular determinants of drought stress. As part of the EPITREE project, natural populations of poplars were planted in common gardens in France, Italy, and Germany (see Figure 4.15) with control over certain environmental variables such as water availability (Sow et al., 2018). The datasets used have already been presented in

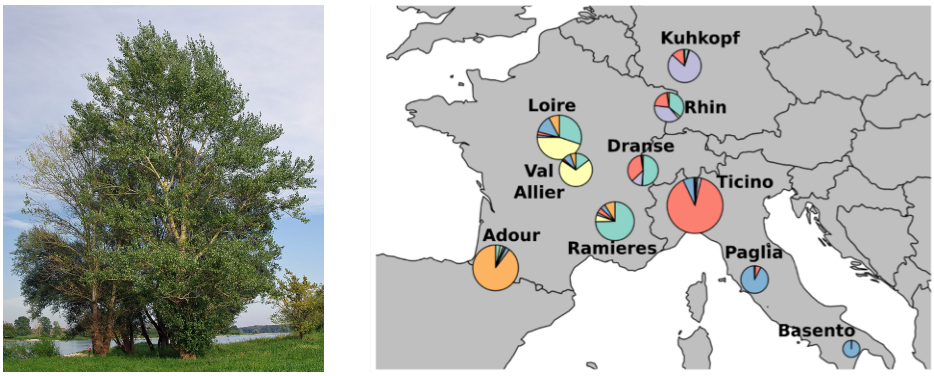


Figure 4.15: Photo of a black poplar (left, source: Christian Fischer) and geographical distribution of the 10 natural populations of poplars (right).

Section 4.2.1. The focus here is on gene expression and methylation data. For each of the 10 natural populations (with two genotypes per population), RNA sequencing data and DNA methylation data for the promoter and the gene-body regions, in the CpG, CHG, and CHH contexts are available. Let A, C, G, and T be the

nucleotides adenine, cytosine, guanine, and thymine, respectively. As a reminder, methylation consists of adding a methyl group to a cytosine of the genome and occurs in three contexts (CpG, CHG, and CHH, where $H \in \{A, C, T\}$). Methylation can be measured on promoter and gene-body regions. In promoter regions, methylation is related to gene silencing, while it is related to tissue-specific expression or alternative splicing in gene-body regions (Sow, 2019).

For the genes methylation data, we used data that have been normalized via the reads by density approach and then passed to a logarithm function $log_2(x+1)$ with $x \in \mathbb{R}$. The reads by density is the product between the number of methylated reads and the proportion of methylated reads. The approach was defined in Sow et al. (2023). RNA-Seq data were normalized via trimmed mean of M-values followed by a correction for experimental design effects via a mixed linear model (Chateigner et al., 2020).

An average value is calculated for each pair of genotypes per population. Integrating the 10 natural populations on the 7 omic entities gives a set of 70 profiles. The analysis was restricted to a group of 151 target genes selected from 24926, which explains the most variability in the data. The gene selection process is described in Section 4.4.2.

4.4.2 Gene selection through sparse PCA

We perform a sparse principal component analysis for variable selection based on singular value decomposition and the LASSO penalty on loading vectors. The most significant genes, i.e., those for which the loading vector entries are non-zero, are selected. The approach is implemented in the R package **mixomics** (Rohart et al., 2017), which is based on a low-rank matrix approximation (Shen and Huang, 2008).

Consider a dataset $\mathbf{X} \in \mathbb{R}^{n \times p}$ with n observations and p variables. Suppose the columns of $\mathbf{X}$ are centered. Vanilla principal component analysis can be performed via the singular value decomposition (SVD) of $\mathbf{X}$. Denote $r = \text{rank}(\mathbf{X})$, the SVD writes:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top, \tag{4.6}$$

where $\mathbf{U} \in \mathbb{R}^{n \times r}$ is the matrix of left singular vectors, $\mathbf{V} \in \mathbb{R}^{p \times r}$ the matrix of right singular vectors, and $\mathbf{D} \in \mathbb{R}^{r \times r}$ is a diagonal matrix whose diagonal elements are ordered singular values. The columns of $\mathbf{UD}$ are the projections of the data i.e. the principal components. The columns of $\mathbf{V}$ are the corresponding loadings or direction vectors, which are used to interpret the principal component axes. Interpretation is improved when loadings are sparse (see Hastie et al. (2009) Section 14.5.5).

The intuition of the sparse PCA approach introduced by Shen and Huang (2008) is to exploit the link between the SVD and the low-rank approximation method (Eckart and Young, 1936) while solving a penalized least-squares regression problem. The matrix $\mathbf{X}$ can be approximated by another matrix $X^{(l)}$ of lower rank

with $l \leq r$, as follows:

$$\mathbf{X}^{(l)} = \sum_{k=1}^{l} d_k \mathbf{u}_k \mathbf{v}_k^\top . \tag{4.7}$$

For $l = 1$, we can write $\mathbf{X}^{(1)} = \tilde{\mathbf{u}}\tilde{\mathbf{v}}^\top$ where $\tilde{\mathbf{u}}$ is a norm-1 n-vector and $\tilde{\mathbf{v}}$ is a p-vector. The rank one matrix $\mathbf{X}^\star = \tilde{\mathbf{u}}^\star\tilde{\mathbf{v}}^\star$ is estimated as follows, with parsimony constraints on the loading vector:

$$\begin{aligned} \underset{\tilde{\mathbf{u}},\,\tilde{\mathbf{v}}}{\text{minimize}} \quad & \left\| \mathbf{X} - \tilde{\mathbf{u}}\tilde{\mathbf{v}}^\top \right\|_F^2 + \lambda \left\| \tilde{\mathbf{v}} \right\|_1 , \\ \text{subject to} \quad & \|\tilde{\mathbf{u}}\|_2 = 1, \end{aligned} \tag{4.8}$$

where $\|.\|_F$ is the Frobenius norm and $\lambda \geq 0$ the LASSO penalty parameter. The authors propose an iterative procedure to obtain the first sparse loading vector $\tilde{\mathbf{v}}/\|\tilde{\mathbf{v}}\|_2$. The other sparse loading vectors are also computed via a rank one approximation.

In practice, the selection of the parameter λ is solved in Rohart et al. (2017) by defining the degree of parsimony as the desired number of significant genes, i.e., the number of non-zero entries in the vectors loading. We applied a sparse PCA on the omics datasets taken separately; then, on the first 3 principal components, we selected 15 genes. This yielded a set of 315 candidate genes. Removing duplicates leaves us with a final set of 151 genes on which we integrate omics data.

4.4.3 MGLASSO learning

We applied the MGLASSO method introduced in Chapter 3, on the material of the study. Natural populations are considered as variables, and genes as observations. The sample size is, therefore, $n = 151$, and the number of variables $p = 70$.

The bounds on the penalization parameters introduced in Section 3.3 were written after the study. They have, accordingly, not been tested in practice with MGLASSO applications. Following the discussion on model selection in the section mentioned above, we selected only the LASSO penalty parameter, and this via the StARS approach (Liu et al., 2010) to provide maximal network stability. In proceeding so, MGLASSO is equivalent to the neighborhood selection method (Meinshausen and Bühlmann, 2006). The LASSO penalty parameter is chosen from a grid of 50 equidistant values on a logarithmic scale $[\lambda_{1\min}, \ldots, \lambda_{1\max}]$ where $\lambda_{1\min} = \lambda_{1\max}/1000$ and $\lambda_{1\max}$ is the maximum of the smallest values of the LASSO tuning parameters for which the node-wise regression coefficients are all zero (El Ghaoui et al., 2010). In practice, we used the **huge** package (Zhao et al., 2012) to compute the StARS selection with a grid of penalization parameters values provided. The StARS cut-point value (variability threshold) was set to 0.05. We drew 50 subsamples from the data with a subsampling block size set to $0.8n$. The sequence of fusion penalty parameters λ_2 is a grid of 20 equally-spaced values in the interval $[0, 30.94]$.

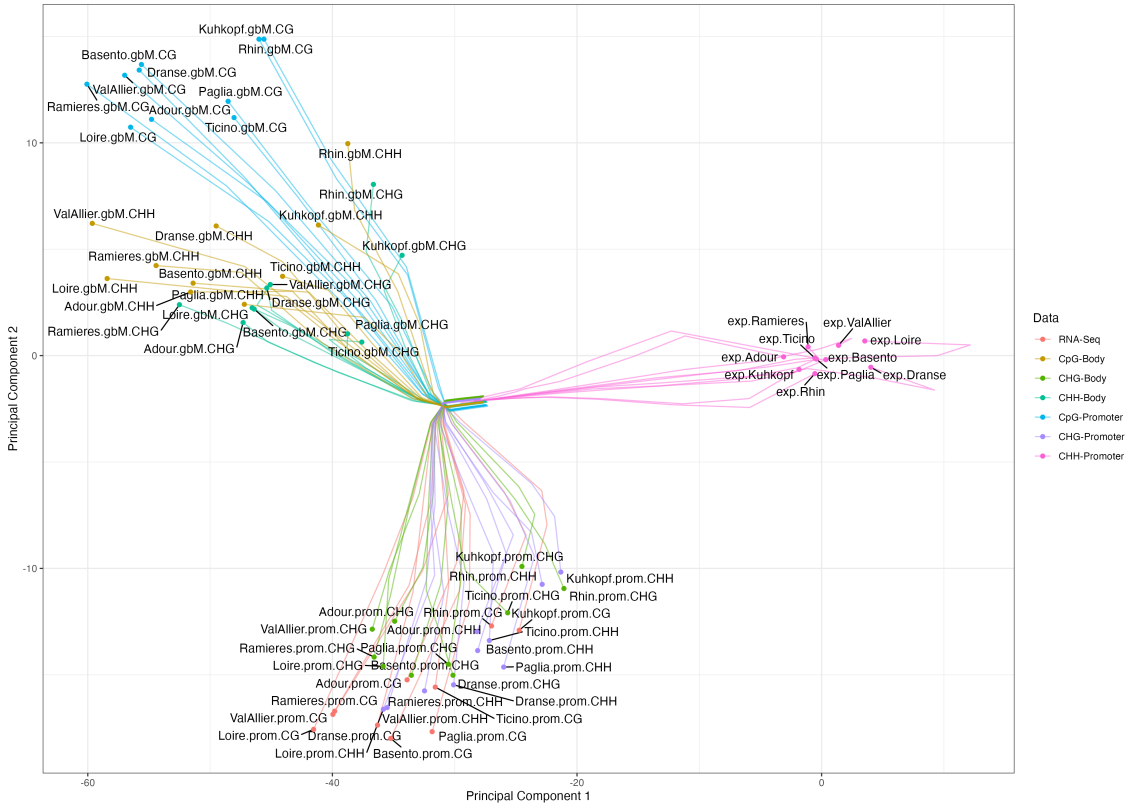
In the convex clustering problem introduced by Hocking et al. (2011), the classic illustration used for the method results is the path of convex clustering solutions over the fusion penalty, projected onto the principal components of the original dataset. In the MGLASSO model, we adopted the same idea with the LASSO penalty fixed. However, the solutions, i.e., the estimated regression vectors, do not belong to the same space as the data. Consequently, the path is drawn on the predicted data.

Graphs computed using MGLASSO are symmetrized using the AND rule (Meinshausen and Bühlmann, 2006) i.e., an edge (i, j) is considered to be present when the estimated coefficient of variable i on variable j and the estimated coefficient of variable j on variable i are both non-zero. The desired precision for stopping the CONESTA-based optimization algorithm (Hadj-Selem et al., 2018) is fixed to $\epsilon = 0.01$.

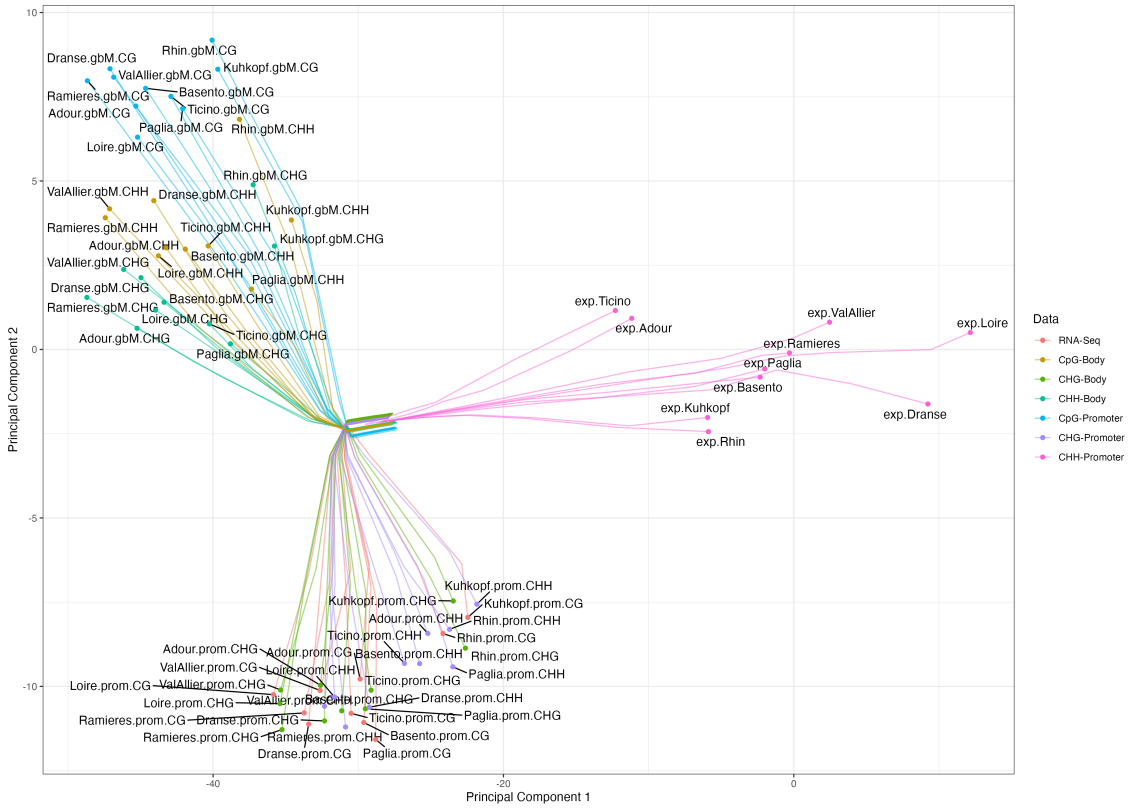
4.4.4 Results

We present the clustering path and the estimated adjacency matrices generated via the MGLASSO model.

Figure 4.16 shows the resulting clustering paths on the entire grid of fusion penalty parameters (Figure 4.16a) and a truncated grid (Figure 4.16b).



(a) Full clustering path

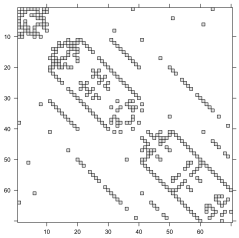


(b) Cutted clustering path

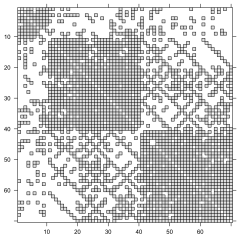
Figure 4.16: Clustering path of solutions on DNA methylation and transcriptomic samples. The figure shows 3 distinct clusters which correspond to omics data of different natures: transcriptomic (right), methylation on the promoter (bottom), and methylation on gene-body (top left).

We observe some observations' splits after their fusion. However, there is a clear separation between the profiles of natural populations corresponding to RNA-Seq data, gene-body methylation data, and promoter methylation data. In the group composed of gene-body profiles, we can see a split between the other methylation contexts and the CpG context. These profiles in the CpG context can be subdivided into two main groups consisting of the natural populations of the Rhine, Kuhkopf, Paglia, and Ticino on the one hand and the natural populations of Basento, Ramières, Loire Adour, Dranse, and Val d'Allier on the other hand. The expression data suggest three groups of natural populations: Tessin and Ardour; Rhine and Kuhkopf; Val d'Allier, Loire, Ramières, Paglia, Basento, and Dranse.

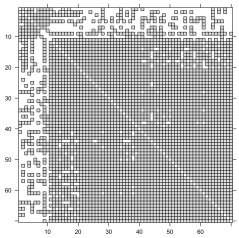
Figure 4.17 shows some graphs (adjacency matrices) of the MGLASSO network inference path. As λ_2 increases, the resulting multi-scale graphs are block-diagonal graphs with the number of blocks in $\{7, 3, 2, 1\}$.



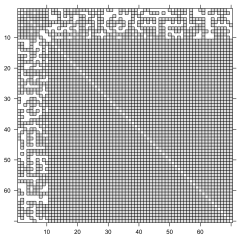
(a) $\lambda_2 = 0$



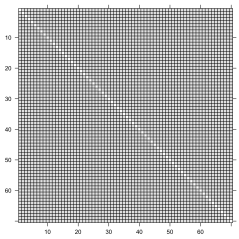
(b) $\lambda_2 = 1.63$



(c) $\lambda_2 = 3.26$



(d) $\lambda_2 = 4.89$



(e) $\lambda_2 = 30.94$

Figure 4.17: Adjacency matrices for different fusion penalty parameters. The first graph shows the inferred network when no fusion penalty is added to the model. In that graph, the first block of size 10×10 variables corresponds to RNA-Seq samples. The second sparser block of size 30×30 corresponds to gene-body DNA methylation data in the three methylation contexts. The last sparse block of the same size corresponds to promoter methylation. The edge bands suggest a relationship between DNA methylation measurements that belong to the same context. For example, the Loire methylation sample in the CpG context is likely related to the Loire samples in the CHG and CHH contexts. The graphs also suggest some relationships between expression and methylation for some natural populations. As the merging penalty increases, the blocks corresponding to the three methylation contexts merge first, then follow the upper left block corresponding to the expression data. For $\lambda_2 = 30.94$, all natural populations merge into a single cluster and complete graph.

Figure 4.17a shows the adjacency matrix with fixed Lasso penalty and $\lambda_2 = 0$. The graphs suggest a higher number of interactions between natural populations belonging to the same omic entity. There are fewer interactions between natural populations from different omics entities. Regarding these interactions between profiles from different omic entities, we observe a clear interaction trend between profiles belonging to the same methylation context.

As the fusion penalty increases, in Figure 4.17b we observe a grouping effect in the edges, which suggests the same 3 clusters identified in the clustering path. The interactions inside the 3 groups are irrelevant because almost all the nodes have merged. However, the relationships between the three groups may suggest potentially natural populations whose expression, gene-body methylation, and methylation in the promoter region are correlated.

As the fusion penalty continues to increase, all methylation profiles merge in Figure 4.17d. The interactions highlighted may suggest potential natural populations with a stronger relationship between their transcriptomic and methylation profiles.

For a maximum fusion penalty value, the clustering effect extends to all nodes in the graph resulting in a complete graph.

4.4.5 Discussion

We conducted an integrative analysis on multiple omics data to model the multi-scale relationships between natural populations of poplar and also illustrate the performance of the MGLASSO method in a context of real data where the results can be proof checked according to the biological context.

We accounted for conditional dependencies within natural populations measuring the same omic entity and conditional dependencies between natural populations belonging to different omic entities. It is well known that methylation regulates the expression of genes (Neuvial, 2020), and some of the results suggest potential poplar natural populations for which there is a strong link between their methylation and expression profiles. When calculating the MGLASSO clustering path, the path was not agglomerative, but the method highlighted three pure clusters that correspond to strictly different omics entities. These results also suggest that gene-body methylation signals for CpG are clearly differentiated from other methylation contexts in an integrative analysis framework. Note, Sow et al. (2023) indicated that methylation in the CpG context has more adaptive power than methylation in the other contexts.

The analysis presented is preliminary; however, it lays the groundwork for further research. It is based on natural populations and not genes and contrasts with integrative analysis approaches performed on genes (Section 5.1 in Rau (2017); Chiquet et al. (2019)).

5 - Conclusion and perspectives

This thesis proposes a novel approach to enhance the inference of Gaussian graphical models by incorporating a structure of the variables. This structure is assumed to be unknown. Multilevel Gaussian graphical models have gained increasing attention in recent years, yet their application has been limited to the inference of networks for single variables and a priori known groups of variables. There is a growing interest in exploring their potential for analyzing heterogeneous data, especially omics data consisting of variables from different probability distributions. In omics projects, such as EPITREE, various types of data, including genetic, genomic, and epigenetic data, have been produced to gain a better understanding of the molecular mechanisms involved in the adaptation of forest trees throughout their lifespan. However, one question that has received less attention is the quantified impact of DNA methylation, an epigenetic mark, on the trees' local adaptation. This question can be addressed by exploring the links that can be established between the different omics data. By proposing an approach that can account for the structure of variables in Gaussian graphical models, this thesis contributes to the advancement of statistical inference methods. Furthermore, by demonstrating the potential of multilevel Gaussian graphical models for analyzing heterogeneous omics data, this research has the potential to inform on the molecular mechanisms of forest tree adaptation and improve our understanding of the role of DNA methylation in this process.

In this research, we proposed a novel inference model that combines the theories of convex clustering and neighborhood selection. Convex clustering is a group-fused LASSO penalized convex relaxation of hierarchical clustering that can produce robust results. Meanwhile, neighborhood selection is a LASSO penalized pseudo-likelihood approximation that reduces the computational burden in estimating graphical models and simplifies extensions to multiple probability distributions. By linking the inference of Gaussian graphical models with the convex clustering theory, we can construct networks with nested structures, similar to the hierarchical tree structure, under certain conditions. While some previous research has combined both theories, the idea of multi-scaling structures has not been clearly demonstrated. Our proposed model overcomes this limitation and provides a more comprehensive framework for analyzing complex networks.

Our proposed model, the Multiscale Graphical LASSO (MGLASSO), aims to improve the interpretability of networks by providing graphs at multiple levels of granularity. This feature is particularly useful in biological data analysis where it's necessary to zoom in and out of inferred networks. The procedure starts by estimating a network based on single variables, followed by the addition of a fuse-group LASSO penalty to encourage similarity in the estimated vectors of parameters and recover networks on groups of variables. The approach uses convex optimization

and minimizes the neighborhood selection objective, penalized by a hybrid regularization that combines a sparsity-inducing norm and a convex clustering penalty. To apply MGLASSO in practice, we developed a complete numerical scheme that includes an optimization algorithm based on CONESTA (Hadj-Selem et al., 2018) and a model selection procedure based on StARS (Liu et al., 2010). We implemented the approach as an R package based on Python libraries (Sanou, 2022). Although the MGLASSO was designed for graph inference, it is also a convex clustering technique. In the literature of convex clustering, this is the first attempt to use a continuation method based on Nesterov smoothing.

Our simulation results on synthetic and real datasets demonstrated that MGLASSO outperforms GLASSO (Friedman et al., 2008) in network support recovery in the presence of groups of correlated variables. We further illustrated the method's effectiveness through its application on heterogeneous data, which combined transcriptomic and DNA methylation data, as well as on non-heterogeneous metagenomic data. Our approach accounted for heterogeneous data and non-Gaussian data by applying variable transformations to return to the Gaussian framework. In an effort to better understand the impact of DNA methylation on the local adaptation of forest trees, specifically the poplar, the results of a joint study suggest that DNA methylation can be utilized as a marker of trees population differentiation based on genetic structure (Sow et al., 2023).

The research work has some limitations that could be addressed in future work. For example, the MGLASSO method struggles to produce tree structures without splits, and the calibration of the threshold for merging variables can be challenging. Data transformation can also introduce biases, and there is currently no selection criterion for the fusion penalty parameter. Additionally, the method is most effective with a relatively low number of variables, which may limit its utility in certain applications. In the framewrk of the EPITREE project, it would be valuable to explore the inference of gene regulatory networks from omics data, in order to identify interactions between epigenetic markers of the genetic structure of natural poplar populations. Furthermore, while the proposed R software is functional, it could benefit from more documentation to help users fully understand and utilize its capabilities. Overall, this research lays the groundwork for further advancements in this area, and future work can build upon these findings to address the existing limitations and extend the method's capabilities.

5.1 Perspectives

5.1.1 Avenues in convex clustering

5.1.1.1 Recovering a tree structure

In the theory of convex clustering, weights w_{ij} are often used in the fusion penalty to measure the similarity between variables $\mathbf{X}^i$ and $\mathbf{X}^j$ in order to improve clustering results. However, as seen in the case of MGLASSO, the convex clustering

solution path may display abrupt fusions or splits of clusters when using identity weights. It should be noted that while convex clustering is a convex relaxation of hierarchical clustering, splits are uncommon in standard hierarchical clustering. Several rules have been proposed for selecting weights that result in a clustering solution path that reflects a tree structure without splits. For example, Hocking et al. (2011) conjectures that for decreasing weights $w_{ij} = \exp(-\left\|\mathbf{X}^i - \mathbf{X}^j\right\|_2^2)$ in the case of ℓ_2 fusions, the clustering does not display splits. In addition, Chiquet et al. (2017a) proposed weights with theoretical guarantees that prevent splits in the clustering solution path. Another study by Chi and Steinerberger (2019) suggests that if the weights w_{ij} reflect a tree structure among the variables, then the expected solution path will exactly reconstruct the tree. However, how to select weights in the MGLASSO framework to ensure paths with no splits is a topic for future research.

5.1.1.2 Selection of clusters' fusion threshold

As mentioned throughout the work, in practice, exact equality between estimated vectors is not required before assigning variables to the same cluster as done in most research work about convex clustering. For example, in Sun et al. (2021), the fusion threshold is arbitrarily set to $\epsilon = 10^{-5}$. Defining the fusion threshold based on a rigorous rule can improve the results. While we assumed a transitivity relation by using an approximate test to derive the clusters, in practice, we have no strong guarantee that $\left\|\boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j\right\|_2 \leq \epsilon$ and $\left\|\boldsymbol{\beta}^j - \boldsymbol{\tau}_{jk}\boldsymbol{\beta}^k\right\|_2 \leq \epsilon$ ensure that $\left\|\boldsymbol{\beta}^i - \boldsymbol{\tau}_{ik}\boldsymbol{\beta}^k\right\|_2 \leq \epsilon$. As suggested by Jiang and Vavasis (2020) for the case of vanilla convex clustering, it may be worth exploring how the MGLASSO is sensitive to threshold selection and how it can be selected more rigorously.

5.1.1.3 Bounds on the regularization parameters

The MGLASSO algorithm is currently implemented as a basic path algorithm in which the LASSO penalty is kept constant and the clustering penalty parameter is varied in a grid of values. Following iterations can benefit from the previous iterations' estimations as warm starts. However, this approach could be further improved by considering theoretical guarantees and incorporating them into the algorithm. In Hoefling (2010), a path following algorithm is proposed for the Fused LASSO signal approximator. This algorithm considers the computation of the next hitting times and violation times as fusion parameter values for which the clusters will fuse or split, respectively. By adopting a similar approach for the MGLASSO, the burden of defining the right grid of values for the fusion penalty parameter could be removed, and the computation burden related to a path algorithm could be optimized. In the worst case scenario, if the hitting and violation times cannot be defined, one could consider deriving an upper-bound on the fusion penalty parameter and avoid the blind definition of the grid of values. Tan and Witten (2015)'s work on convex clustering and Tibshirani and Taylor (2011)'s work on the generalized LASSO could provide some insights on how the upper-bound

can be derived. Overall, incorporating theoretical guarantees into the MGLASSO algorithm could lead to more efficient and accurate clustering results.

5.1.2 Avenues in probabilistic graphical models inference

5.1.2.1 Estimation of precision matrix coefficients

The MGLASSO algorithm is not based on the full Gaussian likelihood of the model, unlike the graphical LASSO (GLASSO). Instead, it focuses on the neighborhood selection scheme of inference of Gaussian graphical models using a pseudo-likelihood approximation. While it would be interesting to aim for an exact estimation of the precision matrix, it is not guaranteed to lead to faster optimization procedures. However, it may overcome uncertainties related to edge estimations. In neighborhood selection approaches, OR or AND rules are typically applied, and the consistency of the approach for the MGLASSO case has not been proven with theoretical properties. To prove the consistency of MGLASSO, the necessary assumptions need to be explicitly addressed, as done in Meinshausen and Bühlmann (2006).

The estimation of the precision matrix can be approached in two ways. The first method involves directly maximizing the full likelihood criterion, which is given by:

$$\log\det \boldsymbol{\Omega} - \mathrm{tr}(\boldsymbol{S\Omega}) - \lambda_1 \sum_{j,k} |\Omega_{jk}| - \lambda_2 \sum_{i<j} \|\boldsymbol{\Omega}_{i.} - \boldsymbol{\tau}_{ij}\boldsymbol{\Omega}_{j.}\|_2 \,, \tag{5.1}$$

where $\boldsymbol{S}$ is the empirical covariance matrix, $\boldsymbol{\Omega}$ is the precision matrix, λ_1 and λ_2 are the regularization parameters for the LASSO and clustering penalties, respectively.

The second method is to first estimate the precision matrix using the vanilla MGLASSO and then estimate the diagonal elements of the precision matrix. Ideas about this approach can be found in Balmand and Dalalyan (2016). An additional two-step approach involves refitting a full likelihood model with the edge constraints found via MGLASSO estimations, as done in Section 17.3.1 of Hastie et al. (2009). In practice, a selection criterion has not been defined for the MGLASSO since our focus was on exploratory applications. However, it may be worth considering how a selection criterion can be defined for the fusion penalty parameter. This could improve the algorithm's performance and make it more efficient in real-world applications.

5.1.2.2 Mixed graphical models

The initial aim of this research was to develop a class of graphical models that can handle variables following different probability distributions. As mentioned earlier, one of the advantages of the pseudo-likelihood formulation of the model is that it can be extended to the class of probability distributions belonging to the exponential family. Some introductory work on this topic can be found in Yang et al. (2012). For handling heterogeneous data, mixed graphical models can be considered (Yang et al., 2014). However, when coupled with convex clustering, defining the fusion penalty term may not be straightforward.

5.2 Conclusion notes

The discussion sections of the thesis directly mention perspectives on biological data analysis. Although some quality control type data analyses for the EPITREE project were carried out as part of the thesis, they were not reported in this manuscript. Moving forward, the MGLASSO package could be significantly improved by enhancing the graphic rendering functions and by implementing object-oriented programming in an S4 framework.

6 - Résumé en français

This chapter is a summary of the present work, written in French.

Chapitre 1

L'inférence de réseaux à partir de données biologiques est utile pour obtenir une compréhension plus complète des mécanismes biologiques sous-jacents à un phénomène particulier. Les réseaux sont un moyen naturel d'intégrer et de décrire la manière dont les variables biologiques interagissent, ce qui permet d'identifier des interactions complexes. Avec l'émergence des techniques de séquençage à haut débit, il est possible de générer de grandes quantités de données omiques liées au génome, au transcriptome, aux variations génétiques et au métabolome. La grande dimension de ces données constitue la principale difficulté d'un point de vue statistique et d'interprétation. L'objectif de la recherche est de proposer et d'étudier une méthode d'inférence de réseau qui prend en compte une structure de groupe ou une hiérarchie entre les variables. L'identification de groupes de nœuds densément connectés dans le réseau peut correspondre à des variables biologiques ayant des fonctions apparentées et offre la possibilité de construire des structures multi-échelles pour synthétiser l'information récupérée par les groupes et améliorer l'interprétabilité. Cette tâche de regroupement peut être envisagée avant ou en même temps que la tâche d'inférence du réseau. Les modèles graphiques probabilistes représentent un modèle bien adapté pour déduire les relations entre les variables. La recherche se concentrera sur une sous-classe générale de modèles graphiques où la distribution conditionnelle aux nœuds est gaussienne et examinera les méthodes d'estimation de groupes basées sur un critère convexe.

La recherche est motivée par le projet EPITREE (Impacts évolutif et fonctionnel de variations épigénétiques chez des arbres forestiers), qui cherche à comprendre comment l'épigénétique, en l'occurrence la méthylation de l'ADN, l'expression des gènes et la variation allélique, influence les mécanismes d'adaptation et la plasticité phénotypique chez les arbres forestiers. L'épigénétique est l'étude des changements héréditaires qui affectent l'expression des gènes sans modifier l'ADN, tandis que la plasticité phénotypique est la capacité d'un génotype individuel à exprimer différentes valeurs d'un trait phénotypique donné dans des conditions environnementales différentes (Rey et al., 2016). Les arbres sont des organismes remarquables qui vivent longtemps, ont des cycles de vie complexes et produisent du bois, tout en fournissant un large éventail de services écosystémiques. Au cours des dernières décennies, un dépérissement généralisé des forêts dû à la sécheresse et au stress thermique a été observé dans le monde entier (Anderegg et al., 2016). Ces événements mettent en évidence la vulnérabilité des écosystèmes forestiers

aux changements environnementaux et le besoin urgent de comprendre comment les arbres réagissent au stress environnemental. En effet, le changement climatique est le facteur qui aura le plus d'impact sur la biodiversité d'ici 2100, après l'utilisation des terres (Chapin Iii et al., 2000). On sait que les arbres forestiers possèdent des mécanismes complexes qui leur permettent de s'adapter aux facteurs de stress environnementaux (Bruce et al., 2007). Il est donc essentiel de comprendre les mécanismes moléculaires qui sous-tendent leur adaptation pour élaborer des stratégies de conservation et de gestion des écosystèmes forestiers. Le projet EPITREE étudie les mécanismes moléculaires qui sous-tendent l'adaptation des arbres, en se concentrant sur deux modèles d'arbres, le peuplier et le chêne. Ces espèces ont été choisies pour leur diversité génétique et leur potentiel d'adaptation à des environnements changeants.

Le projet se compose de plusieurs modules de travail, qui comprennent entre autres le criblage de régions candidates, l'analyse épigénomique et le séquençage du génome à l'aide de technologies omiques modernes. Ces technologies ont permis de générer des données sur les polymorphismes simples méthylés, les régions différentiellement méthylées, l'expression des gènes et les polymorphismes de nucléotides simples. La recherche doctorale est principalement motivée par le quatrième module de travail, qui vise à effectuer une analyse intégrative pour modéliser des relations multi-échelles entre les caractères quantitatifs et leurs déterminants moléculaires. Plus précisément, ce module vise à quantifier la contribution de la diversité génétique et épigénétique à la variation phénotypique, à étudier l'impact de l'évolution du chêne et du peuplier sur la plasticité épigénomique, et à déterminer si les régions différentiellement méthylées au niveau des gènes sont conservées entre les deux espèces. En outre, ce module vise à améliorer les modèles de prédiction de la variation des caractères quantitatifs en combinant les informations génétiques et épigénétiques. Les modèles graphiques sont un outil utile pour déduire les interactions entre l'information génétique et les schémas de méthylation, et peuvent apporter des réponses à certaines des questions soulevées dans le projet.

Les modèles graphiques probabilistes (MGP, Lauritzen (1996); Koller and Friedman (2009)) sont un outil populaire d'analyse de données en grande dimension et de capture d'interactions entre des variables. Ils sont largement utilisés dans diverses applications, telles que la génomique et l'analyse d'images, pour réduire le nombre de paramètres en sélectionnant les interactions les plus pertinentes entre les variables. Une classe de MGP particulièrement utile dans un contexte gaussien est celle des modèles graphiques gaussiens non dirigés. En grande dimension, ces modèles sont souvent supposés parcimonieux. On suppose que seul un petit nombre de variables interagissent par rapport au nombre total d'interactions possibles. Cette hypothèse de parcimonie offre des avantages à la fois statistiques et computationnels. En simplifiant la structure de dépendance entre les variables (Dempster, 1972), elle permet de développer des algorithmes efficaces.

L'inférence du graphe d'indépendance conditionnelle dans les modèles graphiques

gaussiens non dirigés, implique l'identification du support de la matrice de précision $\mathbf{\Omega}$ (l'inverse de la matrice de variance-covariance). Pour apprendre ce graphe, plusieurs méthodes ℓ_1-pénalisées ont été proposées dans la littérature. Une des approches populaires est la méthode de sélection de voisinage (MB, Meinshausen and Bühlmann (2006)) fondée sur des régressions successives par variable en utilisant l'opérateur LASSO. Cette méthode se focalise sur l'apprentissage de la structure du graphe. La méthode MB a donné lieu à une longue série de travaux sur les méthodes de régressions successives par variable, y compris des extensions avec diverses formes de pénalités induisant la parcimonie, comme le sélecteur de Dantzig (Yuan, 2010) et l'estimateur Clime (Cai et al., 2011). Une autre famille de méthodes d'inférence de graphes d'indépendance conditionnelle parcimonieux estime directement $\mathbf{\Omega}$ au travers de la minimisation de la log-vraisemblance négative ℓ_1-pénalisée (Banerjee et al., 2008). Cette méthode, appelée LASSO graphique (GLASSO, Friedman et al. (2008)), bénéficie de nombreux algorithmes d'optimisation (Yuan and Lin, 2007; Rothman et al., 2008a; Banerjee et al., 2008; Hsieh et al., 2014).

Les méthodes discutées précédemment utilisent la structure de groupe pour simplifier le problème d'inférence de graphe et inférer le graphe d'indépendance conditionnelle entre des variables non groupées. Cependant, l'inférence du graphe entre des groupes de variables ou des variables représentatives de ces groupes a reçu peu d'attention. Bien que certains travaux aient abordé ce problème, ils se sont surtout concentrés sur des estimations à deux niveaux, c'est-à-dire au niveau des variables prises individuellement et des groupes connus a priori (voir, par exemple, Cheng et al. (2017)). Le problème de recherche abordé dans ce travail vise à définir une méthode d'inférence qui permet des estimations à plus de deux niveaux de granularité avec des groupes inconnus. Ce problème est principalement motivé par des applications en analyse de données biologiques où des données provenant de sources multiples, typiquement des données multi-omiques, sont analysées. Dans de tels cas, il peut être nécessaire de regrouper des variables partageant les mêmes caractéristiques et de prendre simultanément en compte la structure de regroupement dans la procédure d'inférence de réseau, en utilisant une fonction de coût unique au lieu d'alterner les tâches de clustering et d'inférence de réseau.

Nos recherches se focalisent sur l'inférence de structures de clustering hiérarchique, plus intuitives pour l'interprétation, et sur l'apprentissage de la structure de réseau inférée plutôt que sur l'estimation des coefficients de la matrice de précision. Bien que certaines des applications soient principalement motivées par des questions biologiques du projet EPITREE, répondre à ces questions peut nécessiter l'utilisation d'autres outils d'apprentissage automatique dédiés autre que l'inférence de graphes. D'un point de vue mathématique, nos recherches se situent à la croisée de l'inférence graphique probabiliste, du clustering et de l'optimisation convexe. Du point de vue de la biologie statistique, nous abordons diverses questions biologiques, concernant principalement l'impact de l'épigénétique sur l'adaptation locale du peuplier, ce qui nécessite des outils tels que l'analyse différentielle des gènes, l'analyse

d'enrichissement d'ensembles de gènes, la transformation des données de comptage et les approches de sélection des gènes. Les applications omiques comprennent des applications sur les données transcriptomiques (expression des gènes), épigénétiques (notamment la méthylation de l'ADN), génétiques (notamment les SNP) et un exemple d'illustration sur des données métagénomiques (abondance microbienne).

La méthodologie proposée pour l'inférence de modèle graphique, dénommée LASSO graphique multi-échelle (MGLASSO), est une méthode basée sur la pseudo-vraisemblance pour estimer des structures de clustering hiérarchique et des modèles graphiques qui dépeignent la structure d'indépendance conditionnelle entre des groupes de variables à chaque niveau de la hiérarchie. La méthode MGLASSO combine la sélection de voisinage et le clustering via une pénalité de type fused-LASSO (Pelckmans et al., 2005; Hocking et al., 2011; Lindsten et al., 2011). Bien que l'utilisation de pénalités de fusion dans l'inférence de modèles graphiques gaussiens ait été largement étudiée, les travaux précédents se sont principalement concentrés sur la vraisemblance complète pénalisée et ont étudié les pénalités de fusion pour renforcer la constance locale des nœuds du réseau inféré (Honorio et al., 2009; Yao and Allen, 2019; Lin et al., 2020). Cependant le MGLASSO utilise un critère de pseudo-vraisemblance qui est plus efficace d'un point de vue computationnel et permet de proposer des structures multi-échelles. Bien que le critère utilisé soit similaire à celui utilisé dans le clustering convexe supervisé (Hallac et al., 2015; Chu et al., 2021), l'approche proposée peut-être vue comme un problème d'apprentissage multitâche (Chiquet et al., 2011) en raison de son couplage avec l'inférence graphique gaussienne. Dans les applications biologiques, le MGLASSO s'appuie sur diverses transformations de données adaptées à la nature des données, notamment le log ratio centré (Aitchison, 1982) pour les données compositionnelles.

Le MGLASSO, comme l'approche introduite par Yao and Allen (2019), est une méthode qui combine les modèles graphiques gaussiens et le clustering convexe. Cependant, contrairement à leur travail, nous mettons l'accent sur l'approche sélection de voisinage et nous avons proposé d'ajouter une pénalité induisant la sparsité (LASSO). Nous avons également mis à disposition sur le CRAN une version beta d'un package R qui implémente l'approche (Sanou, 2022). L'algorithme proposé met en évidence les structures multi-échelles estimées en faisant varier le paramètre de fusion. Notre approche peut également être considérée comme une extension de la méthode SpiecEasi aux réseaux multi-échelles lorsqu'elle est appliquée aux données de composition avec la transformation log ratio centrée. Nos applications biologiques dans le cadre du projet EPITREE, ont permis entre autres résultats, de mettre en évidence que la marque épigénétique de la méthylation de l'ADN pour les peupliers, peut être utilisée comme marqueur de la structure génétique (Sow et al., 2023).

Chapitre 2

Ce chapitre passe en revue les éléments de théorie utilisés pour la définition et l'optimisation du modèle présenté dans le chapitre suivant. Le cadre des modèles graphiques gaussiens est d'abord introduit avant d'aborder les approches d'inférence existantes dans la littérature. Dans un second temps, la classification convexe, une méthode de regroupement d'observations basée sur un critère convexe est présentée, ainsi que son importance pour le problème d'inférence du modèle graphique gaussien. Finalement, des techniques d'optimisation convexe pour des critères non différentiables sont présentées en l'occurrence, les méthodes de sous-gradient, proximales, et de lissage.

Modèles graphiques non dirigés

Les modèles graphiques probabilistes résultent d'une fusion entre la théorie des probabilités et la théorie des graphes et sont couramment utilisés dans l'analyse de données en grande dimension pour représenter les interactions entre les variables. Ils trouvent des applications dans des domaines variés tels que la physique statistique, la génomique, l'analyse d'images et l'analyse de réseaux sociaux. Ces modèles simplifient les interactions complexes, réduisent les paramètres en mettant en évidence les connexions les plus pertinentes. Les modèles graphiques se composent de noeuds et d'arêtes.

Il existe deux principaux types de modèles graphiques : les champs aléatoires de Markov (MRF), qui sont des modèles graphiques non dirigés, et les réseaux bayésiens, qui sont des modèles graphiques dirigés. Les modèles graphiques peuvent également inclure des graphes mixtes avec plusieurs types d'arêtes : dirigées, non dirigées et bidirigées. Indépendamment de leur nature, les modèles graphiques gravitent autour du concept central d'indépendance conditionnelle. C'est une notion essentielle permettant de traduire la structure du graphe en contraintes probabilistes.

Propriétés de Markov et Factorisation

Definition (Indépendance Conditionnelle). *Soient $\mathbf{X}^A, \mathbf{X}^B, \mathbf{X}^C$ des ensembles de variables aléatoires. $\mathbf{X}^A$ est indépendant de $\mathbf{X}^B$ étant donné $\mathbf{X}^C$ dans une distribution de probabilité jointe P, si et seulement si :*

$$P(\mathbf{X}^A = \mathbf{x}^A, \mathbf{X}^B = \mathbf{x}^B | \mathbf{X}^C = \mathbf{x}^C) = P(\mathbf{X}^A = \mathbf{x}^A | \mathbf{X}^C = \mathbf{x}^C) P(\mathbf{X}^B = \mathbf{x}^B | \mathbf{X}^C = \mathbf{x}^C)$$

pour toutes les valeurs $\mathbf{x} = (\mathbf{x}^A, \mathbf{x}^B, \mathbf{x}^C) \in Val(\mathbf{X})$. On note $\mathbf{X}^A \perp\!\!\!\perp \mathbf{X}^B | \mathbf{X}^C$.

Soit G, un graphe non dirigé, défini par un ensemble de noeuds V et un ensemble d'arêtes E, et dont les arêtes ne portent pas de flèches directionnelles. La structure du graphe est liée à l'indépendance conditionnelle par la notion de séparation.

Definition (Séparation). *Un sous-ensemble S est dit séparer les ensembles A et B dans un graphe si tout chemin de A à B doit passer par S.*

Cela permet d'introduire les trois propriétés de Markov associées aux modèles graphiques non dirigés :

Definition (Propriété de Markov Globale). *Dans un graphe où S sépare les ensembles A et B, X satisfait la propriété de Markov globale par rapport au graphe G si :*

$$X^A \perp\!\!\!\perp X^B | X^S.$$

où $X^A = (X^k) \setminus k \in A$.

La propriété de Markov globale est liée à deux autres propriétés de Markov : celles de Markov locale et de Markov par paires.

La propriété de Markov locale stipule que chaque noeud est conditionnellement indépendant des noeuds non-voisins, étant donné ses noeuds voisins.

La propriété de Markov par paires quant à elle stipule que deux noeuds non adjacents sont conditionnellement indépendants, étant donné leurs voisins communs.

La factorisation, un autre concept important, joue un rôle significatif dans la compréhension des modèles graphiques. Un graphe est considéré comme complet lorsque chaque arête possible existe entre les noeuds. On note $E = \mathcal{P}_2(V)$.

Definition (Clique). *Dans un graphe, un sous-ensemble $\mathcal{C}$ est appelé clique si le sous-graphe $G_\mathcal{C}$ induit par $\mathcal{C}$ est complet.*

Definition (Factorisation). *La fonction de densité f de la distribution de probabilité P par rapport à une mesure produit ν est dite se factoriser par rapport au graphe G si elle peut être représentée comme suit :*

$$f(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \phi_C(x_C),$$

où $\mathcal{C}$ désigne l'ensemble des cliques maximales, $\phi = (\phi_C, c \in \mathcal{C})$ est une collection de fonctions potentielles positives, et Z est une constante de normalisation.

Les propriétés de factorisation des modèles graphiques permettent d'effectuer des calculs tractables sur des distributions multivariées. En représentant une distribution de probabilité jointe comme un produit de facteurs, où chaque facteur ne dépend que d'un sous-ensemble de variables correspondant à une clique dans le graphe, les calculs peuvent être effectués localement sur les cliques et ensuite combinés à l'aide de la factorisation. Cela permet de développer des algorithmes d'inférence et d'apprentissage efficaces pour des modèles larges et complexes.

Modèles graphiques gaussiens

Les modèles graphiques gaussiens (GGM), également connus sous le nom de modèles de sélection de covariance (Lauritzen, 1996), sont une catégorie spéciale de modèles graphiques non dirigés utilisés dans des cadres gaussiens. Soit $\mathbf{X} = (X^1, \ldots, X^p)^T \in \mathbb{R}^p$ un vecteur aléatoire gaussien de dimension p, de moyenne

nulle et de matrice de covariance $\mathbf{\Sigma} \in \mathbb{S}^p_{\succ 0}$, où $\mathbb{S}^p_{\succ 0}$ désigne l'ensemble des matrices réelles symétriques définies positives de taille $p \times p$. Certaines propriétés des modèles graphiques spécifiques aux distributions gaussiennes sont présentées ci-dessous.

Proposition. *La structure d'indépendance conditionnelle de* $\mathbf{X} \sim \mathcal{N}_p(0, \mathbf{\Sigma})$ *est caractérisée par le graphe* G*, qui est uniquement déterminé par le support de la matrice de précision ou de concentration* $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$*.*

Proposition. *Les entrées de la matrice de précision sont proportionnelles aux coefficients de corrélation partielle.*

En effet, la corrélation partielle entre X^i et X^j étant donné $\mathbf{X} \backslash \{X^i, X^j\}$ est égale à $\frac{-\Omega_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}}$. Le lecteur peut se référer à Lauritzen (1996) pour la preuve.

Le corollaire suivant peut être dérivé du fait que les coefficients de corrélation partielle sont directement liés aux coefficients de régression.

Proposition 6.1. *Dans le cadre du problème de régression* $X^i = \mathbf{X}^{\backslash i}\beta^i + \epsilon^i$*, où* ϵ^i *est le vecteur résiduel gaussien, le coefficient de régression est donné par* $\beta^i_k = -\Omega_{ik}/\Omega_{ii}$*.*

La Proposition 6.1 suggère que les modèles graphiques gaussiens peuvent être estimés par une série de régressions, comme cela a été exposé par Meinshausen and Bühlmann (2006). La prochaine section présentera quelques approches d'inférence de modèles graphiques gaussiens.

Inférence des Modèles Graphiques Gaussiens

Soit P une distribution de probabilité inconnue qui se factorise par rapport au graphe G. En utilisant un ensemble d'échantillons indépendants et identiquement distribués (i.i.d) de P, l'objectif de l'apprentissage d'un modèle graphique gaussien est d'estimer les fonctions potentielles qui correspondent le mieux à la distribution (Maathuis et al., 2018). En d'autres termes, on cherche à estimer les arêtes du graphe et les paramètres de la distribution. Nous regroupons les approches d'apprentissage en trois catégories principales : celles sans contraintes de parcimonie, celles basées sur la parcimonie et celles avec des contraintes supplémentaires sur la structure des noeuds.

Estimation du Maximum de Vraisemblance (MLE)

Une approche courante pour estimer un modèle graphique gaussien est d'utiliser l'Estimateur du Maximum de Vraisemblance. L'objectif est de maximiser la fonction de log-vraisemblance concave basée sur les données observées x provenant du vecteur aléatoire X :

$$l(\mathbf{\Omega}) = \sum_{i=1}^{n} \log f(x_i|\mathbf{\Omega}) \propto \log\det(\mathbf{\Omega}) - \mathrm{tr}(\mathbf{\Omega}\mathbf{S}) \tag{6.1}$$

où $\boldsymbol{S} = \frac{1}{n} X^T X$ est la matrice de covariance empirique. La solution à ce problème est unique et donnée par $\hat{\boldsymbol{\Omega}}_{MLE} = \boldsymbol{S}^{-1}$ pour toute matrice $\boldsymbol{S}$ non singulière. Cependant, le MLE n'est pas calculable lorsque le nombre d'échantillons n est inférieur à la dimension p. Même s'il est calculable, il peut être peu performant et donner lieu à des graphes complets. En général, les modèles graphiques gaussiens sont souvent supposés être parcimonieux, ce qui signifie qu'un petit nombre de variables interagissent par rapport à toutes les interactions possibles. Une approche, introduite par Dempster (1972), consiste à estimer la structure du réseau en fixant certains éléments de la matrice de précision $\boldsymbol{\Omega}$ à zéro, simplifiant ainsi la structure des dépendances entre les variables.

Inférence avec des contraintes de parcimonie

Suivant l'idée de Dempster (1972), plusieurs auteurs ont proposé des approches pour retrouver la structure de la matrice de précision en utilisant la parcimonie.

Estimation de la pseudo-vraisemblance pénalisée par Lasso

Meinshausen and Bühlmann (2006) ont introduit une approche basée sur la régression pour la sélection de voisinage. Ils utilisent la régularisation Lasso et régressent chaque variable sur les autres, en tirant parti du lien entre les coefficients de régression et la matrice de précision. Les différentes régressions sont ensuite combinées pour inférer le graphe d'indépendance conditionnelle. Plus tard, des auteurs comme Rocha et al. (2008), Ambroise et al. (2009) ont montré que la sélection de voisinage pouvait être considérée comme une approximation de la vraisemblance globale via une pseudo-vraisemblance.

Estimation de la vraisemblance globale pénalisée par Lasso

D'autres auteurs ont étendu la pénalisation Lasso aux vraisemblances globales du modèle. Contrairement aux approches de sélection de voisinage, qui s'intéressent principalement à l'estimation de la structure du graphe, les approches basées sur la vraisemblance globale permettent d'apprendre la structure du graphe et d'estimer simultanément ses paramètres de manière cohérente. Elles optimisent une fonction convexe en utilisant divers algorithmes.

Cette approche communément connu sous le nom de Graphical Lasso (Banerjee et al., 2008; Friedman et al., 2008) a été largement étudiée (Yuan and Lin, 2007; Ravikumar et al., 2011; Rothman et al., 2008b) et étendue à diverses applications (Chiquet et al., 2019; Charbonnier et al., 2010; Danaher et al., 2014; Robin et al., 2019). Elle reste un outil puissant pour l'inférence des modèles graphiques gaussiens.

Autres pénalités de parcimonie

Dans la littérature sur les modèles graphiques gaussiens, des pénalités de parcimonie alternatives au Lasso ont été étudiées. En présence de variables fortement corrélées, le Lasso peut être moins performant. Ainsi, différentes pénalités, telles que la pénalité ℓ_2, le SCAD (Smoothly Clipped Absolute Deviation), le Lasso adaptatif (Zou, 2006), le Lasso groupé (Yuan and Lin, 2006) et l'Elastic-net Kovács et al. (2021), ont été explorées selon la nature du problème d'inférence.

Inference tout en prenant en compte la structure sous-jacente

Dans un problème de régression pénalisée, le Lasso a tendance à sélectionner une seule variable parmi un groupe de variables corrélées (Bühlmann et al., 2013). Diverses solutions ont été proposées en utilisant différentes pénalités de parcimonie. Parmi elles, l'Elastic-net (Zou and Hastie, 2005) applique une combinaison linéaire des pénalités Lasso et Ridge, favorisant un effet de regroupement et permettant la sélection de groupes de variables. OSCAR (Bondell and Reich, 2008) y parvient en mélangeant des pénalités Lasso et ℓ_∞. Le clustered Lasso (She, 2008) définit un critère généralisé de Fused Lasso (Tibshirani et al., 2005) où il n'y a pas d'ordre spécifique sur les variables.

Dans le problème de l'inférence des graphes, pour surmonter ce problème, en plus de la sparsité, plusieurs travaux antérieurs tentent d'estimer la CIG en intégrant des structures de regroupement parmi les variables, dans un souci d'équilibre statistique et d'interprétabilité. Une liste non exhaustive de travaux qui intègrent une structure de clustering pour accélérer ou améliorer la procédure d'estimation comprend Honorio et al. (2009); Ambroise et al. (2009); Mazumder and Hastie (2012a); Tan et al. (2015); Yao and Allen (2019); Devijver and Gallopin (2018).

Clustering convexe

Étant donné un ensemble de données $X = \{x_1, \dots, x_n\} \in \mathbb{R}^{n \times p}$ et une matrice de centroïdes $\alpha \in \mathbb{R}^{n \times p}$, l'objectif du clustering convexe est de minimiser le critère suivant :

$$\frac{1}{2} \sum_{i=1}^{n} \|x_i - \alpha_i\|_2^2 + \lambda \sum_{i<j} w_{ij} \|\alpha_i - \alpha_j\|_q \tag{6.2}$$

où λ est un paramètre de pénalisation, $\{w_{ij}\}$ sont des poids symétriques positifs, $\alpha_i \in \mathbb{R}^p$ est le centroïde attribué à l'observation x_i, et $\|\cdot\|_q$ est la norme ℓ_q sur $\mathbb{R}^p$ avec $q \geq 1$.

La relation entre cette formulation (6.2) et le clustering k-means est détaillée dans Lindsten et al. (2011); et la relation avec la classification ascendante hiérarchique est établie dans Hocking et al. (2011). Le clustering convexe résout le problème de clustering de manière indépendante pour une plage de paramètres de pénalisation λ. Étant donné que le problème d'optimisation est convexe, la solution finale est indépendante de l'initialisation des centroides.

Contrairement à la parcimonie dans les modèles graphiques gaussiens, utilisée pour la sélection de variables, dans le clustering convexe, la parcimonie est utilisée pour déterminer une structure de clustering. Le terme d'attache aux données (le premier terme) garantit que les centroïdes restent proches des observations qui composent leur groupe. Le terme de régularisation $\|\alpha_i - \alpha_j\|_q$, également connu sous le nom de "terme de fusion", est une pénalité de type fused-group Lasso (Yuan and Lin, 2006; Tibshirani et al., 2005) lorsque $q > 1$ et un terme fused-Lasso lorsque $q = 1$. Cela encourage la parcimonie dans les différences entre

centroïdes.

Le paramètre de pénalisation λ contrôle le compromis entre l'ajustement du modèle et le nombre de clusters. Deux observations x_i et x_j sont considérées comme appartenant au même cluster lorsque leurs centroïdes estimés sont presque identiques, c'est-à-dire $\hat{\alpha}_i \approx \hat{\alpha}_j$. À mesure que λ augmente, la force de fusion augmente également, ce qui provoque la fusion des centroïdes. Pour une valeur de λ suffisamment grande, tous les clusters fusionnent en un seul.

Le chemin de régularisation des solutions, également connu sous le nom de *clusterpath* (Hocking et al., 2011) et similaire au dendrogramme de la classification ascendante hiérarchique, peut être obtenu après une tâche de clustering convexe.

Chapitre 3

Dans ce chapitre, nous introduisons la méthode d'inférence de graphe dénommée Lasso Graphique multi-échelle (MGLASSO, Sanou et al.), une nouvelle approche permettant d'estimer simultanément une structure de clustering hiérarchique et des graphes d'indépendance conditionnelle entre des variables, à plusieurs niveaux de granularité.

Modèle

Les outils ayant servi à la construction du modèle sont inspirés de la sélection de voisinage via le Lasso introduite par Meinshausen and Bühlmann (2006) et de la théorie du clustering convexe (Hocking et al., 2011; Lindsten et al., 2011; Pelckmans et al., 2005).

Nous proposons de fusionner les p régressions LASSO indépendantes de l'approche de Meinshausen and Bühlmann (2006) en un seul critère d'optimisation avec ajout d'une pénalité de fusion dans ℓ_2 sur les vecteurs de régression considérés comme centres de cluster. Plus précisément, le problème de pseudo-vraisemblance du MGLASSO minimise, dans le cadre gaussien, la quantité suivante :

$$\mathcal{J}_{\lambda_1,\lambda_2}(\boldsymbol{\beta};\mathbf{X}) = \frac{1}{2}\sum_{i=1}^{p}\left\|\mathbf{X}^i - \mathbf{X}^{\setminus i}\boldsymbol{\beta}^i\right\|_2^2 + \lambda_1\sum_{i=1}^{p}\left\|\boldsymbol{\beta}^i\right\|_1 + \lambda_2\sum_{i<j}\left\|\boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j\right\|_2,$$

par rapport à $\boldsymbol{\beta} := [\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^p] \in \mathbb{R}^{(p-1)\times p}$, où $\mathbf{X}^i \in \mathbb{R}^n$ désigne la i-ème colonne de $\mathbf{X}$, λ_1 et λ_2 sont des paramètres de pénalisation, $\boldsymbol{\tau}_{ij} \in \mathbb{R}^{(p-1)\times(p-1)}$ est une matrice de permutation, qui permute les coefficients dans le vecteur de régression $\boldsymbol{\beta}^j$ comme suit :

$$\left\|\boldsymbol{\beta}^i - \boldsymbol{\tau}_{ij}\boldsymbol{\beta}^j\right\|_2 = \sqrt{\sum_{k\in\{1,\ldots,p\}\setminus\{i,j\}}(\beta_k^i - \beta_k^j)^2 + (\beta_j^i - \beta_i^j)^2}.$$

Le critère MGLASSO peut être vu comme un problème de régression multi-tâches où l'ensemble des réponses est identique à l'ensemble des prédicteurs. Le terme

de pénalisation LASSO encourage la parcimonie dans les coefficients estimés. Le terme fused-group LASSO encourage la fusion dans les coefficients de régression $\boldsymbol{\beta}^i$ et $\boldsymbol{\beta}^j$. Le MGLASSO permet ainsi de mettre en évidence de façon simultanée les interactions entre variables tout en proposant des groupes. Les variables qui appartiennent au même groupe sont suceptibles d'avoir la même structure de voisinage.

Optimisation

La fonction objective introduite est la somme de trois composantes convexes : une fonction différentiable (perte quadratique) et deux fonctions de pénalité non différentiables : la fonction de pénalité LASSO, qui est séparable car décomposable en composantes indépendantes correspondant chacune aux coefficients des prédicteurs, et la pénalité group-fused LASSO, qui n'est pas séparable. Plusieurs approches peuvent être utilisées pour résoudre ce type de problème de minimisation. Parmi celles-ci, se trouvent entre autres la méthode du sous-gradient (Shor, 2012), la méthode ADMM Boyd et al. (2011) ou encore les méthodes de continuation combinées avec des techniques de lissage pour les parties non différentiables du critère telle que CONESTA (Hadj-Selem et al., 2018).

Nous avons comparé les performances de convergence empirique de ces 3 approches d'optimisation.

Sauf pour le cas de la méthode du sous-gradient, les algorithmes sont généralement appliqués à des versions reformulées du critère initial MGLASSO, en un problème de régression unique. Avec l'algorithme d'optimisation CONESTA (Hadj-Selem et al., 2018), le MGLASSO atteint une vitesse de convergence plus rapide que les autres approches concurrentes. Une brève analyse empirique de la convergence a été présentée à cette fin. En pratique, le MGLASSO est implémenté dans le package R **mglasso** (version $0.1.2$, Sanou (2022)). La pénalité LASSO est gardée fixe et le paramètre de pénalité group-fused LASSO varie dans une grille de valeurs. Pour un certain seuil ϵ_{fuse}, les variables i et j sont assignées au même groupe si $d(i,j) = \left\| \hat{\boldsymbol{\beta}}^i - \boldsymbol{\tau}_{ij} \hat{\boldsymbol{\beta}}^j \right\|_2 \leq \epsilon_{fuse}$.

La sélection du paramètre de régularisation LASSO est faite via l'approche StARS (Liu et al., 2010) qui est une méthode de sélection basée sur une technique de ré-échantillonnage, en gardant le paramètre de pénalité group-fused LASSO nul. Nous avons également discuté d'autres méthodes de sélection qui ont été proposées pour le problème de sélection de modèle de modèles graphiques gaussiens telles que le BIC étendu (EBIC, Foygel and Drton (2010)), la validation croisée (Bien and Tibshirani, 2011) et la méthode GGMSelect (Giraud et al., 2012).

Performances

Nous avons mené une étude de simulation pour évaluer les performances de la méthode MGLASSO, tant en terme de clustering que de recouvrement de support. Les courbes ROC sont utilisées pour évaluer l'adéquation des graphes inférés avec les graphes de référence, pour le MGLASSO et le GLASSO dans sa version

de sélection de voisinage, dans les cadres des modèles Erdös-Rényi (Erdős et al., 1960), scale-free (Newman et al., 2001), et des modèles blocs stochastiques (SBM, Fienberg and Wasserman (1981)). Les indices de Rand ajustés sont utilisés pour comparer les partitions obtenues avec MGLASSO, la classification hiérarchique ascendante, et la classification k-means pour des modèles bloc stochastique et hiérarchique. Le MGLASSO ne fait pas pire que le GLASSO en terme de performance. En présence de l'ajout d'une pénalité de fusion, l'approche MGLASSO donne lieu à des performances meilleures en terme de courbe ROC.

Chapitre 4

Ce chapitre se concentre sur les applications du modèle MGLASSO et d'autres modèles statistiques sur les données omiques. Diverses questions biologiques sont abordées, en particulier dans le cadre du projet EPITREE, qui vise à étudier l'impact évolutif et fonctionnel de la variation épigénétique chez les arbres forestiers. La méthode MGLASSO est illustrée dans une analyse intégrative des données transcriptomiques et de méthylation d'ADN provenant du projet EPITREE.

L'impact évolutif et fonctionnel des variations épigénétiques chez les arbres forestiers (EPITREE, ANR-17-CE32) est un projet de recherche forestière qui se concentre sur la façon dont les variations génétiques et épigénétiques contribuent à la plasticité phénotypique et à l'adaptation à l'environnement local. EPITREE est né de la nécessité de comprendre les mécanismes sous-jacents à l'adaptation des arbres forestiers afin de mieux gérer les ressources génétiques. En effet, au cours des dernières années, un dépérissement généralisé des forêts a été observé en raison des contraintes liées à la sécheresse. Ces arbres jouent un rôle essentiel dans l'équilibre des écosystèmes de la planète.

Le projet s'intéresse aux bases génétiques de l'adaptation locale des arbres. Cependant, les études existantes se concentrent généralement sur la contribution des SNP. Les mécanismes épigénétiques ne sont pas étudiés en profondeur. La nature dynamique de l'épigénome en fait un sujet intéressant à étudier chez ces organismes à longue durée de vie. EPITREE se concentre sur l'étude des variations de la méthylation de l'ADN, de l'expression des gènes et des variations structurelles du génome afin de mieux comprendre la contribution des variations épigénétiques à l'adaptation locale et à la plasticité phénotypique. Le projet est subdivisé en plusieurs modules qui s'articulent autour de l'identification de régions épigénomiques candidates, de la caractérisation des variations épigénomiques dans les populations naturelles et de leurs conséquences fonctionnelles, de la caractérisation de la plasticité épigénomique et de ses conséquences fonctionnelles en réponse aux contraintes environnementales, de la génération de données et de l'analyse multiomique intégrative.

Comprendre le schéma évolutif des espèces est essentiel en biologie, en particulier pour les organismes à longue durée de vie tels que les arbres. L'étude

des variations génétiques et épigénétiques des populations naturelles de ces arbres peut contribuer à mettre en lumière leurs mécanismes d'adaptation à leur environnement local, ce qui est pertinent dans le contexte actuel du changement climatique. L'intérêt simultané pour les variations génétiques et épigénétiques est un domaine de recherche récemment exploré, centré sur les plantes annuelles, qui sont des organismes à courte durée de vie (Sow et al., 2018). La caractérisation de ces variations pour les arbres reste une question ouverte qui peut apporter une valeur ajoutée dans la gestion forestière (Amaral et al., 2020). L'un des objectifs du projet EPITREE est d'apporter des réponses à cette préoccupation en produisant des données diversifiées sur des populations d'arbres situées sur différents sites géographiques, en se concentrant sur le chêne et le peuplier.

Dans le cadre de la recherche doctorale, nous avons contribué à mettre en évidence la structure génétique et épigénétique des populations de peupliers grâce aux données SNP et aux données de méthylation de l'ADN, respectivement. Nous montrons également comment la méthylation est utilisée comme marqueur de la différenciation des populations en proposant des gènes dont le profil de méthylation suit de près la structure génétique des populations. Nous étudions brièvement le lien entre la méthylation et les profils d'expression pour une classe spécifique de gènes dont les profils d'expression sont stables entre les arbres appartenant aux mêmes métapopulations. Ces domaines d'analyse choisis résultent de discussions avec des experts du projet EPITREE. Nous avons également exploré brièvement la structure et l'interaction entre les profils transcriptomiques et de méthylation dans différents contextes de méthylation à l'aide du modèle MGLASSO.

7 - Appendix

7.1 Link between neighbourhood selection and pseudo-likelihood optimization

Proof. The pseudo log-likelihood pl of a multivariate normal random vector $\mathbf{X} \sim \mathcal{N}_p(\mu, \mathbf{S})$ is given by

$$pl(\mathbf{\Sigma}; \mathbf{X}) = \sum_{i=1}^{n} \sum_{j=1}^{p} \log p(X_i^j | \mathbf{X}_i^{-j}). \tag{7.1}$$

This function is concave with respect to the parameters μ and $\mathbf{\Sigma}$. Now let's derive the normal conditionals.

Given the set of random variables $\mathbf{X}^{\setminus j} = \boldsymbol{Z}$, the conditional distribution of node $\mathbf{X}^j = Y$ is gaussian (Lauritzen, 1996). We can partitioned $\mathbf{\Sigma}$ in four blocks

$$\mathbf{\Sigma} = \begin{pmatrix} \Sigma_{YY} & \Sigma_{YZ} \\ \Sigma_{ZY} & \mathbf{\Sigma}_{ZZ} \end{pmatrix}$$

and thus have,

$$\begin{aligned} Y|\boldsymbol{Z} = z &\sim \mathcal{N}(\mu_{Y|Z}, \quad \Sigma_{Y|Z}) \\ &\sim \mathcal{N}(\mu_Y + (z - \mu_Z)^T \mathbf{\Sigma}_{ZZ}^{-1} \Sigma_{ZY}, \quad \Sigma_{YY} - \Sigma_{ZY}^T \mathbf{\Sigma}_{ZZ}^{-1} \Sigma_{ZY}) \end{aligned} \tag{7.2}$$

Using Schur complement for block inverse matrix and partitionning $\mathbf{\Omega}$ as we did for $\mathbf{\Sigma}$, we have the following

$$Y|\boldsymbol{Z} = z \sim \mathcal{N}(\mu_Y + (z - \mu_Z)^T (-\mathbf{\Omega}_{YZ}/\Omega_{YY}), \quad \Omega_{YY}^{-1})$$

Let $\mu = (\mu_Y, \mu_Z) = 0_p$ for the sake of simplicity. The log-likelihood of the univariate-conditional normal distribution is given by

$$\begin{aligned} \log p(Y|\boldsymbol{Z}) &= \log \left(\frac{1}{(2\pi\Sigma_{Y|Z})^{1/2}} \exp \left(-\frac{1}{2} \frac{(y - \mu_{Y|Z})^2}{\Sigma_{Y|Z}} \right) \right) \\ &= \frac{1}{2} \log(\Sigma_{Y|Z}^{-1}) - \frac{1}{2} \Sigma_{Y|Z}^{-1} (y - \mu_{Y|Z})^2 - \frac{1}{2} \log(2\pi) \\ &= \frac{1}{2} \log(\Omega_{YY}) - \frac{1}{2} \Omega_{YY} \left(y + \frac{\mathbf{\Omega}_{YZ}}{\Omega_{YY}} z \right)^2 + \text{const} \end{aligned}$$

Let's fix Ω_{YY} to a constant and denote $\beta = -\mathbf{\Omega}_{YZ}/\Omega_{YY}$, we have:

$$\log p(Y|\boldsymbol{Z}) = -\frac{1}{2} \Omega_{YY} (y - Z\beta)^2 + \text{const}$$

Penalizing the pseudo-likelihood with a sparsity term and coming back to our original notations amounts to:

$$\begin{aligned} pl(\mathbf{\Omega}; \mathbf{X}) - \lambda_1 ||\mathbf{\Omega}||_1 &= \sum_{i=1}^{n} \sum_{j=1}^{p} \frac{-\Omega_{jj}}{2} \left((X_i^j - X_i^{-j} \boldsymbol{\beta}^j)^2 + \lambda_1 ||\boldsymbol{\beta}^j||_1 \right) \\ &= \sum_{j=1}^{p} \frac{-\Omega_{jj}}{2} \left(||\mathbf{X}^j - X^{-j} \boldsymbol{\beta}^j||_2^2 + \lambda_1 ||\boldsymbol{\beta}^j||_1 \right) \end{aligned} \tag{7.3}$$

with diagonal entries of Ω_{jj} not estimated. □

Bibliography

John Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, 1982.

Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, pages 199–213. Springer, 1998.

Altuna Akalin. *Computational Genomics with r*. Chapman and Hall/CRC, 2020.

Joana Amaral, Zoé Ribeyre, Julien Vigneaud, Mamadou Dia Sow, Régis Fichot, Christian Messier, Gloria Pinto, Philippe Nolet, and Stéphane Maury. Advances and promises of epigenetics for forest trees. *Forests*, 11(9):976, 2020.

Christophe Ambroise, Julien Chiquet, and Catherine Matias. Inferring sparse gaussian graphical models with latent structure. *Electronic Journal of Statistics*, 3(0):205–238, oct 2009. ISSN 19357524. doi: 10.1214/08-EJS314. URL http://arxiv.org/abs/0810.3177http://dx.doi.org/10.1214/08-EJS314http://projecteuclid.org/euclid.ejs/1238078905.

William RL Anderegg, Tamir Klein, Megan Bartlett, Lawren Sack, Adam FA Pellegrini, Brendan Choat, and Steven Jansen. Meta-analysis reveals that hydraulic traits explain cross-species patterns of drought-induced tree mortality across the globe. *Proceedings of the National Academy of Sciences*, 113(18):5024–5029, 2016.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 5:19–53, 2011.

Samuel Balmand and Arnak S Dalalyan. On estimation of the diagonal elements of a sparse precision matrix. *Electronic Journal of Statistics*, 10(1):1551–1579, 2016.

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *JMLR.org*, 9:485–516, June 2008. ISSN 1532-4435.

David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Paul Bastide, Mahendra Mariadassou, and Stéphane Robin. Detection of adaptive shifts on phylogenies by using shifted stochastic processes on a tree. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1067–1093, 2017.

Jean-Patrick Baudry, Cathy Maugis, and Bertrand Michel. Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2):455–470, 2012.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2:183–202, 01 2009. doi: 10.1137/080716542.

Alexandre Belloni, Victor Chernozhukov, and Lie Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

Julian Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3):179–195, 1975.

Antoine Bichat, Christophe Ambroise, and Mahendra Mariadassou. Hierarchical correction of p-values via a tree running ornstein-uhlenbeck process. *ArXiv e-prints*, 2020.

Jacob Bien and Robert J Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, 98(4):807–820, 2011.

Howard D Bondell and Brian J Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1): 115–123, 2008.

Jonathan Borwein and Adrian Lewis. *Convex Analysis*. Springer, 2006.

Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL https://doi.org/10.1561/2200000016.

Toby JA Bruce, Michaela C Matthes, Johnathan A Napier, and John A Pickett. Stressful "memories" of plants: evidence and possible mechanisms. *Plant science*, 173(6):603–608, 2007.

GH Bryan. Elementary principles in statistical mechanics. *Nature*, 66(1708):291–292, 1902.

Alessandro Buccini, Pietro Dell'Acqua, and Marco Donatelli. A general framework for admm acceleration. *Numerical Algorithms*, 85(3):829–848, 2020.

Peter Bühlmann, Philipp Rütimann, Sara Van De Geer, and Cun-Hui Zhang. Correlated variables in regression: clustering and sparse estimation. *Journal of Statistical Planning and Inference*, 143(11):1835–1858, 2013.

Tony Cai, Weidong Liu, and Xi Luo. A constrained l1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011. doi: $10.1198/\mathrm{jasa.2011.tm10155}$. URL https://doi.org/10.1198/jasa.2011.tm10155.

Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The annals of Statistics*, 35(6):2313–2351, 2007.

Stanley H Chan, Xiran Wang, and Omar A Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2016.

F Stuart Chapin Iii, Erika S Zavaleta, Valerie T Eviner, Rosamond L Naylor, Peter M Vitousek, Heather L Reynolds, David U Hooper, Sandra Lavorel, Osvaldo E Sala, Sarah E Hobbie, et al. Consequences of changing biodiversity. *Nature*, 405(6783):234–242, 2000.

Camille Charbonnier, Julien Chiquet, and Christophe Ambroise. Weighted-lasso for structured network inference from time course data. *Statistical applications in genetics and molecular biology*, 9(1), 2010.

Aurélien Chateigner, Marie-Claude Lesage-Descauses, Odile Rogier, Véronique Jorge, Jean-Charles Leplé, Véronique Brunaud, Christine Paysant-Le Roux, Ludivine Soubigou-Taconnat, Marie-Laure Martin-Magniette, Léopoldo Sanchez, et al. Gene expression predictions and networks in natural populations supports the omnigenic theory. *BMC genomics*, 21(1):1–16, 2020.

Lulu Cheng, Liang Shan, and Inyoung Kim. Multilevel Gaussian graphical model for multilevel networks. *Journal of Statistical Planning and Inference*, 190:1–14, nov 2017. doi: $10.1016/\mathrm{j.jspi.2017.05.003}$.

Eric C Chi and Kenneth Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, 2015.

Eric C Chi and Stefan Steinerberger. Recovering trees with convex clustering. *SIAM Journal on Mathematics of Data Science*, 1(3):383–407, 2019.

Julien Chiquet, Yves Grandvalet, and Christophe Ambroise. Inferring multiple graphical structures. *Statistics and Computing*, 21(4):537–553, 2011.

Julien Chiquet, Pierre Gutierrez, and Guillem Rigaill. Fast tree inference with weighted fusion penalties. *Journal of Computational and Graphical Statistics*, 26(1):205–216, 2017a.

Julien Chiquet, Tristan Mary-Huard, and Stéphane Robin. Structured regularization for conditional gaussian graphical models. *Statistics and Computing*, 27(3):789–804, 2017b.

Julien Chiquet, Guillem Rigaill, and Martina Sundqvist. A multiattribute gaussian graphical model for inferring multiscale regulatory networks: an application in breast cancer. In *Gene Regulatory Networks*, pages 143–160. Springer, 2019.

Shuyu Chu, Huijing Jiang, Zhengliang Xue, and Xinwei Deng. Adaptive convex clustering of generalized linear models with application in purchase likelihood prediction. *Technometrics*, 63(2):171–183, 2021.

Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

Alexandre d'Aspremont, Onureena Banerjee, and Laurent El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66, 2008.

James H Degnan and Laura A Salter. Gene tree distributions under the coalescent process. *Evolution*, 59(1):24–37, 2005.

David Degras. Sparse group fused lasso for model segmentation: a hybrid approach. *Advances in Data Analysis and Classification*, 15(3):625–671, 2021.

A. P. Dempster. Covariance Selection. *Biometrics*, 28(1):157, mar 1972. ISSN 0006341X. doi: 10.2307/2528966. URL https://www.jstor.org/stable/2528966?origin=crossref.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Emilie Devijver and Mélina Gallopin. Block-Diagonal Covariance Selection for High-Dimensional Gaussian Graphical Models. *Journal of the American Statistical Association*, 113(521):306–314, jan 2018. ISSN 1537274X. doi: 10.1080/01621459.2016.1247002.

John Duchi, Stephen Gould, and Daphne Koller. Projected subgradient methods for learning sparse gaussians. *arXiv preprint arXiv:1206.3249*, 2012.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

L El Ghaoui, V Viallon, and T Rabbani. Safe feature elimination in sparse supervised learning technical report no. *Technical report, UCB/EECS-2010–126, EECS Department, University of California, Berkeley*, 2010.

Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96 (456):1348–1360, 2001.

Jianqing Fan, Yang Feng, and Yichao Wu. Network exploration via the adaptive lasso and scad penalties. *The annals of applied statistics*, 3(2):521, 2009.

Stephen E Fienberg and Stanley S Wasserman. Categorical data analysis of single sociometric relations. *Sociological methodology*, 12:156–192, 1981.

Rina Foygel and Mathias Drton. Extended bayesian information criteria for gaussian graphical models. *arXiv preprint arXiv:1011.6640*, 2010.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Applications of the lasso and grouped lasso to the estimation of sparse graphical models. Technical report, Technical report, Stanford University, 2010.

Apratim Ganguly and Wolfgang Polonik. Local neighborhood fusion in locally constant gaussian graphical models, 2014.

Christophe Giraud, Sylvie Huet, and Nicolas Verzelen. Graph selection with ggmselect. *Statistical applications in genetics and molecular biology*, 11(3), 2012.

Quentin Grimonprez, Samuel Blanck, Alain Celisse, and Guillemette Marot. Mlgl: An r package implementing correlated variable selection by hierarchical clustering and group-lasso. 2018.

Fouad Hadj-Selem, Tommy Lofstedt, Elvis Dohmatob, Vincent Frouin, Mathieu Dubois, Vincent Guillemot, and Edouard Duchesnay. Continuation of Nesterov's Smoothing for Regression with Structured Sparsity in High-Dimensional Neuroimaging. *IEEE Transactions on Medical Imaging*, 2018, 2018. doi: 10.1109/TMI.2018.2829802. URL https://hal-cea.archives-ouvertes.fr/cea-01883286.

David Hallac, Jure Leskovec, and Stephen Boyd. Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 387–396, 2015.

Luke Harmon. Phylogenetic comparative methods: learning from trees. 2019.

John A Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical learning with sparsity. *Monographs on statistics and applied probability*, 143:143, 2015.

T. Hocking, Jean-Philippe Vert, F. Bach, and Armand Joulin. Clusterpath: an algorithm for clustering using convex fusion penalties. In *ICML*, 2011.

Holger Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010. doi: 10.1198/jcgs.2010.09208. URL https://doi.org/10.1198/jcgs.2010.0920.

Jean Honorio, Dimitris Samaras, Nikos Paragios, Rita Goldstein, and Luis E Ortiz. Sparse and locally constant gaussian graphical models. *Advances in Neural Information Processing Systems*, 22:745–753, 2009.

Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, Pradeep K Ravikumar, and Russell Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. *Advances in neural information processing systems*, 26, 2013.

Cho-Jui Hsieh, Mátyás A. Sustik, Inderjit S. Dhillon, and Pradeep Ravikumar. Quic: Quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research*, 15(83):2911–2947, 2014. URL http://jmlr.org/papers/v15/hsieh14a.html.

Tao Jiang and Stephen Vavasis. On identifying clusters from sum-of-norms clustering computation. *arXiv preprint arXiv:2006.11355*, 2020.

Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

Daphne Koller and Nir Friedman. Probabilistic Graphical Models: Principles. *Italica*, 51(3):327, 2009. ISSN 00213020. doi: 10.2307/478142.

Solt Kovács, Tobias Ruckstuhl, Helena Obrist, and Peter Bühlmann. Graphical elastic net and target matrices: Fast algorithms and software for sparse precision matrix estimation. *arXiv preprint arXiv:2101.02148*, 2021.

MO Kuismin, JT Kemppainen, and MJ Sillanpää. Precision matrix estimation with rope. *Journal of Computational and Graphical Statistics*, 26(3):682–694, 2017.

Zachary D. Kurtz, Christian L. Müller, Emily R. Miraldi, Dan R. Littman, Martin J. Blaser, and Richard A. Bonneau. Sparse and compositionally robust inference of microbial ecological networks. *PLOS Computational Biology*, 11:1–25, 05 2015. doi: 10.1371/journal.pcbi.1004226. URL https://doi.org/10.1371/journal.pcbi.1004226.

Thomas LaFramboise. Single nucleotide polymorphism arrays: a decade of biological, computational and technological advances. *Nucleic acids research*, 37(13): 4181–4193, 2009.

Gina F Lamka, Avril M Harder, Mekala Sundaram, Tonia S Schwartz, Mark R Christie, J Andrew DeWoody, Janna R Willoughby, et al. Epigenetics in ecology, evolution, and conservation. *Frontiers in Ecology and Evolution*, 2022.

Godfrey N Lance and William Thomas Williams. A general theory of classificatory sorting strategies: 1. hierarchical systems. *The computer journal*, 9(4):373–380, 1967.

Steffen L. Lauritzen. *Graphical models*. Clarendon Press, 1996. ISBN 9780198522195. URL https://global.oup.com/academic/product/graphical-models-9780198522195?cc=fr&lang=en&.

Meixia Lin, Defeng Sun, Kim-Chuan Toh, and Chengjing Wang. Estimation of sparse gaussian graphical models with hidden clustering structure. *arXiv preprint arXiv:2004.08115*, 2020.

F. Lindsten, H. Ohlsson, and L. Ljung. Clustering using sum-of-norms regularization: With application to particle filter output computation. In *2011 IEEE Statistical Signal Processing Workshop (SSP)*, pages 201–204, 2011. doi: 10.1109/SSP.2011.5967659.

Han Liu and Lie Wang. Tiger: A tuning-insensitive approach for optimally estimating gaussian graphical models. *Electronic Journal of Statistics*, 11(1):241–294, 2017.

Han Liu, Kathryn Roeder, and Larry Wasserman. Stability approach to regularization selection (StARS) for high dimensional graphical models. *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, pages 1–14, 2010. ISSN 1049-5258.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Zhaosong Lu. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2000–2016, 2010.

Tommy Löfstedt, Vincent Guillemot, Vincent Frouin, Edouard Duchesnay, and Fouad Hadj-Selem. Simulated data for linear regression with structured and sparse penalties: Introducing pylearn-simulate. *Journal of Statistical Software*, 87(3):1–33, 2018. doi: 10.18637/jss.v087.i03. URL https://www.jstatsoft.org/index.php/jss/article/view/v087i03.

Marloes Maathuis, Mathias Drton, Steffen Lauritzen, and Martin Wainwright. *Handbook of graphical models*. CRC Press, 2018.

J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297, 1967.

Claudia Manzoni, Demis A Kia, Jana Vandrovcova, John Hardy, Nicholas W Wood, Patrick A Lewis, and Raffaele Ferrari. Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences. *Briefings in bioinformatics*, 19(2):286–302, 2018.

Benjamin M Marlin and Kevin P Murphy. Sparse gaussian graphical models with unknown block structure. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 705–712, 2009.

Stéphane Maury, Régis Fichot, MD Sow, Alain Delaunay, I Le Jan, G Laskar, Marie-Claude Lesage Descauses, Corinne Buret, Vanina Guerin, Odile Rogier, et al. Epigenetics in forest trees: role in plasticity, adaptation and potential implications for breeding in a context of climate change (epitree). 2019.

Rahul Mazumder and Trevor Hastie. The graphical lasso: New insights and alternatives. *Electronic Journal of Statistics*, 6(none):2125 – 2149, 2012a. doi: 10.1214/12-EJS740. URL https://doi.org/10.1214/12-EJS740.

Rahul Mazumder and Trevor Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *The Journal of Machine Learning Research*, 13(1):781–794, 2012b.

Daniel McDonald, Embriette Hyde, Justine W Debelius, James T Morton, Antonio Gonzalez, Gail Ackermann, Alexander A Aksenov, Bahar Behsaz, Caitriona Brennan, Yingfeng Chen, et al. American gut: an open platform for citizen science microbiome research. *Msystems*, 3(3):e00031–18, 2018.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3):1436–1462, jun 2006. ISSN 00905364. doi: 10.1214/009053606000000281. URL http://projecteuclid.org/euclid.aos/1152540754.

Xochitl C Morgan and Curtis Huttenhower. Chapter 12: Human microbiome analysis. *PLoS computational biology*, 8(12):e1002808, 2012.

Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

Pierre Neuvial. *Contributions to statistical inference from genomic data*. PhD thesis, Université Toulouse III Paul Sabatier, 2020.

Pierre Neuvial, Henrik Bengtsson, and Terence Paul Speed. Statistical analysis of Single Nucleotide Polymorphism microarrays in cancer studies. In *Handbook of Statistical Bioinformatics*, Springer Handbooks of Computational Statistics. April 2011. doi: 10.1007/978-3-642-16345-6_11. URL https://hal.archives-ouvertes.fr/hal-00497273. 35 pages.

Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2): 026118, 2001.

Ashkan Panahi, Devdatt Dubhashi, Fredrik D Johansson, and Chiranjib Bhattacharyya. Clustering by sum of norms: Stochastic incremental algorithm, convergence and cluster recovery. In *International conference on machine learning*, pages 2769–2777. PMLR, 2017.

Emmanuel Paradis, Julien Claude, and Korbinian Strimmer. Ape: analyses of phylogenetics and evolution in r language. *Bioinformatics*, 20(2):289–290, 2004.

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

Mee Young Park, Trevor Hastie, and Robert Tibshirani. Averaged gene expressions for regression. *Biostatistics*, 8(2):212–227, 05 2006. ISSN 1465-4644. doi: 10.1093/biostatistics/kxl002. URL https://doi.org/10.1093/biostatistics/kxl002.

Kristiaan Pelckmans, Joseph De Brabanter, Johan AK Suykens, and Bart De Moor. Convex clustering shrinkage. In *PASCAL workshop on statistics and optimization of clustering workshop*, 2005.

Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009. doi: 10.1198/jasa.2009.0126. URL https://doi.org/10.1198/jasa.2009.0126. PMID: 19881892.

Christophe Plomion, Catherine Bastien, Marie-Béatrice Bogeat-Triboulot, Laurent Bouffier, Annabelle Déjardin, Sébastien Duplessis, Bruno Fady, Myriam Heuertz, Anne-Laure Le Gac, Grégoire Le Provost, et al. Forest tree genomics: 10 achievements from the past 10 years and future prospects. *Annals of Forest Science*, 73(1):77–103, 2016.

Renfrey Burnard Potts. Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pages 106–109. Cambridge University Press, 1952.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022. URL https://www.R-project.org/.

Peter Radchenko and Gourab Mukherjee. Convex clustering via l1 fusion penalization. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1527–1546, 2017.

Andrea Rau. *Statistical methods and software for the analysis of transcriptomic data*. PhD thesis, Université d'Évry-Val-d'Essonne, 2017.

Andrea Rau, Cathy Maugis Rabusseau, and Antoine Godichon-Baggioni. coseq, 2017. URL https://hal.inrae.fr/hal-02786078.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

LKPJ Rdusseeun and P Kaufman. Clustering by means of medoids. In *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*, volume 31, 1987.

Olivier Rey, Etienne Danchin, Marie Mirouze, Céline Loot, and Simon Blanchet. Adaptation to global change: a transposable element–epigenetics perspective. *Trends in ecology & evolution*, 31(7):514–526, 2016.

Genevieve Robin, Christophe Ambroise, and Stéphane Robin. Incomplete graphical model inference via latent tree aggregation. *Statistical Modelling*, 19(5):545–568, 2019.

Stéphane Robin and Christophe Ambroise. Applications in genomics. In *Handbook of Mixture Analysis*, pages 439–461. Chapman and Hall/CRC, 2019.

Mark D Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of rna-seq data. *Genome biology*, 11(3):1–9, 2010.

Mark D. Robinson, Abdullah Kahraman, Charity W. Law, Helen Lindsay, Malgorzata Nowicka, Lukas M. Weber, and Xiaobei Zhou. Statistical methods for detecting differentially methylated loci and regions. *Frontiers in Genetics*, 5, 2014a. ISSN 1664-8021. doi: 10.3389/fgene.2014.00324. URL https://www.frontiersin.org/articles/10.3389/fgene.2014.00324.

Mark D Robinson, Abdullah Kahraman, Charity W Law, Helen Lindsay, Malgorzata Nowicka, Lukas M Weber, and Xiaobei Zhou. Statistical methods for detecting differentially methylated loci and regions. *Frontiers in genetics*, 5:324, 2014b.

Guilherme V. Rocha, Peng Zhao, and Bin Yu. A path following algorithm for sparse pseudo-likelihood inverse covariance estimation (splice), 2008.

Florian Rohart, Benoît Gautier, Amrit Singh, and Kim-Anh Lê Cao. mixomics: An r package for 'omics feature selection and multiple data integration. *PLoS computational biology*, 13(11):e1005752, 2017.

Adam J. Rothman, Peter J. Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2(none): 494 – 515, 2008a. doi: 10.1214/08-EJS176. URL https://doi.org/10.1214/08-EJS176.

Adam J Rothman, Peter J Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008b.

Kayvan Sadeghi and Steffen Lauritzen. Markov properties for mixed graphs. *Bernoulli*, 20(2):676–696, 2014.

Edmond Sanou. *mglasso: Multiscale Graphical Lasso*, 2022. URL https://cran.r-project.org/package=mglasso/. R package version 0.1.2.

Edmond Sanou, Christophe Ambroise, and Geneviève Robin. Inference of Multiscale Gaussian Graphical Models. *Computo*. ISSN 2824-7795. doi: 10.57750/1f4p-7955. URL https://computo.sfds.asso.fr/published-202306-sanou-multiscale_glasso.

Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

Yiyuan She. *Sparse regression with exact clustering*. Stanford University, 2008.

Haipeng Shen and Jianhua Z Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis*, 99 (6):1015–1034, 2008.

Naum Zuselevich Shor. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012.

Charlotte Soneson and Mauro Delorenzi. A comparison of methods for differential expression analysis of rna-seq data. *BMC bioinformatics*, 14(1):1–18, 2013.

Mamadou Dia Sow. *Rôle fonctionnel de l'épigénétique (Méthylation de l'ADN) dans la réponse du peuplier à des variations de disponibilité en eau du sol*. PhD thesis, Université d'Orléans, 2019.

Mamadou Dia Sow, Vincent Segura, Sylvain Chamaillard, Véronique Jorge, Alain Delaunay, Clément Lafon-Placette, Régis Fichot, Patricia Faivre-Rampant, Marc Villar, Franck Brignolas, et al. Narrow-sense heritability and pst estimates of dna methylation in three populus nigra l. populations under contrasting water availability. *Tree Genetics & Genomes*, 14(5):1–12, 2018.

Mamadou Dia Sow, Odile Rogier, Isabelle Lesur, Christian Daviaud, Emile Mardoc, Edmond Sanou, Ludovic Duvaux, Peter Civan, Alain Delaunay, Marie-Claude Lesage-Descauses, Vanina Benoit, Isabelle Le-Jan, Corinne Buret, Celine Besse, Harold Durufle, Régis Fichot, Grégoire Le-Provost, Erwan Guichoux, Christophe Boury, Abel Garnier, Abdeljalil Senhaji-Rachik, Véronique Jorge, Christophe Ambroise, Jorg Tost, Christophe Plomion, Vincent Segura, Stéphane Maury, and Jérôme Salse. Epigenetic variation in tree evolution: a case study in black poplar (populus nigra). *bioRxiv*, 2023. doi: 10.1101/2023.07.16.549253. URL https://www.biorxiv.org/content/early/2023/07/18/2023.07.16.549253.

Douglas Steinley. K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, 59(1):1–34, 2006.

Defeng Sun, Kim-Chuan Toh, and Yancheng Yuan. Convex clustering: Model, theoretical guarantee and efficient algorithm. *J. Mach. Learn. Res.*, 22(9):1–32, 2021.

Tingni Sun and Cun-Hui Zhang. Scaled sparse linear regression. *Biometrika*, 99 (4):879–898, 2012.

Tingni Sun and Cun-Hui Zhang. Sparse matrix inversion with scaled lasso. *The Journal of Machine Learning Research*, 14(1):3385–3418, 2013.

Kean Ming Tan and Daniela Witten. Statistical properties of convex clustering. *Electronic Journal of Statistics*, 9(2):2324 – 2347, 2015. doi: 10.1214/15-EJS1074. URL https://doi.org/10.1214/15-EJS1074.

Kean Ming Tan, Daniela Witten, and Ali Shojaie. The cluster graphical lasso for improved estimation of gaussian graphical models. *Computational Statistics & Data Analysis*, 85:23–36, 2015. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2014.11.015. URL https://www.sciencedirect.com/science/article/pii/S0167947314003387.

Ai Ling Teh, Hong Pan, Xinyi Lin, Yubin Ives Lim, Chinari Pawan Kumar Patro, Clara Yujing Cheong, Min Gong, Julia L MacIsaac, Chee-Keong Kwoh, Michael J Meaney, et al. Comparison of methyl-capture sequencing vs. infinium 450k methylation array for methylome analysis in clinical samples. *Epigenetics*, 11 (1):36–48, 2016.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, pages 91–108, 2005.

Ryan J Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *The annals of statistics*, 39(3):1335–1371, 2011.

Kevin Ushey, JJ Allaire, and Yuan Tang. *reticulate: Interface to Python*, 2020. URL https://github.com/rstudio/reticulate. R package version 1.18.

Lieven Vandenberghe, Stephen Boyd, and Shao-Po Wu. Determinant maximization with linear matrix inequality constraints. *SIAM journal on matrix analysis and applications*, 19(2):499–533, 1998.

Paul M VanRaden. Efficient methods to compute genomic predictions. *Journal of dairy science*, 91(11):4414–4423, 2008.

Guido vanRossum. Python reference manual. *Department of Computer Science [CS]*, (R 9525), 1995.

Evelyne Vigneau. Clustering of variables for enhanced interpretability of predictive models. *arXiv preprint arXiv:2008.07924*, 2020.

Binhuan Wang, Yilong Zhang, Will Wei Sun, and Yixin Fang. Sparse convex clustering. *Journal of Computational and Graphical Statistics*, 27(2):393–403, 2018.

Daniela M Witten, Jerome H Friedman, and Noah Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900, 2011.

Katarzyna Wreczycka, Alexander Gosdschan, Dilmurat Yusuf, Björn Grüning, Yassen Assenov, and Altuna Akalin. Strategies for analyzing bisulfite sequencing data. *Journal of biotechnology*, 261:105–115, 2017.

Sewall Wright. The method of path coefficients. *The annals of mathematical statistics*, 5(3):161–215, 1934.

Eunho Yang, Genevera Allen, Zhandong Liu, and Pradeep Ravikumar. Graphical models via generalized linear models. *Advances in neural information processing systems*, 25, 2012.

Eunho Yang, Yulia Baker, Pradeep Ravikumar, Genevera Allen, and Zhandong Liu. Mixed graphical models via exponential families. In *Artificial intelligence and statistics*, pages 1042–1050. PMLR, 2014.

Tianyi Yao and Genevera I. Allen. Clustered gaussian graphical model via symmetric convex clustering. In *2019 IEEE Data Science Workshop (DSW)*, pages 76–82, 2019. doi: 10.1109/DSW.2019.8755774.

Grace Yoon, Irina Gaynanova, and Christian L Müller. Microbial networks in spring-semi-parametric rank-based correlation and partial correlation estimation for quantitative microbiome data. *Frontiers in genetics*, 10:516, 2019.

Ming Yuan. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11(79):2261–2286, 2010. URL http://jmlr.org/papers/v11/yuan10b.html.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 68:49–67, 2006.

Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 03 2007. ISSN 0006-3444. doi: 10.1093/biomet/asm018. URL https://doi.org/10.1093/biomet/asm018.

Bilin Zeng, Xuerong Meggie Wen, and Lixing Zhu. A link-free sparse group variable selection method for single-index model. *Journal of Applied Statistics*, 44(13): 2388–2400, 2017.

Tuo Zhao, Han Liu, Kathryn Roeder, John Lafferty, and Larry Wasserman. The huge package for high-dimensional undirected graph estimation in r. *The Journal of Machine Learning Research*, 13(1):1059–1062, 2012.

Yingyao Zhou, Bin Zhou, Lars Pache, Max Chang, Alireza Hadj Khodabakhshi, Olga Tanaseichuk, Christopher Benner, and Sumit K Chanda. Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. *Nature communications*, 10(1):1–10, 2019.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.